10.1109/ICICTA.2008.416

2008 International Conference on Intelligent Computation Technology and Automation

A Global Optimization Algorithm Based on Plant Growth Theory: Plant Growth Optimization

Wei Cai, Weiwei Yang and Xiaoqian Chen
Multidisciplinary Aerospace Design Optimization Research Center, College of Aerospace and Materials Engineering, National University of Defense Technology, Changsha, China. 410073
wcai1983@gmail.com

Abstract

A novel optimization algorithm, Plant Growth Optimization (PGO), is proposed in this paper. According to the plant growth characteristics, an artificial plant growth model is built including leaf growth, branching, phototropism and spatial occupancy. Afterward, two mechanisms are introduced and the basic process of PGO is presented in details. Three classical test problems are adopted to test the performance of this new algorithm and the results show that it is very effective to get global optimum solution which benefits from its search mechanism. Furthermore, the effects of population size on optimization performance are also discussed.

1. Introduction

In recent years, great development has been achieved in the application of artificial life especially on optimization, such as Ant Colony Optimization [1], Particle Swarm Optimization and Fish-swarm Algorithm. These bionic optimization algorithms simulate the way that nature adapting the environment and build up a random, positive feedback and distributed computation model. These algorithms are a kind of uncertain algorithm and independent from the mathematical characters of the problems.

The existing bionic optimization algorithms usually imitate animal behavior, but don't pay much attention to the plant. Compared to the animal, the movement of the plant isn't so obvious, it always growth toward the most beneficial direction. Inspired of this idea, some scholar proposed a plant growth simulation algorithm for solving integer programming [2]. In this paper a virtual plant growth model is built to simulate the plant growth process from a new angle and a new algorithm, Plant Growth Optimization, is proposed.

2. Basic concepts of artificial plant growth

In the algorithm discussed in this paper we distribute the search activities over so-called "branches and leaves", that is, agents with very simple basic capabilities which, to some extent, mimic the behavior of real plant. In fact, research on the virtual plant has greatly inspired our work, so we introduce the basic concepts of artificial plant growth in brief first.

2.1. L-systems

L-systems are a mathematical formalism proposed by the biologist Aristid Lindenmayer in 1968 [3] as a foundation for an axiomatic theory of biological development. Central to L-systems, is the notion of rewriting, where the basic idea is to define complex objects by successively replacing parts of a simple object using a set of rewriting rules or productions. The rewriting can be carried out recursively. Figure 1 is an example obtained by the L-system.

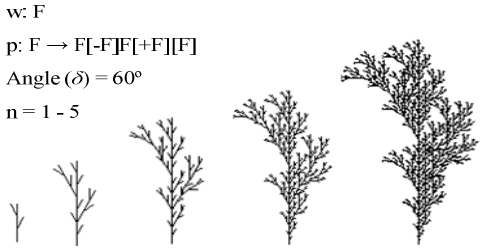


Figure 1. A L-system production

2.2. Branch axes and branch point

The term axis denotes any linear stem structure from its origin to its extremity. An axis may include monopodial or sympodial branch points. A monopodial axis is a sequence of branch segments, each of which extends its predecessor in the terminal position. Sympodial branching indicates that the terminal axis extends its predecessor and the child axis or axes appear in the lateral position [4].

978-0-7695-3357-5/08 $25.00 © 2008 IEEE
DOI 10.1109/ICICTA.2008.416

2.3. Rashevsky model of phyllotaxis

One tree has thousands branch points, which can grow up into new stems is a complex process. The first dynamic models of morphogenesis were suggested by Rashevsky, Turing, and others [5]. Figure 2 shows an idealized vine stalk. At the tip of the growing stalk is a growth bud containing a mass of undifferentiated and totipotent cells. A cell is considered as a bag of fluid with homogeneous chemical composition. One of the chemical constituents is a growth hormone called morphogen. The concentration of this morphogen is the observed parameter of the model. As the parameter varies between 0 and 1, the state space of the model is a line segment. If the concentration of this morphogen exceeds a certain critical value, the growth function of the cell is turned on, the cell divides, and a branchlet comes into existence.

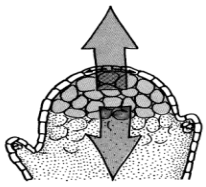


Figure 2. Rashevsky model of phyllotaxis

3. Artificial plant growth modeling

A simple but effective algorithm is presented to simulate plant growth in a realistic way in this paper, taking into account branching, phototropism, leaf growth and spatial occupancy. The main idea of the model is to select the activated points by comparing the concentration of their morphogen to augment the L-system.

3.1. Concentration of the morphogen

It has already been proved by biological experiment that the morphogen is given by the environment of the cell's position and the plant performs the characteristic of phototropism obviously. In the model, a point corresponds to a potential solution instance to the problem and its activity, represented as the concentration of the morphogen, is decided by the result of the objective function called fitness value.

In a minimization problem for example, the concentration of the morphogen A_i is defined as

$$A_i = \begin{cases} 1 - \dfrac{f(x_i) - f_{\min}}{\sum_{j=1}^{N} \left[f(x_j) - f_{\min} \right]} & , f(x_i) > f_{\min} \\ 1 & , f(x_i) = f_{\min} \end{cases} \quad i = 1, 2 \ldots N \tag{1}$$

Where x_i is the position of the point, N is the total number of the points, $f(.)$ is the fitness value of the point and $f_{\min}$ is the minimum of the current points.

3.2. Branching

In artificial plant growth model the branch point is the point that can produce new points by branching, resembles the simple object in L-systems. The purpose of branching is to seek the optimal area. Based on the Rashevsky model of phyllotaxis and inspired by the plant branching theory, we presented a random model to simulate the plant's behavior of branching.

Two different critical values α and β, satisfied $0 \le \alpha < \beta \le 1$, are selected randomly. They divide the state space of morphogen into three pieces. The branch point has three different branching modes accordingly.

$$\begin{cases} \text{Mode 1: Sympodial} & , A_i > \beta \\ \text{Mode 2: Monopodial} & , \alpha < A_i < \beta \\ \text{Mode 3: No branching} & , A_i < \alpha \end{cases} \tag{2}$$

Generally speaking, the current best branch points may not perform sympodial branching every generation. In this paper, we dispose the critical values as follows

$$\alpha = \alpha / 0.9 \quad \beta = \beta / 0.9 \tag{3}$$

The new point produced by branching of the branch points is called growth point in this paper. In order to abstain from getting stuck onto a local optimum, artificial plant produce some new points called random points in the areas with fewer axes randomly. So there are four branching modes in all.

3.2.1. Sympodial. Not only the axis elongates towards the direction of the leaf but at the same time, child axes appear in the lateral position. In this mode, the branch point's concentration of the morphogen is at a high level, that is, the fitness value is better than other branch points. PGO search the optimum solution towards to the direction of the current best point and strengthen the searching around the branch point.

3.2.2. Monopodial. The axis only elongates towards the direction of the leaf. The concentration of the morphogen is at an average level in this mode and PGO search the potential optimum solution towards to the direction of the current best point.

3.2.3. No branching. The concentration of the morphogen is at a low level and the axis doesn't grow and only performs the leaf growth behavior.

3.2.4. Branching randomly. The points appear in the sparse area randomly. This also matches an actual circumstance: The plants always produce smaller branches in the areas where the branches are sparse. The random points help to keep diversity of the points and may lead to a very good result.

3.3. Leaf growth

After producing the new points, artificial plant searches the optimum solution around these points through the operation called leaf growth. This is used to ensure the accuracy of the solution. Two concepts are introduced in this paper

- We call a point "leaf point", around the branch point, when it has a better fitness value;
- The branch point is mature in the current generation when we can't find a point with a better fitness value around the branch point;

If we can find the leaf point, record its information. Otherwise, the branch point is mature. By the way, the area for the leaf growth shouldn't be too large. The way we search the leaf point, the local search strategy, can be any optimization algorithm. In this paper, we give priority to the pattern search method. We applied it to the test problems and found that this strategy can gain higher accuracy with smaller calculation.

3.4. Selection mechanism

One tree only support a certain amount of branch points but the current branch points produce a series of new points including growth points and random points. It is necessary to establish a mechanism to select the branch points for the next generation.

Most selection mechanisms choose the points with better fitness values correspondingly. But under this mechanism, the convergence may occur very rapidly so that it becomes impossible to reach to the optimum. This problem, which is called premature convergence, is similar to the problem of getting stuck onto a local optimum encountered in local search methods.

The actual circumstance that the competition turns more vehement where the points are too crowd and the ones of stronger competition ability will restrain the lower ones. In order to keep the variety of the branch points, we produce a new point in the center of the crowded area and compare its fitness value with the closely points. The ones of lower competition ability will be eliminated and more chance is given to the points in the scanty area.

3.5. Maturity mechanism

When a branch points have grown for a certain generations and still can't find a point with a better eligibility, it is considered to be mature. A mature branch point doesn't take part in the growth. We record its information and compare it with the other branch points in the next generations.

4. The algorithms

As we are not interested in simulation of the natural plant growth process, but in the use of artificial plant as an optimization tool, our system will have some major differences with a real one. The PGO take the solution space of the problem as the growth area of the artificial plant, one point of the plant as one potential solution to the problem. The algorithm searches the optimal point in the solution space through two behaviors:

1. Producing new points by branching to search the optimal area where the optimum solution is;
2. Growing leaves around the branch points to find the accurate solution in the local area;

Given the definitions of the preceding section, formally the Plant Growth Optimization is:

0. Start
1. Initialize:
   Set NG=0 {NG is the generations counter}
   Set NC=0 {NC is the convergence counter}
   Set NM=0 {NM is the Mature points counter}
   Set the upper limit of the branch points N and initialize other parameters.
   Select N_0 branch points at random and perform leaf growth.
2. Assign morphogen
   Calculate the eligibility of the leaf point.
   Assign the concentration of the morphogen of each branch point by Eq.1.
3. Branching
   Select two critical values between 0 and 1 randomly and dispose by Eq.3.
   Produce new points by branching in four modes.
4. Selection mechanism
   Perform leaf growth in all of the points.
   Pick out the mature branch points, the number of which is k ($0 \le k \le N$), by the maturity mechanism.
   Set NM = NM + k
   Produce a new point in the center of the crowded area and select the best point to substitute the crowded points.
   Eliminate the lower competition ability branch points and select N branch points for next generation.
5. Competition
   Compare the current points with the mature points and get the best fitness value $f_{\max}$

Set: $NG = NG + 1$

If $(f_{\max} < f_{\max\_old})$ Set: $f_{\max} = f_{\max\_old}$

If $(|f_{\max} - f_{\max\_old}| < \varepsilon)$ Set: $NC = NC + 1$

else

Set: $NC = 0$

else

Set: $NC = NC + 1$

6. Check the termination criteria:

If $(NG < NG_{\max} \,\&\&\, NC < NC_{\max} \,\&\&\, NM < NM_{\max})$

Goto step2

else

Exit

7. Stop

One execution of the procedure from step2 through step6 is called a generation or a cycle.

5. Experimental study Ⅰ: Performance of the PGO

We apply the proposed algorithm to three classical test problems to test its performance. The Rosenbrock function tests the capability of avoiding premature convergence (Section 5.1). The Schaffer function tests the performance of abstaining from get stuck onto a local optimum (Section 5.2). Finally, the Speed Reducer Problem tests the performance of solving optimization problem with constraints (Section 5.3).

5.1. Rosenbrock Function

The Rosenbrock function is defined as

$$\begin{aligned} \min \quad & f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \\ \text{s.t.} \quad & x_1 \in [-10, 10] \\ & x_2 \in [-10, 10] \end{aligned} \tag{4}$$

It is a continuous unimodal optimization problem with the global minimum 0 at the optimal point (1, 1), which is on the concave of a paraboloid (Fig. 3). When region of search come into the long and narrow place on the bottom, the variation of the object is very tiny. Commonly, algorithms repeat search in the concave and finally converge near the optimal point.

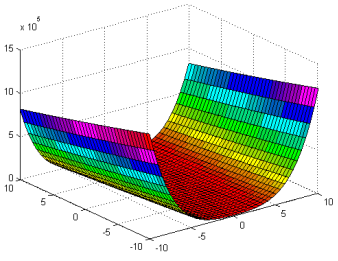


Figure 3. Rosenbrock function

We set the initial point (-10, -10), which is on the edge of the solution space. Figure 4 shows the convergence history of the Rosenbrock function. The object fell below 1 after 21 cycles. In fact, the object was close to the global optimum after 47 cycles and satisfied the termination criteria after 54 cycles. PGO got the optimal point (1.00000029, 1.00000055) at last. The results indicate that the selection mechanism is working and the PGO can overcome the problem of premature convergence.

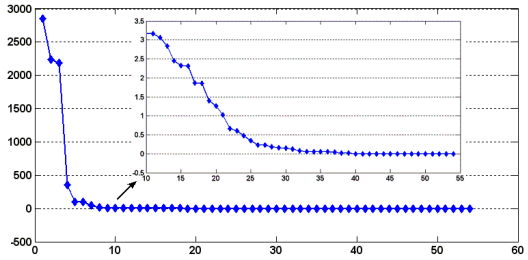


Figure 4. Convergence history of Rosenbrock function

5.2. Schaffer function

The Schaffer function is defined as

$$\begin{aligned} \min \quad & f(x_1, x_2) = -0.5 + \frac{(\sin\sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} \\ \text{s.t.} \quad & x_1 \in [-10, 10] \\ & x_2 \in [-10, 10] \end{aligned} \tag{5}$$

It is a continuous multimodal optimization problem with the global minimum -1 at the optimal point (0, 0). There are countless local optimal points around the global optimal point (Fig. 5) and algorithms usually get stuck onto a local optimum.

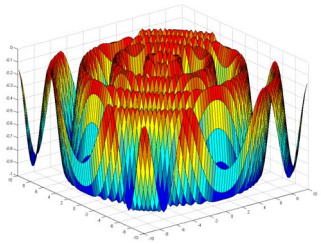
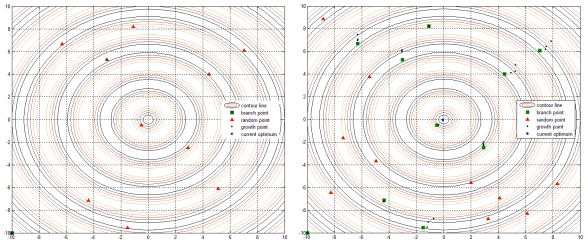


Figure 5. Schaffer function

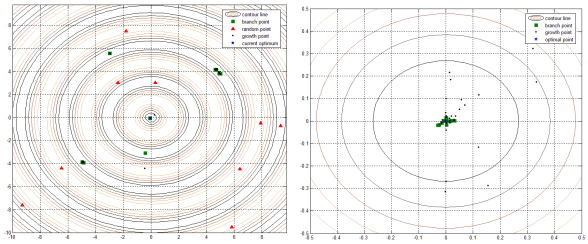
We set the initial point (-10, -10), which is on the edge of the solution space. Algorithm satisfied the termination criteria after 17 cycles and finally got the optimal point (-0.00000003, 0.00000000). We select 4 typical cycles to show how PGO found the optimal point in figure 6.

In cycle 1 the growth points were around initial point (-10, -10) and the random points guided the direction of optimization. In cycle 2 and 6 the growth and the random points distributed in the whole space. The growth points strengthened the search in the local space and the random points helped to escape the local optimum. In cycle 17 the random points had little contribution to the optimization, so we just show the

branch points and the growth points. The growth points helped to search the accurate solution and finally got the global optimum. The results of this test show that PGO can abstain from getting stuck on a local optimum and find the global optimum.



(a) Cycle 1 (b) Cycle 2



(c) Cycle 6 (c) Cycle 17

Figure 6. Typical cycles to solve Schaffer function

5.3. Speed reducer problem

The Speed Reducer Problem (SRP) represents the design of a simple gear box shown in figure 7. The objective is to minimize the speed reducer weight while satisfying a number of constraints imposed by gear and shaft design practices. Its global optimum is on the edge of the solution space. In this paper we used it to test the performance of solving optimization problem with constraints. SRP is defined as

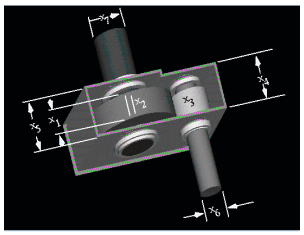


Figure 7. Speed Reducer Problem

$$\min \quad f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.5079x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \quad (6)$$

s.t.

$g_1: 27.0/(x_1x_2^2x_3) - 1 \le 0$ $g_2: 397.5/(x_1x_2^2x_3^2) - 1 \le 0$

$g_3: 1.93x_4^3/(x_2x_3x_6^4) - 1 \le 0$ $g_4: 1.93x_5^3/(x_2x_3x_7^4) - 1 \le 0$

$g_5: A_1/B_1 - 1100 \le 0$ $g_6: A_2/B_2 - 850 \le 0$

$g_7: x_2x_3 - 40.0 \le 0$ $g_8: 5.0 \le x_1/x_2 \le 12.0$

$g_9: (1.5x_6 + 1.9)/x_4 - 1 \le 0$ $g_{10}: (1.1x_7 + 1.9)/x_5 - 1 \le 0$

$$A_1 = \left[\left(\frac{745.0x_4}{x_2x_3}\right)^2 + 16.9\times10^6\right]^{0.5} \quad B_1 = 0.1x_6^3$$

$$A_2 = \left[\left(\frac{745.0x_5}{x_2x_3}\right)^2 + 157.5\times10^6\right]^{0.5} \quad B_2 = 0.1x_7^3$$

$2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8$, $17 \le x_3 \le 28$, $7.3 \le x_4 \le 8.3$

$7.3 \le x_5 \le 8.3$, $2.9 \le x_6 \le 3.9$, $5.0 \le x_7 \le 5.5$

We set the initial point (3.6, 0.8, 28, 8.3, 8.3, 3.9, 5.5), which is breach of constraints. Figure 8 shows the convergence history of the object and constraint of SRP. PGO converged after 74 cycles and found the optimum solution 2994.35502622 at the point (3.5, 0.7, 17, 7.3, 7.71531991, 3.35021467, 5.28665447), satisfied the constraints. The results show that the PGO can solve the optimization problem with constraints.

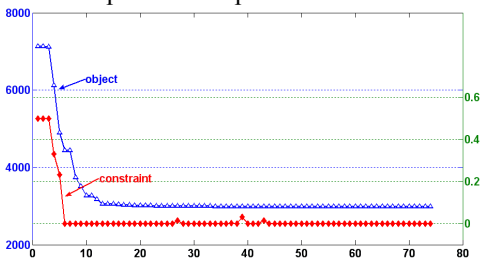


Figure 8. Convergence history of SRP

6. Experimental study Ⅱ: Population size

In this section we discuss experiments which have deepened our understanding of the Plant Growth Optimization. PGO requires parameter tunings of the population size in order to achieve desirable solutions and performance for an arbitrary complex problem. The population size discussed here is:

- N_0: the size of the initial points;
- m: the size of the random points;

The tests were carried out on the Rosenbrock function. Based on these tests we'll give some advices for the tunings of these parameters.

6.1. Size of initial points

We compared the performance of PGO when the initial points are positioned on a unique starting point with the performance obtained when the initial points are positioned on the whole space evenly. In the first case, we set the initial point at (-10, -10), which was on the edge of the solution space (N_0=1). In the second case, we use Latin Hypercube Sampling (LHS) to produce initial branch points (N_0=16). The size of random points m=10 in both cases.

The result is shown in Figure 9. PGO found the optimal points in both cases. Convergence occurred after 54 cycles in case 1 and 36 cycles in case 2. With

uniform distributing initial points PGO had a better performance. The formal Design of Experiments (DOE) techniques may aid in forming the dispersal pattern to achieve a reasonable coverage of the domain. Usually there are too many factors in a complex problem and LHS method will be feasible just need less sampling points.

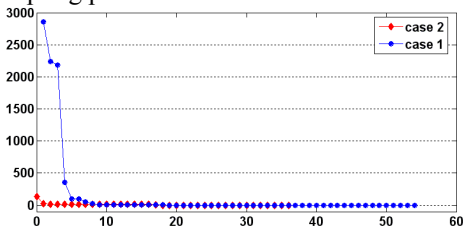


Figure 9. Convergence history of Rosenbrock Problem with different N_0

6.2. Size of random points

The purpose of producing random points is to abstain from getting stuck on a local optimum and we had already proved it in Section 5.2. This experiment was run in order to study whether the size of the random points influences the PGO's performance. We tested whether there is any difference among the cases in which m=2, 10, 50. In all of these three cases the initial point was set (-10, -10) and the number of the branch points N=20.

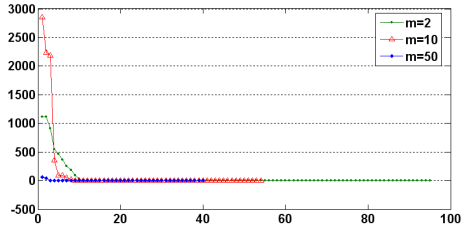


Figure 10. Convergence history of Rosenbrock Problem with different m

The convergence histories are shown in Fig. 10. PGO found the optimal points in all of the three cases. Convergence occurred after 95 cycles in case 1, 54 cycles in case 2 and 40 cycles in case 3. The results indicated that the more the random points is the faster PGO escaped the local optimum. Figure 11 shows the total number of all points (random points, growth points and branch points) in the solution, which can be used to evaluate the elapsed task time. It was least when m=10, the results show that a moderate size (about half of the branch points) of the random points could improve the PGO's performance.

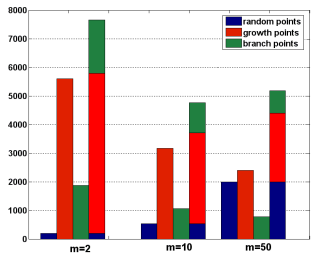


Figure 11. Number of the points with different m

7. Conclusion

A global optimization algorithm, Plant Growth Optimization, based on the plant growth theory is proposed. The main idea of PGO is that each branch point searches optimum in the local space and the plant guides the search direction by refreshing the branch points to enhance the search around the good points which is closer to optimum. It is applied to three classical test problems and the results show that PGO has the desirable characteristics as following:

- It is a global optimization algorithm. The search mechanism of the PGO can help to overcome the problems of getting stuck on a local optimum and premature convergence.
- It is adaptable which can be applied to the optimization problems with constraints or not.
- It has a good openness. The local search strategy of the PGO can be chosen as any optimization algorithm. It means that PGO can be improved to be a hybrid algorithm easily with other excellent algorithm.

Further research is recommended to the study of the practical applications for complex systems.

8. References

[1] M.Dorigo, V.Maniezzo, and A.Colorni "The Ant System: Optimization by a colony of cooperating agents",*IEEE Trans. On Systems, Man, and Cybernetics*, Vol.26, 1996, pp. 1-13.

[2] LI Tong, WANG Chun-feng, WANG Wen-bo, and SU Wei-ling, "A Global Optimization Bionics Algorithm for Solving Integer Programming - Plant Growth Simulation Algorithm", *SYSTEMS ENGINEERING - THEORY & PRACTICE*, No.1, 2005, pp.77-85.

[3] A.Lindenmayer, "Mathematical models of cellular interaction in development, Parts I and II", *Journal of Theoretical Biology*, 1968, 18: 280–315.

[4] P. Prusinkiewicz, W. Remphrey, "Characterization of architectural tree models using L-systems and Petri nets", *The Tree 2000: Papers presented at the 4th International Symposium on the Tree*, pp. 177-186.

[5] M.Klaus, *Thinking in Complexity*, Springer Berlin Heidelberg New York, 2007.