Soft Comput
DOI 10.1007/s00500-013-1073-z

METHODOLOGIES AND APPLICATION

Root growth model: a novel approach to numerical function optimization and simulation of plant root system

Hao Zhang · Yunlong Zhu · Hanning Chen

Abstract This paper presents a general optimization model gleaned ideas from root growth behaviours in the soil. The purpose of the study is to investigate a novel biologically inspired methodology for complex system modelling and computation, particularly for optimization of higher-dimensional numerical function. For this study, a mathematical framework and architecture are designed to model root growth patterns of plant. Under this architecture, the interactions between the soil and root growth are investigated. A novel approach called "root growth algorithm" (RGA) is derived in the framework and simulation studies are undertaken to evaluate this algorithm. The simulation results show that the proposed model can reflect the root growth behaviours of plant in the soil and the numerical results also demonstrate RGA is a powerful search and optimization technique for higher-dimensional numerical function optimization.

Keywords Root growth · Simulation · Numerical function optimization · Higher-dimensional

Communicated by V. Kreinovich.

H. Zhang (✉) · Y. Zhu · H. Chen
Key Laboratory of Industrial Informatics, Shenyang Institute of Automation of Chinese Academy of Sciences,
Shenyang 110016, China
e-mail: zhanghao@sia.cn

Y. Zhu
e-mail: ylzhu@sia.cn

H. Chen
e-mail: chenhanning@sia.cn

H. Zhang
Graduate School of the Chinese Academy of Sciences,
Beijing 100039, China

1 Introduction

Nature ecosystem is a dynamic assemblage of organisms, interacting with each other and with the physical environment they live in. Today, the study of the relations between these elements has become an important research field since it is more and more related to economic issues in agriculture as well as to the protection of the environment. Nature ecosystems have also been the rich source of mechanisms for designing computational systems to solve difficult engineering and computer science problems. Modeling is an important tool for the comprehension of complex systems such as nature ecosystems and the model inspired from nature ecosystem is instantiated as an optimizer for numerical function and engineering optimization.

In this paper, we design a novel model inspired from growth behaviors of plant root system and the model is instantiated as a novel algorithm called "root growth algorithm" (RGA) for higher-dimensional numerical function optimization and simulation of plant root system. The main motivation of the proposed model is to make use of the main characteristics of plant root growth in the soil. In this work, to prove the approach is effective a comprehensive comparative study on the performances of well-known evolutionary and swarm-based algorithms for optimizing a set of numerical functions is presented. The other objective of the study is to simulate the interactive relationships between changing soil environment and the root growth. The results show that the root growth model is feasible and RGA is a powerful search and optimization approach.

This paper is structured as follows: Sect. 2 presents a brief overview of related work about computational root growth models and biologically inspired algorithms. Root growth model is presented in Sect. 3. Root system of plant is simulated in Sect. 4. In Sect. 5, experimental settings and

Published online: 21 June 2013

experimental results are given. Finally, Sect. 6 concludes the paper.

2 Related work

Computational root growth models or "virtual root system of plant" are increasingly seen as a useful tool for comprehending complex relationships between gene function, plant physiology, root system development, and the resulting plant form. Therefore accurate root growth models for simulation of root–soil interactions are of major significance. In the field of modeling root system of plant, there already exists a variety of elaborate approaches (Gerwitz and Page 1974; Pages et al. 1989; White et al. 2005; Lynch 2007; Hodge 2009; Leitner et al. 2010). Simple spatial models of root growth and development have been available for a long time (Gerwitz and Page 1974). In Pages et al. (1989), a quantitative three-dimensional dynamic model of the root system architecture is presented. The model takes into account current observations on the morphogenesis of the maize root system. There is now considerable evidence linking root architecture with water and nutrient acquisition efficiency (White et al. 2005; Lynch 2007; Hodge 2009). The structure and dynamics of root system architecture, however, is complex, and architectural modelling has resulted from the need to incorporate space into models and to take account of the biophysical interactions taking place between roots and their environment. In Leitner et al. (2010), a dynamic root system growth model is proposed based on L-Systems. The modular approach is developed to root growth and architecture modelling with a special focus on soil root interactions. The approaches model root systems from different perspectives. However, the complexity of the root–soil system requires an accurate and detailed description not only of each subsystem, but also of their mutual linkage and influence.

In the past few decades, in pursuit of finding solution to the optimization problems many researchers have been drawing inspiration from the nature (Eberhart et al. 2001). A lot of such biologically inspired algorithms have been developed, such as a basic genetic algorithm (GA), particle swarm optimization (PSO), differential evolution (DE) and so on. The five components are included in a basic GA. They are a fitness evaluation unit, a random number generator and genetic operators for reproduction, crossover and mutation operations (Holland 1975). In PSO, a population of particles starts to move in search space by following the current optimum particles and changing the positions in order to find out the optimal solution. The positions of these particles refer to possible solutions of the function to be optimized.

Table 1 The relationship between biology behaviors and algorithms

Algorithms	Behaviors
GA	The genetic feature in the biology
PSO	Bird flocking behavior
CO	The foraging behavior of ants
AIS	Antibody and antigen in immune system of biology
ABC	The foraging behavior of honeybees
PGSA	Plant growth behavior

Evaluating the function by the particles' positions provides the fitness values of those solutions (Kennedy and Eberhart 1995). The DE algorithm which is also a population-based algorithm is like genetic algorithms using the similar operators: crossover, mutation and selection. The main difference between them in constructing better solutions is that DE relies on mutation operation while genetic algorithm relies on crossover (Price and Storn 1995). There are still many other algorithms based on biology behaviors, such as evolutionary programming (EP) (Fogel et al. 1965), ant colony optimization (ACO) (Dorigo et al. 1991), artificial immune system (AIS) (De Castro and Von Zuben 1999), artificial bee colony (ABC) (Karaboga and Basturk 2008) and so on. The relationship between these biology behaviors and algorithms listed in Table 1. There are some improved hybrid approaches based on swarm intelligence for optimization. The hybrid FPSO + FGA approach that combines PSO and GA using fuzzy logic to integrate the results of both methods and for parameters tuning is a useful tool in solving complex optimization problems (Valdez et al. 2011). We should note that there are some algorithms based on plant growth theory for numerical function. Plant growth simulation algorithm (PGSA) is a powerful evolutionary algorithm simulating plant growth that has been proposed for solving the global optimization problem of integer programming (Tong et al. 2005). Another optimization algorithm, plant growth optimization (PGO) is presented on the basis of the plant growth characteristics including leaf growth, branching, phototropism and spatial occupancy (Cai et al. 2008). These algorithms with their stochastic means are well equipped to handle such problems. However, the algorithms inspired from plant root models are very few.

In this paper, we present a root growth model inspired from growth behaviors of plant root system. The model can be formulated as an optimization algorithm called RGA for higher-dimensional numerical function optimization and simulation of root growth patterns of plant. The two self-adaptive strategies in accordance with growth behaviours of root hairs are employed in RGA to improve the performance of the algorithm.

3 Proposed root growth model

3.1 Description for growth behaviors of root system

A plant begins from its seed and the growth of root is indispensable for plant growth. All roots of a plant can be seen as a system composed by a large number of root hairs and tips. At the end of the 1960s, Lindenmayer introduced Transformational-generative grammar into biology and developed a variant of a formal grammar, namely L-Systems, most famously used to model the growth processes of plant development. L-Systems are based on simple rewriting rules and branching rules and successfully make a formal description for the plant growth. We use L-Systems to describe growth behavior of root system as follows (Sheng and Lei 1995):

(1) A seed germinates in the soil, partly becoming stems of plant above the earth's surface. The other part grows down, becoming root system of plant. New root hairs grow from the root tips.
(2) More new root hairs grow from the root tips of old root hairs. The behavior of root system which is repeated is called as branching of the root tips.
(3) Most root hairs and root tips are similar to each other. The entire root system of plant has self-similar structure. The root system of each plant is composed of numerous root hairs and root tips with similar structure.

3.2 Plant morphology

The impact of roots and rhizosphere traits on plant resource efficiency is important. The uneven concentration of nutrients in the soil makes root hairs growing towards different directions. This characteristic of root growth relates to the morphogenesis model in biological theory. The formation of the model could be considered as a complex process that cells are differentiated and generate new spatial structure with a clear definition. When the rhizome of root system grows, three or four growing points with different rotation directions will be generated at each root tip. The rotation diversifies the growth direction of the root tip. Root tips from which root hairs germinate contain undifferentiated cells. These cells are considered as fluid bags in which there are homogeneous chemical constituents. One of chemical constituents is a version of the growth hormone, called as morphactin. The morphactin concentration is a parameter of the morphogenesis model. The parameter changes between 0 and l. The morphactin concentration determines whether cells start to divide. When cells start to divide, root hairs appear (Mainzer 1997).

With regard to the process of root growth, there have been the following conclusions in biology:

(1) If root system of plant have more than one root tip, which root tips could germinate root hairs depends on their morphactin concentration. The probability of new root hairs germinating is higher from root tips with larger morphactin concentration than root tips with less ones.
(2) The morphactin concentration in cells is not static, but depends on its surroundings, in other words, the spatial distribution of nutrients in the soil. After new root hairs germinate and grow, the morphactin concentration will be reallocated among new root tips in line with the new concentration of nutrients in the soil.

In order to simulate the above process, it is assumed that the sum of the morphactin concentration of is constant (considered as 1) in the morphactin state space of the multi-cellular closed system. If there are n root tips x_i $(i = 1, 2, \ldots, n)$ which are D-dimensional vectors, the morphactin concentration of any cell is defined as E_i $(i = 1, 2, \ldots, n)$. The morphactin concentration of each root tip can be expressed as:

$$E_i = \frac{1/f\,(x_i)}{\sum_{i=1}^{n} 1/f\,(x_i)} \tag{1}$$

where $f(*)$ is objective function which represents the spatial distribution of nutrients in the soil. In expression (1), the morphactin concentration of each root tip is determined by the relative position of each point and environmental information (objective function value) at this position. Therefore, n root tips correspond to n morphactin concentration value. When new root hairs germinate, the morphactin concentration may be changed.

3.3 Branching of the root tips

Branching of the root tips in the root growth model proposed is important for the simulation and the instantiated algorithm. There are four rules for branching of the root as follow:

(1) Plant growth begins from a seed.
(2) In each cycle of growth process, some excellent root tips which have larger morphactin concentration values (The fitness values in the instantiated algorithm) are selected to branch.
(3) The distance should not be close between the selected root tips in order to make spatial distribution of the root system wider and increase the diversity of the fitness values.
(4) If the number of the root tips selected equals the predefined value, the loop of the selection process terminates.

The flowchart of the selection process of the root tips is showed in Fig. 1. In order to produce a new growing point

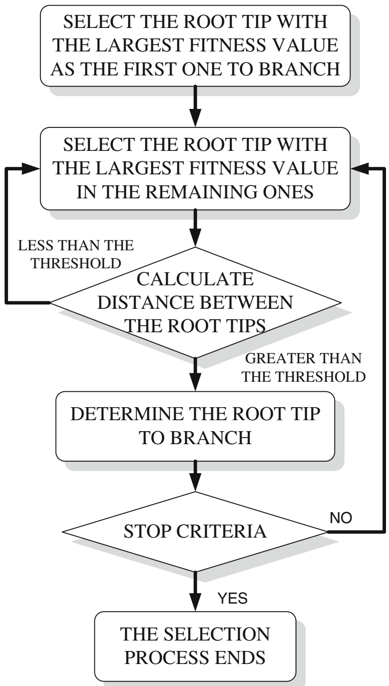

Fig. 1 The selection process of the root tips

from the old root tip in memory, the proposed model uses the following expression:

$$pg_{lj} = \begin{cases} x_{ij} + (2 \times \delta_{ij} - 1) & j = k \\ x_{ij} & j \neq k \end{cases} \tag{2}$$

where $k \in \{1, 2, \ldots, D\}$ are randomly chosen indexes and $j \in \{1, 2, \ldots, D\}$. pg_l $(i = 1, 2, \ldots, S)$ are S new growing points. δ_{ij} is a random number between $[-1, 1]$.

3.4 Root hair growth

After new growing points are produced, root hairs begin to grow from these growing points. Root hair growth depends on its growth angle and growth length. The growth angle is a vector for measuring the growth direction of root hair. The growth angle of each root hair $\varphi_i (i = 1, 2, \ldots, n)$ which is produced randomly can be expressed as:

$$(\phi_1, \phi_2, \ldots, \phi_D) = rand(D) \tag{3}$$

$$\varphi_i = \frac{(\phi_1, \phi_2, \ldots, \phi_D)}{\sqrt{\phi_1^2 + \phi_2^2 + \cdots + \phi_D^2}} \tag{4}$$

The growth length of each root hair is defined as δ_i $(i = 1, 2, \ldots, n)$ which is an important parameter in the root growth model. Some strategies of tuning the parameter can produce multiple versions of the root growth model. After growing, a new root tip is produced by the following expression:

$$x_i = x_i + \delta_i \varphi_i \tag{5}$$

In order to simulate the trophotropism of root system, some rules are defined as follows:

(1) If morphactin concentration (fitness) of a new root tip is better than old one in the same cycle t, the root tip will continue to grow. A new root tip in the inner loop can be expressed as:

$$x_i^t = x_i^t + \delta_i \varphi_i \tag{6}$$

But the number of iterations in the inner loop is a predefined value. While the number of iterations in the inner loop equals a predefined value, the inner loop stops.

(2) If morphactin concentration of a new root tip is worse than old one in the same cycle t, the root tip will stop growing and $t = t + 1$. A new root tip can be expressed as:

$$x_i^{t+1} = x_i^{t+1} + \delta_i \varphi_i \tag{7}$$

3.5 Root growth algorithm

The root growth model proposed is instantiated as RGA for simulation of root system of plant and higher-dimensional numerical function optimization. The threshold of the distance between root tips and the growth length of each root hair are important parameters for RGA. The flowchart of the RGA is presented in Fig. 2. The pseudocode for the RGA is listed in Table 2.

4 Simulation of root system of plant

4.1 Simulating root–soil interaction

As the first stage of a comprehensive root growth model, the objective of the study was to simulate the interactive relationships between changing soil environment and the root growth. In the simulation, Ackley function is used as soil environment in computer. Ackley function is presented in Sect. 4.1 and the setting of the parameters is the same as one in Sect. 4.2. Figures 3 and 4 shows the simulation result of root–soil interaction. The color changes in the simulation of soil indicate the uneven distribution of nutrients in the soil. The root system is represented by points and a point represents a root tip. It is simulated successfully that the roots of plant grow towards the directions with the higher concentration of nutrients. It is also illustrated that RGA can obtain optimal solution for numerical function optimization in Figs. 3 and 4.

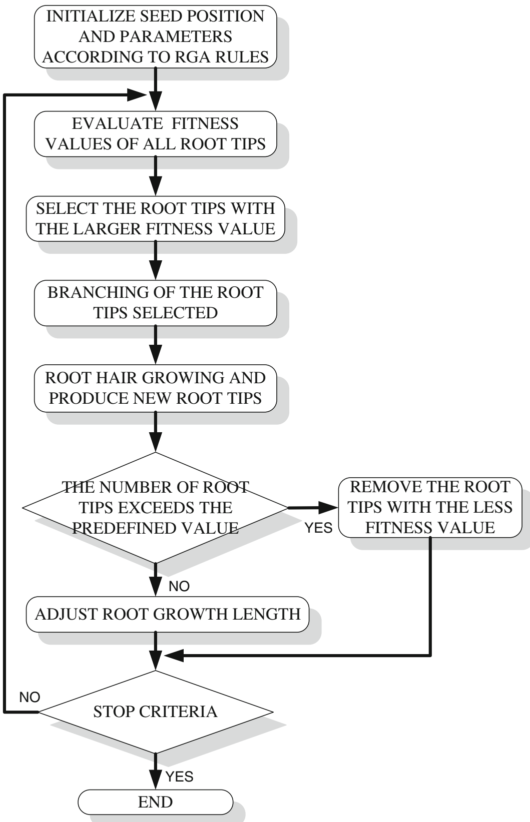

Fig. 2 The flowchart of the RGA

Table 2 Pseudocode for the RGA

```
Set t: = 0;
INITIALIZE. Randomize position of a seed and parameters.
WHILE (the termination conditions are not met)
  FOR (each root tip x)
    Select S root tips pg
  END FOR
  FOR (each root tip selected pg)
    Produce n new growing points using Eq. (2)
  END FOR
  FOR (each root tip x)
    Produce new root tips using Eq. (5)
  END FOR
  Tune the parameter δ
Set t: = t + 1;
END WHILE
```

4.2 Simulating self-adaptive growth of root hairs

The self-adaptive growth is a significant characteristic for the growth of plant root system. Simulating self-adaptive growth of root hairs is important for learning relationships between the nutrient concentration of the soil and the growth length of root hairs. The self-adaptive strategies in accordance with growth behaviours of root hairs can improve the performance of RGA. So the two self-adaptive strategies are proposed to optimize numerical functions and simulate self-adaptive growth of root hairs in this section.

Rastrigin function is used as soil environment in computer. Rastrigin function is presented in Sect. 4.1 and the setting of the parameters is the same as one in Sect. 4.2. The growth length of each root hair δ_i $(i = 1, 2, \ldots, n)$ is the master variable for self-adaptive growth of root hairs. The variable which is an important parameter in root growth model can produce multiple versions of the model and the proper tuning of the parameter can improve the result of the instantiated algorithm for optimization. The two strategies of self-adaptive growth are proposed as follows:

(1) The growth length of all root hairs is the same when initialized. At set intervals of cycle times, root growth length decreases by the specific rate α. If the fitness value of a root tip is not changed at set intervals of cycle times, its root growth length increases by the specific rate β. But root growth length cannot exceed its initial value after increasing. The pseudocode of the first strategy of self-adaptive growth is listed in Table 3. Figure 5 shows the simulation of the first self-adaptive growth strategy. A point represents the growth length of a root tip. The distribution of root tips to polarization is obvious after some generations. One part of root tips grow toward the direction with the best concentration of nutrients and the other part exploit more extensive region of the soil. The simulation indicates the trophotropism of root system and the relationship competition between root hairs. The roots of plant grow towards the directions with the higher concentration of nutrients, but if too many root hairs concentrated in the same area with the higher concentration of nutrients, some week root hairs leave the area. The simulation also shows the instantiated algorithm, RGA, for optimization can find the optimal solution. The variation of RGA with the first self-adaptive growth strategy is named as “RGA-α”.

(2) The growth length of all root hairs change as their fitness values change. The pseudocode and the formula of the second strategy of self-adaptive growth are listed in Table 4 in which τ is a nonnegative number. Figure 6 shows the simulation of the second self-adaptive growth strategy. The roots of plant grow towards the same direction with the high concentration of nutrients. From the view point of the optimization, the second self-adaptive growth strategy can make all points moving toward the area with the largest fitness value and then these points move around back and forth for intensive search. So

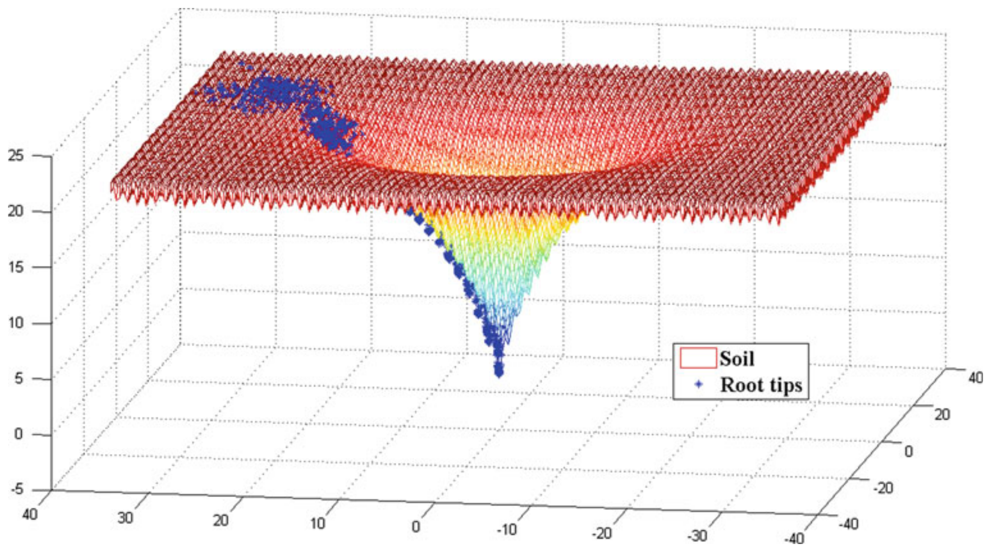

Fig. 3 The simulation of root–soil interaction

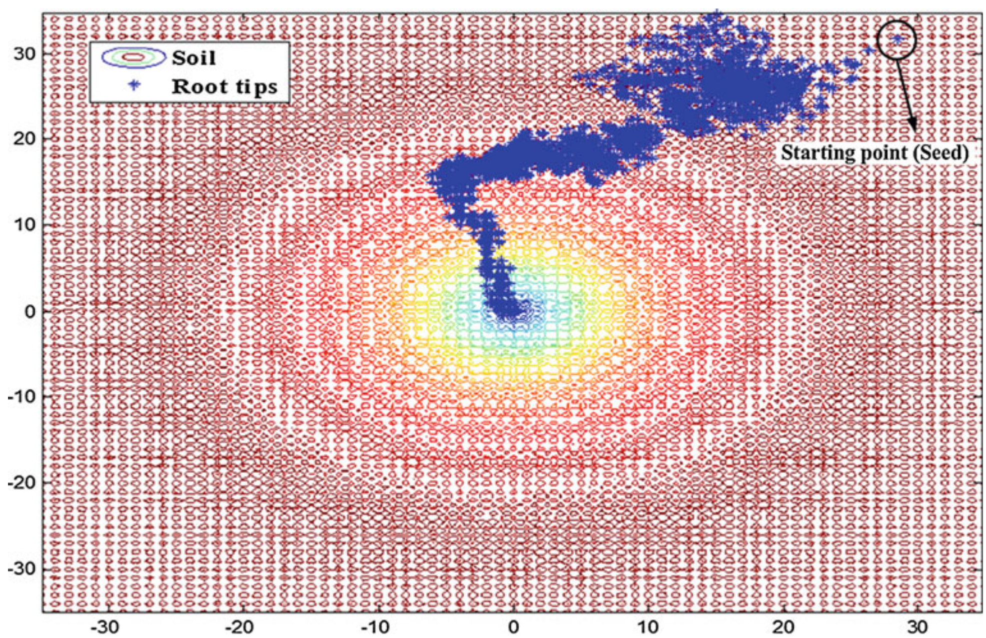

Fig. 4 The contour chart of root–soil interaction

RGA using the strategy of self-adaptive growth can obtain better optimal solution easily and quickly. The variation of RGA with the second self-adaptive growth strategy is named as "RGA-τ".

5 Numerical experiments for optimization

5.1 Benchmark functions

The set of benchmark functions contains eight functions that are commonly used in evolutionary computation literature (Krink et al. 2002; Shi and Ebrehart 1998; Shi and Eberhart 1999; Karaboga and Akay 2009) to show solution quality and convergence rate. The first four functions are unimodal problems and the remaining functions are multimodal. The functions are listed below. The dimensions, initialization ranges, global optimum, and the criterion of each function are listed in Table 5.

(1) Sphere function

$$f_1(x) = \sum_{i=1}^{D} x_i^2 \tag{8}$$

Table 3 Pseudocode for the first self-adaptive growth strategy

```
Set: δi = Initial value
FOR each generation
  FOR each point xi
    IF no change count > default count
      Set: δi = δi×α
      IF δi > Initial value
        Set: δi = Initial value
      END IF
    END IF
  END FOR
  IF λ | iteration
    Set: δi = δi×β
  END IF
END FOR
```

Table 4 Pseudocode for the second self-adaptive growth strategy

```
Set: δi = Initial value
FOR each generation
  FOR each point xi
    Set: δi = |Ei| / |Ei + τ|
  END FOR
END FOR
```

(2) SumSquares function

$$f_2(x) = \sum_{i=1}^{D} i x_i^2 \tag{9}$$

(3) Rosenbrock function

$$f_3(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2) \tag{10}$$

(4) Schwefel 2.22 function

$$f_4(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i| \tag{11}$$

(5) Rastrigin function

$$f_5(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10) \tag{12}$$

(6) Schwefel function

$$f_6(x) = \sum_{i=1}^{D} -x_i \sin(\sqrt{|x_i|}) \tag{13}$$

(7) Ackley function

$$f_7(x) = 20 + e - 20e^{\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)} - e^{\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)} \tag{14}$$

(8) Griewank function

$$f_8(x) = \frac{1}{4{,}000}\left(\sum_{i=1}^{D} x_i^2\right) - \left(\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)\right) + 1 \tag{15}$$

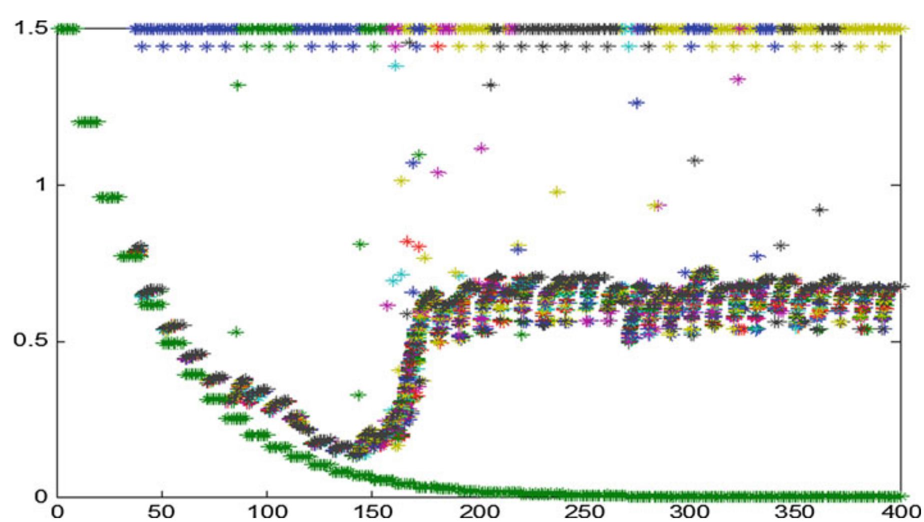

Fig. 5 The simulation of the first self-adaptive growth strategy

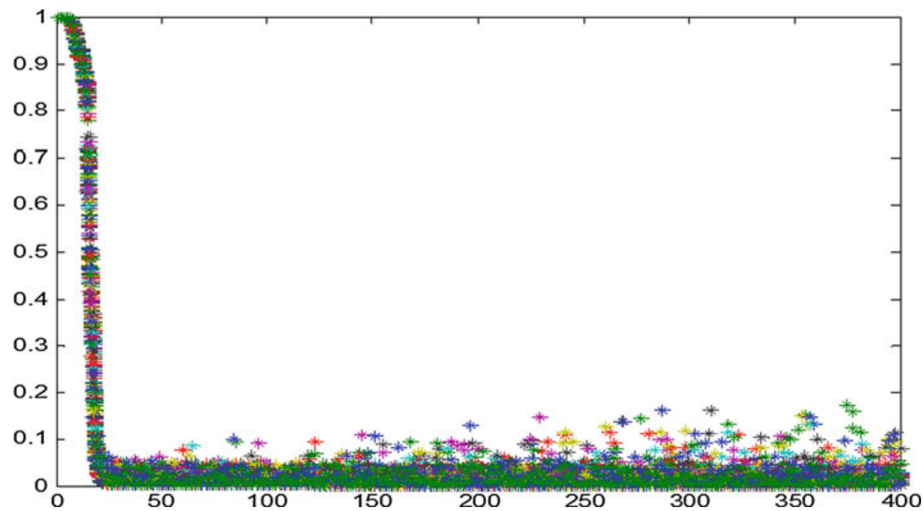

Fig. 6 The simulation of the second self-adaptive growth strategy

Table 5 Parameters of the benchmark functions

Function	Dimensions	Initial range	Minimum
Sphere	30 and 45	$[-100, 100]^D$	0
SumSquares	30 and 45	$[-10, 10]^D$	0
Rosenbrock	30 and 45	$[-30, 30]^D$	0
Schwefel 2.22	30 and 45	$[-10,10]^D$	0
Rastrigin	30 and 45	$[-10,10]^D$	0
Schwefel	30 and 45	$[-500, 500]^D$	−12,569.5
Ackley	30 and 45	$[-32.768, 32.768]^D$	0
Griewank	30 and 45	$[-600, 600]^D$	0

5.2 Settings

The proposed algorithm RGA with the two self-adaptive strategies come into being two variations, RGA-α and RGA-τ. RGA-α and RGA-τ are compared with GA, DE and PSO and they are run in MATLAB 7 on a Personal Computer with a Pentium Dual Core 2.20 GHz processor and 1 GB memory.

For the experiments, the value of the common parameter, total evaluation number, used in each algorithm was chosen to be the same. Population size in GA, DE and PSO was 50. The maximal number of fitness function evaluations was 2,000,000 for all functions.

For GA, a binary coded standard GA were employed. The rate of single point crossover operation was 0.8. Mutation rate was 0.01. The selection method was stochastic uniform sampling technique. Generation gap is the proportion of the population to be replaced. Chosen generation gap value in experiments was 0.9. For DE, F was set to 0.5. Crossover rate was chosen to be 0.9 as recommended in Corne et al. (1999). For PSO, inertia weight was 0.6 and cognitive and social components were both set to 1.8 (Vesterstrom and Thomsen 2004).

The parameter settings of RGA-α and RGA-τ for simulation and optimization are listed in Table 6. All parameter values have been tested many times to obtain better simulation and optimal solution and then were used.

5.3 Numerical results and comparison

The GA, DE, PSO RGA-α and RGA-τ algorithms are compared on eight functions with 30 and 45 dimensions described

Table 6 Parameter setting of RGA-α and RGA-τ

Simulation settings	Optimization settings	Value
The number of seed	The number of initial population	1
The maximum number of root tips	The maximum number of population	100
The initial length of each root fair δ_i	The initial value of the parameter δ_i	1
The distance threshold between root tips	A parameter	1
Branching number of each root tip selected	A parameter	4
S	S	4
α	α	1.2
β	β	0.8
τ	τ	1–1,000
λ	λ	200

Table 7 Comparison of results with 30D obtained by GA, DE, PSO, RGA-α and RGA-τ

Function	GA	DE	PSO	RGA-α	RGA-τ
Sphere					
Mean	0.92505	22.6724	2.25409e−008	0.212980	**1.09852e−008**
Std	1.29811	28.6836	1.76046e−008	0.546753	**1.61596e−008**
Min	0.50220	3.69461	1.06366e−008	6.23595e−003	**5.90561e−024**
Max	3.19916	55.6673	4.27632e−008	1.04037	**2.95410e−008**
SumSquares					
Mean	0.0423408	366.243	**1.15988e−003**	0.0133962	4.07359e−003
Std	6.91778e−003	144.776	6.23641e−004	6.56423e−003	**5.53186e−004**
Min	0.0372955	249.874	**9.67914e−004**	9.37678e−003	3.49710e−003
Max	0.0502266	528.369	**3.28971e−003**	0.0167695	4.60007e−003
Rosenbrock					
Mean	4,300.21	11,859.6	36.0147	1,440.22	**21.2087**
Std	638.258	5,221.57	37.1823	1,837.75	**0.808404**
Min	3,662.62	7,170.21	**12.1717**	150.406	20.3453
Max	4,939.16	17,486.4	78.8579	3,241.94	**21.9481**
Schwefel 2.22					
Mean	0.194223	21.3174	0.0958865	0.122179	**0.0549662**
Std	0.032346	2.48856	0.0709508	0.0209892	**1.99853e−003**
Min	0.162737	19.4056	0.0714948	0.0979435	**0.0460287**
Max	0.227365	24.1311	0.176012	0.134432	**0.0675534**
Rastrigin					
Mean	87.1626	156.074	28.5221	17.2605	**4.28574e−004**
Std	12.4927	70.4707	9.02802	6.67254	**9.69091e−005**
Min	79.1588	96.8458	21.8891	12.8178	**3.35807e−004**
Max	101.558	234.011	38.8034	24.9334	**5.29153e−004**
Schwefel					
Mean	**−8,793.98**	−8,746.91	−5,251.66	−7,808.34	−8,487.84
Std	**587.525**	1,328.42	1,176.508	1,060.61	1,041.19
Min	−9,285.05	**−10,175.6**	−6,469.01	−8,558.30	−9,666.01
Max	**−8,143.08**	−7,548.97	−4,935.01	−7,058.38	−7,691.38
Ackley					
Mean	4.90392	12.8558	**6.44018e−007**	3.03418	3.27635e−004
Std	0.42856	1.06169	**4.68484e−007**	0.655751	1.28737e−005
Min	4.57918	11.6323	**2.42885e−007**	2.54542	1.57042e−004
Max	5.38955	13.5347	**1.1589e−006**	3.77940	4.34591e−004
Griewank					
Mean	1.81473	1.38865	0.0338252	0.659995	**3.70074e−003**
Std	0.20679	0.29033	0.0483038	0.445041	**5.22626e−003**
Min	1.58355	1.05381	4.98359e−006	0.336426	**3.53763e−007**
Max	1.98206	1.57037	0.0891463	1.17155	**4.17191e−003**

Bold values indicate the best values obtained by the five algorithms

in the previous section and are listed in Table 5. Each of the experiments in this section was repeated 30 times. The mean values, the standard deviation, the minimum values and the maximum values produced by the algorithms have been recorded in Tables 7 and 8. The best values obtained by the five algorithms for each function are marked as bold. The mean best function value profiles are shown in Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 and 22.

5.3.1 Analyzing numerical results with 30D

As shown in Table 7, RGA-τ algorithm can obtain better performance than the other algorithms on Sphere, Rosenbrock,

Table 8 Comparison of results with 45D obtained by GA, DE, PSO, RGA- α and RGA-τ

Function	GA	DE	PSO	RGA-α	RGA-τ
Sphere					
Mean	37.5669	290.36	2.27609e−006	12.8228	**1.93873e−006**
Std	4.4795	184.704	**2.62744e−007**	5.36133	1.22332e−006
Min	32.7708	92.9845	2.0238e−006	9.52884	**7.03555e−007**
Max	41.6423	459.032	**2.54817e−006**	19.0092	3.89349e−006
SumSquares					
Mean	2.06001	173.516	**7.25119e−006**	463.418	9.84732e−006
Std	0.687524	49.7097	6.5067e−006	410.908	**2.94736e−006**
Min	1.59746	118.217	2.12194e−006	20.1757	**1.98714e−006**
Max	2.85009	214.492	**1.45703e−005**	831.659	2.84367e−005
Rosenbrock					
Mean	1,532.26	473,057	41.0732	1,511.97	**34.3239**
Std	901.440	645,794	3.8414	2,418.96	**2.11109**
Min	612.841	80,480.9	36.6457	39.9037	**33.9825**
Max	2,414.59	1.2184e+006	43.52	4,303.76	**38.2233**
Schwefel 2.22					
Mean	1.68119	1.73382	0.00681453	21.5694	**0.00268347**
Std	0.210885	1.64727	0.000966737	6.27545	**0.000429901**
Min	1.53852	0.725999	0.00321511	17.6566	**0.00222769**
Max	1.92331	3.63478	0.0120881	28.8077	**0.0030817**
Rastrigin					
Mean	128.885	238.902	74.6218	401.69	**0.55754**
Std	22.188	4.86127	4.33688	277.133	**0.943137**
Min	104.3864	233.291	69.647	82.1326	**0.0130062**
Max	147.629	241.843	77.6067	576.123	**1.64658**
Schwefel					
Mean	−10,057.2	**−13,540.4**	−9,864.92	−11,698.2	−11,494.2
Std	2,512.79	**1,252.87**	1,768.21	1,617.28	1,658.22
Min	−12,957.7	**−14,981**	−11,194.8	−13,460.7	−13,404.1
Max	−8,543.17	**−12,705.2**	−7,858.28	−10,282.5	−10,421.5
Ackley					
Mean	2.52123	14.486	**0.00346664**	2.02747	0.00806774
Std	0.258816	1.58301	**0.00164182**	1.32354	0.00285592
Min	2.37049	12.6661	**0.00175844**	1.19584	0.00348745
Max	2.81999	15.5435	**0.00503285**	3.55371	0.010199283
Griewank					
Mean	1.30435	8.80927	0.0249269	0.189923	**0.00579939**
Std	0.1203	3.24603	0.042542	0.223845	**0.0050909**
Min	1.19407	5.3781	3.42295e−004	0.025093	**8.80639e**−005
Max	1.43262	11.8314	0.0740501	0.444761	**0.0098603**

Bold values indicate the best values obtained by the five algorithms

Schwefel 2.22, Rastrigin, and Griewank functions with 30D. PSO algorithm shows better performance on SumSquares and Ackley functions. GA and DE obtain better optimal solution on Schwefel function. RGA-α can also obtain better performance on every function.

Sphere and SumSquares function are two unimodal variable–separable functions. On the two functions, PSO and RGA-τ obtained satisfying results. However, RGA-τ algorithm showed the best performance on Sphere and the search performance order is RGA-τ > PSO > RGA-α > GA > DE. PSO algorithm shows the better performance on SumSquares and the search performance order is PSO > RGA-τ > RGA-α > GA > DE.

Rosenbrock and Schwefel 2.22 functions are two unimodal non-separable functions. On the two functions, RGA-τ algorithm showed the best performance. On Rosenbrock function, PSO was a little worse than RGA-τ. GA and DE cannot obtain better performance on this function. On Schwe-

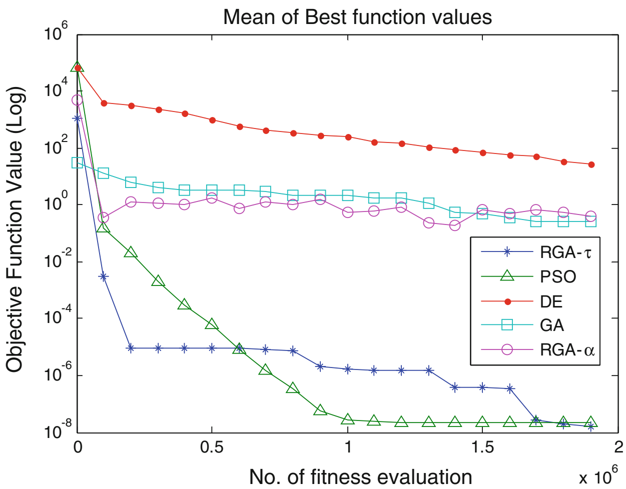

Fig. 7 f_1, Sphere with 30D

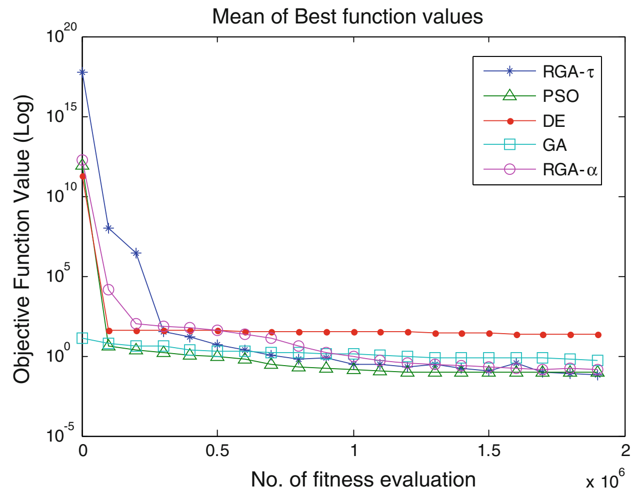

Fig. 10 f_4, Schwefel 2.22 with 30D

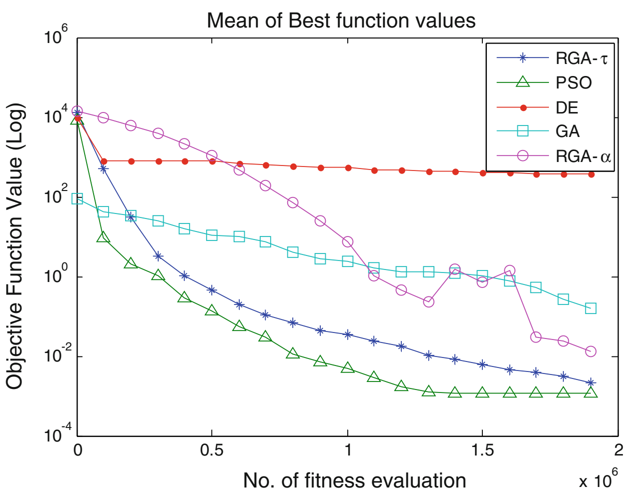

Fig. 8 f_2, SumSquares with 30D

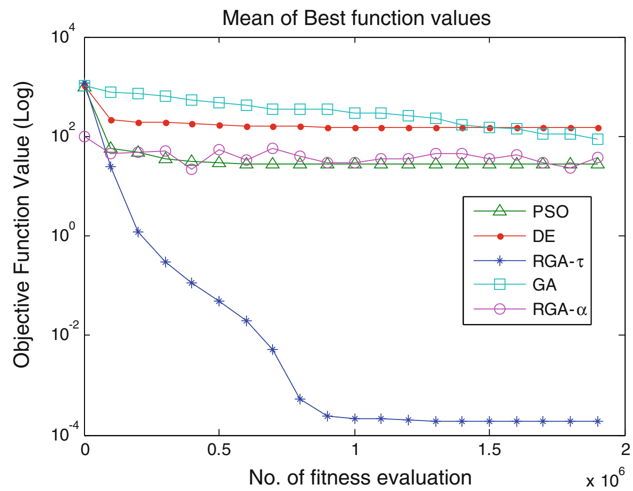

Fig. 11 f_5, Rastrigin with 30D

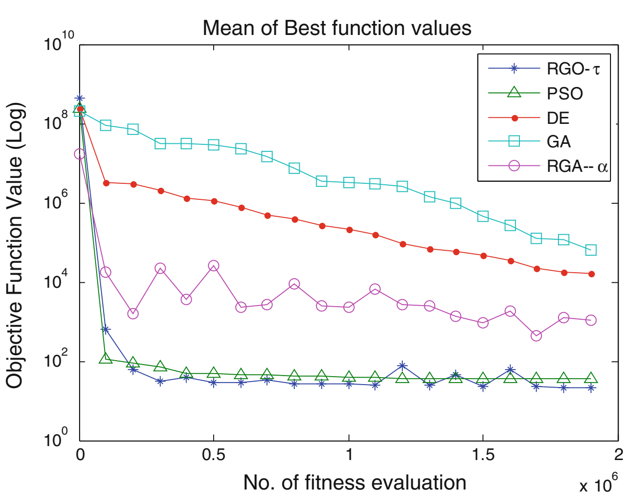

Fig. 9 f_3, Rosenbrock with 30D

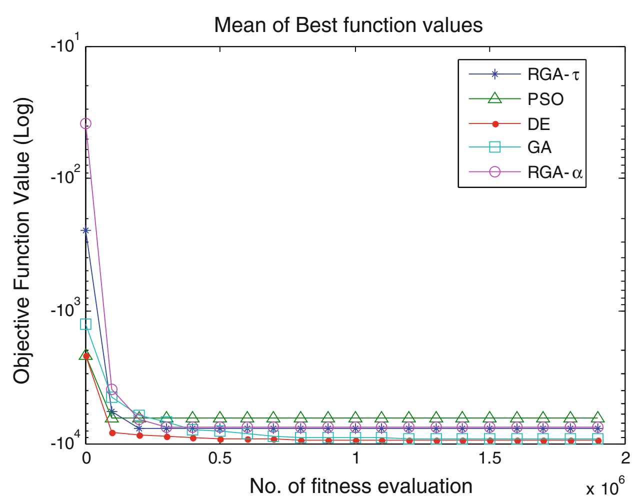

Fig. 12 f_6, Schwefel with 30D

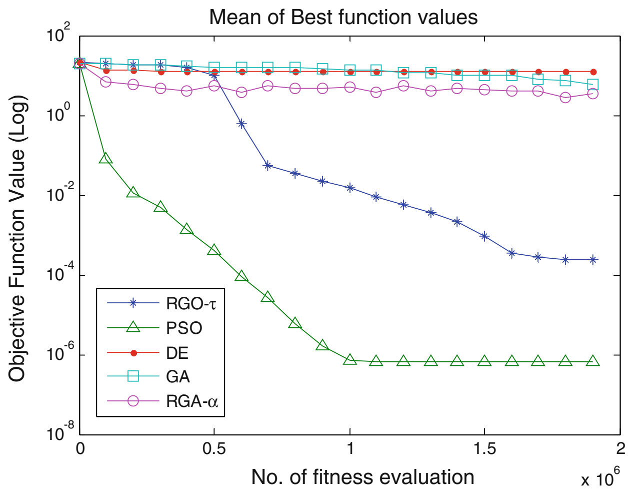

Fig. 13 f_7, Ackley with 30D

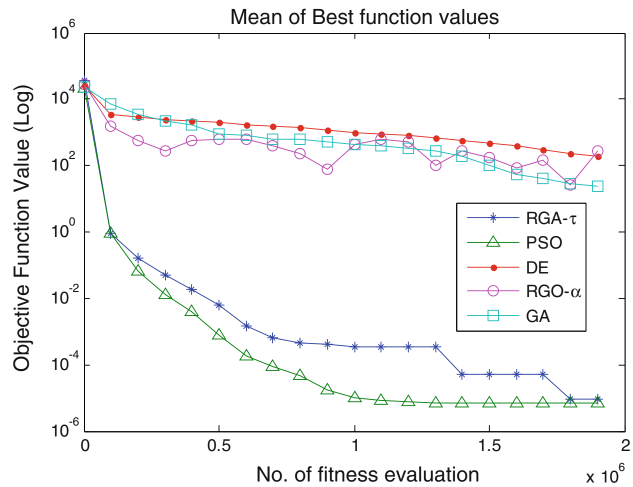

Fig. 16 f_2, SumSquares with 45D

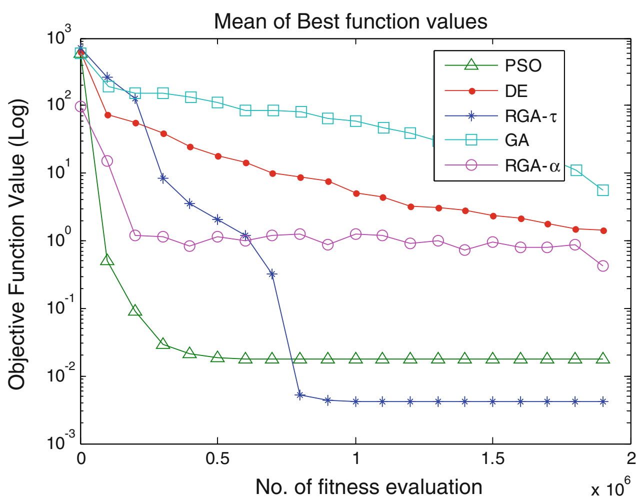

Fig. 14 f_8, Griewank with 30D

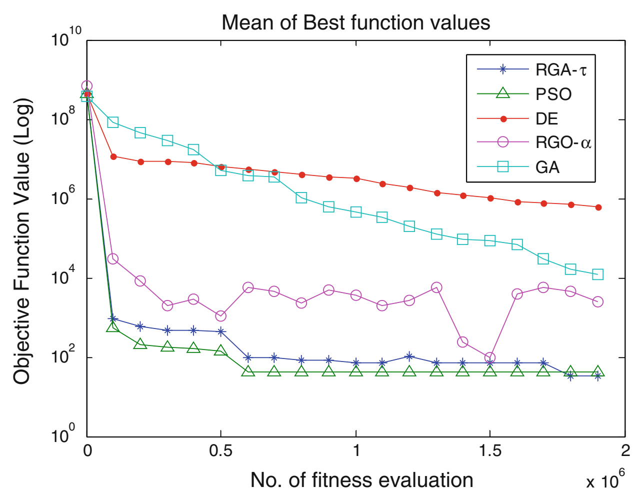

Fig. 17 f_3, Rosenbrock with 45D

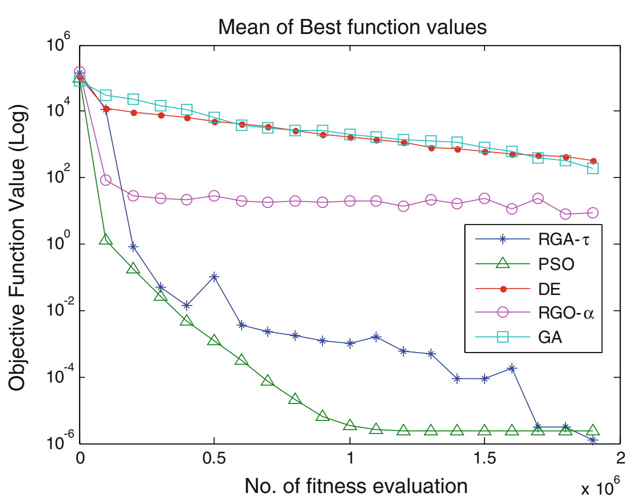

Fig. 15 f_1, Sphere with 45D

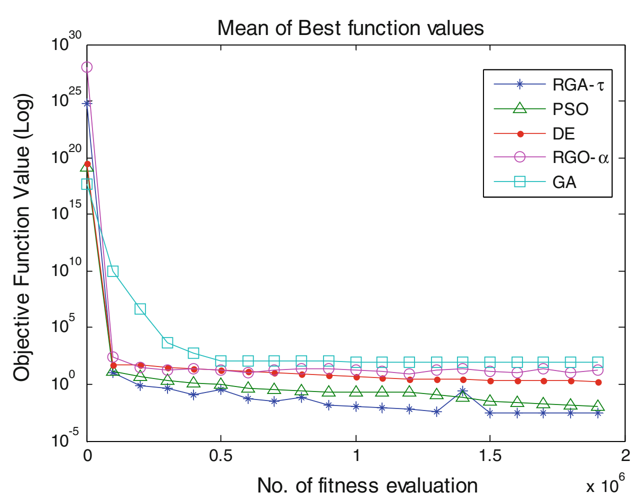

Fig. 18 f_4, Schwefel 2.22 with 45D

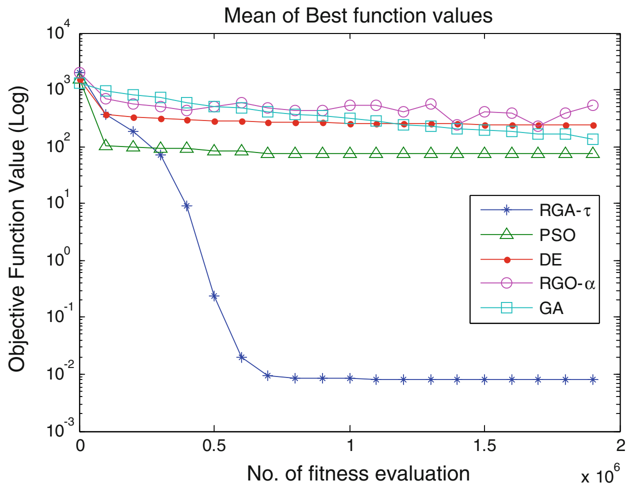

Fig. 19 f_5, Rastrigin with 45D

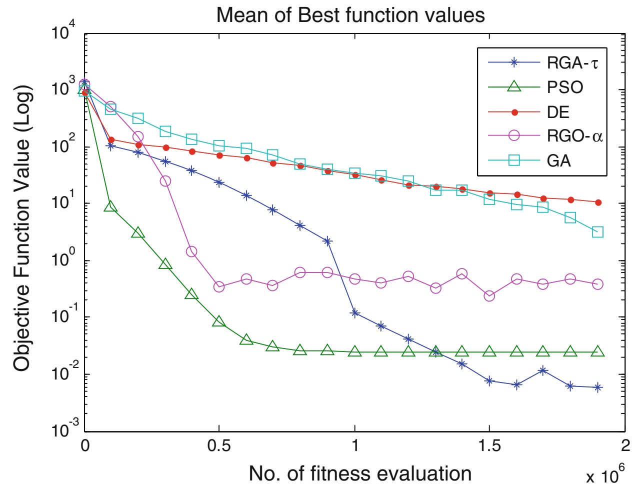

Fig. 22 f_8, Griewank with 45D

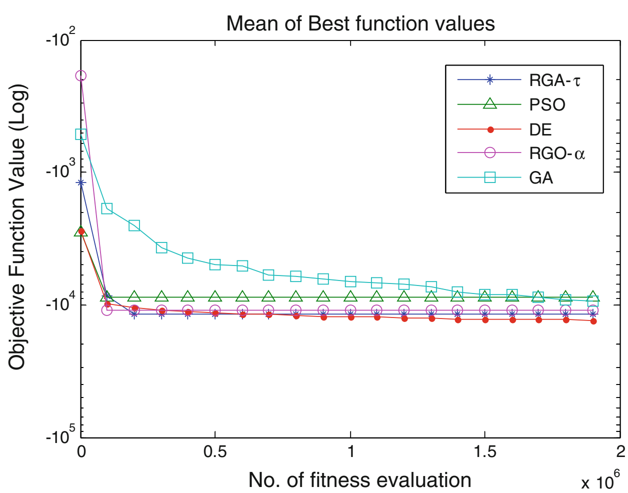

Fig. 20 f_6, Schwefel with 45D

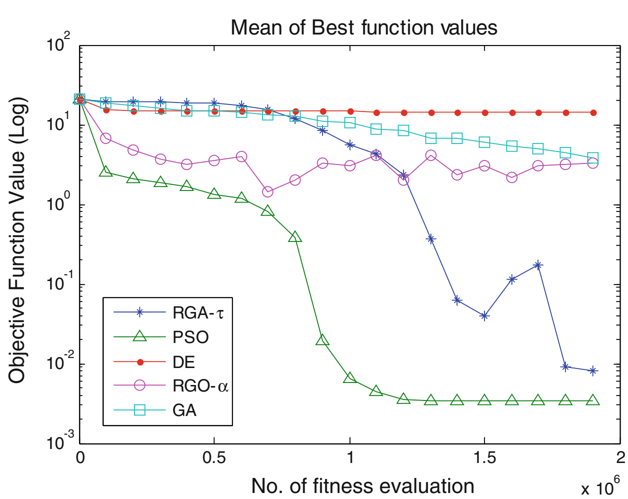

Fig. 21 f_7, Ackley with 45D

fel 2.22 function, PSO, RGA-α and GA can also obtain better optimal solution. The search performance order on the two functions is RGA-τ > PSO > RGA-α > GA > DE.

Rastrigin and Schwefel functions are two multimodal variable–separable functions. As it can be seen in Fig. 11, the convergence profile of RGA-τ was significantly better than the other four algorithms and PSO, DE, GA and RGA-α also trapped in the local optimum soon on Rastrigin function. The search performance order on Rastrigin function is RGA-τ > RGA-α > PSO > GA > DE. On Schwefel, GA and DE show the better performance. The search performance order is GA > DE > RGA-τ > RGA-α > PSO.

Ackley and Griewank functions are two multimodal non-separable functions. On these two functions, the results obtained by RGA-τ and PSO were significantly better than GA and DE, as shown in Figs. 13 and 14. RGA-α can also obtain better results than GA and DE. On Ackley function, PSO showed a better converge performance than RGA-τ and the search performance order is PSO > RGA-τ > RGA-α > GA > DE. But on Griewank function, RGA-τ converged better. PSO algorithm converged fast at the beginning and trapped in the local optimum soon on Griewank function. The search performance order is RGA-τ > PSO > RGA-α > DE > GA.

5.3.2 Analyzing numerical results with 45D

As shown in Table 8, RGA-τ algorithm can obtain better performance than the other algorithms on Sphere, Rosenbrock, Schwefel 2.22, Rastrigin, and Griewank functions with 45D. PSO algorithm shows better performance on SumSquares and Ackley functions. DE obtains the best optimal solution on Schwefel function. RGA-α can also obtain better performance on most functions.

Table 9 Experimental results of RGA-τ under varying τ values

Function	τ				
	1	50	100	500	1,000
Sphere					
Mean	6.03750	1.39662e−005	0.0109505	1.59217e−008	1.48642e−005
Std	0.405303	1.92988e−007	0.0154830	7.79135e−009	1.92248e−006
SumSquares					
Mean	150.623	0.0405131	9.84551e−003	0.0417146	3.61794e−003
Std	44.2301	5.94027e−003	2.01407e−003	0.0581283	4.58169e−004
Rosenbrock					
Mean	2,504.52	1,841.08	3,117.08	60.098	18.6324
Std	2,018.48	184.827	1,758.65	8.36271	0.970460
Schwefel 2.22					
Mean	14.9426	0.175768	0.0628732	0.221833	0.218181
Std	2.13443	0.0349579	1.87490e−003	4.11223e−3	0.0251149
Rastrigin					
Mean	361.247	354.175	355.371	0.0185982	5.05323e−003
Std	42.1935	11.6724	6.09598	0.43982e−003	7.13957e−003
Schwefel					
Mean	−8,372.39	−8,678.21	−8,145.58	−7,977.11	−8,037.08
Std	1,267.7	712.181	2,398.20	1,498.3069	1,872.12
Ackley					
Mean	3.60046	3.71368e−003	9.39340	3.68786e−004	2.80895e−003
Std	0.321164	5.59902e−004	12.7740	2.31268e−006	3.6843e−004
Griewank					
Mean	9.14804e−003	3.70074e−003	5.89445e−003	3.63194	24.9365
Std	1.91572e−003	5.22626e−003	8.29661e−003	1.52999	9.27919

On Sphere and SumSquares function, both PSO and RGA-τ obtained satisfying results. However, RGA-τ algorithm showed the best performance on Sphere and the search performance order is RGA-τ > PSO > RGA-α > GA > DE. PSO algorithm shows the better performance on SumSquares and the search performance order is PSO > RGA-τ > GA > DE > RGA-α.

On Rosenbrock and Schwefel 2.22 functions, RGA-τ algorithm showed the best performance. On the two function, RGA-τ, GA and DE cannot obtain better performance on this function. The search performance order on the two functions is RGA-τ > PSO > GA > RGA-α > DE.

As it can be seen in Fig. 19, the convergence profile of RGA-τ was significantly better than the other four algorithms and PSO, DE, GA and RGA-α also trapped in the local optimum soon on Rastrigin function. The search performance order on Rastrigin function is RGA-τ > PSO > GA > DE > RGA-α. On Schwefel, DE show the better performance. The search performance order is DE > RGA-α > RGA-τ > GA > PSO.

On Ackley and Griewank functions, the results obtained by RGA-τ were significantly better than GA and DE, as shown in Figs. 21 and 22. PSO also obtains better results. RGA-α can also obtain better results than GA and DE. On Ackley function, PSO showed a better converge performance than RGA-τ and the search performance order is PSO > RGA-τ > RGA-α > GA > DE. But on Griewank function, RGA-τ converged better. PSO algorithm converged fast at the beginning and trapped in the local optimum soon on Griewank function. The search performance order is RGA-τ > PSO > RGA-α > GA > DE.

Overall, RGA algorithm with the two self-adaptive strategies can obtain good performance on most functions with higher-dimension, especially on the unimodal non-separable function and the multimodal non-separable functions. RGA-τ shows very good performance. The second self-adaptive strategy is matched with RGA better. So RGA based on root growth model is appropriate for higher dimensional numerical function optimization.

5.4 Parameter τ Tuning

While solving a problem by an optimization algorithm, adjusting algorithm parameters have significant importance on the performance of the algorithm. A fine tuning of control parameters is required for most of the algorithms to obtain desired solutions (Table 9).

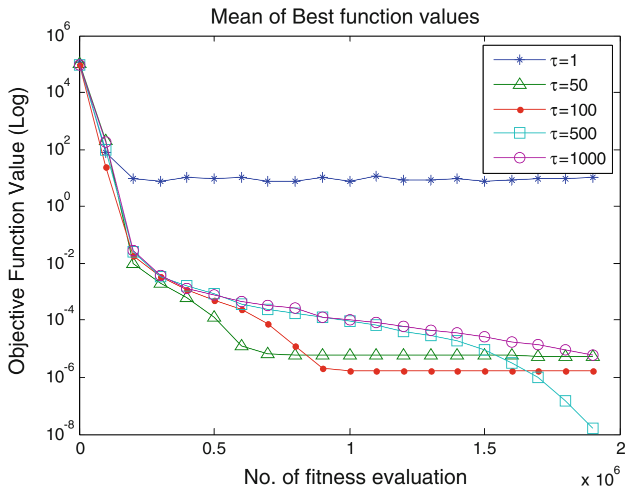

Fig. 23 Effect of τ parameter on Sphere function

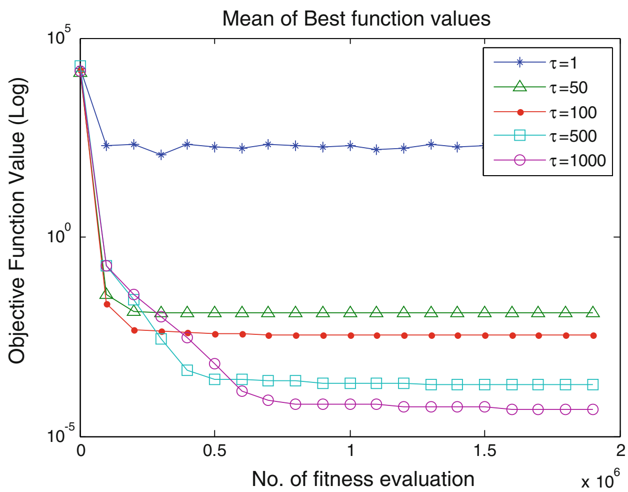

Fig. 24 Effect of τ parameter on SumSquares function

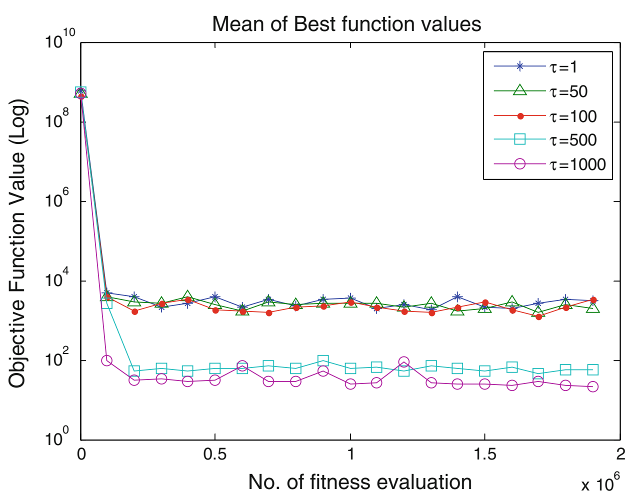

Fig. 25 Effect of τ parameter on Rosenbrock function

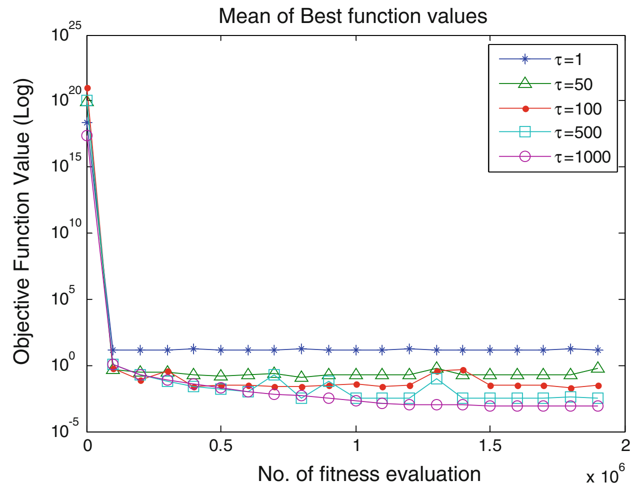

Fig. 26 Effect of τ parameter on Schwefel 2.22 function

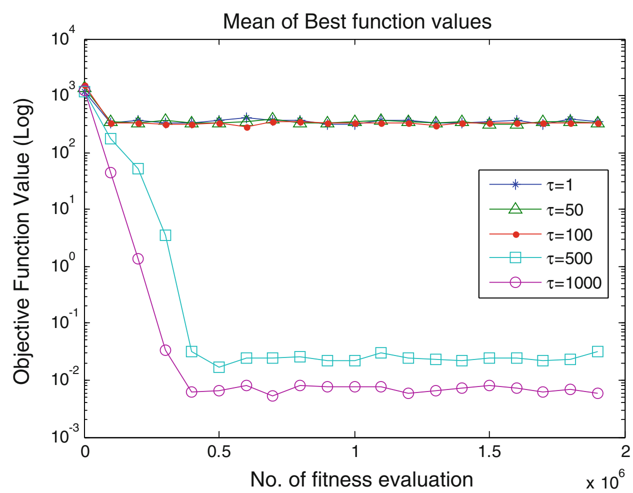

Fig. 27 Effect of τ parameter on Rastrigin function

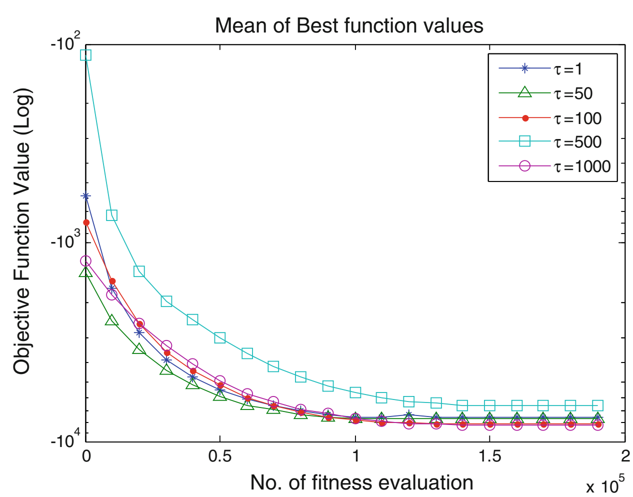

Fig. 28 Effect of τ parameter on Schwefel function

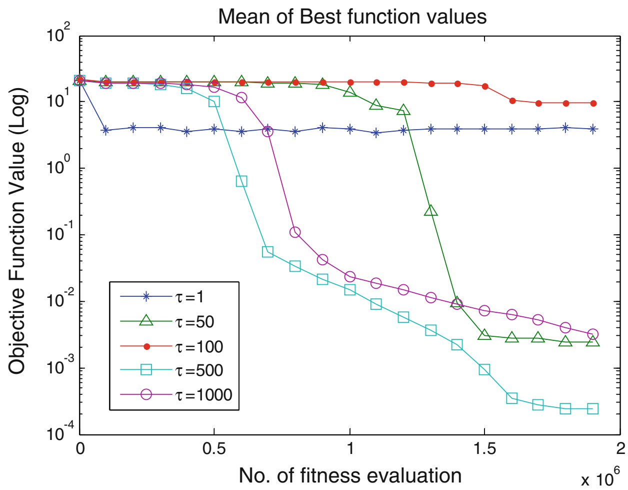

Fig. 29 Effect of τ parameter on Ackley function

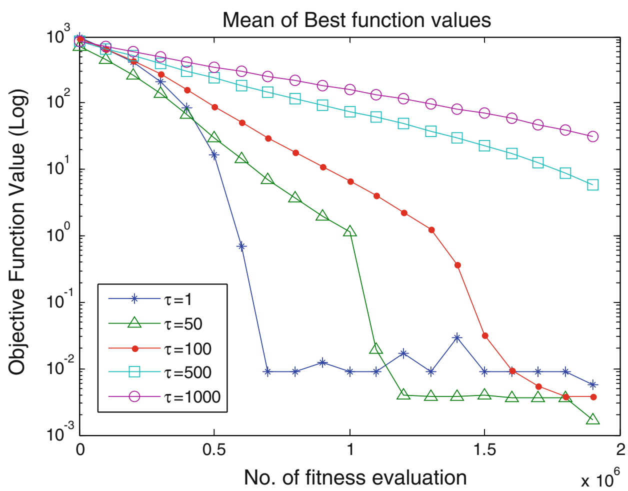

Fig. 30 Effect of τ parameter on Griewank function

As listed Table 6, RGA employs some control parameters, but effect of only parameter τ on the performance of RGA is analyzed in this paper. Because τ is a characteristic parameter in the second self-adaptive growth strategy for RGA and appropriate values of parameter τ can improve performance of RGA notably. In the experiment, the maximal number of fitness function evaluations was 2,000,000 for Sphere, SumSquares, Rosenbrock, Schwefel 2.22, Rastrigin, Ackley and Griewank functions and 200,000 for Schwefel function to make the figure of effect of τ parameter clear. In tables reporting the experimental results, mean values and standard deviations of 20 runs are presented.

In the experiment, the values of parameter τ are changed as 1, 50, 100, 500 and 1,000. Results of this experiment are reported on Table 8. Results in this table show that when other parameters are the same, different values of parameter τ can make the results obtained by RGA distinct on the same function. Reason of this fact is that parameter τ decides the growth length of root hairs which can make RGA exploring new regions better. But the performance of the algorithm on Schwefel function was not highly influenced by the increment in the value of parameter τ. The mean best function value profiles with different values of parameter τ are shown in Figs. 23, 24, 25, 26, 27, 28, 29 and 30 which clearly demonstrate the results in Table 8. So parameter τ has an important effect on RGA.

6 Conclusion

In this paper, we present a root growth model based on root growth behaviours in the soil. By using this model as a computational metaphor, we propose a novel algorithm called RGA for simulation of plant root system and higher-dimensional numerical function optimization. For simulation, some root growth behaviours, root–soil interaction and self-adaptive growth of root hairs, are simulated. The characteristics of root growth are showed in the form of images. For optimization, numerical results obtained from the proposed algorithm have been compared with those obtained from GA, DE and PSO. It is seen from the comparison that RGA performs better than GA, DE and PSO. RGA can optimize higher-dimensional numerical function better.

Acknowledgments This research is partially supported by National Natural Science Foundation of China 61174164, supported by National Natural Science Foundation of China 61003208 and supported by National Natural Science Foundation of China 61105067.

References

Cai W, Yang W, Chen X (2008) A global optimization algorithm based on plant growth theory: plant growth optimization. Int Conf Intell Comput Technol Autom (ICICTA) 1:1194–1199
Corne D, Dorigo M, Glover F (1999) New ideas in optimization. McGraw-Hill, New York
De Castro LN, Von Zuben FJ (1999) Artificial immune systems, Part I. Basic theory and applications, Technical Report Rt Dca 01/99, Feec/Unicamp
Dorigo M, Maniezzo V, Colorni A (1991) Positive feedback as a search strategy, Technical Report 91-016. Dipartimento di Elettronica, Politecnico di Milano
Eberhart RC, Shi Y, Kennedy J (2001) Swarm intelligence. Morgan Kaufmann, Massachusetts
Fogel LJ, Owens AJ, Walsh MJ (1965) Artificial intelligence through a simulation of evolution. In: Maxfield M, Callahan A, Fogel LJ (eds) Biophysics and cybernetic systems. Proceedings of the 2nd cybernetic sciences symposium. Spartan Books, pp 131–155
Gerwitz A, Page ER (1974) An empirical mathematical model to describe plant root systems. J Appl Ecol 11(2):773–781
Hodge A (2009) Root decisions. Plant Cell Environ 32(6):628–640
Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor

Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. Appl Soft Comput 8(1):687–697

Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132

Kennedy J, Eberhart RC (1995) IEEE international conference on neural networks, vol 4, pp 1942–1948

Krink T, Vestertroem JS, Riget J (2002) Particle swarm optimization with spatial particle extension. In: Proceedings of the IEEE congress on evolutionary computation, Honolulu, pp 1474–1479

Leitner D, Klepsch S, Bodner G, Schnepf A (2010) A dynamic root system growth model based on L-Systems. Plant Soil 332:177–192

Lynch J (2007) Roots of the second green revolution. Aust J Bot 55(5):493–512

Mainzer K (1997) Thinking in complexity: the complex dynamics of matter, mind and mankind, 3rd edn. Springer, New York

Pages L, Jordan MO, Picard D (1989) A simulation model of the three-dimensional architecture of the maize root system. Plant Soil 119:147–154

Price K, Storn R (1995) Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report. International Computer Science Institute, Berkley

Sheng WD, Lei C (1995) Fractal and their applications. Publishing Company of University of Science and Technology of China, Beijing

Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation, Washington, DC, pp 1945–1950

Shi Y, Ebrehart RC (1998) A modified particle swarm optimizer. In: Proceeding of the IEEE international conference on computational intelligence, Anchorage, pp 69–73

Tong L, Feng WC, Bo WW, Ling SW (2005) A global optimization bionics algorithm for solving integer programming—plant growth simulation algorithm. Syst Eng 25(1):76–85

Valdez F, Melin P, Castillo O (2011) An improved evolutionary method with fuzzy logic for combining Particle swarm optimization and genetic algorithms. Appl Soft Comput 11(2):2625–2632

Vesterstrom J, Thomsen R (2004) A comparative study of differential evolution particle swarm optimization and evolutionary algorithms on numerical benchmark problems. In: IEEE congress on evolutionary computation (CEC'2004), vol 3, Piscataway, pp 1980–1987

White PJ, Broadley MR, Greenwood DJ, Hammond JP (2005) Genetic modifications to improve phosphorus acquisition by roots. IFS,Proceedings of the International Fertiliser Society, York