A Novel Algorithm Inspired by Plant Root Growth with Self-similarity Propagation

Xiaoxian He[a,b,*], Shigeng Zhang[a], Jie Wang[a]

[a] College of Information Science & Engineering, Central South University, Changsha, China
[b] College of Engineering, University of Tennessee, Knoxville, Tennessee, USA

Abstract—**Most nature-inspired algorithms simulate intelligent behaviors of animals and insects that can move spontaneously and independently. As another species of biology, the survival wisdom of plants has been neglected to some extent until now. This paper presents a novel plant-inspired algorithm which is called root growth optimizer (RGO). RGO simulates the adaptive growth behaviors of plant roots, e.g. self-similar propagation, to optimize continuous space search. In the process, different roots implement different strategies according to their biological roles, so as to cooperate as a whole. Seven well-known benchmark functions are used to validate its optimization effect. We compared RGO with other existing animal-inspired algorithm including artificial bee colony algorithm and particle swarm optimizer. The experimental results show that RGO outperforms other algorithms on most benchmark functions.**

Keywords—**Root growth optimizer; plant-inspired algorithm; self-similarity propagation**

I. INTRODUCTION

In recent years, many heuristic algorithms inspired by collective intelligent behaviors of insects and animals were proposed to solve complex optimization problems. For example, ant colony optimizer (ACO) simulates foraging behaviors of ants[1]. Particle swarm optimizer (PSO) simulates swarm behaviors of birds and fish[2][3]. Bacterial colony optimizer (BCO)[4] and bacterial colony foraging optimizer (BCFO)[5] simulate typical behaviors of bacteria during their lifecycle. Artificial bee colony (ABC) algorithm simulates foraging behaviors of a swarm of bees[6]. Comparing with traditional mathematical methods, these heuristic algorithms have shown substantial advantages when treating on complex, multimodal and irregular problems.

The above-mentioned bio-inspired optimization algorithms all imitate animal's behaviors. As another species of biology, however, plant has gotten little attention in the field[7]. Compared with animals, plant seems insensitive to external information, dull to take actions, and far away from intelligence. But, in some biologists' opinion, plant can also be regarded as "intelligent organisms"[8][9]. During the growth process, plant shows considerable plasticity in its morphology and physiology in response to the variability of environments[10]. For example, plant's root can cope with the prevailing conditions in soil environments, such as avoiding obstacles and exploring nutrient-rich patches of water zones by its hydrotropism, chemotropism, gravitropism, and so on[11]. In a root system, its architecture is well known to be a major determinant of root functions in acquiring soil resources[12][13]. Because most root systems have the characteristics of self-similarity and are considered as approximate fractal objects over a finite range of scales[14] (As shown in Figure 1), fractal geometry has been widely used to assess the architecture and distribution of a root system in soil[15][16].

There is a close correlation between the architecture and propagation strategies of roots as botanists have discovered. During the growing process, root can perceive their external physical environments and implements different strategies. If there are enough resources, it will produce many lateral roots at the same time of elongating forward. Otherwise, few lateral roots are produced. Over time, the similar propagation occurs at different positions in variant scales. As a result, the whole root system will cover the most profitable area with a self-similar architecture.

With the nature-designed growth strategy, it is easy for them to find the best position of water and nutrition in soil. Inspired by the root growth behaviors of plants, this paper presents a new algorithm named root growth optimization (RGO) algorithm.

Figure 1. The self-similarity architecture of a root system

The remainder of this paper is organized as follows. Section 2 models the root growth process. Section 3 presents the root growth optimization algorithm. Experiment and results are given in section 4. Section 5 discusses some unique characteristics of RGO, and section 6 outlines the conclusions.

* Corresponding author. Address: 413 Computer Building, Central South University, Changsha, Hunan, 410083, China. E-mail address: xxhe@csu.edu.cn

II. ARTIFICIAL ROOT GROWTH

2.1. Basic Concept

In the artificial model, an objective function is treated as the growing environments of plant roots, and the initial roots are considered as a homogeneous biomass[17]. Each root apex stands for a feasible solution of the problem. All roots try to adjust their growing directions and propagation strategies in order to search for the optimal growing conditions, which feed back to improve root growth further.

In growing process, all the root apices can select their growth strategies composed of the following three basic actions:

1) Each root apex may elongate forward (or sideways) in the substrate.
2) Each root apex may produce daughter root apices.
3) Each root apex may cease to function as above, and become an ordinary piece of root mass.

In a word, a root apex may regrow itself, produce new roots, or stop growing for some reason.

According to fitness values, the whole root mass are divided into three groups. The group with the best fitness values is called main roots. The group with the worst fitness values is called aging roots. The rest of root mass is called lateral roots. In these three groups, except for aging roots that will stop growing in the next generation, main roots and lateral roots implement different growth strategies.

2.2. The Growth Strategy of Main Roots: Monopodial Branching

According to monopodial branching strategy, a main root itself regrows to form an axis firstly, and then branching roots appear in the lateral position. As a result, the growth strategy contains three operators:

1) Regrowing

This operator means that a root apex regrows towards a local best position where there are better water and nutrient conditions. The operator is formulated as the following expression.

$$x_i^t = x_i^{t-1} + l \cdot rand() \cdot (x_{lbest} - x_i^{t-1}) \qquad (1)$$

where x_i^{t-1} is the original position of the ith root apex, and x_i^t is the new position. l is the local learning constant. $rand()$ is a random number with uniform distribution in [0,1]. x_{lbest} is the local best position in the generation.

2) Branching

This operator means that a root apex produces some new apices around it. The number of new root apices is calculated as follows.

$$w_i = \frac{f - f_{min}}{f_{max} - f_{min}} \cdot (s_{max} - s_{min}) + s_{min} \qquad (2)$$

where f_{max} and f_{min} are the best and the worst fitness value in the generation, respectively. f is the fitness value of the original root apex. s_{max} and s_{min} are the maximal branching number and the minimal branching number which are pre-set.

The positions of new root apices surround the original root apex with Gauss distribution $N(x_i^t, \sigma^2)$. The standard deviation σ is calculated as follows.

$$\sigma_i = ((i_{max} - i)/i_{max})^n \cdot (\sigma_{ini} - \sigma_{fin}) + \sigma_{fin} \qquad (3)$$

where i_{max} and i are the maximal iteration number and current iteration number, respectively. σ_{ini} is the initial standard deviation depending on the value of searching range, and σ_{fin} is the final standard deviation determined by expected accuracy standard in the program.

3) Inhibition mechanism of plant hormones

Because new propagated root apices may be classified into the main root group with high probability in the next generation if the area are really nutrient-rich enough, and all main roots will elongate and propagate again, the number of roots in this area may increase explosively, which we called as "root number explosion".

Root number explosion is absolutely harmful to the adaptability of plants. From the view of optimization, it will make the algorithm plunge into local optimum. In fact, this phenomenon is rarely seen from natural plant roots because plant hormones play an important role in the growth process.

Generally, plant hormones can improve growth speed of plant organisms. However, if some plant organism grows too quickly, one of plant hormones may increase rapidly and the balance between all plant hormones will be broken, then the growth will be inhibited[18].

To simulate the inhibition mechanism of plant hormones in the model, we will calculate the local standard deviation $\sigma_{local}(f)$ of new propagating apices, and then get rid of some apices according to the calculating results by greedy principle. The operator is implemented as formula (4).

$$w_{-i} = \alpha \cdot (1 - \frac{\sigma_{local}(f)}{f_{max} - f_{min}}) \cdot w_i \qquad (4)$$

Where w_{-i} is the number of root apices which should be removed, and α is a control parameter.

From formula (4) we can know that the smaller $\sigma_{local}(f)$ is, the more root apices will be removed in the next generation. On the one hand, rapid local increase of roots is controlled in this way so that root number explosion can be avoided. On the other hand, essential diversity can be kept to prevent the algorithm from pre-maturity.

2.3. The Growth Strategy of Lateral Roots: Sympodial Branching

In sympodial branching mode, the root apex produces a new branching apex at the lateral position instead of regrowing along the original direction, and the new branching apex grows into an axis by replacing the original one. The new branching apex may locate at a random position around the original root with a random angle β. This strategy is formulated as follows.

$$x_i^t = rand() \cdot \beta \cdot x_i^{t-1} \tag{5}$$

where *rand()* is a random number with uniform distribution in [0,1]. β is calculated as follows.

$$\beta = \lambda_i / \sqrt{\lambda_i^T \cdot \lambda_i} \tag{6}$$

where λ_i is a random vector.

III. ROOT GROWTH OPTIMIZATION ALGORITHM

Root growth mechanism offers a wonderful inspiration for proposing a new optimization algorithm. It can be seen that the artificial root growth model has described the original appearance of natural plant roots with few assumptions. Corresponding algorithm is elaborately designed based on the above model.

The psudo code of RGO is listed in table 1.

TABLE 1. PSEUDO CODE OF RGO

1. Initialize the positions of root apices
2. Evaluate the fitness values of root apices
3. While not meet the terminal condition
4.　Divide the root apices into main roots, lateral roots and aging roots
5.　For each main root apex
Regrow with the root regrowing operator
Branch with the root branching operator
Evaluate the fitness value of new root apices
Calculate the deviations, implement inhibition mechanism of plant hormones
End for
6.　For each lateral root apex
Produce a new apex replacing the original one
End for
7.　Rank root apices and label elite roots
8. End while
9. Postprocess results

IV. EXPERIMENTS AND RESULTS

In order to test the performance of RGO algorithm, PSO algorithm and ABC algorithm, which are inspired by the behaviors of animals and popularly used, are employed for comparison. In the experiments, seven benchmark functions are used to test its effectiveness.

4.1. Experiment Sets and Benchmark Functions

The seven classic benchmark functions are widely adopted by other researchers to test their algorithms in many works[19]. Among these functions, sphere is a unimodal function with separable variables which is easy to solve. Schwefel 1.2 and Rosenbrock are unimodal functions with non-separable variables. Rosenbrock function has a narrow valley sloping gently from local optima to the global optimum, thus can be treated as a multimodal function. Rastrigin and Schwefel are multimodal functions with separable variables. Ackley and Griewank are multimodal functions with non-separable variables. They all have a large number of local optima to make it difficult to reach the global optimum.

In this paper, all functions use their standard ranges and variable data. The experiments compare the performance using all accuracies of algorithms for a fixed number of function evaluations. The max evaluation count is 10,000. Experiments have been carried out using Matlab 7.0 on a standard 2.5GHZ desktop computer. All parameters in PSO and ABC are set as their original values. The population size of four algorithms is 50. In order to do meaningful statistical analysis, each algorithm runs for 20 times and the mean value and standard deviation value are taken as final results. In RGO, the number of root apices in main root group is thirty percent of the selected root apices in each generation. s_{max} and s_{min} are set 3.0 and 1.0, respectively. α and l are all set 1.0. All the benchmark functions are listed in Table 2.

TABLE 2. TEST FUNCTIONS

Name	Function	Limits

Sphere	$f_1 = \sum_{i-1}^{D} x_i^2$	$x \in [-100,100]^D$
Schwefel 1.2	$f_2 = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$x \in [-100,100]^D$
Rosenbrock	$f_3 = \sum_{i-1}^{D} (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2)$	$x \in [-30,30]^D$
Rastrigin	$f_4 = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$x \in [-5.12,5.12]^D$
Schwefel	$f_5 = 418.9829 * D + \sum_{i=1}^{D} (-x_i \sin(\sqrt{|x_i|}))$	$x \in [-500,500]^D$
Ackley	$f_6 = 20 + e - 20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2})$ $- \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i))$	$x \in [-32,32]^D$
Griewank	$f_7 = \frac{1}{4000}(\sum_{i=1}^{D} x_i^2) - (\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})) + 1$	$x \in [-600,600]^D$

4.2. Experiment Results and Analysis

The mean fitness values and standard deviation values obtained by the four algorithms with 2, 15, 30 and 50 dimensions are listed in Table 3, Table 4, Table 5 and Table 6. Best values obtained on each function are marked as bold.

As can be seen in Table 3, with dimension of 2, ABC performs better than others on function f_1, f_1, f_4, f_5 and f_6. PSO fail to hit the best results in this round. Though RGO only gets the best results on function f_3 and f_7, it gets

satisfactory accuracy on other functions. From Table 4, we can see that RGO runs much better than ABC and PSO. It gets the best results on most functions except f_3 and f_5. From Table 5 and Table 6, we can see that most of the best results are obtained by RGO. Especially in Table 6, it hits all the best results of multimodal functions, which show that RGO outperforms other algorithms obviously in terms of accuracy on high dimension functions.

TABLE 3. RESULTS ON BENCHMARK FUNCTIONS WITH DIMENSION OF 2

Function		ABC	RGO	PSO
f_1	Mean	**3.58450E-18**	4.36985E-16	5.36985E-12
	Std	**2.24525E-18**	3.96588E-16	8.36521E-12
f_2	Mean	**2.26545E-05**	2.97552E-02	4.85401E-02
	Std	**1.98977E-05**	5.86344E-03	4.32110E-02
f_3	Mean	3.45686E-02	**5.37582E-03**	8.36520E-02
	Std	2.98749E-02	**3.00621E-03**	5.02560E-02
f_4	Mean	**0.00000E+00**	5.98720E-02	3.25811E-02
	Std	**0.00000E+00**	2.58729E-03	6.32862E-02
f_5	Mean	**4.54567E-05**	3.65401E-03	7.32085E-01
	Std	**2.33454E-05**	2.36895E-04	3.56212E-01
f_6	Mean	**2.35681E-01**	3.48756E+02	4.32565E-01
	Std	**3.37654E-01**	2.36598E+01	5.32651E-01
f_7	Mean	5.34686E+04	**7.84526E-02**	3.56824E+01
	Std	3.45685E+04	**4.36521E-02**	4.85014E+01

TABLE 4. RESULTS ON BENCHMARK FUNCTIONS WITH DIMENSION OF 15

Function		ABC	RGO	PSO
f_1	Mean	6.35214E-07	**5.23100E-09**	5.25686E-03
	Std	3.25410E-08	**2.35644E-09**	4.66007E-03

		ABC	RGO	PSO
f_2	Mean	6.00254E-02	**3.25874E-05**	6.53240E-02
	Std	3.85662E-02	**2.01452E-06**	4.38965E-02
f_3	Mean	3.26587E-01	3.59856E-01	**2.45012E-02**
	Std	4.56254E-01	8.69854E-02	**4.69580E-02**
f_4	Mean	3.85410E+01	**7.25845E-01**	4.25632E+02
	Std	6.32544E+00	**3.26222E-02**	6.35210E+02
f_5	Mean	**2.32654E+00**	5.14298E+00	4.78695E+00
	Std	**5.14201E+00**	3.04875E+00	2.03655E+00
f_6	Mean	6.32512E+02	**2.65894E+01**	4.85320E+01
	Std	5.78522E+02	**7.69855E+01**	4.02563E+01
f_7	Mean	2.30120E-01	**8.23540E-02**	5.36627E+02
	Std	3.99851E-01	**1.36555E-02**	3.76952E+01

TABLE 5. RESULTS ON BENCHMARK FUNCTIONS WITH DIMENSION OF 30

Function		ABC	RGO	PSO
f_1	Mean	6.13416E-03	**5.36952E-04**	3.52642E-02
	Std	1.13337E-02	**3.01410E-04**	2.35262E-02
f_2	Mean	2.58658E+02	**5.88652E-02**	8.86454E-01
	Std	5.87078E+01	**4.32125E-02**	4.26532E-01
f_3	Mean	1.23425E+02	**2.87021E-01**	1.32566E+01
	Std	2.42345E+02	**4.63528E-01**	2.36141E+01
f_4	Mean	2.77214E+01	3.65998E+01	**5.75625E-01**
	Std	8.58644E+00	4.87530E-02	**6.35211E-01**
f_5	Mean	2.02340E+03	**2.36325E+01**	2.36452E+02
	Std	3.14094E+02	**1.25635E+00**	3.75632E+02
f_6	Mean	**3.83236E+00**	3.58766E+04	5.39892E+01
	Std	**7.92856E-01**	2.85430E+03	5.56320E+01
f_7	Mean	**2.82235E-01**	4.35621E-01	4.85789E+00
	Std	**3.01241E-01**	1.26020E-01	5.356521E-01

TABLE 6. RESULTS ON BENCHMARK FUNCTIONS WITH DIMENSION OF 50

Function		ABC	RGO	PSO
f_1	Mean	3.56423E-02	6.14524E-04	**4.68297E-04**
	Std	2.36545E-02	5.32009E-04	**3.90120E-04**
f_2	Mean	2.68953E+00	1.98755E-01	**4.78011E-03**
	Std	1.36899E-01	2.36523E-01	**2.52479E-03**
f_3	Mean	3.44532E+01	**3.81927E-02**	4.36025E-01
	Std	6.89530E+00	**2.73918E-02**	2.88152E-01
f_4	Mean	3.56203E+01	**3.46829E-01**	4.26523E-01
	Std	3.63215E+00	**2.73915E-01**	3.84677E-01
f_5	Mean	3.25891E+01	**3.61748E+00**	8.65419E+02
	Std	8.36254E+00	**4.62376E+00**	7.35621E+01
f_6	Mean	5.39911E+01	**2.36250E+01**	3.82761E+02
	Std	2.32562E+01	**3.08943E+01**	2.73102E+02
f_7	Mean	5.36215E+00	**4.73945E-01**	1.36550E+00
	Std	2.20147E+00	**6.18723E-01**	2.54899E+00

In view of the above comparison, we can see RGO is a very promising algorithm. It has very strong optimizing ability on test functions. When the dimension of functions increases, RGO shows more advantages than other evolutionary algorithms.

V. DISCUSSIONS

5.1. Local Learning

In RGO, the formula (1) is similar in form to the iteration formula of PSO, which can be expressed as follows.

$$v_i^{t+1} = v_i^t + c_1 \cdot rand1() \cdot (pbest_i^t - x_i^t)$$
$$+ c_2 \cdot rand2() \cdot (gbest^t - x_i^t) \tag{7}$$

where c_1 and c_2 are local learning factor and social learning factor. Compared with formula (7), the elongation of main root apex is adjusted only by the original position and local learning factor. Social learning factor and global best fitness value will have little influence on the behaviors of a main root apex. The reason contains two aspects. Firstly, as far as natural plant is concerned, there is no proof that a root apex can get global information which may be far away from it, while local information can be acquired though hydrotropism, chemotropism and gravitropism etc. Secondly, there is no evidence that the real global optimum must locates near the temporary global optimum in multimodal problems. So, the temporary global optimum is not worthy for all root apices to follow in RGO algorithm.

5.2. Self-similarity

In early years, biologists have already found that root systems have self-similarity and were considered as approximate fractal objects over a finite range of scales[14~15]. Until now, the architectural characteristic of root systems has been drawing much researchers' attention[20].

In RGO, it can be seen from formula (2) and (3) that when a main root apex has a good fitness value, it will be vigorous for propagation. According to formula (3), along with main roots elongating into the soil, the positions of their new-born daughter root apices comply with the same distribution law at different time points, except that distribution range becomes smaller and smaller. In the meanwhile, new-born roots may become main roots in the next generation and propagate in the same way. With this pattern, approximate fractal architecture with self-similarity characteristics will be shaped.

As far as optimization is concerned, it can be confirm that the self-similar propagation is profitable for roots to exploit resource-rich areas rapidly. As a novel search technique, the correlation between self-similar propagation and searching in multimodal continuous space remains to be an interesting problem which will be investigated in the near future work.

VI. Conclusions

Root Growth Optimization (RGO) algorithm, based on the root growth behavior of plants, is present in this paper. Seven benchmark functions were used to validate its efficiency, and the results were compared with PSO and ABC. The comparing results show that RGO outperforms PSO and ABC on most of benchmark functions in different dimensions. Moreover, RGO is potentially more powerful than PSO and ABC on functions with high dimensions.

A further extension to the current RGO algorithm may lead to even more effective optimization algorithms for solving high-dimension or multi-objective problems. Therefore, future research efforts will be focused on finding new methods to improve our proposed algorithm and applying the algorithm to solve practical engineering problems.

References

[1] E. Bonabeau, M. Dorigo, G. Theraulaz, "Swarm Intelligence-from Natural to Artificial System," Oxford University Press, New York, 1999

[2] J. Kennedy, R. C. Eberhart, "Particle Swarm Optimization", In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942-1948

[3] H. Liu, Z. Cai, Y. Wang, "Hybridizing Particle Swarm Optimization with Differential Evolution for Constrained Numerical and Engineering Optimization," Applied Soft Computing, 2010, 10(2), pp. 629-640

[4] B. Niu, H. Wang, "Bacterial Colony Optimization," Discrete Dynamics in Nature and Society, 2012, Article ID 698057, doi: 10.1155/2012/698057

[5] H. Chen, Y. Zhu, K. Hu, L. Ma, "Bacterial Colony Foraging Algorithm: Combining Chemotaxis, Cell-to-cell Communication, and Self-adaptive Strategy," Information Sciences, 2014, 273, pp. 73–100

[6] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization," Erciyes University, Engineering Faculty, Computer Engineering Department, 2005

[7] W. Cai, W. Yang, X. Chen, "A Global Optimization Algorithm Based on Plant Growth Theory: Plant Growth Optimization," In: International Conference on Intelligent Computation Technology and Automation (ICICTA), 2008, pp. 1194-1199

[8] A. Trewavas, "Green Plants as Intelligent Organisms," Trends in Plant Science, 2005, 10, pp. 413-419

[9] P. C. Struik, X. Yin, H. Meinke, "Plant Neurobiology and Green Plant Intelligence: Science, Metaphors and Nonsense," Journal of the Science of Food and Agriculture, 2008, 88: 363-370

[10] E. D. Brenner, R. Stahlberg, S. Mancuso, J. Vivanco, F. Baluška & E. Van Volkenburgh, "Plant Neurobiology: an Integrated View of Plant Signaling," Trends in Plant Science, 2006, 11, pp. 413-419

[11] A. Hodge, Root Decisions, Plant, Cell and Environment, 2009, 32, pp. 628-640

[12] J. Lynch, "Root Architecture and Plant Productivity," Plant Physiol, 1995, 109, pp. 7−13

[13] H. Wang, Y. Inukai, A.Yamauchi, "Root Development and Nutrient Uptake," Crit. Rev. Plant Sci., 2006, 25, pp. 279−301

[14] J. Tatsumi, A. Yamauchi, Y. Kono, "Fractal Analysis of Plant Root Systems," Ann. Bot., 1989, 64, pp. 499−503

[15] M. Dannowski, A. Block, "Fractal Geometry and Root System Structures of Heterogeneous Plant Communities," Plant Soil, 2005, 272 (1-2), pp. 61-76

[16] H. Wang, J. Siopongco, L. J. Wade, A. Yamauchi, "Fractal Analysis on Root Systems of Rice Plants in Response to Drought Stress," Environmental and Experimental Botany, 2009, 65, pp. 338-344

[17] R. Newson, "A Canonical Model for Production and Distribution of Root Mass in Space and Time," J. Math. Biol., 1995, 33, pp. 477-488

[18] J. Friml, "Auxin Transport-Shaping the Plant," Curr. Opin. Plant Biol., 2003(6), pp. 7-12

[19] D. K. Akay, "A Comparative Study of Artificial Bee Colony Algorithm," Appl. Math. Comput., 2009, pp. 108-132

[20] S. C. Pierce, M. B. Koontz, S. R. Pezeshki, R. Kröger, "Response of Salix Nigra [Marsh.] Cuttings to Horizontal Asymmetry in Soil Saturation," Environmental and Experimental Botany, 2013, 87, pp. 137-147