



ITGO: Invasive tumor growth optimization algorithm

Deyu Tang^{a,b}, Shoubin Dong^{b,*}, Yi Jiang^c, Huan Li^d, Yishuan Huang^{a,b}

^a School of Medical Information and Engineering, Guangdong Pharmaceutical University, Guangzhou 510006, PR China

^b School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, PR China

^c Department of Mathematics & Statistics, Georgia State University, Atlanta, USA

^d Basic Medical College, Guangzhou University of Chinese Medicine, Guangzhou 510006, PR China



ARTICLE INFO

Article history:

Received 11 February 2015

Received in revised form 31 July 2015

Accepted 31 July 2015

Available online 10 August 2015

Keywords:

Invasive tumor growth

Meta-heuristic algorithm

Levy flight

Swarm intelligence

Evolutionary computation

ABSTRACT

This paper proposes a new optimization algorithm named ITGO (Invasive Tumor Growth Optimization) algorithm based on the principle of invasive tumor growth. The study of tumor growth mechanism shows that each cell of tumor strives for the nutrient in their microenvironment to grow and proliferate. In ITGO algorithm, tumor cells were divided into three categories: proliferative cells, quiescent cells and dying cells. The cell movement relies on the chemotaxis, random walk of motion and interaction with other cells in different categories. Invasive behaviors of proliferative cells and quiescent cells are simulated by levy flight and dying cells are simulated through interaction with proliferative cells and quiescent cells. In order to test the effectiveness of ITGO algorithm, 50 functions from CEC2005, CEC2008, CEC2010 and a support vector machine (SVM) parameter optimization problem were used to compare ITGO with other well-known heuristic optimization methods. Statistical analysis using Friedman test and Wilcoxon signed-rank statistical test with Bonferroni–Holm correction demonstrates that the ITGO algorithm is better in solving global optimization problems in comparison to the other meta-heuristic algorithms.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

We often encounter optimization problems in scientific and technological research and development. These problems can be considered as different problems: continuous or discrete; constrained and unconstrained; multimode and unimodal; and multi-objective and single objective. The meta-heuristic methods were widely used to solve these problems, and the essence is to design an algorithm to approximate and solve the optimization problems [1]. In the last three decades, a great interest has been devoted to meta-heuristics method. Metaheuristics method can be divided into three categories [2]: evolutionary computation (also called evolutionary algorithms), swarm intelligence algorithm and physical phenomena algorithms.

Evolutionary computation (EC) is the general term for several optimization algorithms that are inspired by the Darwinian principles of nature's capability to evolve living beings well adapted to their environment. Some classical algorithms such as genetic algorithms [3], evolution strategies [4], evolutionary programming [5], and genetic programming [6] have been widely accepted and used in many research fields and applications. Although the

technologies of these algorithms are different, they share a common ideology, which is to imitate the individual's evolution through mutation, selection, recombination operation, thus resulting in a better solution. There are some new evolutionary algorithms, including Estimation of Distribution Algorithms (EDA) [7], Differential Evolution (DE) [8–11], Simulated Annealing (SA) algorithm [12], Tabu search [13] and Variable Neighborhood Search (VNS) [14]. Coevolutionary Algorithms (CoEA) [15,16], and Cultural Algorithms (CA) [17]. Estimation of Distribution Algorithms, also referred to as Probabilistic Model-Building Genetic Algorithms (PMBGA), was introduced in the field of evolutionary computation, for the first time by Mühlenbein and Paaß. Differential Evolution (DE) algorithm is a very popular algorithm used to solve many continuous optimization problems. Coevolutionary Algorithm (CoEA) is another type of evolutionary algorithm; such algorithm mimics the intimate relationship between different species such as predators and prey, hosts and parasites, or insects and the flowers that they pollinate-influence each other's evolution. Inspired from the cultural evolution process in nature, Cultural Algorithms (CAs) are proposed. The term cultural evolution has been more recently used to refer to the idea that the processes producing cultural stability and change are analogous in important respects to those of biological evolution. CAs consist of three components: a population space; a belief space; the communications protocol [18,17].

* Corresponding author. Tel.: +86 20 87110014.

E-mail addresses: scutdy@126.com (D. Tang), sbdong@scut.edu.cn (S. Dong).

Swarm intelligence (SI) algorithm is a novel distributed model for solving optimization problems, which is inspired by the collective behavior of insects or animals. SI refers to one individual without intelligence, but the intelligent characteristics emerge through cooperation with each other. In intelligent system, the population is a set of individuals that are able to communicate with each other and the local environment, and a group of individuals distributed in search space should work in cooperation to solve problems. For swarm intelligence algorithms, individuals without centralized control are able to solve complex problems. In 1995, inspired by the behavior of birds, James Kennedy and Russell Eberhart put forward particle swarm optimization (PSO) [19–21] as a global optimization technique; M. Dorigo and her colleagues proposed ant colony optimization (ACO) [22,23] for the hard combinatorial optimization problems by mimicking the ants foraging behaviors. Inspired by the social foraging behavior of *Escherichia coli* bacteria, Passino introduced a bacterial foraging optimization algorithm (BFOA) in 2002 [24]. Studies on honey bees' balance exploitation of known food sources and exploration for new and potentially better food sources in a dynamic environment, the bee colony optimization metaheuristic (BCO) [25] and artificial bee colony algorithm (ABC) [26,27] were proposed. Biogeography-based optimization algorithm (BBO), developed by Dan Simon in 2008, imitates the biological system under different environments, and balances the immigration and emigration of different species [28]; Cuckoo Search algorithm (CS) [29] proposed by Xin-she Yang et al. in 2009, was inspired by the cuckoo brood parasitism behavior; Bat algorithm (BA) [30] is an effective global optimization method, which was inspired by the characteristics of the bats which emit ultrasond. Glowworm swarm optimization (GSO) [31] was inspired by the behaviors of a group of firefly which exchanges information by fluorescence. Teaching-learning-based optimization (TLBO) was proposed based on the effect of a teaching on the score of learners in a class [32]; Gray wolf optimizer (GWO) was proposed based on the behavior of capturing prey of the gray wolf swarm [2].

The third main branch of meta-heuristics is the physical phenomena algorithms. Such optimization algorithms typically mimic physical rules due to some physical phenomena. The mechanism of these algorithms is different from EAs or SIs, in which a random set of search agents move and communicate throughout search space according to physical rules. This movement is implemented, for instance, using gravitational force, weights, ray casting, inertia force, electromagnetic force, and so on. Some of the most popular algorithms are Gravitational Search Algorithm (GSA) [33,34], Gravitational Local Search (GLSA) [35], Black Hole (BH) algorithm [36], Big-Bang Big-Crunch (BBCB) [37], Central Force Optimization (CFO) [38], Charged System Search (CSS) [39], Curved Space Optimization (CSO) [40], Ray Optimization (RO) [41] algorithm, and Small-World Optimization Algorithm (SWOA) [42]; Galaxy-based Search Algorithm (GbSA) [43], and Quantum-behaved particle swarm optimization (QPSO) [44,45].

Tree-seed algorithm (TSA) [46] can be seen as a tree growth model based on the relation between trees and their seeds. In this framework, seeds are produced by the trees and the best seed for a tree can substitute for this tree. In other words, the growth of a tree is based on the seeds produced by it. The most important point is the two update rules for the producing seeds for a tree. The first update rule is achieved by the interaction between the best tree location and the randomly selected tree location in population, which can be seen as an exploitation for the local search operation, and the second update rule is achieved by the interaction between the present tree location and the randomly selected tree location in population, which can be seen as an exploration for the global search operation. The search tendency (ST) parameter

controls the selection of the two update rules, which finishes the balances between the exploration and the exploitation.

Gray wolf optimizer (GWO) [2] mimics the leadership hierarchy and hunting mechanism of gray wolves in nature. Four types of gray wolves such as alpha (leader), beta (leader), delta (leader) and omega (commoner) are classified as two layers for simulating the leadership hierarchy; the four main steps of encircling prey, hunting prey, searching for prey and attacking prey are implemented for search in optimization problems. The update rule of encircling prey is achieved by the interaction between the gray wolf and prey, and the position of hunting prey is guided by the three best gray wolfs (alpha, beta and delta). The exploration and the exploitation are implemented by the searching and attacking operation.

Artificial algae algorithm (AAA) [47] is inspired by the living behaviors of microalgae, photosynthetic species. Under sufficient nutrient conditions, the algal colony receives enough light, and it grows and reproduces itself. The algorithm is based on the main three operations: evolutionary process, adaptation process and the movement of microalgae. For the evolutionary process, in any randomly selected dimension, algal cell of the smallest algal colony (corresponding to the worst solution) dies and algal cell of the biggest colony (corresponding to the best solution) replicates itself, which can speed up the convergence. In adaptation process, each algal colony has a starvation value. When one algal colon (solution) has a big starvation value, it will be guided by the best algal colon (solution). In the movement of microalgae, a new mutation method can be seen in this step, which is called 'Helical movement'. For each algal colony (solution), randomly choose the j algal colony (the better solution) from the population and three dimensions to update the old solution by the three different search direction, which is the key in this algorithm.

Cuckoo Search algorithm (CS) [29] was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds (of other species). It can be seen as two main steps. In the first step, it can be seen as the search method of simplified particle swarm optimization. The moving direction of each particle was guided by the use of the global optimal solution. This solution like an attractive factor combined the current best individual with the global best particle. The second step uses the approximate differential evolution rules with a probability P . Select two different individuals randomly from the population to update the current individual, so as to get a better solution. It can enhance the global search ability of the algorithm.

Gravitational Search Algorithm (GSA) [34] is driven by the simulation of a collection of masses which interact with each other based on the Newtonian gravity and laws of motion. The algorithm is a centralized control algorithm, without the need for the guiding by a global best individual. Each individual j in the population attracts the individual i , the attractive force is inversely proportional to the distance of two individuals, and proportional to the product of mass (fitness) of two individuals. Individual i was affected by all other individuals to search for appropriate best solutions. If individual j is near the individual i and the fitness value is better, so it is more important for the i individual. This is a good algorithm that not only considers the distance between individuals, but also considers the fitness value of two individuals.

Teaching-Learning-Based Optimization (TLBO) [32] can be divided into Teaching Phase and Learner Phase. In the teaching stage, the best learner (solution) was recommended from the population as a teacher (best solution), calculates the average scores in a class (average position of the population), and a teacher improves the score of each learner (individual) according to the average scores in a class; in learner stage, for each learner X_i in the class, a new learner X_j was chosen randomly from the class, and a forward direction or opposite direction of study was used to update the solution X_i .



Fig. 1. GBM multicellular tumor spheroid (MTS) gel assay showing dendritic invasive branches. (a) The invasive branches centrifugal evolve from the central MTS. The linear size of central MTS is approximately 400 mm. (b) The invasive branches are composed of chains of invasive cells. The images are adapted from Ref. [52].

In fact, exploration is the process of visiting entirely new regions of a search space, whilst exploitation is the process of visiting those regions of a search space within the neighborhood of previously visited points. The success of meta-heuristic methods to solve optimization problems depends on the balance of exploration (diversification) and the exploitation (intensification) [48]. However, all metaheuristic algorithms employ both the exploration and exploitation aspects but they use different approaches and operators. In other words, all search algorithms have a common framework. So, the No Free Lunch (NFL) theorem [49] is worth mentioning here. This theorem has logically proved that there is no meta-heuristic best suited for solving all optimization problems. That is, a particular meta-heuristic may show very good results on a set of problems, but the same algorithm may show poor performance on a different set of problems. Obviously, NFL makes this field of study highly active which results in enhancing current approaches and proposing new meta-heuristics every year. This also motivates our attempts to develop a new meta-heuristic inspired by the mechanism of invasive tumor growth.

In this context, the first contribution of our work is a new framework for swarm intelligence algorithm. There are three populations with different levels according to their nutrient concentration (fitness), that is, proliferative cell population, quiescent cell population and dying cell population. The cells in the high level population can guide the cells in the low level population. The interactions between three populations accomplish the global search task. The second contribution is the five search strategies (five operations) corresponding to the different four roles of tumor cell. In this search mechanism, the proliferative cells search by a levy flight, which enhances the local search ability; the quiescent cells search by the proliferative cells and mutation, which can be seen as the balance between the global search and local search; the dying cells search by the proliferative cells and quiescent cells, which enhances the global search ability; the invasive cells search by the proliferative cells, which enhances the local search ability and speed up the convergence; all the cells in tumor can achieve the random walk, which enhances the local search ability. The five strategies simulate the growth mechanism of the tumor.

In this paper, we proposed a new metaheuristic algorithm called invasive tumor growth optimization (ITGO). The rest of the paper was organized as follows. Section 2 introduces the mechanism of invasive tumor growth. Section 3 describes ITGO model and algorithm. Section 4 presents experimental simulations compared with well-known meta-heuristic algorithms, results, and analysis. Section 5 discusses the other parameters of ITGO algorithm. Finally, the work is concluded in Section 6.

2. Modeling of invasive tumor growth

Tumor is an abnormal growth of tissue. Induced by the oncogenic disorder, abnormal proliferation of clonal cell clusters lead to formation of tumors, which cannot be controlled by the body's internal regulating mechanism [50,51]. From the point of view of the system science, the tumor can be regarded as a complex linear dynamic system that can self-organize. The tumor cells proliferate and invade into surrounding tissue as seen in Fig. 1 [52]. They can also travel through lymphatic or blood vessel and infiltrate into the distal organs [53]. The growth of tumor is a complex process, influenced by the interactions between tumor cells and their microenvironment, including their surrounding cells, as well as the extracellular matrix (ECM), chemical signals, as well as metabolic substrates such as oxygen and glucose [53].

In the early stage of tumor growth, tumor cells obtain nutrients through diffusion. Nutrient diffuses from nearby blood vessel to tumor, and establishes a gradient from outside to interior of tumors. Diffusion equations have been used to describe this process [54–56,52,57,58]. The distribution of nutrient concentration of tumor cells is shown in Fig. 2; we also use a diffusion equation for nutrient, n :

$$\frac{\partial n}{\partial t} = D_n \nabla^2 n + \sigma e - \varphi c - \lambda n \quad (1)$$

where n , D , σ , φ , and λ are the nutrient concentration, diffusion coefficient, production, consumption, and natural decay rates of the nutrient, respectively. To survive and grow, a tumor cell consumes nutrients at a certain rate, if the nutrient concentration around it in the microenvironment is lower than the specified value, it will become a quiescent cell or dying cell or apoptosis (dead). The consumption of nutrient also reduces nutrient concentration [59]. Many cellular signaling pathways that regulate migration have been described in recent years. A cell has a persistent random walk of motion [57,58,60]. Cell migration can be driven by various factors such as the directed response to gradients in pressure and chemical substances (nutrient concentrations), etc.

Therefore, it is generally assumed that tumor can be regarded as an isotropic sphere. A concentric spatial layered structure consists of an outer rim of proliferating cells, an intermediate layer of viable, but dormant, quiescent cells, and an inner necrotic core [57]. In the outer layer of tumor with relatively higher nutrient concentration, cells were called 'proliferative cells' (these cells move fast); in the middle layer of the tumor with medium nutrient concentrations, these cells were called 'quiescent cells' (they are not still, they move slowly); in the inside layer with lower nutrient

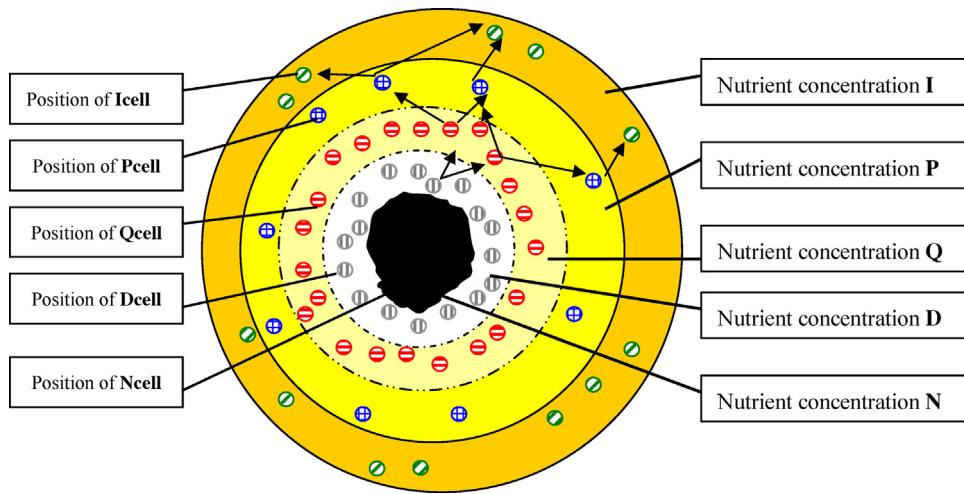


Fig. 2. The optimization model of invasive tumor growth.

concentrations, these cells were called dying cells (they move very slowly). Tindall and Please [58] proposed a tumor growth model which considers that the tumor has a chemotaxis phenomenon, that is to say, tumor cells will move in the direction toward the nutrient gradient. Dorie et al. [56] use experiments to explain that proliferative cells and quiescent cells move toward the outer layer of the tumor. Proliferative cells, quiescent cells and dying cells have different behaviors. Vitro experiments show that proliferative cells may turn into quiescent cells or dying cells, and quiescent cells may also turn into proliferative cells or dying cells according to the nutrient concentration in microenvironment [56,52]. Most studies on continuum mathematical models of avascular tumor growth accept the assumption that all cells within a tumor have a common velocity. Some studies supposed that the tumor is porous media and the velocity of the tumor can be expressed in terms of the pressure [61]. Pettet et al. [62] and Tindall et al. [58] have considered the possibility of different cell velocities due to chemotaxis, movement of cells along a chemical concentration gradient either toward or away from the chemical stimulus. For simplicity they assumed two types of cells, proliferating and necrotic. Moreover, Sherratt and Chaplain [60] have assumed that the cell movement is due to “contact inhibition of migration”. They have considered that the cells can be in three states: proliferation, quiescence and necrosis and the diffusion is taken into account. It is generally accepted that the invasive behavior of tumor is the outcome of many complex interactions occurring among the tumor cells, and between a tumor and its host microenvironment [53]. Tumor invasion itself is a complex multi-step process involving homotypic detachment, enzymatic matrix degradation, integrin-mediated heterotypic adhesion, as well as active, directed and random motility [63]. Yang Jiao et al. [52] proposed an invasion of tumor growth model to simulate the behavior of tumor cell.

Summarized from the mechanism of tumor growth, tumor cells have the following characteristics:

- (1) In order to survive and grow, tumor cells must get enough nutrients, and the tumor cells have invasive ability.
- (2) Tumor cells will move toward the direction of higher nutrient concentrations.
- (3) There is a complex interaction between tumor cells.
- (4) They have a random walk of motivation.
- (5) When nutrient concentration is lower than a certain value, the proliferative cells may turn into quiescent cells; when nutrient concentration is lower than the minimum value, the quiescent cells turn into dying cells, or even apoptosis.

- (6) Nutrient concentration corresponds to different types of tumor cells.

3. Design of invasive tumor growth optimization (ITGO) algorithm

A cell may be one of the two classes of phenotypes: either invasive or non-invasive [52]. The invasive cells (it is a special cell with invasive behavior) live at the outer of tumor. The non-invasive cells remain in the primary tumor and can be proliferative, quiescent, dying, or necrotic cell (dead), depending on the nutrient supply they get. For avascular tumor growth, our focus here, the nutrients available to the tumor cells are essentially the nutrient concentrations that diffuse into the tumor through its surface. As the tumor grows, the amount of nutrient supply, which is proportional to the tumor, cannot meet the needs of all cells, leading to the development of necrotic, dying and quiescent regions.

3.1. Invasive tumor growth optimization model

In order to solve the optimization problem, we propose an optimization model of invasive tumor growth. In this model, we assume that the tumor cells have the following characteristics:

- (1) Considering the invasion of tumor cells, tumor cells were divided into four layers: invasive cells, proliferative cells, quiescent cells, and dying cells.
- (2) Tumor is a complex dynamic system; there are interactions between different cell layers. All the cells move toward the direction of high nutrient concentration.
- (3) Invasive cells will guide the moving direction of proliferative cells.
- (4) The proliferative cells have the intrusive behavior by mutation.
- (5) Proliferative cells will guide the moving direction of quiescent cells; quiescent cells will have the interaction with neighbors.
- (6) In dying cells region, nutrient concentration is low, so dying cells will move toward the direction of the proliferative cells and the quiescent cell.
- (7) Dying cells would be death due to the low nutrient concentration.
- (8) The state of invasive cells, proliferative cells, quiescent cells and dying cell varies with the nutrient concentration.
- (9) If tumor cells cannot get sufficient nutrient in the chemotaxis function, cell will perform a random walk behavior.

In our model, 'nutrient' corresponds to 'fitness' of heuristic algorithm, and 'nutrient concentration' corresponds to the 'fitness value'. A tumor cell growth means the search of an agent (or particle). As shown in Fig. 2, the black area presents the necrotic cells (each cell is called 'Ncell') of tumor; in this region, cells have been death without adequate nutrient. The vertical line gray balls represent the dying cells (each cell is called 'Dcell'), if they get enough nutrient, they will move and turn into quiescent cells or the proliferative cells; the horizontal line red balls present the quiescent cells (each cell is called 'Qcell'), which is located in the middle layer of tumor, in this region, nutrient concentration is at the medium level, and these cells will move toward the direction of proliferative cells (each cell is called 'Pcell'). At the same time, quiescent cells have interaction among themselves. The grid line blue and oblique line green ball are located in the outermost layer of tumor; they represent the proliferative cell and invasive cells (each cell is called 'Icell'); in the outermost layer, they can obtain highest nutrient and they are active. Fig. 2 shows the diffusion of nutrient in tumor corresponding to the tumor growth of each cell. 'I, P, Q, D, N' in Fig. 2 present the different nutrient concentration region corresponding to the 'Icell' (invasive cell), 'Pcell' (proliferative cell), 'Qcell' (quiescent cell), 'Dcell' (dying cell), 'Ncell' (necrotic cell), respectively. The deep and shallow color from outside to inside presents the different nutrient concentration for different cells in tumor.

In the process of tumor growth, each cell wants to get enough nutrients to facilitate its growth. In order to survive and grow, each cell moves with a variety of ways to get the most nutrients (fitness). In our model, the cells of tumor were divided into four levels according to the nutrient concentration from high to low: Proliferative cells (Pcell), Quiescent cells (Qcell), Dying cells (Dcell) and Invasive cells (Icell). In order to grow, different cells have different behaviors and interactions. The detail is described as following.

3.1.1. The growth of proliferative cells

The proliferative cells must have adequate nutrient to grow. Study on the mechanism of tumor growth indicates that nutrient is conveyed to each cell of tumor by diffusion, just like the 'heat conduction'. The nutrient spreads from the outer layer to the inner layer of the tumor. The proliferative cells were located in the outermost layer of a tumor; therefore, cells in this layer can easily obtain sufficient nutrients, they are more active and they grow fast. These cells grow by a special way and have a strong invasive ability. In order to reflect the intrusive behavior of proliferative cells, we use the levy flight [64,65] to simulate it. In fact, many natural phenomena

$$sQcell_{i,j}(t+1) = \begin{cases} Qcell_{i,j}(t) + \beta \cdot step \cdot (hPcell_{p,j}(t) - Qcell_{i,j}(t)) + \beta \cdot step \cdot ((Qcell_{x,j}(t) - Qcell_{y,j}(t))), & rand < 0.5 \\ Qcell_{i,j}(t) + \beta \cdot step \cdot (cPcell_{p,j}(t) - Qcell_{i,j}(t)) + \beta \cdot step \cdot ((Qcell_{x,j}(t) - Qcell_{y,j}(t))), & rand > 0.5 \end{cases} \quad (9)$$

exhibit the features of Levy flight, such as the flying of albatross, the flying of Drosophila and the feeding way of other animals, even the hunting mode of human is a Levy flight. Some physical phenomena also exhibit these characteristics of Levy flight, such as fluorescent particle diffusion, noise, etc. The intrusive behavior of proliferative cells can also be viewed as a Levy flight.

$$Pcell_{i,j}(t+1) = Pcell_{i,j}(t) + \alpha \cdot Levy(s) \quad (2)$$

$Pcell_{i,j}$ presents the position of proliferative cell, $Levy(s)$ presents the intrusive behavior of the proliferative cell. We can see that, it moves occasionally short distance or long distance; α is the control size of step.

$$\alpha = rand \cdot \left(\frac{Fes}{Max_fes} \right) \quad (3)$$

where Fes is the current number of fitness evaluation consumed, and Max_fes is the max number of fitness evaluations consumed.

Researches of tumor mechanism show that, in early stage of tumor growth, tumor cells grow in a linear velocity, and in the late stage, the bulk of quiescent cells and dying cells almost no longer move or move slowly, only the proliferative cells grow fast in a special way. Formula (3) shows that, α control the size of the moving step of each proliferative cell, in the early stage, proliferative cells moves not very fast; but in the late stage, proliferative cell moves more long distance according to the step of $Levy(s)$ and α . It conforms to the mechanism of tumor growth.

Levy distribution usually appears in a simplified form, as follows:

$$Levy(s) \sim s^{-1-\omega}, \quad (0 < \omega \leq 2) \quad (4)$$

Broadly speaking, Levy flight is a random walk and the step size obeys Levy distributions, and walk direction obeys uniform distribution. To perform the Levy distributions, we use the Levy distribution characterized with Mantegna law [64,65] to select the step length vector. In Mantegna law, the step size is designed as follows:

$$step = \frac{u}{|\nu|^{1/\omega}} \quad (5)$$

where u, ν obeys normal distribution, i.e.,

$$u \sim N(0, \sigma_u^2), \quad \nu \sim N(0, \sigma_\nu^2) \quad (6)$$

$$\sigma_\nu = \sqrt{\frac{\Gamma(1+\omega)\sin(\pi\omega/2)}{\Gamma((1+\omega)/2)\omega^{2(\omega-1)/2}}} \quad (7)$$

$$\sigma_u = 1 \quad (8)$$

ω is a constant, where $\omega \in [0.3, 1.99]$ according to the literary [65,66].

3.1.2. Quiescent cells of tumor (Qcell)

Quiescent cells are located in the middle layer of the tumor. According to the nutrient diffusion mechanism of the tumor, we can see that these cells can obtain nutrient only at a moderate level of nutrient concentration. Because the nutrient is not enough, the quiescent cell grows relatively slowly. In order to grow, these cells mainly move through interaction with proliferative cells and themselves to get more nutrients. Due to insufficient nutrient concentration, these cells have the intrusive behavior by mutation.

The Quiescent cells move toward the higher nutrient concentrations, which is guided by proliferative cells and they interact with each other.

$$Qcell_{i,j}(t+1) = \begin{cases} Qcell_{i,j}(t), & rand < e^{\left(\frac{Fes}{Max_fes} - 1 \right)} \\ sQcell_{i,j}(t+1), & \text{else} \end{cases} \quad (10)$$

$$\beta = rand(0, 1) \cdot normal(0, 1) \quad (11)$$

Here, $hPcell_{p,j}(t)$ is the historical best position of proliferative cell, $cPcell_{p,j}(t)$ is the current position of proliferative cell. $Qcell_{i,j}(t)$ is the old position of quiescent cell and $sQcell_{i,j}(t+1)$ is the current position of quiescent cell. The $step$ presents the step size of Levy flight such as formula (5); β presents the control size of step, $rand(0, 1)$ means the random distribution, $normal(0, 1)$ means the normal distribution as formula (11); p is integer, which indicates a proliferative cell chosen randomly from the subpopulation of proliferative cell; x, y are integers ($x \neq y$), which indicate two quiescent cells near the i quiescent cell. Formula (10) presents the mutation of Quiescent cell.

3.1.3. Dying cells of tumor (Dcell)

The dying cells are located in the third layer of the tumor. They can only obtain the rare nutrient with lower nutrient concentration, are not dead, but still alive and grow. In order to survive, these cells mainly get more nutrients through interaction with proliferative cells and quiescent cells. Because the living environment is not good (rare nutrient), some cells in this layer will die, and a few cells grow through interaction, are lucky to survive and may even turn into the quiescent cell or proliferative cell.

In dying cells region, nutrient concentration is very low, so these cells move toward the direction of quiescent cells and proliferative cells with a higher nutrient concentration.

$$\begin{aligned} Dcell_{i,j}(t+1) = & Dcell_{i,j}(t) + \gamma \cdot (cPcell_{p,j}(t) - Dcell_{i,j}(t)) \\ & + \gamma \cdot ((Qcell_{x,j}(t) - Dcell_{i,j}(t))) \end{aligned} \quad (12)$$

$Dcell_{i,j}(t)$ indicates the position of the old dying cell, $Dcell_{i,j}(t+1)$ indicates the position of the current dying cell, $Qcell_{x,j}(t)$ indicates the position of quiescent cell, $cPcell_{p,j}(t)$ indicates the position of proliferative cell (the current best position), p is integer, which indicates a proliferative cell chosen randomly from the subpopulation of proliferative cells; x is integer, which indicates a quiescent cell chosen randomly from the subpopulation of quiescent cells, where $\gamma = \text{rand}[-1, 1]$. Formula (12) shows that $cPcell_{p,j}(t)$ and $Qcell_{x,j}(t)$ guide $Dcell_{i,j}(t)$ to move for growth.

3.1.4. Invasive cells

The invasive cells are considered as mutant daughters of the proliferative cells [52], which can gain nutrient and motility that allow them to leave the outer layer of tumor and invade into surrounding microenvironment (other layer of the tumor). We consider that the invasive cells can move from the outer layer to the inner layer of tumor.

In the dying cells region, if the nutrient concentration of dying cells was lower than the average nutrient concentration of dying cell subpopulation, the dying cell will be apoptosis (dead); when the proliferative cell has obtained enough nutrients and completed the growing behavior, cloning will happen, that is, a mutant daughter of it called invasive cell will be born. When dying cell is apoptosis, the invasive cell will replace it. Formula (13) presents a new cell which is born randomly in the tumor, Formula (14) shows that a mutant daughter of a current best proliferative cell is born, which is an invasive cell $Icell_{i,j}(t+1)$. Formula (15) shows that a dying cell is apoptosis and replaced by the invasive cell $Icell_{i,j}(t+1)$.

$$\text{newCell}_{i,j}(t) = \text{Random}(i, D), \quad D \in [D \min, D \max] \quad (13)$$

$$Icell_{i,j}(t+1) = cPcell_{p,j}(t) + \eta \cdot (\text{newCell}_{i,j}(t) - cPcell_{p,j}(t)) \quad (14)$$

$$Dcell_{i,j}(t+1) = Icell_{i,j}(t+1) \quad (15)$$

Here, $\eta \in \text{rand}[0, 1]$ presents the growth speed, $\text{Random}(i, D)$ produces a new solution randomly in the search space.

3.1.5. Random walk of cells

Cell migration is most commonly and widely described as a persistent random walk motion [66,67]. All the cells in the tumor have the characteristics of random walk. When the proliferative cells, quiescent cells and dying cells are unable to get enough nutrients through the interaction or mutation, these cells grow with a random walk way to get the nutrients.

Each cell in tumor has a cycle of life. When a cell is not affected by other chemotaxis within the specified cycle of life, the cell performs

a random walk as formula (16). Here, $\text{newCell}_{i,j}(t)$ presents a new cell born using formula (13).

$$Cell_{i,j}(t+1) = Cell_{i,j}(t) + \lambda \cdot \left(\frac{\text{newCell}_{i,j}(t)}{\|\text{newCell}_{i,j:D}(t)\|} \right) \quad (16)$$

where $\|\text{newCell}_{i,j:D}(t)\|$ is an Euclidean norm. $\lambda \in \text{rand}[-1, 1]$.

3.2. Invasive tumor growth algorithm

Based on the invasive tumor growth model, we design the invasive tumor growth optimization algorithm. The main function of our algorithm was shown in [Algorithm 1](#) (ITGO()). In order to realize the invasive tumor growth optimization algorithm, six steps were adopted. In the first step, we set the size of the population (a population consists of many tumor cells), and max number of fitness evaluations consumed (Max_fes), etc. The second step sorts according to the nutrient concentration (fitness value) of each cell, the high nutrient concentrations of cells were identified as proliferative cells (Pcell, accounted for 20% of the total number of cells), the middle nutrient concentration of cells were identified as quiescent cells (Qcell, accounted for 60% the total number of cells), the low nutrient concentration of cells were identified as dying cells (Dcell, accounted for 20% the total number of cells). In order to grow, Pcell, Qcell and Dcell use a different way to get adequate nutrients. The third step implements the growth of proliferative cells (Pcell). Because the 'Pcell' has the highest nutrient concentration, it grows fast, and the detail of its growth is described by [Algorithm 2](#) (ProliferativeCellGrowth()). The fourth step implements the growth of quiescent cell. As the nutrient concentration of these cells belongs to the middle level, most of them grow through interaction with proliferative cell and their neighbors to get nutrients, as seen in [Algorithm 3](#) (QuiescentCellGrowth()). The fifth step implements the growth of dying cells. As nutrient concentration of these cells is very low, most of them run through interaction with proliferative cells and quiescent cells to get enough nutrients, as seen in [Algorithm 4](#) (DyingCellGrowth()). The sixth step implements the growth of invasive cell. In this step, the invasive cell is born due to the mutation of proliferative cell; because the tumor in the space is limited, the new intrusive cell will replace the dying cell with too lower nutrient concentrations for survival, as described in [Algorithm 5](#) (InvasiveCellGrowth()). In the process of tumor growth, all the cells will occasionally perform random walks, and this is the characteristic of tumor cell, as described in [Algorithm 6](#) (RandomWalk()).

In [Algorithm 2](#) (ProliferativeCellGrowth()), the lines 2–6 present the possibility of proliferative cell growth. $\frac{Fes}{Max_fes}$ can be seen as a liner function (Formula (3)). α is the control size of step, and $\text{Levy}(\lambda)$ is the step length of Levy flight. As other meta-heuristic algorithms, at the late of the iteration, most individuals (cells) gather into a local optimum. So the proliferative cells growing with a greater probability ensure that the algorithm can jump out of the local optimal solution. As seen as the line 7–line 15, when the value of $Gc(i)$ is greater than Max_gc (growth cycle threshold), the cell performs random walks. This means that when the proliferative cell cannot update the historical position ($hPcell$) for a long time, and it performs a random walk, so as to find the position with rich nutrient (fitness).

(2) For [Algorithm 3](#) (QuiescentCellGrowth()) performs the growth process of quiescent cells. Codes represent the interaction between quiescent cells and other cells. Historical best position $hPcell(p,j)$ and current best position $cPcell(p,j)$ lead the quiescent cell to move alternately so that the quiescent cell can get adequate nutrients. Code lines 15–19 show the mutation of quiescent cell, and $\exp(Fes/Max_fes - 1)$ is the rate of mutation.

Algorithm 1

ITGO (Np, Psize, Qsize, Dsize, Gc, Max_fes, Fes, Err, Beta, Max_gc).

| | |
|----------------|---|
| Input: | population size (N_p), number of proliferative cells ($Psize = 20\% N_p$), number of quiescent cells ($Qsize = 60\% N_p$), number of dying cells ($Dsize = 20\% N_p$), growth cycle of tumor cell (Gc), max number of fitness evaluations consumed (Max_fes), number of fitness evaluation consumed (Fes), stopping criteria (Err), Levy parameter ($Beta$), max number of growth cycle (Max_gc). |
| Output: | the global best cell in population. |
| 1. | $Fes = 0;$ |
| 2. | Generate an initial tumor cells population by uniformly and randomly sampling from the feasible solution space; |
| 3. | $Fes = Fes + Np;$ //compute the fitness evaluations |
| 4. | While ($Fes < Max_fes$ and absolute error $> Err$) |
| 5. | Sort the tumor cells according to the fitness value of each cell in the population (ascending order); |
| 6. | Choose $Psize$ cells (20%) with the best fitness value in tumor cells as proliferative cells ($PCells$); |
| 7. | Choose $Qsize$ cells (60%) with the middle fitness value in tumor cells as quiescent cells ($QCells$); |
| 8. | Choose $Dsize$ cells (20%) with the low fitness value in tumor cells as dying cells ($DCells$); |
| 9. | ProliferativeCellGrowth ($Pcells$, $Psize$, Gc , Fes , $Beta$, Max_Gc); |
| 10. | QuiescentCellGrowth ($Pcells$, $Psize$, $Qcells$, $Qsize$, Gc , Fes , $Beta$, Max_Gc); |
| 11. | DyingCellGrowth ($Pcells$, $Psize$, $Qcell$, $Qsize$, $Dcells$, $Dsize$, Gc , Fes , Max_Gc); |
| 12. | InvasiveCellGrowth ($Pcells$, $Psize$, $Dells$, $Dsize$, Fes); |
| 13. | Combine three subpopulations ($PCells$, $QCells$, $DCells$) as a new tumor cells population; |
| 14. | End while |
| 15. | Output the global best Cell in tumor cells population; |

Algorithm 2

ProliferativeCellGrowth (Pcells, Psize, Gc, Fes, Beta, Beta, Max_Gc).

| | |
|----------------|--|
| Input: | proliferative cells subpopulation ($PCells$), number of proliferative cells ($Psize = 20\% N_p$), growth cycle of tumor cell (Gc), number of fitness evaluation consumed (Fes), Levy parameter ($Beta$), max number of growth cycle (Max_gc). |
| Output: | historical proliferative cells subpopulation ($hPCells$), current proliferative cells subpopulation ($cPCells$) |
| 1. | for $i = 1: Psize // p$ is the number of Pcells |
| 2. | for $j = 1: D // Dimension D$ |
| 3. | now $Pcell(i,j) = Pcells(i,j) + \alpha \cdot Levy(\lambda)$ //Formulae (3)–(8) |
| 4. | end for |
| 5. | Calculate the fitness evaluation of now $Pcell(i)$; |
| 6. | $Fes = Fes + 1$; |
| 7. | If now $Pcell(i)$ has more Nutrient (best fitness value) |
| 8. | $cPcell(i) = \text{now } Pcell(i)$; |
| 9. | else |
| 10. | $hPcell(i) = Pcell(i)$; |
| 11. | $Gc(i) = Gc(i) + 1$; |
| 12. | end if |
| 13. | If $Gc(i) > Max_gc$ |
| 14. | RandomWalk ($cPcell(i)$, $Gc(i)$); |
| 15. | end if |
| 16. | end for |

(3) **Algorithm 4** (DyingCellGrowth()) performs the process of dying cells. Lines 4–7 show that the position of dying cell was updated according to Formula (12), and a new proliferative cell and a new quiescent cell lead the dying cell to move for a better nutrient. In lines 10–17, the dying cell will run random walks like the proliferative cell or the quiescent cell.

(4) **Algorithm 5** (InvasiveCellGrowth()) performs the growth process of the invasive cells. If the nutrient concentration of a dying cell is lower than the average nutrient concentrations of all the dying cells, a invasive cell ' $Icell(i,j)$ ' is born by the mutation of proliferative cell ' $cPcell(i,j)$ ', and the dying cell was replaced by the invasive cell.

Algorithm 3

QuiescentCellGrowth (Pcell, Psize, Qcells, Qsize, Gc, Fes, Beta, Max_Gc).

| | |
|----------------|---|
| Input: | historical proliferative cells subpopulation ($hPCells$), current proliferative cells subpopulation ($cPCells$), quiescent cells subpopulation ($QCells$), number of proliferative cells ($Psize = 20\% N_p$), number of quiescent cells ($Qsize = 60\% N_p$), growth cycle of tumor cell (Gc), number of fitness evaluation consumed (Fes), Levy parameter ($Beta$), max number of growth cycle (Max_gc). |
| Output: | updated $Qcells$ (array) |
| 1. | for $i = 1: Qsize // Qsize$ is the number of $Qcells$ |
| 2. | Select the p cell randomly from $PCells$ subpopulation as $Pcells(p)$; |
| 3. | Calculate the distance between all the $Qcells$ (except $Qcells(i)$) and the $Qcells(i)$ and sort; |
| 4. | Choose two $Qcells$ nearest to the $Qcells(i)$ as Neighbors: $Qcell(x)$, $Qcell(y)$, $x \neq y$; |
| 5. | Using levy flight 'step' by βeta //Formulae (4)–(8) |
| 6. | Calculates the control size of 'step' b//Formula (11) |
| 7. | for $j = 1:D$ |
| 8. | If $\text{rand} < 0.5$ |
| 9. | now $Qcell(i,j) = Qcells(i,j) + b * \text{step} * (hPcells(p,j) - Qcells(i,j)) + b * \text{step} * (Qcells(x,j) - Qcells(y,j))$; |
| 10. | else |
| 11. | now $Qcell(i,j) = Qcells(i,j) + b * \text{step} * (cPcells(p,j) - Qcells(i,j)) + b * \text{step} * (Qcells(x,j) - Qcells(y,j))$; |
| 12. | end if |
| 13. | //formulae (9) and (11) |
| 14. | end for |
| 15. | for $j = 1:D$ |
| 16. | If $\text{rand} < \exp(Fes/Max_fes - 1)$ //formula (10) |
| 17. | now $Qcell(i,j) = Qcells(i,j)$; |
| 18. | end if |
| 19. | end for |
| 20. | Calculate the fitness evaluation of now $Qcell(i)$; |
| 21. | $Fes = Fes + 1$; |
| 22. | If now $Qcell(i)$ has more Nutrient (Fitness value) |
| 23. | $Qcells(i) = \text{now } Pcell(i)$; |
| 24. | else |
| 25. | $Gc(Psize + i) = Gc(Psize + i) + 1$; |
| 26. | end if |
| 27. | If $Gc(Psize + i) > Max_gc$ |
| 28. | RandomWalk($Qcells(i)$, $Gc(Psize + i)$); |
| 29. | end if |
| 30. | end for |

(5) **Algorithm 6** (RandomWalk()) performs the random walk of any tumor cell. When a tumor cell moves by a random walk, the growth cycle ' Gc ' was set to zero. It means that the random walk will stop, and a normal moving will begin.

3.3. Discussion of the ITGO algorithm

In order to explain the mechanism of ITGO algorithm, we discuss four aspects about our algorithm. Firstly, we discuss the feature of Levy flight. One dimension of the step size for a proliferative cell and a quiescent cell was test. This can help us to understand the mechanism of search by Levy flight. Secondly, we discuss the search trajectory of a proliferative cell, a quiescent cell and a dying cell. Here, we finished a simple experiment with Fun 4 in Table 1. This helps us to understand the search mechanism of our algorithm. Thirdly, we discuss the exploration and exploitation of the different search strategies for the different tumor cells. Fourth, we analyze the convergence of our algorithm.

3.3.1. Analysis of Levy flight in tumor

Levy flight is a Markov process, and walking step satisfies a heavy tailed (heavy-tailed) Levy distribution. In Levy flight, local search of short walking distance and occasionally global search of longer walking distance occurs, so some solutions in the near optimum value can search and enhance the local search ability. On the

Algorithm 4DyingCellGrowth (*Pcells*, *Psize*, *Qcells*, *Qsize*, *Dcells*, *Dsize*, *Gc*, *Fes*, *Max_Gc*).

Input: current proliferative cells subpopulation (*cPcells*), number of proliferative cells (*Psize* = 20% *Np*), quiescent cells subpopulation (*QCells*), number of quiescent cells (*Qsize* = 60% *Np*), Dying cells subpopulation (*Dcells*), number of dying cells (*Dsize* = 20% *Np*), growth cycle of tumor cell (*Gc*), number of fitness evaluation consumed (*Fes*), max number of growth cycle (*Max_gc*).
Output: updated *Dcells*

```

1.   for i = 1:Dsize%Dsize is the number of Dcells
2.     Select the u cell randomly from Pcells subpopulation as Pcells(u);
3.     Select the v cell randomly from Qcells subpopulation as Qcells(v);
4.     for j = 1:D
5.       now Dcell(i,j) = Dcells(i,j) + y * (cPcells(u,j) – Dcells(i,j)) + y * (Qcells(v,j) – Dcells(i,j));
6.       //Formula (12)
7.     end for
8.     Calculate fitness evaluation of now Dcell(i);
9.     Fes = Fes + 1;
10.    If now Ncell(i) has more Nutrient (better Fitness value)
11.      Dcell(i) = now Dcell(i);
12.    else
13.      Gc(Psize + Qsize + i) = Gc(Psize + Qsize + i) + 1;
14.    end if
15.    If Gc(Psize + Qsize + i) > Max_gc
16.      Random Walk (Dcells(i), Gc(Psize + Qsize + i));
17.    end if
18.  end for

```

Algorithm 5InvasiveCellGrowth (*Pcells*, *Psize*, *Dcells*, *Dsize*, *Fes*).

Input: current proliferative cells subpopulation (*cPcells*), number of proliferative cells (*Psize* = 20% *Np*), Dying cells subpopulation (*Dcells*), number of dying cells (*Dsize* = 20% *Np*), number of fitness evaluation consumed (*Fes*).
Output: updated *Dcells*

```

1.   for i = 1:Dsize//Dsize is the number of Dcells
2.     Select the u cell randomly from cPcells subpopulation as cPcell(u);
3.     Calculate average nutrient concentration (fitness value) of all cells in Dcells subpopulation as M;
4.     If the Nutrient concentration of Dcells(i) is not better than M;
5.       Now cell(i) = rand(1,D); //Formula (13)
6.       for j = 1:D
7.         Icell(i,j) = cPcell(i,j) + rand * (Nowcell(1,j) - cPcell(i,j)); //Formula (14)
8.       end for
9.       Calculate fitness evaluation of Icell(i);
10.      Fes = Fes + 1;
11.      If Icell(i) has more Nutrient (better fitness value)
12.        Dcell(i) = Icell(i); //Formula (15)
13.      end if
14.    end for

```

Algorithm 6RandomWalk (*cell*, *Gc*, *Fes*).

Input: any *cell* in the three tumor cells subpopulations, the growth cycle of one cell (*Gc*), number of fitness evaluation consumed (*Fes*).
Output: *cell*, *Gc*

```

1.   Nowcell = rand(1,D); //Formula (13)
2.   newCell = cell + a * Nowcell / (Nowcell * Nowcell); //Formula (16)
3.   Calculate fitness evaluation of newCell;
4.   Fes = Fes + 1;
5.   If the Nutrient concentration of newCell is better than cell;
6.     cell = newCell;
7.     Gc = 0;
8.   end if

```

other hand, some solutions can obtain the global optimal value with far enough searching space, so as to ensure that the algorithm will not fall into a local optimum. As shown in Fig. 3, *x* coordinate is the numbers of fitness evaluations (FEs, 100,000), *y* coordinate is the search step value of one-dimensional Levy flight. It can be seen in Fig. 3 that Levy flight has a jumping search step length. Fig. 3(a) shows the search step of proliferative cell, when the iteration number (Fes numbers) increases, the control step size grows linearly (seen as Eq. (3)), so the Levy step size is gradually increased, which reflects the stronger invasive behavior of proliferative cells in the latter. We know that at the later search of our algorithm, all the solutions (the position of proliferative cells) are accumulated at the local optimum. To jump it out, Levy step size must be increased for the proliferative cell in a large search space. Fig. 3(b) shows the searching step of quiescent cells, using the control step size as formula (11); it can be seen that the control step size of quiescent cell is relatively steady. This conforms to the growth mechanism of invasive tumor growth in reality. In the whole search process of our algorithm, quiescent cells were guided by the proliferative cells, we use the same levy step size, that is to say, local search of short walking distance and occasionally global search of longer walking distance occurs in the whole search process for quiescent cells, which achieved the balance of local search and global search.

3.3.2. Analysis of invasive tumor growth algorithm

Here, we present the evolution process of our algorithm. Fig. 4(a) is a shifted unimodal function with Noise. In 2-dimension space, ITGO algorithm was ran for 186 fitness evaluation numbers, which obtains the global optimum value –450, the global optimal solution is *X* = 35.6267, *y* = –82.9124. As mentioned above, we know that the population was divided into the main three subpopulations: proliferative cell subpopulation, quiescent cell subpopulation and dying cell subpopulation. Different cell has different behavior corresponding to different search pattern. Fig. 4(b) presents the searching trajectory of a proliferative cell. All the positions of proliferative cells located in the better solution spaces (better solution set). So they can obtain better solution. The levy flight behaviors help them to jump out the local optimum easy. Fig. 4(c) presents the searching trajectory of a quiescent cell. All the positions of quiescent cells located in the general solution spaces (general solution set). These cells search in relatively wide solution space, and they can obtain better solution guided by the proliferative cell, which helps them obtain the global optimum solution. Fig. 4(d) presents the searching trajectory of a dying cell. All the positions of dying cells located in the large solution spaces (large solution set), because these cells have lower nutrient concentration (fitness value). The search in a large solution space increases the diversity of our algorithms. A dying cell obtains global optimal solution guided by a proliferative cell and a quiescent cell.

3.3.3. Exploration and exploitation

To better explain our algorithm, we discuss more aspects about the exploration and exploitation. In the search process of the proliferative cell, *Levy(s)* presents the search in different regions and control size of step *α* achieves the balance of exploration and exploitation. In the early search stage, when $\frac{Fes}{Max_Fes} \leq 0.5$, *α* obtains a more small value, each proliferative cell visits those regions of a search space within the neighborhood of previously visited points, it can be seen as an exploitation, whilst in the late search stage, when $\frac{Fes}{Max_Fes} > 0.5$, *α* perhaps obtains a more large value, each proliferative cell visits entirely new regions of a search space, it can be seen as an exploration. In the search process of quiescent cells, we consider the balance of exploration and exploitation too. As quiescent cell population is the main body of the tumor, many cells should search including exploration and exploitation. In formula

Table 1

Details of CEC2005 and CEC2008.

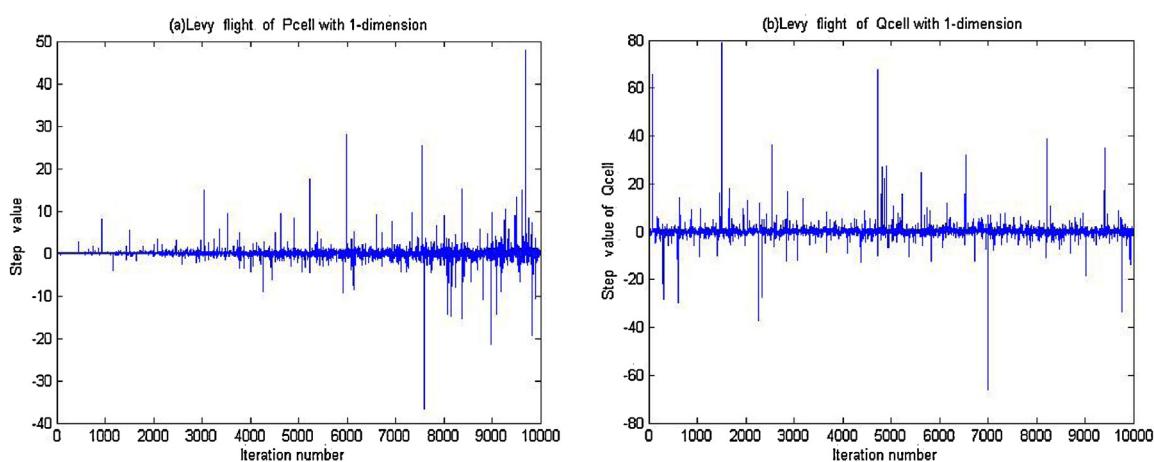
| Fun | Benchmark problem | Type low | Low | Up | Dim | Optimum solution |
|-----|---|----------|------|-----|-----|------------------|
| 1 | Shifted sphere | U | -100 | 100 | 10 | -450 |
| 2 | Shifted Schwefel's problem 1.2 | U | -100 | 100 | 10 | -450 |
| 3 | Shifted rotated high conditioned elliptic function | U | -100 | 100 | 10 | -450 |
| 4 | Shifted Schwefel's problem 1.2 with noise | U | -100 | 100 | 10 | -450 |
| 5 | Schwefel's problem 2.6 | U | -100 | 100 | 10 | -310 |
| 6 | Shifted Rosenbrock | M | -100 | 100 | 10 | 390 |
| 7 | Shifted rotated Griewank's | M | 0 | 600 | 10 | -180 |
| 8 | Shifted rotated Ackley | M | -32 | 32 | 10 | -140 |
| 9 | Shifted Rastrigin | M | -5 | 5 | 10 | -330 |
| 10 | Shifted rotated Rastrigin | M | -5 | 5 | 10 | -330 |
| 11 | Shifted rotated Weierstrass | M | -0.5 | 0.5 | 10 | 90 |
| 12 | Schwefel's problem 2.13 | M | -100 | 100 | 10 | -460 |
| 13 | Expanded extended Griewank's + Rosenbrock's | M | -3 | 1 | 10 | -130 |
| 14 | Expanded rotated extended Scaffer's F6 | M | -100 | 100 | 10 | -300 |
| 15 | Hybrid composition function 1 | M | -5 | 5 | 10 | 120 |
| 16 | Rotated hybrid comp. Fn 1 | M | -5 | 5 | 10 | 120 |
| 17 | Rotated hybrid comp. Fn 1 with noise | M | -5 | 5 | 10 | 120 |
| 18 | Rotated hybrid comp. Fn 2 | M | -5 | 5 | 10 | 10 |
| 19 | Rotated hybrid comp. Fn 2 with narrow global optimal | M | -5 | 5 | 10 | 10 |
| 20 | Rotated hybrid comp. Fn 2 with the global optimum | M | -5 | 5 | 10 | 10 |
| 21 | Rotated hybrid comp. Fn 3 | M | -5 | 5 | 10 | 360 |
| 22 | Rotated hybrid comp. Fn 3 with high condition number matrix | M | -5 | 5 | 10 | 360 |
| 23 | Non-continuous rotated hybrid comp. Fn 3 | M | -5 | 5 | 10 | 360 |
| 24 | Rotated hybrid comp. Fn 4 | M | -5 | 5 | 10 | 260 |
| 25 | Rotated hybrid comp. Fn 4 | M | -5 | 5 | 10 | 260 |
| 26 | Shifted sphere function | U | -100 | 100 | 100 | -450 |
| 27 | Shifted Schwefel's problem 2.21 | U | -100 | 100 | 100 | -450 |
| 28 | Shifted Rosenbrock's function | M | -100 | 100 | 100 | 390 |
| 29 | Shifted Griewank's function | M | -600 | 600 | 100 | -180 |
| 30 | Shifted Ackley's function | M | -32 | 32 | 100 | -140 |

(9), we can see that each quiescent cell is guided by the historical best position of proliferative cell $hPcell_{pj}(t)$ and the current position of proliferative cell $cPcell_{pj}(t)$ alternately, which means that, each quiescent cell searches in a large region to find a best position, and it can be seen as an exploration, whilst the two neighbor's operation ensures that the quiescent cell visits the neighbor regions, which can be seen as an exploitation. In the search process of the dying cells, formula (12) means that each dying cell is guided by the proliferative cell and the quiescent cell, and it can be seen as an exploration. Because each dying cell (each agent) visits new regions according the position of the proliferative cell and the quiescent cell. Two leaders (proliferative cell and the quiescent cell) were chosen in different populations randomly, which can enhance the diversity of our algorithm. In the search process of invasive cells, an exploration is achieved. Formulae (13) and (14) mean that an invasive cell visits a more larger search region nearby the proliferative cell. Especially, for the multimodal problems, in the late search process, many proliferative cells may obtain different local

optimums. So the invasive cell ensures that they can jump out the local optimum, and find the best position, that is, they can enhance the exploration ability of our algorithm. Formula (15) means that some bad solution (dying cell) can be replaced by the best solution (invasive cell). In addition, the random walk can be seen as an exploitation. $\frac{newCell_{i,j}(t)}{\|newCell_{i,1:D}(t)\|}$ means a small value in the search region. So the random walk can be seen as a small fluctuation for all the tumor cells, that is, each tumor cell can visit its neighbor region. Due to this mechanism of exploration and exploitation, our algorithm has better search ability.

3.3.4. Convergence analysis

Our proposed algorithm is a population-based optimization algorithm, in which the state of the tumor cell population will transfer to a new state with different probability, so we can use a Markov chain model [68] for the theoretical analysis. The mathematical description is as follows:

**Fig. 3.** Levy flight process.

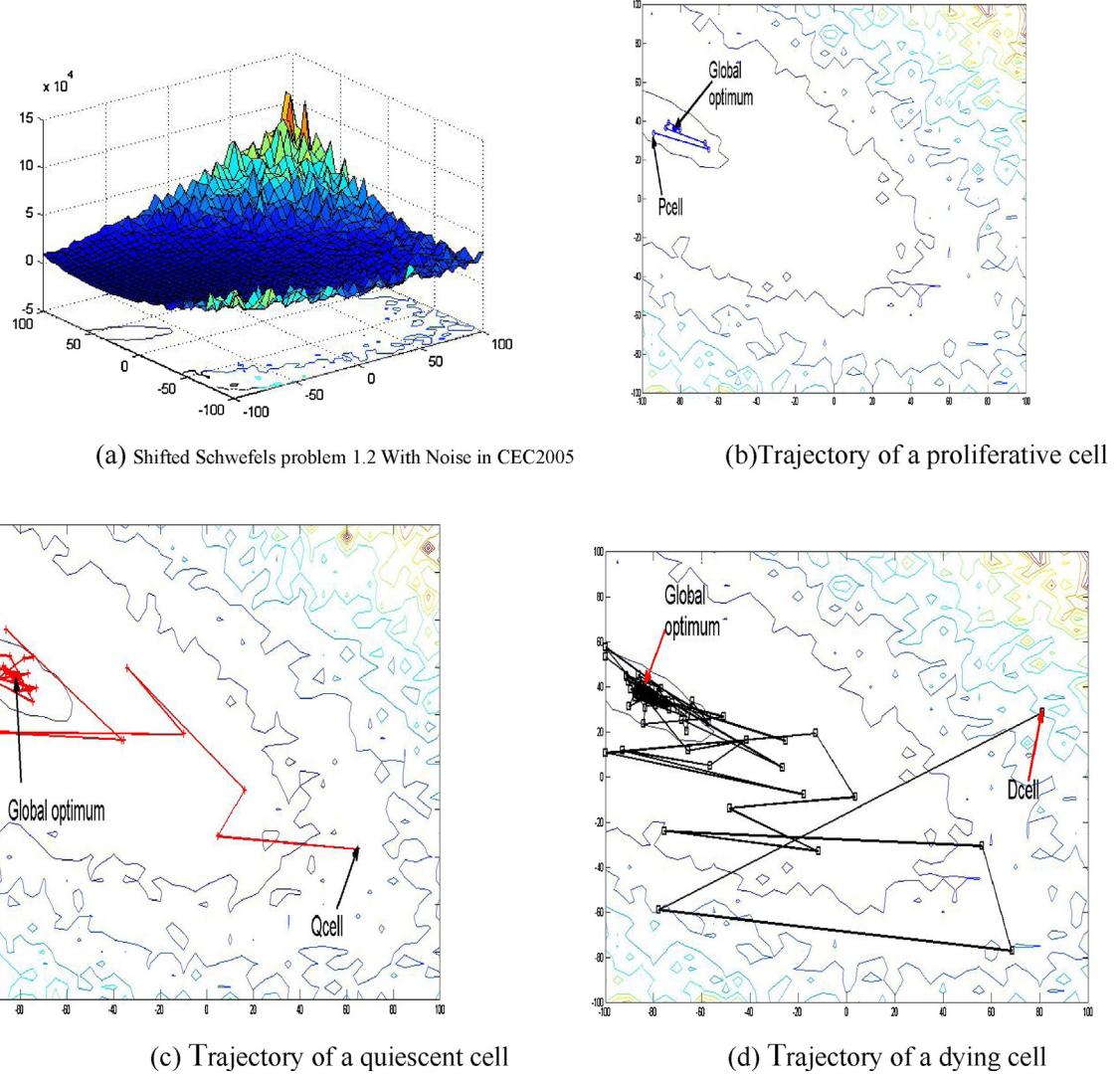


Fig. 4. Evolution process of ITGO algorithm.

Definition 1. Each individual state is decided by the position of each tumor cell, that is, $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, D is the dimension.

All the state of each individual constitutes the individual state space: $X = \{X, X \in \Omega\}$, Ω is all the feasible solution set.

Definition 2. All the individual's present state constitutes the present tumor cell population state space, that is,

$$Y = \{X_1, X_2, \dots, X_i, \dots, X_{NP}\}, NP$$
 is the population size.

All the state of tumor cells population is described as

$$Y = \{y = \{X_1, X_2, \dots, X_i, \dots, X_{NP}\}\}, X_i \in X, 1 \leq i \leq NP.$$

Definition 3. One step transition probability for individuals is

$$P(T_y(X_i) = X_j),$$

Here, $T_y(X_i) = X_j$ means that the state of individual transfers from X_i to X_j .

Definition 4. One step transition probability for tumor cell populations is

$$P(T_Y(y_i) = y_j)$$

Here, $T_Y(y_i) = y_j$ means that the state of tumor cell population transfers from y_i to y_j .

Probability distribution is:

$$P(T_Y(y_i) = y_j) = \prod_{n=1}^{NP} P(T_y(X_{in}) = X_{jn}).$$

Definition 5 ((Convergence in Probability [68,69])). Let $\{X(t), t = 0, 1, 2, \dots\}$ be a population sequence generated by a population-based stochastic algorithm, the stochastic sequence $\{X(t)\}$ weakly converges in probability to the global optimum, if and only if:

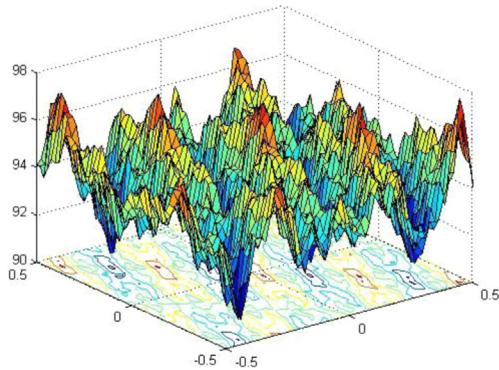
$$\lim_{t \rightarrow \infty} P\{X(t) \cap G^* \neq \emptyset\} = 1$$

where G^* is the set of the global optima of an optimization problem. That is, the algorithm holds with convergence in probability.

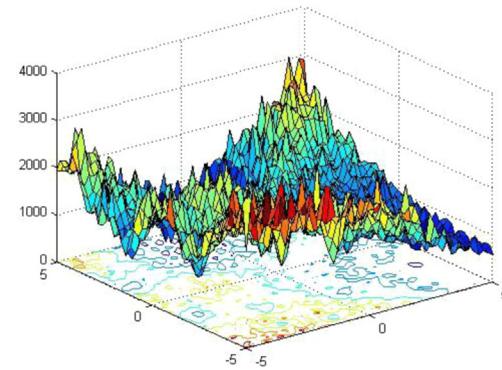
According to the definitions above, we can discuss the convergence of our proposed algorithm. In the search space, each individual search in a finite dimension and each tumor cell population state space is composed of all the finite individuals. We can also see that one step transition probability of the tumor cell populations $P(T_y(y_i) = y_j)$ is dependent on the one step transition probability of all the individuals. In addition, one step transition

probability of one individual $P(T_y(X_i) = X_j)$ is only related to the X_i at the i time. We can infer that it is a finite homogeneous Markov chain model. Especially, the search steps of our proposed algorithm are dependent on the five operations for different tumor cell. All the search steps of our proposed algorithm cannot search the whole search space except the invasive cell and the random walk. Because new individuals can be generated in the global search space for the invasive cell growth and the random walk of the tumor cells, we can infer that our proposed algorithm can converge in probability to the global optimum, that is, $\lim_{t \rightarrow \infty} P\{X(t) \cap G^* \neq \emptyset\} = 1$. The proof of our proposed algorithm can be a new subject in future.

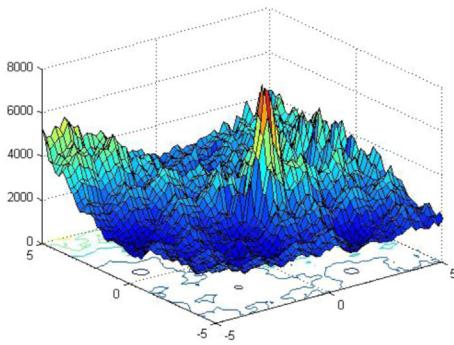
4. Experimental results and analysis



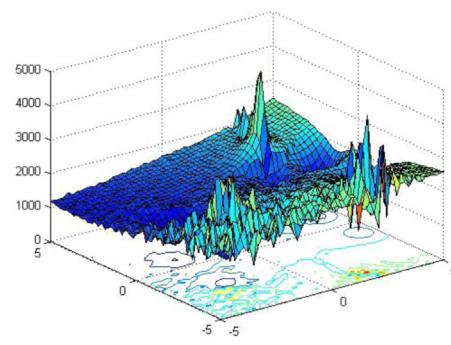
F8 .Shifted rotated Weierstrass



F12 Rotated hybrid comp. Fn 1 With Noise



F13 Rotated hybrid comp. Fn 3 with high condition number matrix



F15 Rotated hybrid comp. Fn 4 without Bounds

In between these two extreme cases, there are various kinds of partially separable problems. Matter of fact, real-world optimization problems will most likely consist of different groups of parameters with strong dependencies within but little interaction between the groups. In CEC2010, four types of high-dimensional problems were designed to reflect the real problem, that is: separable functions; partially separable functions, in which a small number of variables are dependent while all the remaining ones are independent; partially separable functions consist of multiple independent subcomponents, each of which is m -nonseparable; and fully nonseparable functions. F1–F3 are separable functions, F4–F18 are partially separable functions and F19–F20 are fully nonseparable functions. Details are shown in Table 2.

For these algorithms, parameter settings are shown in Table 3. The tests of all algorithms are rooted on a uniform standard:

- (1) For CEC2005, dimension is 10, population size is 30, and MAX.FES (total fitness evaluations number)=1e5 [70].
- (2) For CEC2008, dimension is 100, population size is 30, and MAX.FES=5e5 [71].
- (3) For CEC2010, dimension is 1000, population size is 100, and MAX.FES=3e5 and 3e6 [72].

For purposes of avoiding any negative effects of the structure of the initial population during the tests, the benchmark problems were solved 25 times by using a different initial population each time. If the absolute error obtained by the algorithm is smaller

In order to verify the performance of the proposed algorithm, we compare ITGO to the seven well-known algorithms (PSO, DE, ABC, BBO, GSA, CS and TLBO) and other improved algorithms (cPSO, QIGSA and rcGA). We choose the 25 Benchmark functions from CEC 2005 [70]. In these functions, F1–F5 are unimodal functions and F6–F25 are multimodal functions. Especially, F6–F12 are basic functions and F13–F25 are expanded functions. The global optimums of F7 and F25 functions are outside of the initialization range. To solve the large scale global optimization problem, we choose the five benchmark functions from CEC2008 [71] and 20 benchmark functions from CEC2010 [72]. In CEC 2008, F1 and F2 are unimodal functions. F3, F5, F6 are Multimodal Functions. Details are shown in Tables 1 and 2.

In general, separable problems are considered to be easy, while the fully nonseparable ones usually are the most difficult problems.

Table 2
Details of CEC 2010.

| Fn | Benchmark problem | Type | Low | Up | Dim | Optimum solution |
|----|--|------|------|-----|------|------------------|
| 1 | Shifted elliptic function | U | -100 | 100 | 1000 | 0 |
| 2 | Shifted Rastrigin's function | M | -5 | 5 | 1000 | 0 |
| 3 | Shifted Ackley's function | M | -32 | 32 | 1000 | 0 |
| 4 | Single-group shifted and m-rotated elliptic function | U | -100 | 100 | 1000 | 0 |
| 5 | Single-group shifted and m-rotated Rastrigin's function | M | -5 | 5 | 1000 | 0 |
| 6 | Single-group shifted and m-rotated Ackley's function | M | -32 | 32 | 1000 | 0 |
| 7 | Single-group shifted m-dimensional Schwefel's problem 1.2 | U | -100 | 100 | 1000 | 0 |
| 8 | Single-group shifted m-dimensional Rosenbrock's function | M | -100 | 100 | 1000 | 0 |
| 9 | $\frac{D}{m}$ group shifted and m-rotated elliptic function | M | -100 | 100 | 1000 | 0 |
| 10 | $\frac{D}{m}$ group shifted and m-rotated Rastrigin's function | M | -5 | 5 | 1000 | 0 |
| 11 | $\frac{D}{m}$ group shifted and m-rotated Ackley's function | M | -32 | 32 | 1000 | 0 |
| 12 | $\frac{D}{m}$ group shifted m-dimensional Schwefel's problem 1.2 | U | -100 | 100 | 1000 | 0 |
| 13 | $\frac{D}{m}$ group shifted m-dimensional Rosenbrock's function | M | -100 | 100 | 1000 | 0 |
| 14 | $\frac{D}{m}$ group shifted and m-rotated Elliptic function | U | -100 | 100 | 1000 | 0 |
| 15 | $\frac{D}{m}$ group shifted and m-rotated Rastrigin's function | M | -5 | 5 | 1000 | 0 |
| 16 | $\frac{D}{m}$ group shifted and m-rotated Ackley's function | M | -32 | 32 | 1000 | 0 |
| 17 | $\frac{D}{m}$ group shifted m-dimensional Schwefel's problem 1.2 | U | -100 | 100 | 1000 | 0 |
| 18 | $\frac{D}{m}$ group shifted m-dimensional Rosenbrock's function | M | -100 | 100 | 1000 | 0 |
| 19 | Shifted Schwefel's problem 1.2 | U | -100 | 100 | 1000 | 0 |
| 20 | Shifted Rosenbrock's function | M | -100 | 100 | 1000 | 0 |

than $\text{err} = 1e-15$ then stop. The data such as the global minimum value, maximum, mean, standard deviation and median obtained as a result of the tests have been saved for detailed statistical analysis. All the tests were implemented by using Matlab 7.8. Statistical analyses were realized by SPSS software.

The Wilcoxon signed-rank test [82–84] and the Friedman's test [84] were used to compare ITGO with other algorithms. In the statistical tests conducted for this paper, Bonferroni–Holm correction method is based on the correction of p -value [84]. When using Wilcoxon's test in our experimental study, the first step is to compute the R_+ and R_- related to the comparisons between ITGO and the rest of algorithms. Let R_+ be the sum of ranks for the problems in which the first algorithm outperformed the second, and R_- the sum of ranks for the opposite. Once they have been obtained, their associated p -values can be computed. The null hypothesis H_0 was used for the Wilcoxon Signed-Rank Tests for purposes of this paper. The statistical significance value used to test H_0 hypothesis is $\tau = 0.05$. If in any test a p -value that is smaller than or equal to significance level τ value is produced, then the H_0 hypothesis for that test is rejected and the alternative hypothesis is selected.

In order to compare the performance of all the algorithms, Friedman's test was used. It is a multiple comparisons test that aims to detect significant differences between the behaviors of two or more algorithms. The first step in calculating the test statistic is to convert the original results to ranks. At the end, average ranks are used to comparison.

The experiments were divided into four groups: (1) experiment on basic global optimization; (2) comparison with the improved other algorithms; (3) experiment on large scale global optimization compared with classical algorithms; (4) experiment on large scale global optimization compared with improved PSO. In experiments (1) and (2), we use the 25 benchmark functions from CEC 2005 and 5 benchmark functions from CEC 2008 to test for PSO, DE, BBO, GSA, TLBO, CLPSO, cPSO, QIGSA, rcGA and ITGO. In these benchmark functions, it maybe unimodal or Multimodal or Non-separable or separable or Shifted or Rotated, and it is not easy to obtain the global optimum of these functions, shown as Fig. 7. In experiment (3), in order to solve the Large Scale Global Optimization, we choose the 20 benchmark functions from CEC2010 to test the performance of CLPSO, DE, ABC, BBO, GSA, CS, TLBO and ITGO (MAX_FES = 3e5). In these functions, it may be Separable or Partially separable or Fully nonseparable functions; In experiment (4), for the higher dimensional 20 benchmark functions, MAX_FES = 3e6, we compare ITGO with the CLPSO, GOPSO [72], DNSCLPSO [72] and DNSPSO [72].

4.1. Parameter tuning

Two parameters were used in ITGO, namely beta, and Max_gc. Nevertheless, a complete evaluation on all possible combinations of these parameters is impractical. The parameter tuning strategy, as reported in [85,86], is used to find an appropriate parameter combination of ITGO for the good performances. In order to find

Table 3
Parameter setting of algorithms.

| No. | Algorithm | Parameter setting |
|-----|---------------|--|
| 1 | PSO | $c_1 = c_2 = 2$ and inertia factor (w) is decreasing linearly from 0.9 to 0.2 [19] |
| 2 | DE/best/2/bin | $F = 0.5$, $CR = 0.9$ [73] |
| 3 | DE/2/exp | $F = 0.5$, $CR = 0.9$ [73] |
| 4 | GbABC | $SN = 12$, $If = 1.12$, $C = 1.507$, tuned GbABC according to the paper [74] |
| 5 | TLBO [32,75] | No parameter, duplicate elimination step was not included |
| 6 | BBO | pmodify = 1, PMutate = 0.01, and elitism parameter = 2 as in the Matlab code given in [28] |
| 7 | GSA | Rpower = 2, Rnorm = 2, ElitistCheck = 1 as in [34] |
| 8 | GWO [2] | – |
| 9 | CS | $b = 1.50$, $p_0 = 0.25$ as in [76]. Matlab Code provided by author Xin-She Yang in [29] |
| 10 | CLPSO [77] | $c = 1.49335$, $m = 0$, $p_c = 0.5 \cdot (e^t - e^{t(1)})/(e^{t(p_s)} - e^{t(1)})$, $t = 0.5 \cdot (0 : \frac{1}{p_s-1} : 1)$ as in the Matlab code given in [78] |
| 11 | cPSO | $V_p = 300$, $\varphi_1 = -0.2$, $\varphi_2 = -0.007$, $\varphi_3 = 3.74$, $\gamma_1 = 1$, $\gamma_2 = 1$ as in [79] |
| 12 | QIGSA | $SS = 200$, $g = 1.3$, as in [80] |
| 13 | rcGA | $V_p = 300$ [81] |
| 14 | ITGO | $\betaeta = 1.1$, $Max_gc = 0.7 * \text{Dimension}$. βeta is the parameter of Levy flight; Max_gc is the cycle size of the cell |

Table 4

Mean fitness evaluation values for different Beta values. Bold font means the best mean fitness evaluation value.

| ITGO | Order | Frandman test (mean rank value) | F2 | F9 | F12 | F16 | F19 | F22 |
|-----------|----------|------------------------------------|-------------------|-------------------|----------------|-----------------|-----------------|-----------------|
| Beta=0.3 | 14 | 12.17 | 4.6652e-4 | 0.1694 | 165.8918 | 118.5228 | 661.6615 | 674.1582 |
| Beta=0.4 | 11 | 10.83 | 1.9976e-04 | 0.0823 | 129.1490 | 119.2651 | 565.3326 | 695.7637 |
| Beta=0.55 | 12 | 11.00 | 6.6015e-05 | 4.7447e-05 | 118.8651 | 122.5309 | 632.4498 | 725.1910 |
| Beta=0.6 | 10 | 10.50 | 5.3372e-05 | 2.8736e-07 | 162.2354 | 117.0856 | 621.9661 | 707.3697 |
| Beta=0.7 | 5 | 8.17 | 4.3568e-05 | 4.5084e-05 | 124.6896 | 112.9711 | 654.2683 | 707.3697 |
| Beta=0.8 | 10 | 10.50 | 1.9296e-05 | 6.7919e-07 | 143.0816 | 117.1556 | 701.1381 | 709.5440 |
| Beta=0.9 | 13 | 11.50 | 1.8424e-05 | 0.0031 | 111.7124 | 127.1517 | 672.2489 | 725.3870 |
| Beta=1.0 | 4 | 8.00 | 1.2685e-05 | 0.0498 | 85.2803 | 114.9324 | 659.1948 | 698.3361 |
| Beta=1.1 | 1 | 5.33 | 9.5703e-06 | 4.1483e-08 | 79.6823 | 113.5747 | 739.1236 | 630.9221 |
| Beta=1.2 | 6 | 8.33 | 9.1044e-06 | 2.6560e-05 | 100.3773 | 117.0066 | 707.6682 | 700.1438 |
| Beta=1.3 | 6 | 8.33 | 7.8374e-06 | 4.2394e-07 | 104.0764 | 114.7072 | 730.6909 | 727.9724 |
| Beta=1.4 | 5 | 8.17 | 6.0058e-06 | 1.6402e-04 | 80.1614 | 114.3240 | 756.4484 | 722.3373 |
| Beta=1.5 | 9 | 9.67 | 6.6581e-06 | 0.0585 | 73.1952 | 115.4082 | 748.8755 | 746.2393 |
| Beta=1.6 | 2 | 6.33 | 2.8927e-06 | 0.1497 | 50.8136 | 111.9165 | 765.3370 | 686.3084 |
| Beta=1.7 | 3 | 7.00 | 4.1623e-06 | 0.2815 | 62.2841 | 111.4671 | 858.6937 | 652.0878 |
| Beta=1.8 | 7 | 8.50 | 3.6273e-06 | 1.2114 | 59.0461 | 117.04 | 776.0147 | 654.2346 |
| Beta=1.9 | 8 | 8.67 | 8.9698e-06 | 2.9518 | 43.5387 | 115.3828 | 886.6828 | 651.0057 |

the better parameter setting, we choose the six problems with different characteristics, that is, one unimodal function (F2) and five multimodal functions (F9, F12, F16, F19 and F22). Especially, F9 is a shifted function, F16, F19 and F22 are rotated hybrid functions. Parameter beta ranges in [0.3,1.9] and MAX_gc ranges in dimension*[0.1,1.1] (dimension=10). We first initialize the parameter combination of [beta, MAX_gc] as the mean values of their respective lower and upper boundary values, i.e., [1.1, 0.6 * 10], beta and MAX_gc are integers. Based on this initial combination, we adjust the parameters one at a time. We first vary parameter beta from 0.3 to 1.9 with MAX_gc are being fixed. Then we choose the best beta value due to the best fitness evaluation value. In the next step, we fix the beta value and adjust MAX_gc from 1 to 11. ITGO algorithm with each different parameter setting is ran 20 times for the six functions. Mean fitness evaluation values are recorded in [Tables 4 and 5](#). [Figs. 5 and 6](#) show the curve of mean fitness evaluation values for each function. Firstly, we discuss the influence of parameter Beta for the performance of ITGO. From [Fig. 5](#) and [Table 4](#), we can see that when Beta values are 1.6, 1.1, 1.9, 1.7, 0.4 and 1.1, F2, F9, F12, F16, F19 and F22 functions obtain the best fitness evaluation value. It means that for most problems (F2, F9, F16, F22), Beta value should be set as a middle value. For especial problem, Beta value can be set as a more big or small value (F19 and F12). In order to choose the best Beta parameter value, Frandman test is used here. The three columns of [Table 4](#) show the mean rank value of Frandman test. The two columns of [Table 4](#) show the order of ITGO for the different Beta values. Black font shows the best one. We can see that when Beta = 1.1, the performance of ITGO is the best. Secondly, we discuss the influence of parameter Max_gc for the performance of ITGO. From [Fig. 6](#) and [Table 5](#), we can see that, when MAX_gc values are 3, 9, 1, 7, 3 and 5, F2, F9, F12, F16, F19 and

F22 functions obtain the best fitness evaluation value. It means that MAX_gc should not be set as too large value (F2, F12, F16, F19 and F22) except F9. It can be explained according to the formula [\(16\)](#). We know that MAX_gc is the cycle of growth for each tumor cell and it is a disturbance for the tumor cells. If MAX_gc is set too large, the random walk operator is less to be used, and it is not good to help the ITGO leave from the local optimum. Then, we use the Frandman tests to choose the best MAX_gc value. From [Table 5](#), we can see that, when MAX_gc = 7, mean rank value is best and the order of ITGO is 1. At the end, the experimental findings suggest that we can set the parameter combination [beat, MAX_gc] as [1.1, 0.7 * 10] in the performance evaluation.

4.1.1. Experiments on basic global optimization

In this experiment, [Table 6](#) shows the minimum, maximum, means, standard deviation and the median of the fitness evaluation values after 25 times runs for PSO, DE/best/2/bin, BBO, GSA, TLBO, GWO and ITGO algorithms. The black font in [Table 6](#) indicates the winner. In [Table 7](#), we use the Wilcoxon signed-rank test by the means in [Table 6](#) to compare the PSO, DE/best/2/bin, BBO, GSA, TLBO, GWO and ITGO algorithm. The comparison results among ITGO and PSO, DE/best/2/bin, BBO, GSA, TLBO, GWO algorithms are summarized as in the last two rows of [Table 7](#), which means that ITGO wins in w functions, ties in t functions and loses in l functions, compared with its competitors. 'ITGO' shown in [Table 7](#) means that ITGO is the winner and blank in grid unit of [Table 7](#) means that ITGO is the loser. The 'ITGO' with black font shows the comparison results of ITGO and PSO, DE/best/2/bin, BBO, GSA, TLBO, GWO algorithms with the means. The statistical results show that the performance of the ITGO algorithm outperforms other algorithms.

Table 5

Mean fitness evaluation values for different MAX_gc values. Bold font means the best mean fitness evaluation value.

| ITGO | Order | Frandman test (mean rank value) | F2 | F9 | F12 | F16 | F19 | F22 |
|-------------------|----------|------------------------------------|--------------------|-------------------|----------------|-----------------|-----------------|-----------------|
| Max.gc = 0.1 * 10 | 6 | 6.00 | 1.2308 e-05 | 0.8222 | 34.8015 | 116.7309 | 650.1895 | 676.7296 |
| Max.gc = 0.2 * 10 | 5 | 5.83 | 1.0448 e-05 | 0.0801 | 50.9338 | 112.6588 | 755.6286 | 673.3788 |
| Max.gc = 0.3 * 10 | 2 | 5.00 | 7.3810 e-06 | 0.0015 | 61.6157 | 117.3754 | 585.8494 | 705.2133 |
| Max.gc = 0.4 * 10 | 4 | 5.67 | 1.1377e-05 | 0.0498 | 61.5047 | 114.3881 | 650.8025 | 700.4475 |
| Max.gc = 0.5 * 10 | 3 | 5.50 | 1.2452e-05 | 4.9316e-04 | 79.4621 | 113.4392 | 718.6826 | 629.2496 |
| Max.gc = 0.6 * 10 | 5 | 5.83 | 8.5656e-06 | 0.0017 | 88.2924 | 112.5883 | 755.0617 | 665.0431 |
| Max.gc = 0.7 * 10 | 1 | 4.00 | 9.6999e-06 | 4.6930e-06 | 84.5745 | 109.8842 | 709.1705 | 648.6243 |
| Max.gc = 0.8 * 10 | 7 | 6.83 | 1.1731e-05 | 3.1365e-08 | 85.3898 | 122.3725 | 630.0155 | 727.0747 |
| Max.gc = 0.9 * 10 | 5 | 5.83 | 1.0176e-05 | 1.1047e-08 | 88.0885 | 115.1158 | 744.3821 | 700.4267 |
| Max.gc = 1.0 * 10 | 9 | 8.17 | 1.3259e-05 | 6.6664e-07 | 150.8997 | 118.7674 | 678.4408 | 719.8443 |
| Max.gc = 1.1 * 10 | 8 | 7.33 | 1.3429e-05 | 1.7278e-08 | 82.6406 | 116.3355 | 716.1840 | 748.5158 |

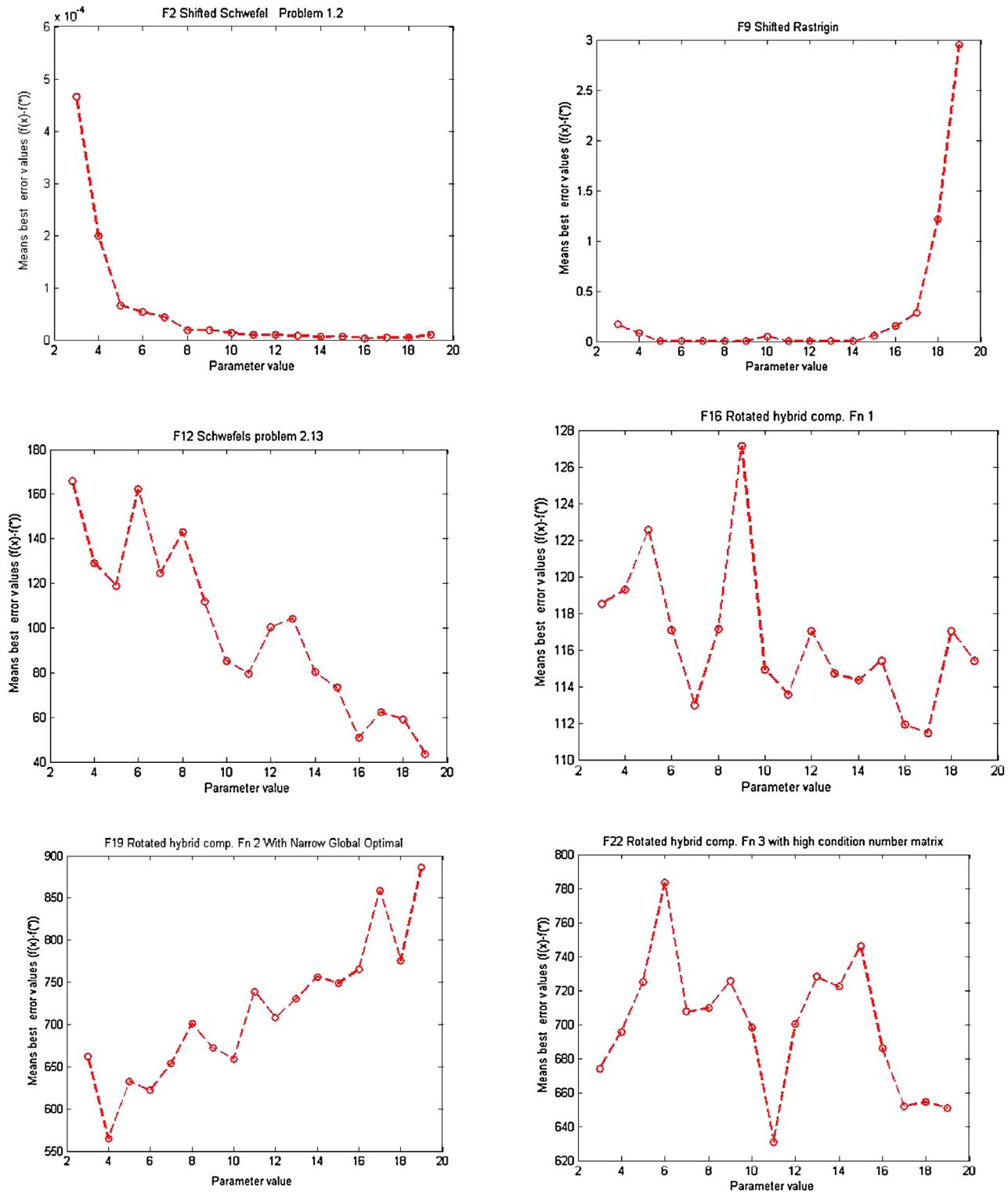


Fig. 5. Curve of mean fitness evaluation values for different Beta values.

In Table 8, we use the Wilcoxon Signed-Rank Test and Bonferroni-Holm correction method to compare ITGO with PSO, DE/best/2/bin, BBO, GSA, TLBO and GWO. The Bonferroni-Holm procedure adjusts the value of α in a step-down manner. Let p_1, p_2, \dots, p_{k-1} be the ordered p -values (from smallest to largest), so that $p_1 \leq p_2 \leq \dots \leq p_{k-1}$, and let H_1, H_2, \dots, H_{k-1} be the corresponding hypotheses. The Bonferroni-Holm procedure

rejects H_1 to H_{i-1} if i is the smallest integer such that $p_i > \alpha/(k-1)$. In Tables 9 and 10, we use the Friedman's test. In order to achieve better comparison, the means data set was selected from Table 6 to be analyzed by Wilcoxon Signed-Rank Test, Friedman's test and Bonferroni-Holm correction method. As for the Wilcoxon Signed-Rank Test and Bonferroni-Holm correction method, the comparison results show that the null hypothesis was rejected, that

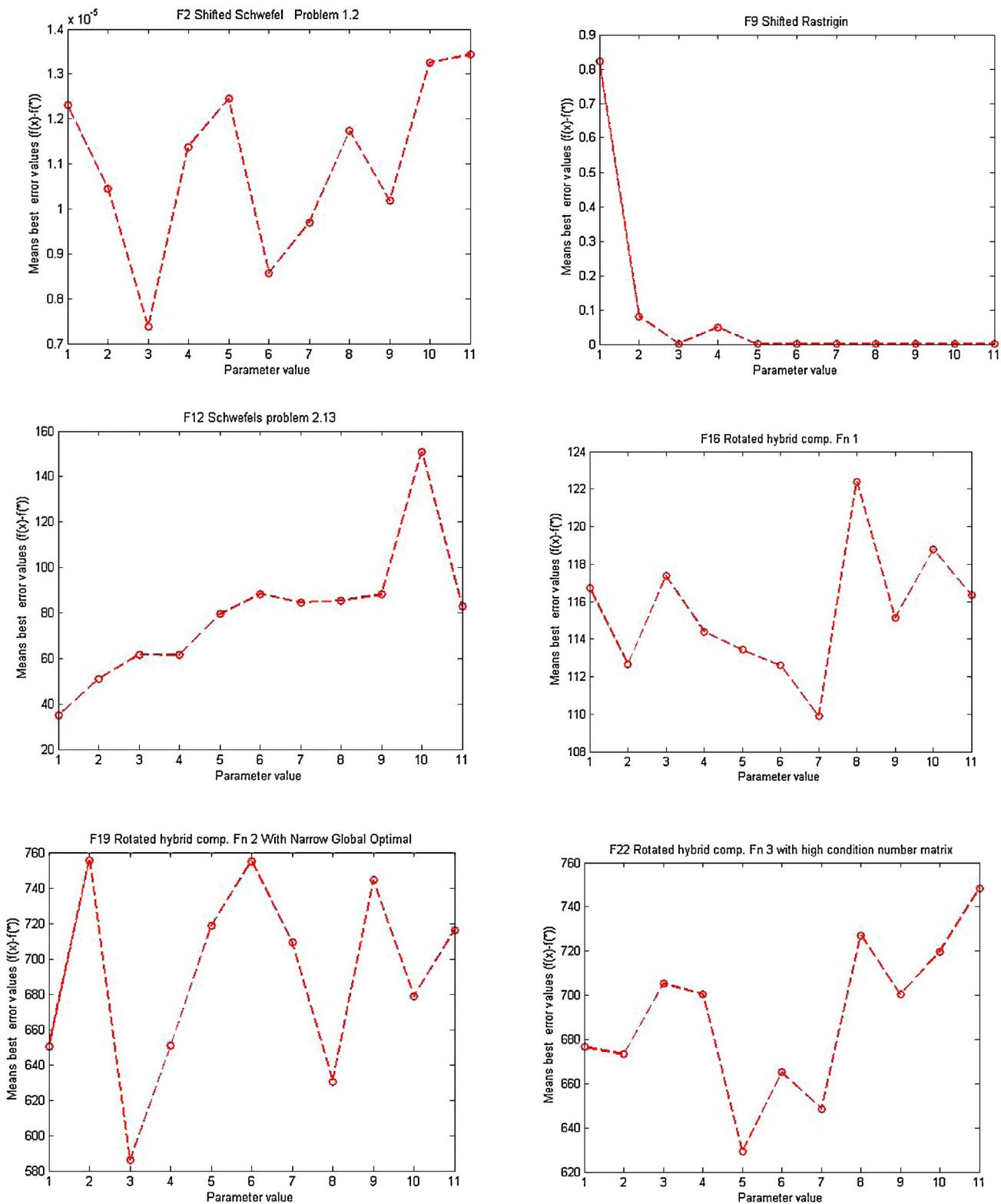


Fig. 6. Curve of mean fitness evaluation values for different MAX.gc values.

is to say, the alternative hypothesis was accepted and the performance of ITGO is better than the others. As for the Friedman's test, the average ranks show that ITGO is better than the others.

Fig. 7 is the convergence curves of the 7 algorithms for the 10 benchmark functions. These 10 functions are multimodal functions. Especially, F13, F15, F18, F19, F20, F22, F23, F24 and F25

are expanded functions, while F12 are basic functions. For multimodal problems, the performance of ITGO is better than PSO, DE/best/2/bin, BBO, GSA, TLBO, GWO. For F13, F18, F19, F20, F24 and F25, the performance of TLBO is better than PSO, DE/best/2/bin, BBO, GSA, GWO. For F12, the performance of BBO is better than PSO, DE/best/2/bin, GSA, GWO and TLBO. For F15 and F22, the

Table 6

The results of PSO, DE/best/2/bin, BBO, GSA, TLBO, GWO and ITGO for CEC 2005 and CEC2008. Bold font means the best fitness evaluation value.

| Fun | PSO | DE/best/2/bin | BBO | GSA | TLBO | GWO | ITGO | |
|-----|-----|---|--|--|--|---|--|--|
| (a) | F1 | Min 6.4591e+000 Max 3.8833e+002 Means 5.7997e+001 SD 8.0374e+001 Median 2.4941e+001 | 3.0518e-009 9.6141e-009 6.4641e-009 1.5174e-009 6.5183e-009 | 1.5375e-002 4.5698e-001 8.6583e-002 9.3137e-002 6.3415e-002 | 7.7266e+003 2.7422e+004 1.7987e+004 5.0912e+003 1.8110e+004 | 4.2552e-009 9.9755e-009 7.7228e-009 1.3808e-009 7.6904e-009 | 1.1498e+000 7.4077e+002 1.0597e+002 1.9230e+002 1.4569e+001 | 5.6843e-014 5.6843e-013 1.5689e-013 1.3854e-013 1.1369e-013 |
| | F2 | Min 1.4502e+001 Max 2.1126e+002 Means 5.5605e+001 SD 5.3670e+001 Median 2.3541e+001 | 3.6197e-009 9.9446e-009 7.6674e-009 1.6416e-009 7.5828e-009 | 1.8380e+000 3.0322e+001 1.0638e+001 7.1644e+000 9.1617e+000 | 1.0900e+004 3.2988e+004 2.3332e+004 5.8819e+003 2.3012e+004 | 5.3422e-009 9.6346e-009 2.7391e+005 1.1086e+005 8.5191e-009 | 7.0155e+000 1.5743e+002 3.6158e+006 9.7686e+005 5.0965e+001 | 1.4188e-006 3.6235e-005 1.0066e-005 8.6954e-006 7.2833e-006 |
| | F3 | Min 1.7264e+005 Max 1.1753e+006 Means 3.6225e+005 SD 2.1583e+005 Median 3.1649e+005 | 2.1496e-004 1.2585e+002 1.4351e+001 3.4876e+001 1.2132e+000 | 4.3885e+004 3.3051e+006 1.3871e+006 8.5520e+005 1.3333e+006 | 4.5977e+006 8.8090e+008 2.9925e+008 2.0339e+008 2.4863e+008 | 2.2737e+003 2.7391e+005 1.1086e+005 7.4994e+004 9.6490e+004 | 1.7387e+005 3.6158e+006 9.7686e+005 9.1945e+005 5.8014e+005 | 1.2546e+004 3.3150e+005 9.8286e+004 8.2820e+004 7.3113e+004 |
| | F4 | Min 1.8848e+001 Max 4.0460e+002 Means 6.9620e+001 SD 8.1818e+001 Median 4.3178e+001 | 2.1750e-009 5.5894e+001 2.2358e+000 1.1179e+001 8.4592e-009 | 2.8604e+002 1.9103e+003 9.3255e+002 4.8451e+002 8.7248e+002 | 1.1830e+004 3.3621e+004 2.2058e+004 5.8528e+003 2.0151e+004 | 6.5751e-009 9.8784e-009 8.6137e-009 1.0468e-009 8.9166e-009 | 8.4371e+000 1.2625e+002 5.5514e+001 3.7737e+001 4.0136e+001 | 6.6939e-006 1.2688e-002 8.8885e-004 2.5484e-003 2.5207e-004 |
| | F5 | Min 3.0833e+000 Max 4.4155e+003 Means 1.8249e+002 SD 8.8188e+002 Median 5.4948e+000 | 4.7476e-009 9.1422e-009 7.1903e-009 1.2878e-009 7.5743e-009 | 4.7480e+001 5.2773e+002 2.1879e+002 1.4074e+002 1.9423e+002 | 1.5194e+004 2.4332e+004 2.0352e+004 2.4306e+003 2.0591e+004 | 4.8021e-009 9.6006e-009 8.2485e-009 1.1329e-009 8.5674e-009 | 9.2147e+000 8.3721e+002 3.5985e+002 2.5815e+002 4.3597e+002 | 0 1.8695e-002 3.7306e-003 5.6552e-003 4.7087e-004 |
| (b) | F6 | Min 2.5869e+003 Max 4.4332e+004 Means 9.1520e+003 SD 9.3045e+003 Median 6.9988e+003 | 4.3678e-009 3.9866e+000 1.1162e+000 1.8269e+000 8.1349e-009 | 1.0480e+001 2.5730e+002 9.0019e+001 6.3094e+001 7.9840e+001 | 2.9112e+008 2.5231e+009 9.0197e+008 5.3149e+008 7.6882e+008 | 7.5693e-009 1.1098e+000 9.6592e-002 2.9067e-001 4.5887e-008 | 0 1.0324e+002 7.6650e+000 7.5591e+000 2.7804e-006 | 2.1268e-003 4.0145e+001 7.5591e+000 8.8272e+000 5.5128e+000 |
| | F7 | Min 2.6353e+003 Max 4.4970e+003 Means 3.5962e+003 SD 4.6087e+002 Median 3.6825e+003 | 1.2670e+003 1.2673e+003 1.2671e+003 1.0479e-001 1.2670e+003 | 1.2681e+003 1.2715e+003 1.2697e+003 8.6803e-001 1.2697e+003 | 2.7938e+003 4.2457e+003 3.5899e+003 3.4202e+002 3.6037e+003 | 1.2670e+003 1.2672e+003 1.2671e+003 3.6006e-002 1.2670e+003 | 1.2670e+003 1.2678e+003 1.2677e+003 2.8624e-001 1.2678e+003 | 1.2670e+003 1.2670e+003 1.2670e+003 4.2790e-013 1.2670e+000 |
| | F8 | Min 2.0129e+001 Max 2.0472e+001 Means 2.0347e+001 SD 8.6290e-002 Median 2.0341e+001 | 2.0216e+001 2.0461e+001 2.0346e+001 6.4751e-002 2.0357e+001 | 2.0184e+001 2.0490e+001 2.0349e+001 8.1648e-002 2.0361e+001 | 2.0003e+001 2.0084e+001 2.0037e+001 2.4662e-002 2.0029e+001 | 2.0145e+001 2.0512e+001 2.0347e+001 8.1152e-002 2.0357e+001 | 2.0232e+001 2.0457e+001 2.0373e+001 6.0273e-002 2.0385e+001 | 2.0135e+001 2.0365e+001 2.0240e+001 5.2215e-002 2.0239e+001 |
| | F9 | Min 1.0535e+001 Max 4.0368e+001 Means 2.2126e+001 SD 7.3691e+000 Median 2.2359e+001 | 1.9899e+000 1.1939e+001 5.9698e+000 2.6167e+000 4.9748e+000 | 3.7103e-009 9.9927e-009 7.6095e-009 1.7099e-009 7.9018e-009 | 9.9496e-001 8.9546e+000 4.4972e+000 1.6529e+000 4.9748e+000 | 2.9849e+000 1.7909e+001 8.8258e+000 3.2832e+000 8.9546e+000 | 3.4863e+000 2.6234e+001 1.3951e+001 6.6540e+000 1.3012e+001 | 5.6275e-012 2.9889e-005 1.3202e-006 5.9764e-006 4.5270e-009 |
| | F10 | Min 1.0739e+001 Max 3.1626e+001 Means 2.4897e+001 SD 4.6228e+000 Median 2.5956e+001 | 5.9697e+000 3.2984e+001 1.6751e+001 7.2853e+000 1.5919e+001 | 8.2902e+000 3.2943e+001 1.9723e+001 6.2341e+000 1.8840e+001 | 9.9496e-001 7.9597e+000 4.7758e+000 2.1301e+000 4.9748e+000 | 3.7979e+000 3.1625e+001 1.2418e+001 6.7207e+000 9.9498e+000 | 6.1623e+000 3.6995e+001 2.1673e+001 9.9525e+000 2.2650e+001 | 2.9849e+000 4.3778e+001 1.3929e+001 8.7778e+000 1.0945e+001 |
| (b) | F11 | Min 2.8981e+000 Max 6.2518e+000 Means 4.1754e+000 SD 1.0422e+000 Median 4.1193e+000 | 2.0041e-001 8.6315e+000 3.6237e+000 1.9866e+000 3.3611e+000 | 2781e+000 9.0348e+000 7.9404e+000 7.2357e-001 8.0140e+000 | 3.9732e-002 6.9540e-002 5.4078e-002 7.4615e-003 5.4650e-002 | 2.5775e+000 8.0098e+000 5.1605e+000 1.4171e+000 5.0184e+000 | 2.5318e+000 1.0364e+001 5.6073e+000 2.4527e+000 5.3414e+000 | 1.2260e+000 5.3970e+000 3.1096e+000 1.2573e+000 3.0102e+000 |
| | F12 | Min 1.6676e+004 Max 4.3105e+004 Means 2.8302e+004 SD 7.1476e+003 Median 4.1193e+000 | 1.5066e+002 2.3375e+004 9.3353e+003 6.8889e+003 3.3611e+000 | 1.9393e+001 6.0291e+003 7.8047e+002 1.3200e+003 1.9752e+002 | 2.5824e+004 1.3872e+005 9.1739e+004 2.8149e+004 8.9767e+004 | 1.0402e+004 4.0839e+004 2.3192e+004 7.8028e+003 2.1245e+004 | 2.1213e+004 4.1668e+004 2.4800e+004 7.3247e+003 2.3652e+004 | 2.8603e+001 1.8991e+002 8.3562e+001 4.3167e+001 7.7623e+001 |
| | F13 | Min 1.8526e+000 Max 3.3813e+000 Means 2.5973e+000 SD 3.8598e-001 Median 2.5980e+000 | 3.4924e-001 2.5807e+000 1.3667e+000 6.7189e-001 1.3414e+000 | 1.0622e+000 1.8261e+000 1.3495e+000 2.6117e-001 1.3607e+000 | 7.4708e-001 1.9640e+000 1.3614e+000 3.4717e-001 1.3013e+000 | 1.4975e-001 1.0089e+000 6.1755e-001 2.3051e-001 5.7969e-001 | 1.1253e-001 2.4508e+000 1.4377e+000 6.6850e-001 1.5721e+000 | 1.8864e-001 5.3970e+000 3.1096e+000 1.2573e+000 3.0102e+000 |
| | F14 | Min 2.7153e+000 Max 4.0152e+000 Means 3.1446e+000 SD 3.2858e-001 Median 3.1929e+000 | 2.1405e+000 3.4988e+000 2.6764e+000 3.6004e-001 2.6299e+000 | 2.6001e+000 3.8690e+000 3.3300e+000 2.8946e-001 3.3883e+000 | 4.0244e+000 4.5418e+000 4.3326e+000 1.4385e-001 4.3613e+000 | 1.7224e+000 3.3121e+000 2.7634e+000 4.4496e-001 2.8318e+000 | 2.2399e+000 3.6176e+000 2.8544e+000 3.6738e-001 2.7716e+000 | 1.8824e+000 3.5173e+000 2.8188e+000 3.4327e-001 2.8580e+000 |
| (b) | F15 | Min 1.1376e+002 Max 8.0254e+002 Means 3.5338e+002 SD 1.9769e+002 Median 2.6460e+002 | 7.3459e+001 9.0000e+002 3.8012e+002 2.0408e+002 4.2836e+002 | 7.9840e+000 4.4454e+002 2.0922e+002 2.0404e+002 3.6996e+002 | 4.1509e+001 4.0984e+002 2.9980e+002 1.6157e+002 1.1932e+002 | 9.9627e-009 5.9240e+002 3.5551e+002 7.5560e+001 4.0984e+002 | 5.7560e+001 7.0290e+002 3.5551e+002 5.4143e-012 3.3965e+002 | 5.4143e-012 4.2817e+002 1.1662e+002 1.8197e+002 4.0185e-007 |

Table 6 (Continued)

| Fun | | PSO | DE/best/2/bin | BBO | GSA | TLBO | GWO | ITGO |
|-----|--------|--------------------|--------------------|-------------|--------------------|--------------------|--------------------|--------------------|
| F16 | Min | 1.1828e+002 | 9.9266e+001 | 1.1762e+002 | 3.1597e-007 | 9.7060e+001 | 1.0582e+002 | 9.9776e+001 |
| | Max | 6.3268e+002 | 8.0000e+002 | 1.9647e+002 | 1.2983e+002 | 1.8465e+002 | 2.0180e+002 | 1.5747e+002 |
| | Means | 1.7158e+002 | 1.8022e+002 | 1.5063e+002 | 8.7659e+001 | 1.1889e+002 | 1.5032e+002 | 1.1782e+002 |
| | SD | 9.7699e+001 | 1.5467e+002 | 2.1871e+001 | 2.6008e+001 | 1.9867e+001 | 2.5854e+001 | 1.2943e+001 |
| | Median | 1.5567e+002 | 1.3264e+002 | 1.4559e+002 | 9.3473e+001 | 1.1550e+002 | 1.4727e+002 | 1.1626e+002 |
| (c) | Min | 1.3565e+002 | 1.0337e+002 | 1.5301e+002 | 6.4711e-007 | 9.9138e+001 | 9.8665e+001 | 9.5703e+001 |
| | Max | 2.4865e+002 | 6.2553e+002 | 2.2283e+002 | 1.2344e+002 | 1.3840e+002 | 2.1315e+002 | 1.5719e+002 |
| | Means | 1.7373e+002 | 1.6179e+002 | 1.9812e+002 | 8.9968e+001 | 1.1933e+002 | 1.6004e+002 | 1.2149e+002 |
| | SD | 2.6511e+001 | 1.1635e+002 | 1.7787e+001 | 3.0030e+001 | 1.1411e+001 | 3.3656e+001 | 1.5277e+001 |
| | Median | 1.6891e+002 | 1.2475e+002 | 1.9986e+002 | 1.0012e+002 | 1.2118e+002 | 1.6699e+002 | 1.1831e+002 |
| F18 | Min | 6.8934e+002 | 6.0156e+002 | 3.1870e+002 | 8.0000e+002 | 4.4952e+002 | 3.9945e+002 | 3.0000e+002 |
| | Max | 1.0796e+003 | 1.0388e+003 | 1.0265e+003 | 1.0058e+003 | 9.9050e+002 | 1.0858e+003 | 1.0233e+003 |
| | Means | 9.3038e+002 | 8.7413e+002 | 7.9553e+002 | 8.7472e+002 | 7.8031e+002 | 8.5925e+002 | 7.2973e+002 |
| | SD | 1.1831e+002 | 1.4599e+002 | 2.5507e+002 | 8.0829e+001 | 1.7979e+002 | 1.5340e+002 | 2.3971e+002 |
| | Median | 9.6723e+002 | 9.0256e+002 | 9.3108e+002 | 8.0000e+002 | 8.0007e+002 | 8.8235e+002 | 8.0001e+002 |
| F19 | Min | 6.8877e+002 | 3.0000e+002 | 3.1870e+002 | 8.0000e+002 | 3.0000e+002 | 3.2188e+002 | 3.0000e+002 |
| | Max | 1.0368e+003 | 1.0356e+003 | 1.0485e+003 | 1.0078e+003 | 9.9664e+002 | 1.0281e+003 | 9.8209e+002 |
| | Means | 9.2018e+002 | 8.7642e+002 | 8.9694e+002 | 9.1700e+002 | 7.4648e+002 | 8.3228e+002 | 6.8678e+002 |
| | SD | 9.1917e+001 | 1.6361e+002 | 1.9799e+002 | 6.9790e+001 | 1.9529e+002 | 1.7779e+002 | 2.7300e+002 |
| | Median | 9.6363e+002 | 8.8657e+002 | 9.7287e+002 | 9.3938e+002 | 8.0000e+002 | 8.7223e+002 | 8.0002e+002 |
| F20 | Min | 6.9935e+002 | 6.2237e+002 | 5.3110e+002 | 8.0000e+002 | 3.4736e+002 | 3.1850e+002 | 3.0000e+002 |
| | Max | 1.0532e+003 | 1.0475e+003 | 1.0624e+003 | 1.0183e+003 | 1.0586e+003 | 1.0907e+003 | 9.9253e+002 |
| | Means | 9.3647e+002 | 8.9489e+002 | 9.3072e+002 | 9.0925e+002 | 7.7655e+002 | 7.6262e+002 | 7.4049e+002 |
| | SD | 1.1977e+002 | 1.3827e+002 | 1.2471e+002 | 7.9997e+001 | 2.0027e+002 | 2.3460e+002 | 2.4383e+002 |
| | Median | 9.7687e+002 | 9.5606e+002 | 9.7314e+002 | 9.3940e+002 | 8.0000e+002 | 8.0546e+002 | 8.0002e+002 |
| F21 | Min | 3.6195e+002 | 3.0000e+002 | 3.1793e+002 | 8.0000e+002 | 3.0000e+002 | 3.0196e+002 | 3.0000e+002 |
| | Max | 1.2822e+003 | 1.2247e+003 | 1.1712e+003 | 8.0000e+002 | 1.2401e+003 | 1.2371e+003 | 1.1708e+003 |
| | Means | 9.9518e+002 | 8.5335e+002 | 7.3202e+002 | 8.0000e+002 | 6.4136e+002 | 8.0845e+002 | 7.6831e+002 |
| | SD | 3.0444e+002 | 3.1972e+002 | 3.2175e+002 | 5.7990e-007 | 3.6061e+002 | 3.9555e+002 | 3.1146e+002 |
| | Median | 1.0887e+003 | 9.4841e+002 | 5.0428e+002 | 8.0000e+002 | 5.0000e+002 | 9.1042e+002 | 8.6215e+002 |
| F22 | Min | 7.7316e+002 | 7.3690e+002 | 7.6587e+002 | 5.6667e+002 | 7.4132e+002 | 3.4938e+002 | 3.0000e+002 |
| | Max | 9.7478e+002 | 1.0000e+003 | 8.9117e+002 | 8.0000e+002 | 8.6242e+002 | 9.6582e+002 | 8.7141e+002 |
| | Means | 8.5338e+002 | 8.4363e+002 | 8.0868e+002 | 7.3502e+002 | 7.7437e+002 | 8.0476e+002 | 6.6933e+002 |
| | SD | 6.3028e+001 | 8.2622e+001 | 3.7150e+001 | 5.5346e+001 | 3.2884e+001 | 1.0822e+002 | 2.1400e+002 |
| | Median | 8.4877e+002 | 8.2798e+002 | 8.0064e+002 | 7.3450e+002 | 7.5922e+002 | 8.1794e+002 | 7.5793e+002 |
| F23 | Min | 4.2517e+002 | 5.5947e+002 | 5.5997e+002 | 9.7050e+002 | 5.5947e+002 | 5.5947e+002 | 5.5947e+002 |
| | Max | 1.2667e+003 | 1.2929e+003 | 1.2707e+003 | 9.7050e+002 | 1.2626e+003 | 1.3048e+003 | 1.2517e+003 |
| | Means | 9.4835e+002 | 1.0427e+003 | 9.3660e+002 | 9.7050e+002 | 9.4157e+002 | 9.0455e+002 | 8.5401e+002 |
| | SD | 3.0066e+002 | 2.8297e+002 | 2.8362e+002 | 0 | 2.4190e+002 | 2.6525e+002 | 2.2749e+002 |
| | Median | 9.7050e+002 | 1.2059e+003 | 1.0909e+003 | 9.7050e+002 | 9.5742e+002 | 7.5513e+002 | 7.2122e+002 |
| F24 | Min | 3.6714e+002 | 2.0000e+002 | 2.0513e+002 | 2.0000e+002 | 2.0000e+002 | 2.0099e+002 | 2.0000e+002 |
| | Max | 1.0180e+003 | 1.2785e+003 | 6.1946e+002 | 9.0000e+002 | 5.0000e+002 | 8.0272e+002 | 5.0000e+002 |
| | Means | 7.6950e+002 | 6.7780e+002 | 2.9233e+002 | 3.1600e+002 | 2.6000e+002 | 3.2945e+002 | 2.2400e+002 |
| | SD | 2.3155e+002 | 3.0950e+002 | 1.2790e+002 | 2.1346e+002 | 1.2247e+002 | 1.7795e+002 | 8.3066e+001 |
| | Median | 9.1383e+002 | 6.1388e+002 | 2.3756e+002 | 2.0000e+002 | 2.0000e+002 | 2.1330e+002 | 2.0000e+002 |
| (d) | Min | 1.0037e+003 | 8.1147e+002 | 8.3023e+002 | 1.2844e+003 | 8.1314e+002 | 6.0831e+002 | 8.1145e+002 |
| | Max | 1.3699e+003 | 1.3423e+003 | 9.6918e+002 | 1.3354e+003 | 1.3490e+003 | 1.3565e+003 | 8.2218e+002 |
| | Means | 1.1893e+003 | 9.3959e+002 | 8.8783e+002 | 1.3125e+003 | 8.6057e+002 | 8.8502e+002 | 8.1685e+002 |
| | SD | 1.5550e+002 | 1.8393e+002 | 3.5008e+001 | 1.1833e+001 | 1.4244e+002 | 1.6336e+002 | 2.3989e+000 |
| | Median | 1.2607e+003 | 8.3195e+002 | 8.8292e+002 | 1.3144e+003 | 8.1913e+002 | 8.3623e+002 | 8.1728e+002 |
| F26 | Min | 1.2142e+004 | 1.2948e+002 | 6.1313e+001 | 3.8265e+005 | 7.6773e-009 | 3.3103e+004 | 6.0719e-006 |
| | Max | 5.4072e+004 | 2.0295e+004 | 1.5292e+002 | 5.5340e+005 | 1.0821e+000 | 9.5508e+004 | 1.2649e-005 |
| | Means | 2.4172e+004 | 3.7187e+003 | 9.3691e+001 | 4.8282e+005 | 4.3749e-002 | 5.7671e+004 | 9.0554e-006 |
| | SD | 1.1174e+004 | 5.4111e+003 | 2.4590e+001 | 3.3457e+004 | 2.1632e-001 | 1.4269e+004 | 1.7484e-006 |
| | Median | 2.2219e+004 | 1.6254e+003 | 8.8541e+001 | 4.8365e+005 | 2.9933e-006 | 5.6133e+004 | 8.7628e-006 |
| F27 | Min | 2.3372e+001 | 6.0445e+001 | 1.5970e+001 | 1.4082e+002 | 7.9657e+001 | 6.0380e+001 | 7.6431e-001 |
| | Max | 8.3165e+001 | 1.0434e+002 | 2.0647e+001 | 1.5776e+002 | 9.2809e+001 | 7.5027e+001 | 2.2084e+000 |
| | Means | 4.4496e+001 | 8.0126e+001 | 1.8142e+001 | 1.5071e+002 | 8.7230e+001 | 6.5296e+001 | 1.5589e+000 |
| | SD | 1.3286e+001 | 1.2400e+001 | 1.0790e+000 | 5.2621e+000 | 3.8558e+000 | 3.1450e+000 | 3.5590e-001 |
| | Median | 4.3515e+001 | 7.8131e+001 | 1.7938e+001 | 1.5171e+002 | 8.7592e+001 | 6.5795e+001 | 1.6072e+000 |
| F28 | Min | 1.3374e+008 | 9.2588e+001 | 5.8441e+004 | 2.7647e+011 | 9.3162e+001 | 3.8759e+009 | 1.1396e+002 |
| | Max | 4.5604e+009 | 4.3010e+009 | 2.6374e+005 | 4.3054e+011 | 2.3800e+006 | 1.1692e+010 | 8.4859e+003 |
| | Means | 1.0845e+009 | 5.1707e+008 | 1.2253e+005 | 3.7139e+011 | 9.6352e+004 | 7.1097e+009 | 1.2672e+003 |
| | SD | 1.1886e+009 | 1.1055e+009 | 5.6626e+004 | 3.6833e+010 | 4.7577e+005 | 2.2383e+009 | 1.8831e+003 |
| | Median | 3.9835e+008 | 7.0057e+007 | 1.0867e+005 | 3.7308e+011 | 3.6014e+002 | 7.2525e+009 | 4.4501e+002 |
| F29 | Min | 8.2591e+001 | 1.2574e+000 | 1.6011e+000 | 4.0615e+003 | 1.6295e+003 | 2.1060e+002 | 1.2531e-007 |
| | Max | 3.4674e+002 | 1.3252e+002 | 2.2202e+000 | 5.2115e+003 | 1.6304e+003 | 7.2658e+002 | 1.4780e-002 |
| | Means | 1.9743e+002 | 1.9516e+001 | 1.8055e+000 | 4.6247e+003 | 1.6295e+003 | 4.7618e+002 | 3.3508e-003 |
| | SD | 6.2689e+001 | 2.8431e+001 | 1.7194e-001 | 2.5963e+002 | 1.7519e-001 | 1.1922e+002 | 5.6391e-003 |
| | Median | 2.0571e+002 | 8.2604e+000 | 1.7791e+000 | 4.6140e+003 | 1.6295e+003 | 4.9433e+002 | 2.6989e-007 |
| F30 | Min | 1.0031e+001 | 1.5628e+001 | 2.1197e+000 | 1.8998e+001 | 2.1069e+001 | 1.4315e+001 | 1.1623e-003 |
| | Max | 1.4024e+001 | 2.0655e+001 | 2.9134e+000 | 1.9332e+001 | 2.1353e+001 | 1.7071e+001 | 2.2327e-003 |
| | Means | 1.1578e+001 | 1.8129e+001 | 2.5120e+000 | 1.9184e+001 | 2.1301e+001 | 1.5833e+001 | 1.4823e-003 |
| | SD | 8.5236e-001 | 1.2147e+000 | 2.0988e-001 | 7.1797e-002 | 6.0001e-002 | 6.3984e-001 | 2.5047e-004 |
| | Median | 1.1605e+001 | 1.8225e+001 | 2.5620e+000 | 1.9205e+001 | 2.1316e+001 | 1.5870e+001 | 1.4142e-003 |

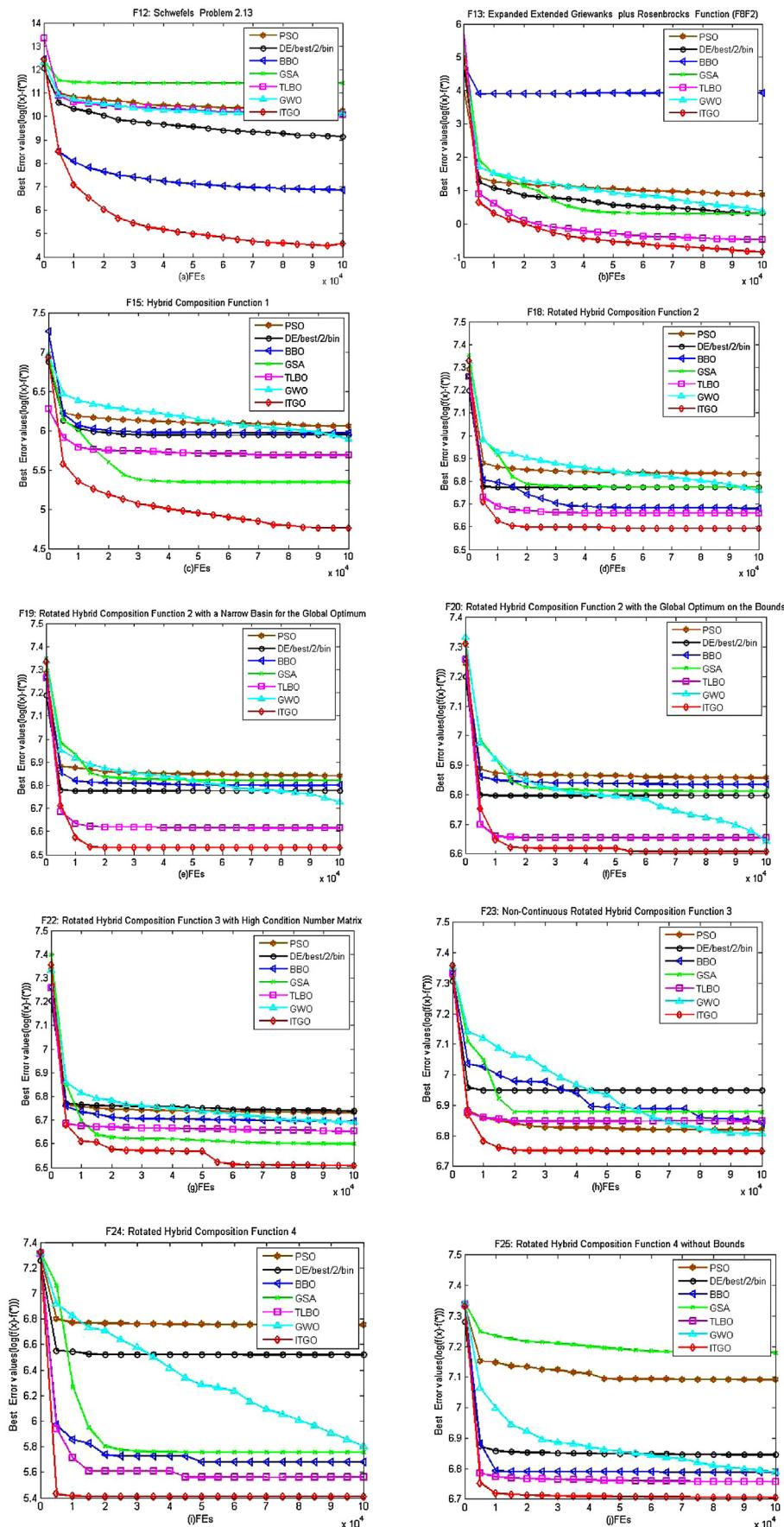


Fig. 7. The convergence curve.

Table 7

The results of compared PSO, DE/best/2/bin, BBO, GSA, TLBO, GWO with ITGO by Wilcoxon signed ranks test.

| | PSO vs ITGO | DE/best/2/bin vs ITGO | BBO vs ITGO | GSA vs ITGO | TLBO vs ITGO | GWO vs ITGO |
|--------------|-------------|-----------------------|-------------|-------------|--------------|-------------|
| F1 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F2 | ITGO | | ITGO | ITGO | | ITGO |
| F3 | ITGO | | ITGO | ITGO | ITGO | ITGO |
| F4 | ITGO | ITGO | ITGO | ITGO | | ITGO |
| F5 | ITGO | | ITGO | ITGO | | ITGO |
| F6 | ITGO | | ITGO | ITGO | | ITGO |
| F7 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F8 | ITGO | ITGO | ITGO | | ITGO | ITGO |
| F9 | ITGO | ITGO | | ITGO | ITGO | ITGO |
| F10 | ITGO | ITGO | ITGO | | | ITGO |
| F11 | ITGO | ITGO | ITGO | | ITGO | ITGO |
| F12 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F13 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F14 | ITGO | | ITGO | ITGO | | ITGO |
| F15 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F16 | ITGO | ITGO | ITGO | | ITGO | ITGO |
| F17 | ITGO | ITGO | ITGO | | | ITGO |
| F18 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F19 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F20 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F21 | ITGO | ITGO | | ITGO | | ITGO |
| F22 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F23 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F24 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F25 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F26 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F27 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F28 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F29 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| F30 | ITGO | ITGO | ITGO | ITGO | ITGO | ITGO |
| w/tl (means) | 30/0/0 | 25/0/5 | 28/0/2 | 25/0/5 | 22/0/8 | 30/0/0 |

Table 8

Comparison with ITGO and PSO, DE/best/2/bin, GSA, BBO, TLBO and GWO for CEC 2005 and CEC 2008 by using Wilcoxon signed-rank test and Bonferroni–Holm correction (means).

| i | Algorithm | R+ | R- | p-value | α/i | Hypothesis |
|---|-----------------------|-------|-------|---------|------------|------------|
| 5 | ITGO vs PSO | 15.50 | 0.00 | 1.73e–6 | 0.01 | Reject |
| 5 | ITGO vs GWO | 15.50 | 0.00 | 1.73e–6 | 0.01 | Reject |
| 4 | ITGO vs BBO | 16.07 | 7.50 | 7.69e–6 | 0.0125 | Reject |
| 3 | ITGO vs GSA | 17.48 | 5.60 | 2.59e–5 | 0.0167 | Reject |
| 2 | ITGO vs DE/best/2/bin | 16.52 | 10.10 | 2.05e–4 | 0.025 | Reject |
| 1 | ITGO vs TLBO | 17.59 | 9.75 | 0.0015 | 0.05 | Reject |

performance of GSA is better than PSO, DE/best/2/bin, BBO, TLBO and GWO. For F23, the performance of GWO is better than PSO, DE/best/2/bin, BBO, GSA and TLBO.

4.1.2. Comparison with the other improved algorithms

In this experiment, Table 11 shows the means and standard deviation of the error values after 25 times runs for cPSO, rcGA, QIGSA and ITGO algorithms. The black font in Table 11 presents the winner. We use the Wilcoxon signed-rank test for the means in Table 11 to compare the cPSO, rcGA and QIGSA with ITGO algorithm. The comparison results among ITGO and cPSO, QIGSA, rcGA algorithms are summarized as in the last rows of Table 11, which means that ITGO wins in **w** functions, ties in **t** functions

Table 9

Average ranks of 7 algorithms for CEC 2005 and CEC 2008 by Friedman test (means). Bold font means the best performance.

| Algorithm | Average ranks |
|---------------|---------------|
| ITGO | 1.67 |
| TLBO | 3.00 |
| DE/best/2/bin | 3.95 |
| BBO | 4.10 |
| GWO | 4.73 |
| GSA | 5.00 |
| PSO | 5.55 |

and loses in **l** functions, compared with its competitors. Table 12 shows the comparison results by Wilcoxon signed-rank test with Bonferroni–Holm correction method. For ITGO vs cPSO, ITGO vs rcGA and ITGO vs QIGSA, p-values are less than α/i , so all the H0 hypotheses were rejected. That is, the performance of ITGO is better than cPSO, rcGA and QIGSA significant. Tables 13 and 14, show the results by Friedman's test, which shows that ITGO is better than all the others.

Fig. 8 is the convergence curves of the 4 algorithms for the six benchmark functions. F3 is unimodal function and F6, F10, F14, F16 and F20 are multimodal functions. Especially, F14, F16 and F20 are expanded functions. Regardless of the problems being unimodal or multimodal problems, the performance of ITGO is better than cPSO, rcGA and QIGSA. For F3 and F14, the performance of cPSO is better than rcGA and QIGSA. For F6, F10 and F16, the performance of rcGA is better than cPSO and QIGSA. For F20, the performance of QIGSA is better than cPSO and rcGA.

Table 10

Results of Friedman's method for ITGO, TLBO, PSO, DE/best/2/bin, BBO, GWO and GSA.

| Method | Statistical value | p-value |
|---------------|-------------------|----------|
| Friedman test | 66.919 | 1.74e–12 |

Table 11

Comparison with ITGO and cPSO, QIGSA, rcGA for CEC 2005 and CEC 2008. Bold font means the best fitness evaluation values.

| Fun | cPSO | QIGSA | rcGA | ITGO |
|-------|----------------------------------|----------------------------------|---|---|
| F1 | 9.3289e−002 ± 6.1316e−002 | 5.9778e+002 ± 1.1447e+003 | 4.5929e−013 ± 1.0875e−013 | 1.5689e−013 ± 1.3854e−013 |
| F2 | 3.5034e+000 ± 1.688e+000 | 1.1040e+003 ± 1.009e+003 | 1.2023e−001 ± 5.2817e−001 | 1.0066e−005 ± 8.6954e−006 |
| F3 | 1.1159e+006 ± 8.3638e+005 | 7.2049e+006 ± 5.8635e+006 | 6.1970e+006 ± 4.7594e+006 | 9.8286e+004 ± 8.2820e+004 |
| F4 | 3.6461e+001 ± 4.777e+001 | 1.6546e+003 ± 2.0341e+003 | 9.8990e+001 ± 1.532e+002 | 8.8885e−004 ± 2.5484e−003 |
| F5 | 1.1149e+002 ± 4.635e+001 | 1.7583e+003 ± 2.8372e+003 | 1.2750e−004 ± 4.0708e−005 | 3.7306e−003 ± 5.6552e−003 |
| F6 | 2.3576e+002 ± 5.0793e+002 | 4.4017e+005 ± 3.7372e+005 | 1.6286e+002 ± 2.5776e+002 | 7.5591e+000 ± 8.8272e+000 |
| F7 | 1.2708e+003 ± 5.908e+000 | 1.2674e+003 ± 3.1785e−001 | 1.2670e+003 ± 1.7698e−005 | 1.2670e+000 ± 4.2790e−013 |
| F8 | 2.0324e+001 ± 5.4350e−002 | 2.0358e+001 ± 7.7946e−002 | 2.0168e+001 ± 9.9427e−002 | 2.0240e+001 ± 5.2215e−002 |
| F9 | 1.0769e+001 ± 5.664e+000 | 3.4633e+001 ± 1.0185e+001 | 1.8825e+001 ± 1.052e+001 | 1.3202e−006 ± 5.9764e−006 |
| F10 | 4.2113e+001 ± 1.6210e+001 | 5.1696e+001 ± 1.3140e+001 | 3.8250e+001 ± 1.3857e+001 | 1.3929e+001 ± 8.7778e+000 |
| F11 | 4.7772e+000 ± 1.273e+000 | 7.7592e+000 ± 1.3845e+000 | 5.4987e+000 ± 2.279e+000 | 3.1096e+000 ± 1.2573e+000 |
| F12 | 1.9342e+004 ± 7.354e+003 | 1.2101e+004 ± 6.7914e+003 | 6.7908e+002 ± 8.0078e+002 | 8.3562e+001 ± 4.3167e+001 |
| F13 | 1.3611e+000 ± 5.8489e−001 | 5.2962e+000 ± 1.0674e+000 | 3.1158e+000 ± 1.8852e+000 | 4.2341e−001 ± 1.3816e−00 |
| F14 | 3.2258e+000 ± 2.3524e−001 | 3.5461e+000 ± 4.1580e−001 | 3.6181e+000 ± 3.1397e−001 | 2.8188e+000 ± 3.4327e−001 |
| F15 | 2.9001e+002 ± 1.7745e+002 | 5.2912e+002 ± 1.2079e+002 | 3.7064e+002 ± 1.1495e+002 | 1.1662e+002 ± 1.8197e+002 |
| F16 | 1.6356e+002 ± 2.860e+001 | 2.3090e+002 ± 2.7962e+001 | 1.4610e+002 ± 2.8556e+001 | 1.1782e+002 ± 1.2943e+000 |
| F17 | 1.7620e+002 ± 3.935e+001 | 2.4087e+002 ± 1.9438e+001 | 1.7028e+002 ± 3.0606e+001 | 1.2149e+002 ± 1.5277e+001 |
| F18 | 8.4799e+002 ± 2.1316e+002 | 9.3516e+002 ± 1.2070e+002 | 9.3546e+002 ± 3.788e+002 | 7.2973e+002 ± 2.3971e+002 |
| F19 | 8.3469e+002 ± 2.2268e+002 | 9.1185e+002 ± 1.6495e+002 | 9.8161e+002 ± 7.8424e+001 | 6.8678e+002 ± 2.7300e+002 |
| F20 | 9.4020e+002 ± 1.5386e+002 | 9.0109e+002 ± 1.2940e+002 | 9.7909e+002 ± 7.6837e+001 | 7.4049e+002 ± 2.4383e+002 |
| F21 | 8.5908e+002 ± 3.4042e+002 | 1.0793e+003 ± 2.7205e+002 | 1.0711e+003 ± 1.9112e+002 | 7.6831e+002 ± 3.1146e+002 |
| F22 | 7.4935e+002 ± 1.373e+002 | 9.1273e+002 ± 6.8041e+001 | 8.1681e+002 ± 5.8666e+001 | 6.6933e+002 ± 2.1400e+002 |
| F23 | 9.3385e+002 ± 3.0771e+002 | 1.2262e+003 ± 1.2826e+002 | 1.2412e+003 ± 1.2722e+002 | 8.5401e+002 ± 2.2749e+002 |
| F24 | 2.7639e+002 ± 1.923e+002 | 9.6329e+002 ± 1.1232e+001 | 2.8800e+002 ± 1.763e+002 | 2.2400e+002 ± 8.3066e+001 |
| F25 | 8.8627e+002 ± 2.646e+001 | 1.0258e+003 ± 1.2372e+002 | 2.0633e+001 ± 5.187e+002 | 8.1685e+002 ± 2.3989e+000 |
| F26 | 3.6148e+003 ± 7.1106e+002 | 1.6702e+003 ± 3.5235e+002 | 3.2016e+002 ± 3.5235e+002 | 9.0554e−006 ± 1.7484e−006 |
| F27 | 7.2628e+001 ± 4.5268e+000 | 2.1202e+003 ± 3.5235e+002 | 1.1668e+002 ± 8.5111e+000 | 1.5589e+000 ± 3.5590e−001 |
| F28 | 4.4810e+005 ± 1.6810e+005 | 1.7302e+003 ± 3.5235e+002 | 5.4842e+007 ± 8.8087e+007 | 1.2672e+003 ± 1.8831e+003 |
| F29 | 2.0333e+001 ± 4.0792e+000 | 1.9102e+003 ± 3.5235e+002 | 6.6463e+000 ± 4.5434e+000 | 3.3508e−003 ± 5.6391e−003 |
| F30 | 1.9079e+001 ± 6.8580e−001 | 2.0502e+003 ± 3.5235e+002 | 1.9281e+001 ± 7.4620e−001 | 1.4823e−003 ± 2.5047e−004 |
| w/t/l | 30/0/0 | 30/0/0 | 26/1/3 | |

Table 12

Comparison with ITGO and rcGA, CLPSO, QIGSA and cPSO for CEC 2005 and CEC 2008 by using Wilcoxon signed-rank test and Bonferroni–Holm correction (means).

| i | Algorithm | R+ | R− | p-value | α/i | Hypothesis |
|---|---------------|-------|-------|---------|------------|------------|
| 2 | ITGO vs QIGSA | 15.50 | 0.00 | 1.73e−6 | 0.025 | Reject |
| 2 | ITGO vs cPSO | 15.50 | 0.00 | 1.73e−6 | 0.025 | Reject |
| 1 | ITGO vs rcGA | 15.50 | 10.67 | 6.04e−5 | 0.05 | Reject |

Table 13

Average ranks of 4 algorithms for CEC 2005 and CEC 2008 by Friedman test (means). Bold font means the best performance.

| Algorithm | Average ranks |
|-------------|---------------|
| ITGO | 1.12 |
| cPSO | 2.60 |
| rcGA | 2.65 |
| QIGSA | 3.63 |

4.1.3. Experiment on large scale global optimization compared with classical algorithms

In order to verify the capability of ITGO algorithm for solving large scale optimization problems, we use the 20 benchmark functions from CEC 2010 to compare ITGO with PSO, DE/best/2/bin, DE/2/exp, GbABC, BBO, GSA, TLBO, GWO, CLPSO and CS. Some of these functions may be singly separable or partially separable or fully nonseparable, which reflects the parameter optimization problem in reality. We set MAX_FES=3E5. The minimum, maximum, means, standard deviation and the median of the

fitness evaluation values of the 11 algorithms are shown in Table 15. Tables 16–18 show the comparison results of ITGO, PSO, DE/best/2/bin, DE/2/exp, GbABC, BBO, GSA, TLBO, GWO, CLPSO and CS using the Wilcoxon signed-rank test, Bonferroni–Holm correction method and Friedman's test for the means data set from Table 15. As for Wilcoxon signed-rank test and Bonferroni–Holm correction method, shown in Table 16, we consider H0 hypothesis where the results of two algorithms A and B have no significant difference. As seen from the statistical results, when $\alpha/i=8.857E−5$, $8.857E−5$, $8.857E−5$, 0.0071 , 0.0083 , 0.01 , 0.0125 , 0.0167 , 0.025 and 0.05 , which compare ITGO with PSO, DE/best/2/bin, DE/2/exp, GbABC, BBO, GSA, TLBO, GWO, CLPSO and CS, p-values are less than α/i , so all the H0 hypotheses were rejected. As for the Friedman's test, average ranks show that ITGO is better than the others.

Fig. 9 is the convergence curves of the 11 algorithms for the 10 benchmark functions. F1 function is separable function; F6, F8, F9, F12, F13, F17 and F18 are partially separable functions. All these functions are high-dimensional problems. Fig. 9 shows that whether it is separable or partially separable or fully nonseparable, the performance of ITGO outperforms other 10 algorithms. For F1 function, the performance of GbABC outperforms other algorithms except ITGO. For F6 function, the performance of DE/best/2/bin outperforms other algorithms except ITGO. For F8 function, the performance of DE/2/exp outperforms other algorithms except ITGO. For F9 function, the performance of GbABC outperforms other algorithms except ITGO. For F12 function, the performance of GbABC outperforms other algorithms except ITGO. For F13 function, the

Table 14

Results of Friedman's method for ITGO, rcGA, cPSO and QIGSA by CEC 2005 and CEC 2008 (means).

| Method | Statistical value | p-value |
|---------------|-------------------|----------|
| Friedman test | 58.344 | 1.32e−12 |

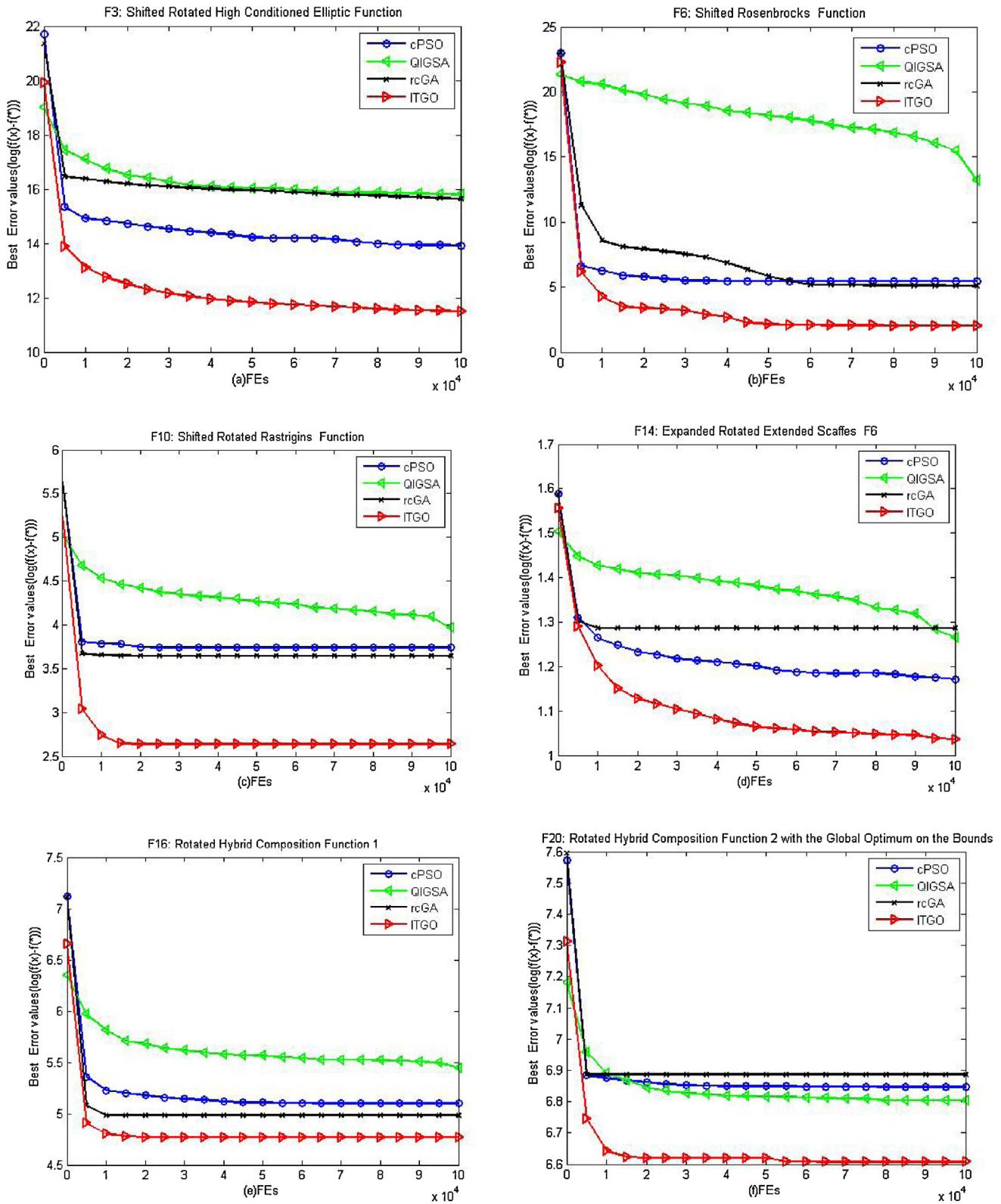


Fig. 8. The convergence curve.

performance of DE/2/exp outperforms other algorithms except ITGO. For F17 function, the performance of GbABC outperforms other algorithms except ITGO. For F18 function, the performance of GbABC outperforms other algorithms except ITGO. It shows that GbABC has good performance to solve many high-dimensional problems except ITGO.

4.1.4. Experiment on large scale global optimization compared with improved PSO

In this experiment, we choose the current improved PSO algorithms such as CLPSO, GOPSO, DNSCLPSO and DNSPSO to compare with ITGO. The results of CLPSO, GOPSO, DNSCLPSO and DNSPSO are copied from [71]. The results are shown in Table 19. We set

Table 15

Comparison with ITGO and PSO, DE/best/2/bin, DE/2/exp, GbABC, BBO, GSA, TLBO, GWO, CLPSO, CS for CEC 2010. Bold font means the best fitness evaluation value.

| Fun | PSO | De/best/2/bin | DE/2/exp | GbABC | BBO | GSA | TLBO | GWO | CLPSO | CS | ITGO | |
|-----|--------|---------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|-------------|--------------------|-------------|--------------------|
| F1 | Min | 2.2016e+010 | 5.2212e+010 | 2.2335e+010 | 6.2884e+008 | 8.4754e+009 | 3.0760e+011 | 6.0310e+009 | 6.8241e+010 | 1.6242e+010 | 9.8948e+009 | 1.0716e+009 |
| | Max | 3.3954e+010 | 7.0602e+010 | 2.8873e+010 | 1.1324e+010 | 1.1299e+010 | 3.6404e+011 | 1.3339e+010 | 1.1747e+011 | 1.9136e+010 | 1.2272e+010 | 1.2778e+009 |
| | Means | 2.7934e+010 | 6.1271e+010 | 2.5064e+010 | 4.4445e+009 | 9.8104e+009 | 3.4016e+011 | 9.1313e+009 | 8.8856e+010 | 1.7625e+010 | 1.0871e+010 | 1.1792e+009 |
| | SD | 3.0164e+009 | 4.5155e+009 | 1.6114e+009 | 2.7078e+009 | 7.4695e+008 | 1.6274e+010 | 1.7513e+009 | 1.0185e+010 | 8.0904e+008 | 5.8255e+008 | 5.9898e+007 |
| | Median | 2.4422e+010 | 5.6927e+010 | 2.3474e+010 | 2.0259e+009 | 9.9594e+009 | 3.4038e+011 | 9.1257e+009 | 8.0815e+010 | 1.7616e+010 | 1.0823e+010 | 1.1218e+009 |
| F2 | Min | 1.6269e+004 | 1.7107e+004 | 8.7306e+003 | 1.8690e+003 | 9.5964e+003 | 7.4903e+003 | 9.9781e+003 | 1.4724e+004 | 1.0751e+004 | 1.4060e+004 | 9.4258e+003 |
| | Max | 1.7513e+004 | 1.8521e+004 | 9.0180e+003 | 2.4863e+003 | 1.0238e+004 | 7.8937e+003 | 1.0556e+004 | 1.5419e+004 | 1.1133e+004 | 1.4913e+004 | 9.9292e+003 |
| | Means | 1.6889e+004 | 1.7753e+004 | 8.8814e+003 | 2.1311e+003 | 9.8999e+003 | 7.6210e+003 | 1.0285e+004 | 1.4966e+004 | 1.0925e+004 | 1.4385e+004 | 9.6849e+003 |
| | SD | 2.9310e+002 | 3.6884e+002 | 6.4506e+001 | 1.7134e+002 | 1.6591e+002 | 9.8862e+001 | 1.3688e+002 | 1.5424e+002 | 1.1386e+002 | 1.9215e+002 | 1.6015e+002 |
| | Median | 1.6714e+004 | 1.7356e+004 | 8.8042e+003 | 1.9565e+003 | 9.9028e+003 | 7.5923e+003 | 1.0286e+004 | 1.4850e+004 | 1.0907e+004 | 1.4381e+004 | 9.5216e+003 |
| F3 | Min | 1.9970e+001 | 2.1163e+001 | 1.8333e+001 | 1.7026e+001 | 1.9240e+001 | 2.1487e+001 | 2.1590e+001 | 2.0765e+001 | 2.0184e+001 | 2.0905e+001 | 2.0331e+001 |
| | Max | 2.1064e+001 | 2.1340e+001 | 1.8624e+001 | 1.8919e+001 | 1.9673e+001 | 2.1522e+001 | 2.1612e+001 | 2.0973e+001 | 2.0434e+001 | 2.1008e+001 | 2.0731e+001 |
| | Means | 2.0806e+001 | 2.1263e+001 | 1.8491e+001 | 1.8349e+001 | 1.9434e+001 | 2.1507e+001 | 2.1603e+001 | 2.0839e+001 | 2.0304e+001 | 2.0964e+001 | 2.0599e+001 |
| | SD | 2.7627e-001 | 4.9454e-002 | 7.5422e-002 | 5.6077e-001 | 6.9359e-002 | 1.1232e-002 | 6.3251e-003 | 4.3760e-002 | 6.8622e-002 | 2.6468e-002 | 1.3662e-001 |
| | Median | 2.0691e+001 | 2.1222e+001 | 1.8414e+001 | 1.7803e+001 | 1.9449e+001 | 2.1508e+001 | 2.1603e+001 | 2.0792e+001 | 2.0299e+001 | 2.0970e+001 | 2.0440e+001 |
| F4 | Min | 4.9359e+013 | 1.9475e+014 | 1.0523e+014 | 9.3290e+013 | 4.2962e+013 | 3.0502e+015 | 2.6917e+012 | 7.9053e+013 | 2.0210e+014 | 3.0202e+013 | 7.2818e+012 |
| | Max | 8.9074e+013 | 4.0738e+014 | 1.9222e+014 | 2.7587e+014 | 1.6483e+014 | 7.6811e+015 | 6.6745e+012 | 2.3842e+014 | 3.9512e+014 | 4.3478e+013 | 1.9589e+013 |
| | Means | 6.4501e+013 | 2.7437e+014 | 1.6127e+014 | 1.8809e+014 | 9.9635e+013 | 5.5865e+015 | 3.7125e+012 | 1.4910e+014 | 2.6151e+014 | 3.7238e+013 | 1.2007e+013 |
| | SD | 1.1075e+013 | 5.1676e+013 | 2.0435e+013 | 5.2900e+013 | 3.3414e+013 | 1.1443e+015 | 1.0029e+012 | 4.8551e+013 | 4.4005e+013 | 3.6532e+012 | 2.9534e+012 |
| | Median | 5.4729e+013 | 2.2816e+014 | 1.4354e+014 | 1.3588e+014 | 9.4441e+013 | 5.6151e+015 | 3.5082e+012 | 1.0014e+014 | 2.6006e+014 | 3.7836e+013 | 9.3791e+012 |
| F5 | Min | 3.5411e+008 | 4.1156e+008 | 4.9082e+008 | 5.5970e+008 | 1.2272e+008 | 3.7909e+008 | 1.0709e+008 | 1.9488e+008 | 4.7153e+008 | 3.1132e+008 | 1.5026e+008 |
| | Max | 5.1327e+008 | 4.6979e+008 | 6.6127e+008 | 8.7882e+008 | 3.0898e+008 | 4.3879e+008 | 2.5773e+008 | 4.5908e+008 | 5.7505e+008 | 4.5440e+008 | 3.2736e+008 |
| | Means | 4.2160e+008 | 4.3459e+008 | 5.7618e+008 | 6.9937e+008 | 2.1831e+008 | 4.1264e+008 | 1.6394e+008 | 3.4157e+008 | 5.2144e+008 | 3.8647e+008 | 2.2966e+008 |
| | SD | 3.7237e+007 | 1.3853e+007 | 4.7615e+007 | 8.5576e+007 | 4.1292e+007 | 1.5738e+007 | 4.2348e+007 | 8.6845e+007 | 2.3868e+007 | 4.0171e+007 | 4.7349e+007 |
| | Median | 3.7827e+008 | 4.2154e+008 | 5.2043e+008 | 5.9816e+008 | 2.1761e+008 | 4.1491e+008 | 1.5027e+008 | 2.5167e+008 | 5.1732e+008 | 3.8323e+008 | 1.8610e+008 |
| F6 | Min | 7.9623e+006 | 5.3801e+006 | 1.6416e+007 | 2.0371e+007 | 4.5983e+006 | 2.0874e+007 | 2.1112e+007 | 1.0362e+007 | 1.6805e+007 | 1.4219e+007 | 1.3744e+006 |
| | Max | 1.2145e+007 | 7.2299e+006 | 2.0833e+007 | 2.0861e+007 | 9.8275e+006 | 2.1234e+007 | 2.1205e+007 | 1.5716e+007 | 1.9715e+007 | 2.0336e+007 | 2.4262e+006 |
| | Means | 9.4347e+006 | 6.2737e+006 | 1.9657e+007 | 2.0648e+007 | 7.1778e+006 | 2.1115e+007 | 2.1157e+007 | 1.3084e+007 | 1.8473e+007 | 1.8530e+007 | 1.9411e+006 |
| | SD | 1.0271e+006 | 4.0663e+005 | 1.4021e+006 | 1.1705e+005 | 1.3684e+006 | 9.1583e+004 | 2.9844e+004 | 1.0903e+006 | 7.5800e+005 | 1.8825e+006 | 3.2697e+005 |
| | Median | 8.3829e+006 | 5.9404e+006 | 1.8083e+007 | 2.0550e+007 | 6.9052e+006 | 2.1128e+007 | 2.1159e+007 | 1.1963e+007 | 1.8631e+007 | 1.9345e+007 | 1.6463e+006 |
| F7 | Min | 7.8128e+009 | 2.5424e+010 | 6.6302e+010 | 1.0588e+011 | 1.6683e+010 | 7.3509e+011 | 2.0462e+013 | 2.4479e+010 | 5.8950e+010 | 8.0679e+009 | 9.9850e+008 |
| | Max | 4.5060e+010 | 4.7108e+010 | 9.9146e+010 | 1.8844e+011 | 4.3063e+010 | 5.2381e+012 | 2.0462e+013 | 6.2729e+010 | 8.8751e+010 | 1.2592e+010 | 2.2132e+009 |
| | Means | 1.8042e+010 | 3.9497e+010 | 8.3277e+010 | 1.4145e+011 | 2.7603e+010 | 2.6315e+012 | 2.0462e+013 | 4.4181e+010 | 7.3964e+010 | 1.0457e+010 | 1.4824e+009 |
| | SD | 8.5559e+009 | 4.9206e+009 | 9.0711e+009 | 2.2202e+010 | 7.0918e+009 | 1.1750e+012 | 5.1708e+004 | 9.1158e+009 | 7.5617e+009 | 1.4578e+009 | 2.8075e+008 |
| | Median | 1.0010e+010 | 3.5455e+010 | 7.4099e+010 | 1.1674e+011 | 2.8425e+010 | 2.4938e+012 | 2.0462e+013 | 3.5382e+010 | 7.3279e+010 | 1.0579e+010 | 1.2850e+009 |
| F8 | Min | 1.9905e+013 | 5.0548e+012 | 5.0039e+008 | 9.1227e+011 | 5.5028e+010 | 1.1482e+017 | 8.6958e+015 | 8.6958e+015 | 2.3894e+012 | 5.4455e+008 | 5.0124e+007 |
| | Max | 1.3786e+015 | 2.4577e+013 | 8.7443e+008 | 2.7005e+014 | 3.8828e+011 | 2.2850e+017 | 8.6958e+015 | 8.6958e+015 | 4.8537e+012 | 3.2417e+009 | 8.9949e+008 |
| | Means | 1.8760e+014 | 1.3585e+013 | 7.1616e+008 | 4.0442e+013 | 1.3337e+011 | 1.8176e+017 | 8.6958e+015 | 8.6958e+015 | 4.0711e+012 | 1.2942e+009 | 2.4133e+008 |
| | SD | 3.0910e+014 | 5.4134e+012 | 9.5955e+007 | 5.7557e+013 | 8.4847e+010 | 3.2159e+016 | 1.1601e+009 | 1.2395e+009 | 1.3029e+012 | 6.7237e+008 | 1.9278e+008 |
| | Median | 2.4310e+013 | 8.5095e+012 | 5.9441e+008 | 3.3903e+012 | 1.0988e+011 | 1.8373e+017 | 8.6958e+015 | 8.6958e+015 | 3.8541e+012 | 1.1324e+009 | 1.0420e+008 |
| F9 | Min | 2.3504e+010 | 7.7145e+010 | 2.2627e+010 | 2.4227e+009 | 2.0541e+010 | 3.4822e+011 | 1.8432e+011 | 1.8432e+011 | 3.2348e+010 | 1.5329e+010 | 3.3739e+009 |
| | Max | 3.2359e+010 | 1.1728e+011 | 2.6234e+010 | 7.5876e+009 | 2.5292e+010 | 4.1157e+011 | 1.8531e+011 | 1.8603e+011 | 3.6662e+010 | 1.9558e+010 | 4.1460e+009 |
| | Means | 2.7714e+010 | 9.8518e+010 | 2.4637e+010 | 4.5956e+009 | 2.3077e+010 | 3.8633e+011 | 1.8482e+011 | 1.8497e+011 | 3.4374e+010 | 1.7267e+010 | 3.7003e+009 |
| | SD | 2.7140e+009 | 8.3490e+009 | 8.7138e+008 | 1.2217e+009 | 1.2672e+009 | 1.3465e+010 | 2.8697e+008 | 4.1852e+008 | 1.2780e+009 | 1.0318e+009 | 1.6754e+008 |
| | Median | 2.4954e+010 | 9.1122e+010 | 2.4040e+010 | 3.4409e+009 | 2.2897e+010 | 3.8339e+011 | 1.8482e+011 | 1.8486e+011 | 3.4237e+010 | 1.7253e+010 | 3.5370e+009 |
| F10 | Min | 1.6462e+004 | 1.6934e+004 | 1.6338e+004 | 8.9833e+003 | 1.5027e+004 | 7.8209e+003 | 1.0059e+004 | 1.4828e+004 | 1.4661e+004 | 1.4883e+004 | 1.0479e+004 |
| | Max | 1.7668e+004 | 1.8464e+004 | 1.7221e+004 | 1.0167e+004 | 1.6301e+004 | 8.2193e+003 | 1.0685e+004 | 1.5743e+004 | 1.5239e+004 | 1.5762e+004 | 1.1144e+004 |
| | Means | 1.6929e+004 | 1.7872e+004 | 1.6892e+004 | 2.3409e+002 | 9.5581e+003 | 1.5594e+004 | 8.0350e+003 | 1.0389e+004 | 1.5217e+004 | 1.4972e+004 | 1.5331e+004 |
| | SD | 2.8111e+002 | 3.3876e+002 | 1.6701e+004 | 3.5280e+002 | 3.7781e+002 | 1.0491e+002 | 1.5754e+002 | 2.0082e+002 | 1.4284e+002 | 2.1054e+002 | 1.8523e+002 |
| | Median | 1.6728e+004 | 1.7629e+004 | | 9.1752e+003 | 1.5611e+004 | 8.0465e+003 | 1.0405e+004 | 1.5036e+004 | 1.5001e+004 | 1.5315e+004 | 1.0628e+004 |

Table 15 (Continued)

| Fun | PSO | De/best/2/bin | DE/2/exp | GbABC | BBO | GSA | TLBO | GWO | CLPSO | CS | ITGO |
|-------|--------|--------------------|-------------|--------------------|--------------------|-------------|--------------------|-------------|-------------|--------------|--------------------------------|
| F11 | Min | 2.0349e+002 | 2.2649e+002 | 2.3293e+002 | 2.2366e+002 | 2.2585e+002 | 2.3576e+002 | 2.2003e+002 | 2.2299e+002 | 2.3133e+002 | 2.2758e+002 1.7189e+002 |
| | Max | 2.2222e+002 | 2.3266e+002 | 2.3355e+002 | 2.2749e+002 | 2.3350e+002 | 2.3636e+002 | 2.2360e+002 | 2.2894e+002 | 2.3297e+002 | 2.3064e+002 2.2590e+002 |
| | Means | 2.1373e+002 | 2.2957e+002 | 2.3325e+002 | 2.2633e+002 | 2.3064e+002 | 2.3611e+002 | 2.2163e+002 | 2.2525e+002 | 2.3213e+002 | 2.2986e+002 2.0754e+002 |
| | SD | 5.7977e+000 | 1.5096e+000 | 1.7159e−001 | 8.1935e−001 | 2.1246e+000 | 1.3707e−001 | 1.1581e+000 | 1.4016e+000 | 3.9697e−001 | 6.0078e−001 2.1238e+001 |
| | Median | 2.0677e+002 | 2.2846e+002 | 2.3310e+002 | 2.2579e+002 | 2.3134e+002 | 2.3614e+002 | 2.2127e+002 | 2.2377e+002 | 2.3221e+002 | 2.2992e+002 1.7373e+002 |
| F12 | Min | 4.4085e+006 | 1.0241e+007 | 4.5090e+006 | 1.6410e+006 | 3.8624e+006 | 2.2065e+007 | 1.9679e+006 | 4.2147e+006 | 5.9643e+006 | 2.7642e+006 1.3917e+006 |
| | Max | 5.4874e+006 | 1.1738e+007 | 5.1487e+006 | 2.0800e+006 | 4.7205e+006 | 4.2450e+007 | 2.5522e+006 | 5.6306e+006 | 6.9282e+006 | 3.0257e+006 1.5649e+006 |
| | Means | 4.9149e+006 | 1.0912e+007 | 4.8493e+006 | 1.8469e+006 | 4.2460e+006 | 3.3598e+007 | 2.1934e+006 | 4.7279e+006 | 6.4754e+006 | 2.9203e+006 1.4687e+006 |
| | SD | 2.5665e+005 | 4.9488e+005 | 1.6890e+005 | 1.1705e+005 | 2.2917e+005 | 5.5222e+006 | 1.4216e+005 | 2.9584e+005 | 2.4899e+005 | 6.9409e+004 4.9900e+004 |
| | Median | 4.6721e+006 | 1.0403e+007 | 4.6944e+006 | 1.7442e+006 | 4.2399e+006 | 3.4229e+007 | 2.1559e+006 | 4.4592e+006 | 6.4759e+006 | 2.9278e+006 1.4189e+006 |
| F13 | Min | 2.1677e+010 | 4.6311e+011 | 1.8614e+009 | 1.1569e+009 | 2.5313e+010 | 3.1332e+012 | 5.9782e+011 | 3.9874e+011 | 5.9703e+010 | 5.8670e+010 6.1593e+008 |
| | Max | 4.2960e+010 | 6.9003e+011 | 2.3968e+009 | 5.0756e+010 | 4.2581e+010 | 3.5666e+012 | 5.9930e+011 | 4.8980e+011 | 7.7578e+010 | 8.1583e+010 1.1153e+009 |
| | Means | 3.0907e+010 | 5.6592e+011 | 2.1331e+009 | 1.2870e+010 | 2.9778e+010 | 3.3600e+012 | 5.9823e+011 | 4.4078e+011 | 6.9107e+010 | 6.8521e+010 8.1438e+008 |
| | SD | 5.6079e+009 | 6.2904e+010 | 1.4478e+008 | 1.2852e+010 | 3.9726e+009 | 1.1755e+011 | 3.5225e+008 | 2.2270e+010 | 4.0687e+009 | 5.3827e+009 1.0913e+008 |
| | Median | 2.6130e+010 | 5.1724e+011 | 1.9765e+009 | 2.1968e+009 | 2.9536e+010 | 3.3470e+012 | 5.9814e+011 | 4.1943e+011 | 6.9706e+010 | 6.8242e+010 7.1398e+008 |
| F14 | Min | 1.9565e+010 | 1.0935e+011 | 3.6320e+010 | 6.2007e+009 | 2.7120e+010 | 3.4473e+011 | 8.2995e+009 | 8.2556e+010 | 3.8661e+010 | 1.4331e+010 5.9112e+009 |
| | Max | 2.7994e+010 | 1.4256e+011 | 4.0269e+010 | 8.2100e+009 | 3.6521e+010 | 4.4187e+011 | 1.2534e+010 | 1.1935e+011 | 4.4051e+010 | 1.8321e+010 7.2781e+009 |
| | Means | 2.2538e+010 | 1.2649e+011 | 3.8445e+010 | 7.0125e+009 | 3.1600e+010 | 4.1233e+011 | 1.0926e+010 | 1.0145e+011 | 4.1359e+010 | 1.6691e+010 6.6483e+009 |
| | SD | 2.4262e+009 | 9.8585e+009 | 9.6504e+008 | 4.1243e+008 | 2.1510e+009 | 1.9521e+010 | 1.0396e+009 | 7.8803e+009 | 1.5186e+009 | 9.5479e+008 3.2660e+008 |
| | Median | 2.0307e+010 | 1.1543e+011 | 3.7505e+010 | 6.6961e+009 | 2.9674e+010 | 4.1404e+011 | 1.1225e+010 | 9.5417e+010 | 4.1408e+010 | 1.6802e+010 6.4128e+009 |
| F15 | Min | 1.6287e+004 | 1.7326e+004 | 1.9954e+004 | 1.6037e+004 | 1.7406e+004 | 7.9304e+003 | 1.0207e+004 | 1.4647e+004 | 1.6145e+004 | 1.5528e+004 1.1170e+004 |
| | Max | 1.7368e+004 | 1.8788e+004 | 2.0950e+004 | 1.7367e+004 | 1.9104e+004 | 8.3350e+003 | 1.0860e+004 | 1.5700e+004 | 1.6670e+004 | 1.6082e+004 1.1815e+004 |
| | Means | 1.6900e+004 | 1.7961e+004 | 2.0499e+004 | 1.6742e+004 | 1.8273e+004 | 8.0751e+003 | 1.0491e+004 | 1.5168e+004 | 1.6356e+004 | 1.5804e+004 1.1555e+004 |
| | SD | 2.7547e+002 | 4.1300e+002 | 2.2118e+002 | 3.1259e+002 | 4.8684e+002 | 1.0482e+002 | 1.4998e+002 | 2.6092e+002 | 1.2132e+002 | 1.7656e+002 1.7815e+002 |
| | Median | 1.6647e+004 | 1.7542e+004 | 2.0327e+004 | 1.6462e+004 | 1.7843e+004 | 8.0630e+003 | 1.0481e+004 | 1.4941e+004 | 1.6379e+004 | 1.5832e+004 1.1340e+004 |
| F16 | Min | 3.9290e+002 | 4.1994e+002 | 4.2519e+002 | 4.1474e+002 | 4.2342e+002 | 4.2862e+002 | 4.1644e+002 | 4.1323e+002 | 4.2127e+002 | 4.1829e+002 4.0599e+002 |
| | Max | 4.1981e+002 | 4.2572e+002 | 4.2577e+002 | 4.1620e+002 | 4.2717e+002 | 4.2986e+002 | 4.1805e+002 | 4.1745e+002 | 4.22369e+002 | 4.1964e+002 4.1132e+002 |
| | Means | 4.0432e+002 | 4.2373e+002 | 4.2551e+002 | 4.1543e+002 | 4.2566e+002 | 4.2952e+002 | 4.1747e+002 | 4.1563e+002 | 4.2259e+002 | 4.1905e+002 4.0782e+002 |
| | SD | 8.8747e+000 | 1.3665e+000 | 1.8634e−001 | 4.1458e−001 | 7.7818e−001 | 2.9147e−001 | 4.7223e−001 | 9.6281e−001 | 5.4169e−001 | 4.0607e−001 1.3134e+000 |
| | Median | 3.9642e+002 | 4.2243e+002 | 4.2533e+002 | 4.1504e+002 | 4.2523e+002 | 4.2958e+002 | 4.1755e+002 | 4.1500e+002 | 4.2258e+002 | 4.1903e+002 4.0647e+002 |
| F17 | Min | 5.6440e+006 | 1.9350e+007 | 7.4608e+006 | 3.4290e+006 | 6.6230e+006 | 7.2905e+007 | 3.4337e+007 | 6.6394e+006 | 1.0329e+007 | 3.8800e+006 2.1959e+006 |
| | Max | 7.7651e+006 | 2.6404e+007 | 8.7419e+006 | 4.0134e+006 | 8.2733e+006 | 1.0725e+008 | 3.5376e+007 | 9.6842e+006 | 1.2461e+007 | 4.5183e+006 2.6320e+006 |
| | Means | 6.6949e+006 | 2.3220e+007 | 8.3104e+006 | 3.6743e+006 | 7.5769e+006 | 8.6301e+007 | 3.4686e+007 | 8.0816e+006 | 1.1210e+007 | 4.1718e+006 2.4004e+006 |
| | SD | 4.9253e+005 | 1.6207e+006 | 2.6608e+005 | 1.5848e+005 | 4.9181e+005 | 8.4311e+006 | 2.2981e+005 | 7.8292e+005 | 4.8652e+005 | 1.6547e+005 1.1586e+005 |
| | Median | 6.2046e+006 | 2.2591e+007 | 8.1229e+006 | 3.4763e+006 | 7.0634e+006 | 8.4380e+007 | 3.4613e+007 | 7.3505e+006 | 1.1154e+007 | 4.2066e+006 2.2938e+006 |
| F18 | Min | 1.5507e+011 | 2.1602e+012 | 7.0252e+010 | 7.5037e+009 | 3.1190e+011 | 6.8134e+012 | 1.1357e+012 | 1.1298e+012 | 3.5033e+011 | 3.5157e+011 2.9864e+010 |
| | Max | 2.4622e+011 | 3.3674e+012 | 8.3329e+010 | 1.1322e+011 | 4.3949e+011 | 7.7377e+012 | 1.1423e+012 | 1.2388e+012 | 4.1473e+011 | 4.1226e+011 3.6260e+010 |
| | Means | 1.9928e+011 | 2.6804e+012 | 7.7982e+010 | 3.9720e+010 | 3.7347e+011 | 7.4074e+012 | 1.1387e+012 | 1.1868e+012 | 3.7962e+011 | 3.8656e+011 3.3201e+010 |
| | SD | 2.3679e+010 | 2.9751e+011 | 2.8603e+009 | 2.4802e+010 | 3.0723e+010 | 2.4093e+011 | 1.7000e+009 | 2.7119e+010 | 1.7340e+010 | 1.5483e+010 1.6522e+009 |
| | Median | 1.7598e+011 | 2.3946e+012 | 7.5970e+010 | 1.7792e+010 | 3.5387e+011 | 7.4356e+012 | 1.1382e+012 | 1.1636e+012 | 3.7895e+011 | 3.8634e+011 3.1355e+010 |
| F19 | Min | 1.0275e+007 | 3.4045e+007 | 1.4913e+007 | 1.3191e+007 | 1.9056e+007 | 1.4950e+008 | 8.1001e+007 | 1.3068e+007 | 2.7985e+007 | 1.5724e+007 5.5539e+006 |
| | Max | 1.5056e+007 | 5.1205e+007 | 2.0179e+007 | 1.7681e+007 | 2.6142e+007 | 3.6823e+008 | 8.8805e+007 | 1.6888e+007 | 3.6770e+007 | 1.8222e+007 7.5097e+006 |
| | Means | 1.2468e+007 | 4.6013e+007 | 1.8220e+007 | 1.5604e+007 | 2.2777e+007 | 2.6463e+008 | 8.4166e+007 | 1.4718e+007 | 3.1648e+007 | 1.7203e+007 6.5017e+006 |
| | SD | 1.1130e+006 | 4.2004e+006 | 1.1636e+006 | 1.2285e+006 | 2.2204e+006 | 5.4682e+007 | 2.0099e+006 | 1.0587e+006 | 2.4543e+006 | 7.5591e+005 4.3085e+005 |
| | Median | 1.1522e+007 | 4.1953e+007 | 1.7559e+007 | 1.4301e+007 | 2.0042e+007 | 2.5982e+008 | 8.4309e+007 | 1.3626e+007 | 3.1513e+007 | 1.7348e+007 6.1192e+006 |
| F20 | Min | 1.4854e+011 | 2.5160e+012 | 7.2388e+010 | 1.2751e+010 | 3.7698e+011 | 7.5003e+012 | 1.2764e+012 | 1.2889e+012 | 4.2324e+011 | 4.2290e+011 3.7591e+010 |
| | Max | 2.6518e+011 | 6.5177e+012 | 8.6529e+010 | 1.0473e+011 | 4.9955e+011 | 8.4727e+012 | 1.2869e+012 | 1.4222e+012 | 4.6955e+011 | 5.1785e+011 4.6425e+010 |
| | Means | 2.2069e+011 | 3.0955e+012 | 7.9534e+010 | 4.1995e+010 | 4.3078e+011 | 8.1016e+012 | 1.2804e+012 | 1.3551e+012 | 4.4858e+011 | 4.6315e+011 4.2370e+010 |
| | SD | 2.6124e+010 | 9.5185e+011 | 3.1999e+009 | 2.3639e+010 | 3.2740e+010 | 2.2550e+011 | 2.8109e+009 | 3.1240e+010 | 1.2488e+010 | 2.4782e+010 2.3644e+009 |
| | Median | 2.0144e+011 | 2.6388e+012 | 7.74467e+010 | 2.0610e+010 | 3.9565e+011 | 8.1277e+012 | 1.2799e+012 | 1.3274e+012 | 4.4784e+011 | 4.6408e+011 3.9746e+010 |
| w/t/l | | 18/2/0 | 20/0/0 | 18/2/0 | 16/4/0 | 18/2/0 | 17/3/0 | 16/4/0 | 20/0/0 | 19/1/0 | 20/0/0 |

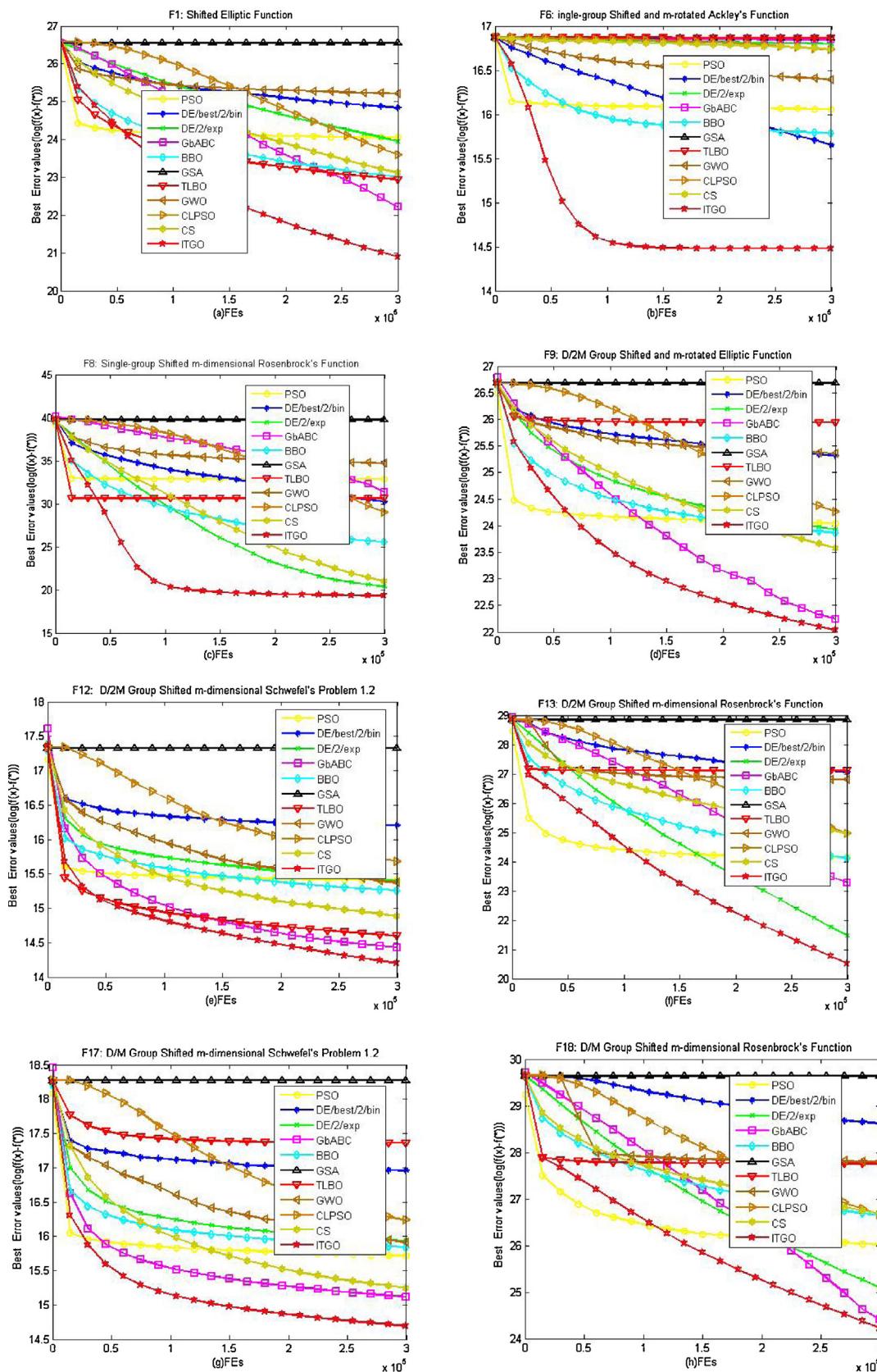


Fig. 9. The convergence curve.

MAX_FES=3E6. Table 20 shows the comparison results of ITGO, CLPSO, GOPSO, DNSCLPSO and DNSPSO by the Wilcoxon signed ranks test and Bonferroni-Holm correction method. The result shows that there are significant differences for ITGO vs CLPSO,

ITGO vs GOPSO, ITGO vs DNSCLPSO except ITGO vs DNSPSO. Tables 21 and 22 show the results of the statistical analysis by the Friedman test. Average rank of ITGO is equal to 1.85, less than CLPSO, GOPSO, DNSCLPSO.

Table 16

Comparison with ITGO and PSO, DE/best/2/bin, DE/2/exp, GbABC, BBO, GSA, TLBO, GWO, CLPSO, CS for CEC 2010 by using Wilcoxon signed-rank test and Bonferroni–Holm correction (means).

| i | Algorithm | R+ | R- | p-value | α/i | Hypothesis |
|---|-----------------------|--------|-------|----------|------------|------------|
| 8 | ITGO vs CS | 210.00 | 0.00 | 8.857e-5 | 6.25e-004 | Reject |
| 8 | ITGO vs DE/BEST/2/BIN | 210.00 | 0.00 | 8.857e-5 | 6.25e-004 | Reject |
| 8 | ITGO vs GWO | 210.00 | 0.00 | 8.857e-5 | 6.25e-004 | Reject |
| 7 | ITGO vs CLPSO | 209.00 | 1.00 | 1.033e-4 | 0.0071 | Reject |
| 6 | ITGO vs PSO | 206.00 | 4.00 | 1.55e-4 | 0.0083 | Reject |
| 5 | ITGO vs DE/2/EXP | 205.00 | 5.00 | 1.89e-4 | 0.01 | Reject |
| 4 | ITGO vs BBO | 199.00 | 11.00 | 2.19e-4 | 0.0125 | Reject |
| 3 | ITGO vs GSA | 195.00 | 15.00 | 7.79e-4 | 0.0167 | Reject |
| 2 | ITGO vs GbABC | 187.00 | 23.00 | 0.002 | 0.025 | Reject |
| 1 | ITGO vs TLBO | 172.00 | 38.00 | 0.012 | 0.05 | Reject |

Table 17

Average ranks of 11 algorithms for CEC 2010 by Friedman test (means).

| Fun | Average ranks |
|---------------|---------------|
| ITGO | 1.90 |
| GbABC | 4.20 |
| PSO | 5.15 |
| CS | 5.20 |
| BBO | 5.35 |
| DE/2/EXP | 6.30 |
| TLBO | 6.45 |
| GWO | 6.90 |
| CLPSO | 7.00 |
| DE/BEST/2/BIN | 8.40 |
| GSA | 9.15 |

4.1.5. Optimization of real-world problems

Support vector machine (SVM) is a statistical classification method based on the VC (Vapnik–Chervonenkis) statistical learning theory and the structural risk minimization principle proposed by Vapnik et al. [87]. As a state-of-the-art algorithm, it has been used widely in many fields, such as novelty detection [88], image recognition [89], fault diagnosis [90], and so on. Given a training dataset $\{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\} \in (\mathcal{X} \times \mathcal{Y})^l$, $i = 1, 2, \dots, l$. $x_i \in \mathcal{X} \subset \mathbb{R}^n$ is the input vector, $y_i = \{-1, +1\}$ is the class number. $w \cdot x + b = 0$ is the maximum margin separating hyper plane. w is a orthogonal vector to the hyper plane. In order to separate the training samples into different categories, we should maximize the margin $1/\|w\|$, that is to minimize $\|w\|^2$. SVM can be seen as a quadratic optimization problem:

$$\begin{aligned} & \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \\ & \text{s.t. } y_i(w \cdot \phi(x_i)) + b + \xi_i \geq 1 \quad (\xi_i \geq 0, i = 1, 2, \dots, l) \end{aligned} \quad (17)$$

C is the penalty parameter which imposes a trade-off between training error and generalization, ξ_i is a slack variable.

The optimal problem of Eq. (17) can be converted to the dual problem as follows by Lagrange method:

$$\begin{aligned} & \min \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j K(x_i \cdot x_j) - \sum_{j=1}^l \alpha_j \\ & \text{s.t. } \sum_{i=1}^l y_i \alpha_i = 0 \quad (0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, l) \end{aligned} \quad (18)$$

Table 18

Results of Friedman test for ITGO, PSO, DE/best/2/bin, DE/2/exp, GbABC, BBO, GSA, TLBO, GWO, CLPSO and CS using CEC 2010 (means).

| Method | Statistical value | p-value |
|---------------|-------------------|----------|
| Friedman test | 72.03 | 1.79e-11 |

$\varphi(\cdot)$ denotes the mapping function which maps the input vectors into the high dimensional feature space and $K(\cdot, \cdot)$ denotes the kernel function. The generally used kernel functions are given as follows:

Linear kernel $K(x_i, x_j) = x_i^T x_j$.

Polynomial kernel $K(x_i, x_j) = (x_i^T x_j + l)^d$, d , l is parameter.

RBF kernel $K(x_i, x_j) = \exp(-\|x_i - x_j\|^2/2\sigma^2)$, σ is parameter.

Sigmoid kernel $K(x_i, x_j) = \tanh(\beta x_i^T x_j + b)$, β, b is parameter.

In our instance, we choose RBF kernel function to solve practical problem. The setting of parameters (C, σ) directly influences the performance of SVM. It is a SVM parameter optimization problem. We choose two benchmark datasets with different dimensions from the repository of the machine learning databases [91], namely, Glass and Wine. The glass identification dataset contains six different types of glass, that is, float-processed building windows, non-float processed building windows, float-processed vehicle windows, containers, tableware and headlamps. There are nine attributes, namely, refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium and iron. The wine dataset contains chemical analyses of wines derived from three different cultivars, with 13 attributes, namely, alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavonoids, nonflavonoid phenols, proanthocyanidins, color intensity, hue, OD280/OD315 of diluted wines and praline.

In order to verify the performance of our proposed algorithm, we compare ITGO-SVM with PSO-SVM, DE-SVM, WQPSO-SVM, GS-SVM, BBDE-SVM and ICDF-GS-SVM. The average accuracy of 10-fold cross-validation is used for the fitness. Population size is 30 and the max fitness evaluations MAX.FES = 1000. For ITGO, parameter beta = 1.1, MAX.gc = 7. Parameter setting of PSO, DE/best/2, WQPSO are adopted by the paper [19,73,92].

Table 23 shows the description of glass and wine data set and it indicates that they are multi-class classification problems. For this, we use the LIBSVM tool provided by Chih-Jen Lin [93]. **Table 24** shows the comparison results of ITGO-SVM, DE-SVM, PSO-SVM and QWPSO-SVM. It is obvious that ITGO-SVM has better accuracy than DE-SVM, PSO-SVM and QWPSO-SVM for the glass and wine dataset. **Fig. 10** shows the convergence curve of ITGO-SVM, DE-SVM, PSO-SVM and QWPSO-SVM. It means that ITGO-SVM has not only better accuracy but also fast convergence speed. **Table 25** shows the comparison results of ITGO-SVM, GS-SVM, BBDE-SVM and ICDF-GS-SVM. The results of GS-SVM, BBDE-SVM and ICDF-GS-SVM are obtained by the paper [94]. ITGO-SVM has better accuracy than GS-SVM, BBDE-SVM and ICDF-GS-SVM too.

5. Discussion of other parameters

In our proposed algorithm, we use two parameters (beta and Max.gc) under the following assumption: the number of proliferative cells (p), quiescent cells (q) and dying cells (d) are 20%, 60% and 20% of the numbers of tumor cells, in which (p, q, d) can be seen as a

Table 19

Comparisons with ITGO and other algorithms for CEC 2010 [68]. Bold font means the best fitness evaluation value.

| Fun | CLPSO | GOPS0 | DNSCLPSO | DNSPSO | ITGO |
|-----------|-----------------|----------|-----------------|-----------------|-------------------------|
| F1 Means | 7.80e+08 | 1.13e+10 | 1.67e+08 | 1.87e+07 | 4.1123e+006 |
| SD | 6.79e+07 | 2.37e+09 | 7.30e+06 | 1.73e+06 | 1.4907e+005 |
| F2 Means | 6.06e+03 | 9.25e+03 | 5.66e+03 | 5.85e+03 | 5.0257e+003 |
| SD | 1.40e+02 | 5.62e+03 | 8.29e+01 | 4.23e+02 | 9.8261e+001 |
| F3 Means | 2.04e+01 | 1.97e+01 | 1.91e+01 | 1.93e+01 | 1.9289e+001 |
| SD | 8.29e−02 | 7.21e−02 | 7.89e−01 | 4.44e−02 | 1.5069e−002 |
| F4 Means | 9.98e+13 | 1.92e+13 | 8.99e+12 | 2.25e+12 | 2.2274e+012 |
| SD | 9.44e+12 | 1.04e+12 | 1.22e+12 | 1.96e+11 | 7.5434e+011 |
| F5 Means | 1.44e+08 | 1.62e+08 | 8.52e+07 | 1.57e+08 | 1.8508e+008 |
| SD | 2.63e+07 | 1.46e+08 | 1.43e+07 | 2.32e+07 | 1.6544e+007 |
| F6 Means | 6.68e+06 | 1.10e+07 | 2.41e+01 | 1.75e+06 | 2.7333e+006 |
| SD | 7.99e+05 | 2.33e+07 | 9.95e−01 | 2.75e+05 | 2.1919e+005 |
| F7 Means | 9.03e+09 | 1.41e+09 | 6.57e+08 | 8.60e+06 | 1.7018e+006 |
| SD | 8.80e+08 | 2.62e+09 | 2.44e+08 | 2.55e+05 | 8.6709e+004 |
| F8 Means | 1.11e+08 | 9.68e+07 | 9.19e+07 | 1.31e+07 | 1.3398e+008 |
| SD | 1.40e+07 | 1.24e+10 | 3.50e+07 | 4.65e+06 | 2.7740e+007 |
| F9 Means | 8.96e+09 | 4.28e+10 | 1.57e+09 | 3.16e+08 | 4.0255e+008 |
| SD | 1.29e+08 | 2.73e+10 | 6.43e+07 | 2.49e+06 | 1.1608e+007 |
| F10 Means | 1.24e+04 | 9.69e+03 | 7.63e+03 | 6.90e+03 | 6.3378e+003 |
| | 2.80e+02 | 1.28e+04 | 2.57e+01 | 1.88e+02 | 4.5140e+001 |
| F11 Means | 2.28e+02 | 2.01e+02 | 1.68e+02 | 1.76e+02 | 2.1217e+002 |
| SD | 1.03e−01 | 1.35e+02 | 9.13e+00 | 5.74e+00 | 4.7122e−001 |
| F12 Means | 7.26e+06 | 5.91e+06 | 1.00e+06 | 2.34e+05 | 2.7552e+005 |
| SD | 1.08e+06 | 2.61e+06 | 5.83e+03 | 1.81e+04 | 1.1325e+004 |
| F13 Means | 6.73e+08 | 5.23e+09 | 2.20e+08 | 1.15e+06 | 4.0784e+004 |
| | 1.55e+08 | 3.28e+09 | 3.41e+07 | 1.81e+05 | 3.4967e+003 |
| F14 Means | 2.09e+10 | 5.48e+10 | 3.15e+09 | 3.81e+09 | 1.0920e+009 |
| | 7.32e+08 | 4.16e+10 | 8.62e+07 | 7.15e+07 | 4.6092e+007 |
| F15 Means | 1.47e+04 | 1.24e+04 | 8.26e+03 | 7.52e+03 | 7.6444e+003 |
| SD | 2.49e+02 | 9.40e+03 | 8.78e+01 | 4.34e+01 | 1.5411e+002 |
| F16 Means | 4.24e+02 | 3.97e+02 | 4.04e+02 | 3.86e+02 | 3.8710e+002 3.3088e−001 |
| SD | 2.85e−01 | 2.58e+02 | 9.27e−01 | 4.15e−01 | |
| F17 Means | 1.15e+07 | 1.05e+07 | 1.56e+06 | 6.13e+05 | 7.5107e+005 |
| | 1.04e+06 | 8.18e+06 | 2.41e+04 | 6.03e+03 | 1.9676e+004 |
| F18 Means | 5.69e+10 | 5.46e+11 | 2.81e+10 | 2.29e+07 | 9.8767e+004 |
| SD | 8.51e+09 | 7.19e+10 | 3.05e+09 | 9.39e+06 | 1.0029e+004 |
| F19 Means | 3.82e+07 | 8.78e+06 | 5.05e+06 | 5.33e+06 | 2.3228e+006 |
| | 8.43e+06 | 5.42e+06 | 7.09e+04 | 1.66e+06 | 1.9984e+005 |
| F20 Means | 8.44e+10 | 2.22e+11 | 4.23e+10 | 2.02e+08 | 5.1957e+004 |
| SD | 2.18e+09 | 3.15e+10 | 3.61e+09 | 1.92e+08 | 4.6986e+003 |
| w/t/l | 18/2/0 | 17/3/0 | 15/5/0 | 11/9/0 | |

Table 20

Comparison with ITGO and CLPSO, GOPSO, DNSCLPSO and DNSPSO for CEC 2010 (means) by using Wilcoxon signed-rank test and Bonferroni–Holm correction.

| i | Algorithm | R+ | R− | p-value | α/i | Hypothesis |
|---|------------------|--------|-------|-----------------------|--------|------------|
| 4 | ITGO vs GOPSO | 184.00 | 26.00 | 0.0019442974027892967 | 0.0125 | Reject |
| 3 | ITGO vs CLPSO | 188.00 | 22.00 | 0.0031851432686595074 | 0.0167 | Reject |
| 2 | ITGO vs DNSCLPSO | 173.00 | 37.00 | 0.011129014082381275 | 0.025 | Reject |
| 1 | ITGO vs DNSPSO | 129.00 | 81.00 | 0.3702612704529144 | 0.05 | Accept |

Table 21

Average ranks of 5 algorithms for CEC 2010 by Friedman test.

| Fun | Average ranks |
|----------|---------------|
| ITGO | 1.95 |
| DNSPSO | 1.95 |
| DNSCLPSO | 2.45 |
| GOPS0 | 4.25 |
| CLPSO | 4.40 |

percentage as 1:3:1 for the total number of the tumor cells. Though the detailed analysis of many experiments has verified the performance of ITGO algorithm in Section 4, the other three parameters (p , q , d) have not been discussed. According to the mechanism of

Table 23

Details of two dataset.

| Dataset | Number of class | Dimension | Number of samples |
|---------|-----------------|-----------|-------------------|
| Glass | 6 | 10 | 214 |
| Wine | 3 | 14 | 178 |

real world tumor growth, the quiescent cells are the main body of the tumor with more cells. Meanwhile, the proliferative cells and dying cell are the secondary parts of the tumor with few cells. So, the parameter tuning is to vary the percentage of $p:q:d$. For simplicity, p and d equals to 1, and q ranges in [1, 10]. As mentioned above, ITGO

Table 22

Results of Friedman test for ITGO, DNSPSO, DNSCLPSO, GOPSO and CLPSO using CEC 2010.

| Method | Statistical value | p-value |
|---------------|-------------------|----------|
| Friedman test | 48.24 | 8.41e−10 |

Table 24

Fitness value of four algorithms (accuracy rate of SVM). Bold font means the best fitness value.

| Dataset | PSO-SVM (%) | DE-SVM (%) | WQPSO-SVM (%) | ITGO (%) |
|---------|-------------|------------|---------------|----------------|
| Glass | 67.2430 | 68.4579 | 70.2804 | 73.4112 |
| Wine | 62.8652 | 97.7528 | 86.0112 | 98.8764 |

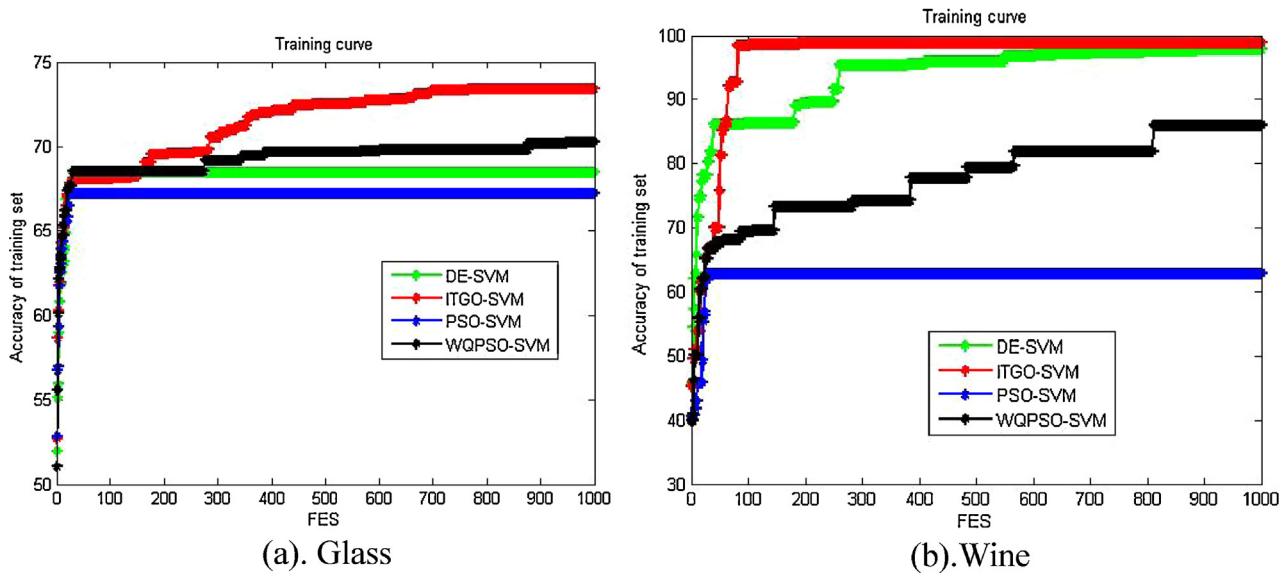


Fig. 10. Convergence curve.

Table 25

Fitness value of four algorithms (accuracy rate of SVM). Bold font means the best fitness value.

| Dataset | GS-SVM (%) [94] | BBDE-SVM (%) [94] | ICDF-GS-SVM (%) [94] | ITGO (%) |
|---------|-----------------|-------------------|----------------------|----------------|
| Glass | 69.23 | 70.77 | 72.31 | 73.4112 |
| Wine | 98.08 | 96.15 | 98.08 | 98.8764 |

Table 26Mean fitness evaluation values for different $p:q:d$ values. Bold font means the best fitness evaluation value.

| ITGO | Order | Frandman test (mean rank value) | F2 | F9 | F12 | F16 | F19 | F22 |
|------------------|----------|------------------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| $p:q:d = 1:1:1$ | 8 | 7.83 | 6.1363e-005 | 8.5569e-004 | 8.6988e+001 | 1.2073e+002 | 8.6217e+002 | 1.0071e+003 |
| $p:q:d = 1:2:1$ | 7 | 6.50 | 3.4566e-005 | 5.4222e-004 | 6.4310e+001 | 1.2079e+002 | 7.5860e+002 | 9.7600e+002 |
| $p:q:d = 1:3:1$ | 7 | 6.50 | 9.6525e-006 | 4.9833e-005 | 8.8981e+001 | 1.1405e+002 | 7.8296e+002 | 1.0298e+003 |
| $p:q:d = 1:4:1$ | 5 | 6.00 | 5.9739e-006 | 1.3155e-007 | 7.4440e+001 | 1.1898e+002 | 6.1306e+002 | 1.0814e+003 |
| $p:q:d = 1:5:1$ | 3 | 4.33 | 4.2999e-006 | 9.4379e-009 | 6.3882e+001 | 1.1407e+002 | 6.9253e+002 | 1.0382e+003 |
| $p:q:d = 1:6:1$ | 6 | 6.17 | 3.7474e-006 | 1.6955e-006 | 1.2460e+002 | 1.1333e+002 | 6.6523e+002 | 1.0809e+003 |
| $p:q:d = 1:7:1$ | 4 | 5.00 | 3.0664e-006 | 9.9507e-009 | 8.0023e+001 | 1.1899e+002 | 5.5848e+002 | 1.0895e+003 |
| $p:q:d = 1:8:1$ | 1 | 3.67 | 1.9643e-006 | 3.6803e-009 | 7.3046e+001 | 1.1120e+002 | 6.4997e+002 | 1.0931e+003 |
| $p:q:d = 1:9:1$ | 2 | 4.00 | 2.3962e-006 | 1.4198e-009 | 8.7756e+001 | 1.1503e+002 | 5.9800e+002 | 1.0577e+003 |
| $p:q:d = 1:10:1$ | 4 | 5.00 | 3.5100e-006 | 1.1341e-007 | 9.5617e+001 | 1.1631e+002 | 5.6738e+002 | 1.0300e+003 |

algorithm with each different parameter setting is ran 20 times for the six functions with one unimodal function (F2) and five multimodal functions (F9, F12, F16, F19 and F22), in which parameter beta = 1.1 and MAX_gc = 7 due to the before tuning. Mean fitness evaluation values are recorded in Table 26 and Frandman test is used too. From the results in Table 26, we can see that the best parameter combination is $p:q:d = 1:8:1$ (order = 1). It means that the performance of ITGO algorithm is better when the number of proliferative cells (p), quiescent cells (q) and dying cells (d) are 10%, 80% and 10% of the numbers of tumor cells (population size = 30). In addition, it shows that the performance is the worst when the parameter combination is as $p:q:d = 1:1:1$ (1/3,1/3,1/3). It means that the number of quiescent cells is must large than the number of proliferative cells and dying cells for the better performance of our proposed algorithm, which can provide that one proliferative cell has more than one times opportunity to guide one quiescent cell for the better solution. At the end, the experimental findings suggest that we can set the parameter combination [β , MAX_gc, p , q , d] as [1.1, 0.7 * 10, 1, 8, 1] in the performance evaluation. In particular, for some multimodal functions (F12, F22), parameter combination [β , MAX_gc, p , q , d] as [1.1, 0.7 * 10, 1, 2, 1] can also be considered due to the best performance for ITGO algorithm.

6. Conclusions

This paper proposes a new metaheuristic algorithm, named ITGO. This algorithm mimics the process of invasive tumor growth. The tumor cells were divided into four types: proliferative cells, quiescent cells, dying cells and invasive cells. Different cells have different growing patterns. In order to imitate the intrusive behavior of tumor cells, we use the levy flight. According to the value of nutrient concentration, only the proliferative cells, quiescent cells and invasive cells have invasive behavior. When the nutrient concentration is too lower, the dying cell loses its invasive ability, even, may die without enough nutrients for living.

In ITGO algorithm, a quiescent cell grows by guidance of a proliferative cell and interaction with its neighbors. In order to increase the diversity, a proliferative cell was selected randomly from proliferative cell population in each iteration. The historical best position and the current position were used together. The quiescent cells have an adaptive probability of mutation. The proliferative cell in the algorithm has strong intrusive behavior. Invasion probability is represented as a linear function, which describes the process of proliferative cell growth in tumor. This indicates that, at late growth stage, the proliferative cell will have stronger

invasive ability. As for a dying cell, because it can only get nutrient of low concentration, it must be dead at a certain probability. Death was decided by the average nutrient concentrations. In general, the search strategy inspired by tumor growth mechanism enhances the exploration ability of ITGO algorithm.

There are some points about ITGO algorithm we can discuss, because it is similar to some other metaheuristic algorithms, for instance, particle swarm optimization or differential evolution or artificial bee colony, etc. Indeed, there are many differences between ITGO algorithm and them. For particle swarm optimization, each agent (particle) searches by the global agent ('leader', only one) and historical information, while there are five different operations for the different agent (cell) in ITGO algorithm. For the quiescent cell, it searches by many 'global agents' (many leaders), not only one global agent; for the dying cell, it searches by many 'global agents' (many high level leaders) and many 'local agents' (many middle level leaders). For differential evolution, cross and mutation operations are used in the whole population. Different from it, only the quiescent cells in ITGO algorithm use the cross and mutation operations, and the cross rate is a self-adaptive method. For the artificial bee colony (ABC), there are two populations with three roles (employed bees, onlookers and scouts). Employed bees belong to the actual populations and onlookers belong to the virtual population (it is the same population as employed bees). In addition, onlookers and employed bees use almost the same operation with cross and mutation and it is like the differential evolution. When the onlooker in the population cannot search a better solution for a long time, it will be replaced by a new bee called scout. Different from it, in ITGO algorithms, there are three actual populations (proliferative cells population, quiescent cells population and dying cells population) with four roles (proliferative cells, quiescent cells, dying cells and invasive cells). There are five different operations in ITGO algorithms, that are simulated to the different behaviors of tumor cells including the growth of proliferative cell, the growth of quiescent cell, the growth of dying cell, the growth of invasive cell and the random walk for all the cells. So, ITGO algorithm provides a new framework with three populations, four roles and five search rules.

The performance of our proposed ITGO algorithm is checked with the recent and well-known optimization algorithms such as PSO, DE/best/2/bin, DE/2/exp, CLPSO, BBO, GbABC, CS, GSA, TLBO, GWO, cPSO, QIGSA, rcGA, GOPSO, DNSCLPSO and DNSPSO by experimenting with different benchmark problems with different characteristics such as, multimodality, separability, regularity, and dimensionality. The effectiveness of ITGO method is also checked for different performance criteria, mean solution, median solution, min solution, predetermined stopping criteria, convergence rate, etc. with different statistical methods such as Friedman test and Wilcoxon signed-rank statistical test with Bonferroni–Holm correction. The results show better performance of ITGO algorithm over other natured-inspired optimization methods for the considered benchmark functions. Also, the ITGO method shows better performance with less computational efforts for the large scale problems, i.e., problems with high dimensions. This method can be used for the optimization of engineering design applications.

Acknowledgments

This work is supported by the National Natural Science Foundation Project (No. 61070092/F020504); the building of strong Guangdong Province for Chinese Medicine Scientific Research (20141165); the Humanities and social science fund project for Guangdong Pharmaceutical University (RWSK201409); NSFC, Research on reasoning of behavior trust for resisting collusive reputation attack (71401045); GuangDong Provincial Natural

fund (2014A030313585), Ukraine Senate Xingnao neuroprotective effect mechanism of dynamic network based on network pharmacology; Guangdong Province Youth Innovation Talent Project, based on the cognitive rules of the semi supervised key algorithm and its cancer pattern recognition (2014KQNCX139); Major Science And Technology project of Guangdong province (2014B010112006); Guangdong Natural Science Foundation, major basic research and training talents project.

References

- [1] A. Lazar, R.G. Reynolds, Heuristic Knowledge Discovery for Archaeological Data Using Genetic Algorithms and Rough Sets, Artificial Intelligence Laboratory, Department of Computer Science, Wayne State University, 2003.
- [2] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [3] J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, USA, 1975.
- [4] I. Rechenberg, *Evolution Strategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog, Stuttgart, 1973.
- [5] L. Kota, K. Jarmai, Mathematical modeling of multiple tour multiple traveling salesmen problem using evolutionary programming, *Appl. Math. Model.* 39 (2015) 3410–3433.
- [6] J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*, first ed., The MIT Press, 1992.
- [7] H. Mühlenbein, G. Paas, From recombination of genes to the estimation of distributions I. Binary parameters, in: *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, PPSN IV*, Springer-Verlag, London, UK, 1996, pp. 178–187.
- [8] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, *IEEE Trans. Evol. Comput.* 10 (6) (2006) 646–657.
- [9] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE Trans. Evol. Comput.* 13 (2) (2009) 398–417.
- [10] R.M. Storn, K.V. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *J. Glob. Optim.* 11 (1997) 341–359.
- [11] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, *IEEE Trans. Evol. Comput.* 13 (5) (2009) 945–958.
- [12] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, *Science* 220 (1983) 671–680.
- [13] F. Glover, *Tabu Search*, Kluwer Academic Publishers, 1997.
- [14] Y.C. Liang, Y.C. Chen, Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm, *Reliab. Eng. Syst. Saf.* 92 (2007) 323–331.
- [15] W.D. Hillis, Co-evolving parasites improve simulated evolution as an optimization procedure, *Physica D* 42 (1990) 228–234.
- [16] P. Civicioglu, Artificial cooperative search algorithm for numerical optimization problems, *Inf. Sci.* 229 (2013) 58–76.
- [17] R.G. Reynolds, Cultural algorithms: theory and applications, in: *New Ideas in Optimization*, McGraw-Hill Ltd., UK Maidenhead, UK, England, 1999, pp. 367–378.
- [18] R.G. Reynolds, T.A. Kohler, Z. Kobti, The effects of generalized reciprocal exchange on the resilience of social networks: an example from the prehispanic Mesa Verde region, *Comput. Math. Organ. Theory* 9 (2003) 227–254.
- [19] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: *Proceeding IEEE International Conference Neural Network, Perth, Western Australia*, 1995, pp. 1942–1948.
- [20] F. Valdez, P. Melin, O. Castillo, An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms, *Appl. Soft Comput.* 11 (2) (2011) 2625–2632.
- [21] N.A. El-Hefnawy, Solving bi-level problems using modified particle swarm optimization algorithm, *Int. J. Artif. Intell.* 12 (2) (2014) 88–101.
- [22] M. Dorigo, M. Birattari, T. Stützle, U. Libre, D. Bruxelles, A.F.D. Roosevelt, Ant colony optimization-artificial ants as a computational intelligence technique, *IEEE Comput. Intell. Mag.* 1 (2006) 28–39.
- [23] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. B* 26 (1996) 29–41.
- [24] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, *IEEE Control Syst. Mag.* 22 (2002) 52–67.
- [25] P. Lucic, D. Teodorovic, Computing with bees: attacking complex transportation engineering problems, *Int. J. Artif. Intell. Tools* 12 (2003) 375–394.
- [26] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report TR06, Erciyes University, 2005.
- [27] M.S. Kiran, H. Hakli, M. Gunduz, H. Uguz, Artificial bee colony algorithm with variable search strategy for continuous optimization, *Inf. Sci.* 300 (2015) 140–157.
- [28] D. Simon, Biogeography-based optimization, *IEEE Trans. Evol. Comput.* 12 (2008) 702–713.
- [29] X.S. Yang, S. Deb, Cuckoo search via Levy flights, in: *Proceedings of World Congress on Nature & Biologically Inspired Computing*, IEEE Publications, USA, 2009, pp. 210–214.

- [30] X.S. Yang, A new metaheuristic bat-inspired algorithm, in: *Nature Inspired Cooperative Strategies for Optimization*, vol. 284, Springer, SCI284, 2010, pp. 65–74.
- [31] K.N. Krishnanand, D. Ghose, Detection of multiple source locations using a glowworm metaphor with applications to collective robotics, in: *Proc. of IEEE Swarm Intelligence Symposium*, IEEE Press, Piscataway, 2005, pp. 84–91.
- [32] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems, *Inf. Sci.* 183 (2012) 1–15.
- [33] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, *Inf. Sci.* 179 (2009) 2232–2248.
- [34] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, A.S. Paul, Gravitational search algorithm-based tuning of fuzzy control systems with a reduced parametric sensitivity, in: A. Gaspar-Cunha, et al. (Eds.), *Soft Computing in Industrial Applications*, AISC, vol. 96, Springer-Verlag, Berlin Heidelberg, 2011, pp. 141–150.
- [35] P.J. Bernhard, B. Webster, A local search optimization algorithm based on natural principles of gravitation, in: *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE'03)*, Las Vegas, Nevada, USA, 2003, pp. 255–261.
- [36] A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, *Inf. Sci.* 222 (2013) 175–184.
- [37] O.K. Erol, I. Eksin, A new optimization method: big bang-big crunch, *Adv. Eng. Softw.* 37 (2006) 106–111.
- [38] R.A. Formato, Central force optimization: a new metaheuristic with applications in applied electromagnetics, *Prog. Electromagn. Res.* 77 (2007) 425–491.
- [39] K.A. Tahatara S, A novel heuristic optimization method: charged system search, *Acta Mech.* 213 (2010) 267–289.
- [40] F.F. Moghaddam, R.F. Moghaddam, M. Cheriet, Curved space optimization: a random search based on general relativity theory, *arXiv*, preprint arXiv (2012) 1208–2214.
- [41] K.A. Khayatazar M., A new meta-heuristic method: ray optimization, *Comput. Struct.* 112 (2012) 283–294.
- [42] H. Du, X. Wu, J. Zhuang, Small-world optimization algorithm for function optimization, in: *Advances in Natural Computation*, Springer, 2006, pp. 264–273.
- [43] H. Shah-Hosseini, Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation, *Int. J. Comput. Sci. Eng.* 6 (2011) 132–140.
- [44] J. Sun, B. Feng, W.B. Xu, Particle swarm optimization with particles having quantum behavior, *IEEE Congr. Evol. Comput.* 1 (2004) 325–331.
- [45] D.Y. Tang, Y.M. Cai, J. Zhao, Y. Xue, A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous non-linear large scale problems, *Inf. Sci.* 289 (2014) 162–189.
- [46] M.S. Kiran, TSA: tree-seed algorithm for continuous optimization, *Expert Syst. Appl.* 42 (2015) 6686–6698.
- [47] S.A. Uymaz, G. Tezel, E. Yel, Artificial algae algorithm (AAA) for nonlinear global optimization, *Appl. Soft Comput.* 31 (2015) 153–171.
- [48] M. Crepinšek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, *ACM Comput. Surv.* 45 (3) (2013) 1–33.
- [49] W. DH, W.G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1997) 67–82.
- [50] M. Abercrombie, Contact inhibition in tissue culture, *In Vitro* 6 (1970) 128–140.
- [51] D.S. Coffey, Self organization, complexity and chaos: the new biology for medicine, *Nat. Med.* 4 (1998) 882–885.
- [52] Y. Jiao, S. Torquato, Emergent behaviors from a cellular automaton model for invasive tumor growth in heterogeneous microenvironments, *PLoS Comput. Biol.* 7 (2011) 1–14.
- [53] E.R. Fearon, B. Vogelstein, A genetic model for colorectal tumorigenesis, *Cell* 61 (1990) 759–767.
- [54] A.R. Kansal, S. Torquato, G.R. Harsh, E.A. Chiocca, T.S. Deisboeck, Simulated brain tumor growth using a three-dimensional cellular automaton, *J. Theor. Biol.* 203 (2000) 367–382.
- [55] M. Byrne, J.R. King, D.L.S. McElwain, L. Preziosi, A two-phase model of solid tumour growth, *Appl. Math. Lett.* 16 (2003) 567–573.
- [56] M.J. Dorie, R.F. Kallman, D.F. Rapachietta, D. van Antwerp, Y.R. Huang, Migration and internalization of cells and polystyrene microspheres in tumour cell spheroids, *Exp. Cell Res.* 141 (1982) 201.
- [57] R.M. Sutherland, Cell and environment interactions in tumor microregions: the multicell spheroid model, *Science* 240 (1988) 177–184.
- [58] M.J. Tindall, C.P. Please, Modelling the cell cycle and cell movement in multi-cellular tumour spheroids, *Bull. Math. Biol.* 69 (2007) 1147.
- [59] J.J. Casciari, S.V. Sotirchos, R.M. Sutherland, Variations in tumor cell growth rates and metabolism with oxygen concentration, glucose concentration, and extracellular pH, *J. Cell. Physiol.* 151 (1992) 386–394.
- [60] J.A. Sherratt, M.A.J. Chaplain, A new mathematical model for avascular tumour growth, *J. Math. Biol.* 43 (2001) 291–312.
- [61] D. Selmezi, S. Mosler, H. Flyvbjerg, Cell motility as persistent random motion: theories from experiments, *Biophys. J.* 89 (2005) 912–931.
- [62] G. Pettet, C.P. Please, M.J. Tindall, D. McElwain, The migration of cells in multi-cell tumor spheroids, *Bull. Math. Biol.* 63 (2001) 231–257.
- [63] T.S. Deisboeck, M.E. Berens, A.R. Kansal, S. Torquato, A. Rachamimov, et al., Patterns of self-organization in tumor systems: complex growth dynamics in a novel brain tumor spheroid model, *Cell Prolif.* 34 (2001) 115–134.
- [64] R.N. Mantegna, Levy walks and enhanced diffusion in Milan stock exchange, *Physica A* 179 (1991) 232–242.
- [65] R.N. Mantegna, Fast, accurate algorithm for numerical simulation of Levy stable stochastic process, *Phys. Rev. E* 5 (49) (1994) 4677–4683.
- [66] M.J. Kipper, H.K. Kleinman, F.W. Wang, New method for modeling connective-tissue cell migration: improved accuracy on motility parameters, *Biophys. J.* 93 (2007) 1797–1808.
- [67] W.M. Saltzman, *Tissue Engineering: Engineering Principles for the Design of Replacement Organs and Tissues*, Oxford University Press, New York, 2004.
- [68] Z.B. Hu, S.W. Xiong, Q.H. Su, Z.X. Fang, Finite Markov chain analysis of classical differential evolution algorithm, *J. Comput. Appl. Math.* 268 (2014) 121–134.
- [69] M.B. Robert, Pointwise properties of convergence in probability, *Stat. Probab. Lett.* 6 (1985) 315–316.
- [70] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, Technical Report, Nanyang Technological University, Singapore and KanGAL Report Number 2005005 (Kanpur Genetic Algorithms Laboratory, IIT Kanpur), 2005.5, 2005.
- [71] K. Tang, X. Yao, P.N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen, Z. Yang, “Benchmark Functions for the CEC’2008 Special Session and Competition on Large Scale Global Optimization,” University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL), China, Tech. Rep., 2007.
- [72] H. Wang, H. Sun, C. Li, S. Rahnamayan, J.-s. Pan, Diversity enhanced particle swarm optimization with neighborhood search, *Inf. Sci.* 223 (2013) 119–135.
- [73] N. Vecek, M. Mernik, M. Crepinsek, A chess rating system for evolutionary algorithms: a new method for the comparison and ranking of evolutionary algorithms, *Inf. Sci.* 277 (2014) 656–679.
- [74] T. Liao, D. Aydin, T. Stützle, Artificial bee colonies for continuous optimization: experimental analysis and improvements, *Swarm Intell.* 7 (2013) 327–356.
- [75] <https://sites.google.com/site/tlbocodes>
- [76] P. Civicioglu, E. Besdok, A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms, *Artif. Intell. Rev.* (2013), <http://dx.doi.org/10.1007/s10462-011-9276-0>, in press.
- [77] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Trans. Evol. Comput.* 10 (3) (2006) 281–295.
- [78] Q. Zhang, 2011, <http://dces.essex.ac.uk/staff/qzhang/> (accessed 10.06.13).
- [79] F. Neri, E. Mininno, G. Iacca, Compact particle swarm optimization, *Inf. Sci.* 239 (2013) 96–121.
- [80] M. Soleimanpour-moghadam, H. Nezamabadi-pour, M.M. Farsangi, A quantum inspired gravitational search algorithm for numerical function optimization, *Inf. Sci.* 267 (2014) 83–100.
- [81] E. Mininno, F. Cupertino, D. Naso, Real-valued compact genetic algorithms for embedded microcontroller optimization, *IEEE Trans. Evol. Comput.* 12 (2008) 203–219.
- [82] J.D. Gibbons, S. Chakraborti, *Nonparametric Statistical Inference*, 5th ed., Chapman & Hall, 2010.
- [83] M. Hollander, D. Wolfe, *Nonparametric Statistical Methods*, 2nd ed., Wiley-Interscience, 1999.
- [84] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of non-parametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.* 1 (2011) 3–18.
- [85] W.H. Lim, N.A.M. Isa, Bidirectional teaching and peer-learning particle swarm optimization, *Inf. Sci.* 280 (2014) 111–134.
- [86] A.Y.S. Lam, V.O.K. Li, J.J.Q. Yu, Real-coded chemical reaction optimization, *IEEE Trans. Evol. Comput.* 16 (2012) 339–353.
- [87] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.
- [88] H.G. Zhou, C.D. Yang, Using immune algorithm to optimize anomaly detection based on SVM, in: *Proceedings of IEEE International Machine Learning and Cybernetics Conference*, Dalian, China, 2006, pp. 4257–4261.
- [89] J. Bo, T. Yuchun, Z. Yang-Qing, L. Chung-Dar, I. Weber, Support vector machine with the fuzzy hybrid kernel for protein subcellular localization classification, in: *Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ’05)*, Reno, NV, 2005, pp. 420–423.
- [90] H.J. Shin, D.H. Eom, S.S. Kim, One-class support vector machine—an application in machine fault detection and classification, *Comput. Ind. Eng.* 48 (2) (2005) 395–408.
- [91] C.J. Merz, C.L. Blake, UCI Repository of Machine Learning Databases, 2015, <http://www.ics.uci.edu/~mlearn/MLRepository.html>
- [92] M.L. Xi, J. Sun, W.B. Xu, An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position, *Appl. Math. Comput.* 205 (2008) 751–759.
- [93] C.-J. Lin, 2015, <http://www.csie.ntu.edu.tw/~cjlin/>
- [94] X.Y. Zhang, D.Y. Qiu, F. Chen, Support vector machine with parameter optimization by a novel hybrid method and its application to fault diagnosis, *Neurocomputing* 149 (2015) 641–651.