https://doi.org/10.1007/978-3-642-39482-9_66

An Idea Based on Plant Root Growth for Numerical Optimization

Xiangbo Qi[1,2], Yunlong Zhu[1], Hanning Chen[1], Dingyi Zhang[1], and Ben Niu[3]

[1] Shenyang Institute of Automation Chinese Academy of Sciences 110016, Shenyang, China
[2] University of Chinese Academy of Sciences 100039, Beijing, China
[3]College of Management, Shenzhen University 518060, Shenzhen, China

Abstract. Most bio-inspired algorithms simulate the behaviors of animals. This paper proposes a new plant-inspired algorithm named Root Mass Optimization (RMO). RMO simulates the root growth behavior of plants. Seven well-known benchmark functions are used to validate its optimization effect. We compared RMO with other existing animal-inspired algorithms, including artificial bee colony (ABC) and particle swarm optimization (PSO). The experimental results show that RMO outperforms other algorithms on most benchmark functions. RMO provides a new reference for solving optimization problems.

Keywords: Root mass optimization, artificial bee colony algorithm, particle swarm optimization algorithm.

1 Introduction

In order to solve tough optimization problems, many researchers have been drawing inspiration from the nature and many meta-heuristic algorithms inspired by biology behavior are proposed. Genetic Algorithm (GA) mimics the process of natural evolution[1]. Particle Swarm Optimization (PSO) algorithm simulates the swarm behavior of birds and fish[2]. Firefly algorithm (FA) simulates the bioluminescent communication behavior of fireflies[3]. Bacterial Foraging Optimization (BFO) simulates the foraging behavior of bacteria[4]. BCO is based on a lifecycle model that simulates some typical behavior of E. coli bacteria[5].Artificial Fish Swarm Algorithm (AFSA) simulates prey, swarm and follow behavior of a school of fish[6]. Ant Colony Optimization(ACO) algorithm modeled on the actions of an ant colony[7]. Artificial Bee Colony (ABC) algorithm inspired by the foraging behavior of a swarm of bees is proposed by Karaboga[8].

Can be seen from above, most bio-inspired algorithms simulate some behaviors of animals. Algorithms simulating the growth behavior of plants are rarely seen. However, plants also have ‘brain-like’ control[9, 10]. Some researchers proposed a plant growth simulation algorithm (PGSA) simulating plant growth [11]. Every algorithm has its advantages and disadvantages. “No free Lunch “theorems [12, 13] suggests one algorithm impossibly shows the best performance for all problems. Many strategies including improving existed algorithms or studying new algorithms can get better

D.-S. Huang et al. (Eds.): ICIC 2013, LNAI 7996, pp. 571–578, 2013.

optimization effect. Inspired by the root growth behavior of plants, this paper proposes a new algorithm named Root Mass Optimization (RMO) algorithm.

The remainder of the article is organized as follows. Section 2 introduces some researches about root growth model. Section 3 proposes Root Mass Optimization (RMO) algorithm and gives the pseudo code. Section 4 gives the experiment process in detail, presents the experimental results and gives the analysis. Section 5 discusses a similarity in form between PSO and RMO. Finally, section 6 gives the conclusions.

2 Research of Root Growth Model

To understand the biological process of plant roots, many researchers built all kinds of models as a means of simulating the growth behaviors of plants. Different models show different purposes. In order to describe the process of plant growth, some mathematical models were constructed and were useful for the investigation of the effects of soil, water usage, nutrient availability and many other factors on crop yield [14]. There is increasing evidence that root–root interactions are much more sophisticated [15]. Roger Newson characterized the root growth strategy of plants in a model [16]: (1) Each root apex may migrate downwards (or sideways) in the substrate. (2) Each root apex, as it migrates, leaves behind it a trail of root mass, which stays in place.(3) Each root apex may produce daughter root apices.(4) Each root apex may cease to function as above, and “terminally differentiate” to become an ordinary, on-migrating, on-reproducing piece of root mass.

From the above point of view, the meaning of each root apex growth contains two aspects. One is root apex itself grows. The other is producing branch roots. These two kinds of growth may stop for some reasons. Existing root system models can be divided into pure root growth models, which focus on describing the root system’s morphology, and more holistic models, which include several root-environment interaction processes, e.g. water and nutrient uptake[17]. However, we don’t want to pay attention to root system’s biological significances and agricultural significances. Plant roots are fascinating as they are able to find the best position providing water and nutrient in soil depending on their growth strategy. These strategies include gravitropism, Hydrotropism, chemotropism, and so on. We link the root growth process to the optimizing process for an objective function.

3 Root Mass Optimization Algorithm

Root growth offers a wonderful inspiration for proposing a new optimization algorithm. The objective function is treated as the growth environment of plant roots. The initial root apices forms a root mass. Each root apex can be treated as the solution of the problem. Roots turn to the direction that provides the optimal soil water and fertilizer conditions, so they may proliferate. That process can be simulated as an optimizing process in the soil replaced with an objective function. In view of this, we proposed the root mass optimization (RMO) algorithm. Some rules are made to idealize the root growth behaviors in RMO:(1)All root apices forms a root mass. Two operators including root regrowing and root branching are needed to idealize the root growth behavior. Each root apex grows using one of these two operators. (2)Root mass are divided into three groups according to the fitness. The group with better fitness is called regrowing group. The group with worse fitness called stopping group stops growing. The rest of root mass is called branching group.

The meanings of two operators including root regrowing and root branching are listed below.

(a) Root regrowing: this operator means that the root apex regrows along the original direction. The root apex may migrate downwards (or sideways) the best position which provides the optimal soil water and fertilizer conditions. This operator is formulated using the expression (1).

$$n_i = x_i + r + (g_{best} - x_i) \tag{1}$$

Where r is a random vector each element of which is between [-1, 1]. n_i is the new position of the ith root apex. x_i is the original position of the ith root apex. g_{best} is the root apex with the best fitness in each generation.

(b) Root branching: this operator means that the root apex produces a new growth point instead of regrowing along the original direction. The growth point may be produced at a random position of the original root with a random angle β. It is worth noting that, a growth point is seen as a root apex in this paper. This operator is formulated using the expression (2).

$$n_i = \beta \alpha x_i \tag{2}$$

Where α is a random number between (0, 1). n_i is the new position of the ith root apex. x_i is the original position of the ith root apex. β is calculated using the expression (3).

$$\beta = \lambda_i / \sqrt{\lambda_i^T \lambda_i} \tag{3}$$

Where λ_i is a random vector.

The pseudo code of RMO algorithm is listed in Table 1. In each generation, sort the root apices in descending order according to the fitness. The selection of root apices participating in the next generation employs the linear decreasing way according to the expression (4). This way makes the root apices with better fitness perform root regrowing or root branching and makes the worse ones stop going on growing. In the selected part, select a percentage of the root apices from the front and let these root apices (growing group) regrow using the operator root growing; the rest of root apices (branching group) branch using the operator root branching.

$$ratio = sRatio - (sRatio - eRation)\frac{eva}{mEva} \quad (4)$$

Where eva is the current function evaluation count and mEva is the maximum function evaluation count. sRatio is the initial percentage and eRatio is the last percentage.

Table 1. Pseudo Code of RMO

1.	Initialize the position of root apices to form a root mass and Evaluate the fitness values of root apices
2.	While not meet the terminal condition
3.	Divide the root apices into regrowing group , branching group and stopping group
4.	Regrowing phase
5.	For each root apex in regrowing group
	Grow using the operator root regrowing
	Evaluate the fitness of the new root apex
	Apply greedy selection
	End for
7.	Branching phase
8.	For each root apex in branching group
	Produce two growing point using the operator root branching
	Evaluate the fitness
	Apply greedy selection
	End for
9.	Rank the root apices and memorize the current best root apex
10.	End while
11.	Postprocess results

4 Validation and Comparison

In order to test the performance of RMO, PSO and ABC were employed for comparison.

4.1 Experiments Sets and Benchmark Functions

The max evaluations count is 10000. In PSO, inertia weight varied from 0.9 to 0.7 and learning factors c1 and c2 were set 2.0. The population size of three algorithms was 40. Each algorithm runs for 30 times and takes the mean value and the standard deviation value as the final result. In RMO, the number of root apices in regrowing group is thirty percent of the selected root apices in each generation. sRatio is 0.9 and eRatio is 0.4. Seven well-known benchmark functions which are widely adopted by other researchers[18] are listed as follows.

Sphere function: $f_1 = \sum_{i=1}^{n} x_i^2$ $\quad x \in [-5.12, 5.12]$

Quadric function: $f_2 = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$ $\quad x \in [-10, 10]$

Rosenbrock function: $f_3 = \sum_{i=1}^{n} (100(x_i^2 - x_{i+1})^2 + (1-x_i)^2)$ $\quad x \in [-15, 15]$

Rastrigin function: $f_4 = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$ $\quad x \in [-15, 15]$

Schwefel function: $f_5 = n * 418.9829 + \sum_{i=1}^{n} (-x_i \sin(\sqrt{|x_i|}))$ $\quad x \in [-500, 500]$

Ackley function: $f_6 = 20 + e - 20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))$ $\quad x \in [-32.768, 32.768]$

Griewank function: $f_7 = \frac{1}{4000}(\sum_{i=1}^{n} x_i^2) - (\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})) + 1$ $\quad x \in [-600, 600]$

4.2 Experiment Results and Analysis

As can be seen in Table2, on function with dimension of 2, ABC performs better than RMO and PSO on f1, f2, f4, f5 and f6. PSO shows the best performance on function f3. RMO not only gets the best result on f7 but also gets satisfactory accuracy on f1, f2, and f4. From Table4 and Table5, we can see that RMO performs much better than ABC and PSO on most functions except f5. From Table3, RMO performs much better than ABC and PSO on most functions except f3 and f5. On most benchmark functions with multiple dimensions, RMO is much superior to other algorithms in terms of accuracy.

In view of the above comparison, we can see RMO is a very promising algorithm. It has a very strong optimizing ability on test functions with multiple dimensions.

Table 2. Results of ABC, RMO and PSO on benchmark functions with dimension of 2

Function		ABC	RMO	PSO
f_1	Mean	**2.67680e-018**	6.31869e-017	3.13912e-014
	Std	**2.30179e-018**	1.15724e-016	5.99525e-014
f_2	Mean	**2.21443e-017**	1.57823e-016	4.18295e-013
	Std	**1.94285e-017**	1.78346e-016	1.03217e-012
f_3	Mean	2.44453e-002	2.48765e-002	**2.17246e-010**
	Std	2.90649e-002	3.54920e-002	**7.02026e-010**
f_4	Mean	**0**	7.56728e-014	4.28023e-010
	Std	**0**	1.11558e-013	1.60122e-009
f_5	Mean	**2.54551e-005**	2.24088e+002	7.83010e+001
	Std	**1.37842e-020**	1.27893e+002	6.31680e+001
f_6	Mean	**8.88178e-016**	6.38552e-008	3.00013e-006
	Std	**2.00587e-031**	6.33102e-008	3.06260e-006
f_7	Mean	6.01118e-007	**1.18805e-013**	2.25163e-003
	Std	3.28120e-006	**2.56555e-013**	3.61174e-003

Table 3. Results of ABC, RMO and PSO on benchmark functions with dimension of 15

Function		ABC	RMO	PSO
f_1	Me	4.99706e-007	**3.17457e-017**	3.00736e-004
	Std	1.24507e-006	**3.16538e-017**	2.55414e-004
f_2	Me	3.87111e+001	**4.33235e-017**	5.60171e-002
	Std	1.48006e+001	**3.33955e-017**	3.01647e-002
f_3	Me	**5.75084e+000**	1.39190e+001	3.39054e+001
	Std	**5.25142e+000**	3.52324e-002	4.30184e+001
f_4	Me	2.16707e+000	**0**	2.79516e+001
	Std	1.22010e+000	**0**	1.11010e+001
f_5	Me	**3.43993e+002**	4.46660e+003	2.73225e+003
	Std	**1.08184e+002**	3.50720e+002	5.48055e+002
f_6	Me	3.38283e-002	**1.08484e-010**	1.73043e+000
	Std	2.37433e-002	**1.13511e-010**	8.12113e-001
f_7	Me	3.67391e-002	**2.96059e-017**	2.18996e-001
	Std	2.13001e-002	**4.99352e-017**	9.42148e-002

Table 4. Results of ABC, RMO and PSO on benchmark functions with dimension of 30

Function		ABC	RMO	PSO
f_1	Me	6.12616e-003	**3.09257e-017**	2.93943e-002
	Std	1.10377e-002	**3.06840e-017**	1.14802e-002
f_2	Me	2.55908e+002	**3.02047e-017**	6.67699e+000
	Std	5.10078e+001	**2.89243e-017**	2.76913e+000
f_3	Me	1.66525e+002	**2.89378e+001**	1.62023e+002
	Std	2.46282e+002	**4.43782e-002**	6.00192e+001
f_4	Me	2.77714e+001	**0**	1.04094e+002
	Std	8.58554e+000	**0**	1.76451e+001
f_5	Me	**2.07710e+003**	1.02135e+004	6.17883e+003
	Std	**3.14194e+002**	5.28514e+002	8.70712e+002
f_6	Me	3.83016e+000	**1.70717e-011**	3.54363e+000
	Std	7.92153e-001	**2.68388e-011**	6.32969e-001
f_7	Me	5.82595e-001	**2.96059e-017**	1.09786e+000
	Std	3.01591e-001	**4.99352e-017**	4.47926e-002

5 Discussion

Eberchart & Kennedy (1995) firstly presented PSO algorithm (PSO) which imitated the swarm behavior of birds and fish [2]. In the mathematical models of PSO, particle swarm optimizer adjusts velocities by the following expression:

$$v_i^{t+1} = v_i^t + c_1 rand1()(pbest_i^t - x_i^t) + c_2 rand2()(gbest^t - x_i^t) \qquad (5)$$

i :1,2,...,N, N :population size; t :iterations; c_1 and c_2 :positive constants; $rand1()$ and $rand2()$:uniform distribution in [0, 1].The expression (5) means that the new velocity of a particle is adjusted by the original velocity,own learning factor and social learning factor. $pbest_i^t$ is the best previous position of the ith particle. $gbest^t$ is the best position among all the particles in the swarm. In RMO algorithm, the expression (1) is simialr in form to the expression (5) .However, the expression (1) has its own biological significance for roots growth in essence. A root apex regrows depending many factors including gravity,water,soil nutrient,and so on. We use the root apex g_{best} with the best fitness as the position providing the best water and soil nutrient. The original position of the root apex x_i means inertia of growth. In addition, we add a random factor r to make root apices grow more naturally.

Table 5. Results of ABC, RMO and PSO on benchmark functions with dimension of 128

Function		ABC	RMO	PSO
f_1	Mean	2.05907e+002	**3.85926e-017**	9.06713e+000
	Std	2.99124e+001	**3.08812e-017**	1.48865e+000
f_2	Mean	4.71130e+003	**3.31398e-017**	5.78462e+002
	Std	7.44964e+002	**3.37742e-017**	1.62295e+002
f_3	Mean	1.10722e+007	**1.26939e+002**	3.38841e+004
	Std	3.03572e+006	**2.82328e-002**	8.03505e+003
f_4	Mean	2.67134e+003	**0**	1.11605e+003
	Std	2.71514e+002	**0**	7.93187e+001
f_5	Mean	**2.59243e+004**	4.84723e+004	3.25326e+004
	Std	**1.31763e+003**	1.05336e+003	2.73701e+003
f_6	Mean	1.88570e+001	**3.16843e-013**	1.05328e+001
	Std	2.58762e-001	**4.12864e-013**	6.95615e-001
f_7	Mean	6.61743e+002	**1.85037e-017**	3.32923e+001
	Std	1.08413e+002	**4.20829e-017**	4.18386e+000

6 Conclusion

Root Mass Optimization (RMO) algorithm, based on the root growth behavior of plants, is presented in this paper. Seven benchmark functions were used to compare with PSO and ABC. The numerical experimental results show the performance of RMO outperforms PSO and ABC on most benchmark functions. RMO is potentially more powerful than PSO and ABC on functions with multiple dimensions.

A further extension to the current RMO algorithm may lead to even more effective optimization algorithms for solving 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.

Acknowledgments. This work is partially supported by The National Natural Science Foundation of China (Grants nos. 61174164, 61105067, 51205389, 71001072, 71271140).

References

1. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
2. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: IEEE Int. Conf. Neural Networks, pp. 1942–1945 (1995)
3. Yang, X.-S.: Nature-inspired Metaheuristic Algorithms. Luniver Press (2008)
4. Passino, K.M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine 22, 52–67 (2002)
5. Niu, B., Wang, H.: Bacterial Colony Optimization. In: Discrete Dynamics in Nature and Society 2012 (2012)
6. Li, X., Shao, Z., Qian, J.: An optimizing method based on autonomous animats: fish-swarm algorithm. Systems Engineering Theory & Practice 22, 32–38 (2002)
7. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed Optimization by Ant Colonies. In: The 1st European Conference on Artificial Life, Paris, France, pp. 134–142 (1991)
8. Karaboga, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization. Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
9. Trewavas, A.: Green Plants as Intelligent Organisms. Trends in Plant Science 10, 413–419 (2005)
10. Trewavas, A.: Response toAlpi et al.: plant neurobiology – all metaphors have value. Trends in Plant Science 12, 231–233 (2007)
11. Cai, W., Yang, W., Chen, X.: A Global Optimization Algorithm Based on Plant Growth Theory: Plant Growth Optimization. In: International Conference on Intelligent Computation Technology and Automation (ICICTA), pp. 1194–1199 (2008)
12. Wolpert, D.H., Macready, W.G.: No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation 1, 67–82 (1997)
13. Wolpert, D.H.: The Supervised Learning No-free-lunch Theorems. In: The Sixth Online World Conference on Soft Computing in Industrial Applications (2001)
14. Gerwitz, A., Page, E.R.: An Empirical Mathematical Model to Describe Plant Root Systems. Journal of Applied Ecology 11, 773–781 (1974)
15. Hodge, A.: Root Decisions. Plant, Cell and Environment 32, 628–640 (2009)
16. Newson, R.: A Canonical Model for Production and Distribution of Root Mass in Space and Time. J. Math. Biol. 33, 477–488 (1995)
17. Leitner, D., Klepsch, S., Bodner, G.: A Dynamic Root System Growth Model Based on L-Systems. Plant Soil 332, 177–192 (2010)
18. Karaboga, D., Akay, B.: A Comparative Study of Artificial Bee Colony algorithm. Applied Mathematics and Computation 214, 108–132 (2009)