A Hybrid Artificial Life System for Optimization of Functions

*CHEN YANG, **HONGYU SU, ***HAO YE
*Dept. of Automation, Tsinghua Univ., Beijing, 100084, P.R. CHINA
* yangchen00@mails.tsinghua.edu.cn
**Tsinghua Tongfang Co. Ltd., Beijing 100084, P.R. CHINA.
**suhy@it.thtf.com.cn
***Dept. of Automation, Tsinghua Univ., Beijing, 100084, P.R. CHINA
*** haoye@tsinghua.edu.cn

Abstract: In this paper, a hybrid Artificial Life (ALife) system for function optimization that combines ALife colonization with Genetic Algorithm (GA) is proposed. The method includes two stages. In the first stage, the emergent colonization of ALife system is used to provide an excellent initial population for GA, and GA is further used to find the optimal solution in the second stage. The hybrid method is compared with ordinary ALife based optimization approach and ordinary GA approach respectively by simulation.

Key-Words: Artificial Life, Evolution Computation, Genetic Algorithm, Emergent Colonization

1 Introduction

Artificial Life (ALife) emerged through the interaction of Biology and Computer Sciences [1]. AL not only contributes to the understanding of the secret of life, but also provides innovative concepts and important approaches to practical applications of AI, engineering, computer science and so on.

Some bio-inspired mathematical models in AL such as emergent colonization [2] in the artificial ecology have provided the ideas of distributed algorithms that can be successfully applied to the parallel optimization and design. The underlying idea is that some complex tasks may be performed by distributed activities over massively parallel systems composed of computationally simple elements.

The Artificial Life system, which was proposed by Andrew and Norman [2], is a computational model of organisms in an artificial ecology where colonization emerges through a process of resource gathering and exchange amongst an evolving population. However, except some work done by Hayashi et al. [3] and Yang and Lee [4], Yang and Ye [5], the Artificial Life system has not been widely applied to the optimization problem.

Based on the Artificial World (AWorld) first proposed by Andrew and Norman [2], Yang and Ye [5] has proposed an optimization approach that combines the Artificial Life system with Ant Algorithm. Due to the introduction of Ant Algorithm, the searching speed of the approach has been improved greatly compared with other optimization approaches based on Alife.

However, although colonization emerges in the optimum area in that approach, the ALife system cannot always give out a solution that is good enough because it is just chosen from a limited number of Artificial Organisms (Artorgs) in the AWorld. So in this paper, the classic evolution computation methods such as GA is further introduced into the optimization method based on ALife and Ant Algorithm [5], and a hybrid optimization approach that combines the Alife system and GA is proposed, in which the emergent colonization of ALife system is used to provide an excellent initial population for GA to start, and GA is further used to find the optimal solution.

In order to keep the diversity of GA, the concept of minimal distance is introduced to the process of choosing the initial population from the emergent colonization of the ALife System. After such processing, the individuals of the initial population of GA will keep a certain distance from each other.

This proposed ALife optimization method utilizes no gradient information and so that can be applied to very wide class of optimization problems even if the functions are not continuous and have no derivatives.

In the paper, after the introduction, the procedures of ALife system based optimization and GA will be introduced in Section 2 and 3 respectively. Section 4 will give the two-phase process as a whole and test the algorithm with Functions. Conclusions will be given in the last section.

2 Optimization approach based on ALife system and Ant Algorithm [5]

2.1 Artificial World [2][6]

AWorld first proposed is square Cartesian plane containing N-by-N discrete points [2]. In their approach, each point in the AWorld can contain energy resources, and can also be inhabited by members of the four different species of Artorgs.

The Artorgs move in the AWorld, seeking different types of food, consuming food, trading with each other according to a genetically encoded evolving strategy, mating, reproducing and dying. The four species of Artorgs (white, red, green, and blue) compose a circular food chain where one species waste is another's food. An Artorg must maintain a minimum internal energy level by consuming enough food in order to exist; otherwise it will die and is removed from the AWorld. If the Artorgs can survive to the age of sexual maturity, they can reproduce, and their offspring's strategy is changed using crossover and mutation of the parents' trading strategies. The Artorgs have sensor systems to find the food in their neighborhoods in which they can move in one step.

In our algorithm for function optimization, the AWorld is defined as n-dimensional continuous Euclidean space with lower and upper limits, i.e.,

$$X = \left\{ x \in R^n \middle| x_{\min} \le x \le x_{\max} \right\} \tag{1}$$

The neighborhood of the point $\bar{x}$ at which an Artorg is inhabited is defined by

$$C = \left\{ x \in R^n \middle| \left\| x - \bar{x} \right\| \le D \right\} \tag{2}$$

2.2 Basic idea of optimization

If the AWorld where the artificial organisms live is regarded as the solution space of an optimization problem, the artificial organisms can be agents in the space to search for the optimal solution. The motivation for choosing emergent colonization of artificial organisms as an optimization mechanism such as minimizing the value of a function is that the colonization may occur in the promising area with lower function values if the ability of artificial organisms in the Alife system is properly selected.

In the process, the locations of both Artorgs and resources become the variables of the object function to be optimized. So, the objective function values for Artorgs and resources can proceed by substituting the location into the objective function. Each Artorg moves to the location of the optimum resource within the neighborhood region. Therefore, Artorgs can produce an emergent colonization at the locations where the optima of objective functions exist.

2.3 Optimization Procedure [5]

According to the above idea, the optimization problem to be solved is to minimize a multimode nonlinear function formulated as the following.

$$\underset{x_{\min} \le x \le x_{\max}}{Min} \quad f(x) \tag{3}$$

The procedure of the ALife algorithm for optimization is as follows.

Step 1 Initialization: Equal numbers of each of the four species of Artorgs and resources are randomly placed in the AWorld, randomly generating sixteen bit strings as trading strategies of the initial population. Each Artorg is initially given an internal energy.

Step 2 Search resource: Each Artorg searches all the resources in its own neighborhood and finds its food considering distance and the pheromone on it laid by other Artorgs.

$$F(\tilde{x}, \bar{x}) = 1/\|\tilde{x} - \bar{x}\| + W \cdot \{p(\bar{x})\} \quad (4)$$

Where $\bar{x}$ is the present location of an Artorg and $\tilde{x}$ is that of the food. $p(\bar{x})$ is the pheromone on the Artorg which is decided by the Artorgs which have visited the food . The constant W>0 is a weight factor.

Step 3 Metabolism and lay pheromone: If an Artorg finds the resource it wants, it metabolizes the resource to get the energy. Then it produces waste on which it lays some pheromone according to the objective function value. The Artorg can also carry the resource it cannot metabolize to trade with others in the next cycle according its trade strategy.

Step 4 Increasing age: All the Artorgs' age is increased by 1.

Step 5 Reproduction: If the age and sexual maturity is reached, two Artorgs with the same color can mate and produce an offspring. The offspring inherits its trading strategy from both parents.

Step 6 Reducing energy: All Artorgs' energy is decreased. If an Artorg's energy drops below a lower limit, it is considered to be dead and is removed from AWorld.

Step 7 Increasing generation: Generation is increased by 1. In returning to Step 2, this hybrid algorithm is iterated until the final generation.

3 Float Encoding Genetic Algorithm

Michalewicz [7] has done extensive experimentation comparing real-valued and binary GAs and shows that the FGA (Floating Encoding Genetic Algorithm) is an order of magnitude more efficient in terms of CPU. In our experiments, it is found that FGA converges to the global optimum quickly, hardly gets stuck and gets more precise results. So the FGA is adopted for GA in this paper and the corresponding genetic operators (selection, crossover and mutation) are given as follows.

Selection Operator [8]: The reproductive probability of individual x is

$$P(x) = \left[Min + (Max - Min)\frac{Pop - rank(k)}{Pop - 1} \right] \Big/ Pop \quad (5)$$

Where Rank (x) is the rank of x in the population, Pop is the population size, Max ∈ (1,5,2), Max+Min=2, rank (x)=1,2…pop.

Then roulette wheel selection is used.

Crossover Operator: The Crossover Operator is defined as:

$$\begin{cases} a' = (1-\alpha)\cdot a + \beta b, \\ b' = \alpha a + (1-\beta)\cdot b, \\ if \quad a'(b') < L \quad then \quad a'(b') = L, \\ if \quad a'(b') > R \quad then \quad a'(b') = R, \end{cases} \quad (6)$$

Where a, b are the two corresponding variables of two parent individuals, a', b' as offsprings, α, β are uniform random numbers between (0,1), L, R are the lower and upper bound respectively.

Mutation Operator [9]: The Mutation Operator is defined as

$$c' = \begin{cases} c + \kappa\cdot(R - c)\cdot\gamma, \alpha \le 0.5 \\ c - \kappa\cdot(c - L)\cdot\gamma, \alpha > 0.5 \end{cases} \quad (7)$$

Where c is one variable from individual before mutation, c' is after mutation, κ is a parameter and α is a uniform random numbers between (0,1).

4 Two-phase Optimization Process

4.1 Basic idea

Although colonization can emerge in the optimum area in the above optimization approach based on ALife system and Ant Algorithm, it is found that the ALife system cannot always give out a solution good enough because it is just chosen from a limited number of Artorgs in the AWorld. So in this section, a hybrid optimization approach with two phases that combines the ALife system and GA is proposed, in which the emergent colonization of ALife system is used to provide an excellent initial population for GA in the first stage, and GA is further used to find the optimal solution in the second stage.

4.2 The Two Phases of the algorithm

The algorithm includes two phases:

Phase I: ALife Algorithm is used to find the emergent colonization of Artorgs in the ALife system

Phase II: GA is used to find the optimal solution, which further includes two steps:

Step 1:Select enough Artorgs form the emergent colony with distance as the initial population

Step 2: Execute GA and output the best solution found so far

4.3 Selecting with Distance in GA

Obviously, the ALife system initialization can greatly improve the efficiency of GA. In order to keep the diversity while selecting initial population in a reduced space, the concept of distance is introduced. The selecting process is as follows:

Step 1: Get the best Artorg in the colonization as the first individual in GA population.

Step 2: Set L (the least distance between any two individuals in the initial population) or obtain it by some simple calculations.

Step 3: Search in the colonization until find one Artorg whose distance is no less than L from the first individual.

Step 4: Continue Step 3 until having found N (population size) individuals from the colonization, each new selected individual must have far than L from all the existing individuals.

Step 5: If cannot find N individuals satisfying L, set L a lower value and go to Step 3.

As a result, the initial population of GA will disperse as separately as possible in the emergent colony of ALife system.

5 Simulation

5.1 Test Function

In order to demonstrate the effective of the proposed method, The Banana Function is used as a test function that is defined by

$$f(x) = 100.0 \times (x_2 - x_1^2)^2 + (1.0 - x_1)^2 \qquad (8)$$

The function has one global minimum at (1.0, 1.0) and its optimal function value is 0.0.

5.2 Simulation Results

In the simulation, values of parameters are selected as W=0.5 and number of Artorgs in the initial configuration are 160=70(Artorgs)×4（species）. The search space is $x_{\min}$ =(-2, -2), $x_{\max}$ =(+2,+2).The least distance in selecting process L=(2+2)/28.GA population size is 50, $p_c = 0.8$, $p_m = 0.08$.

The initial distribution of Artorgs and the configuration after 200 circles are shown in Fig.1, 2. The initial populations selected randomly and with least distance L are also shown in the down-left and down-right of Fig. 3 respectively.

Fig.2 shows that the emergent colonization occurs at banana-shaped area that is the area of lower value of banana function. Compared with the population selected randomly, the one selected with distance has a disperser distribution that may make it more possible to keep the global optimum from ALife Algorithm and is beneficial to the diversity of GA.

For comparison, numerical results corresponding to ordinary ALife approach, the ordinary GA, the Hybrid approach based on ALife and GA but selecting initial population randomly, and the Hybrid approach which select initial population with distance are shown in Table 1 respectively. The analytical optimal solution is also given in the table.

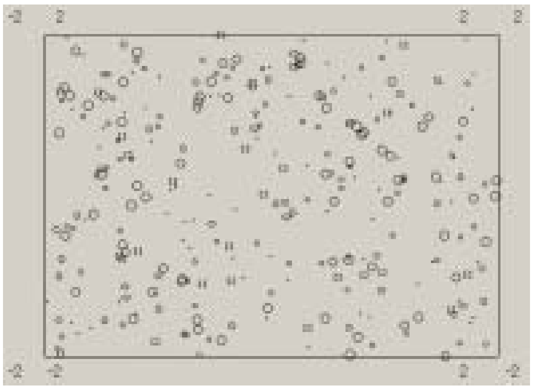

Fig.1. Initial Distribution (280 Artorgs)

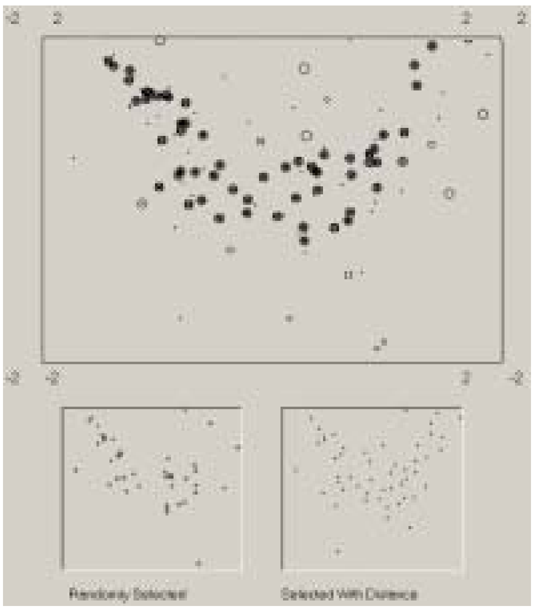

Fig.2. Colonization and the Initial Population of GA

	Best Sol.	Av. of 25 Runs
ALife	1.45E-02	1.19E-01
GA	1.99E-05	7.90E-03
Hybrid (Randomly)	1.02E-06	1.29E-02
Hybrid (Distance)	1.97E-07	3.28E-03
Analytical Optimal	0.00	0.00

Table 1. Comparison of Results

Remark: From table 1 it can be seen that compared with ALife Algorithm, GA and Hybrid ALife System can find the solution with much higher accuracy, and the Hybrid approach has higher accuracy than ordinary GA, because ALife Algorithm can provide a good start for GA that can greatly improve the GA's ability to find optima. In addition, selecting with distance is also important in the two-phase process. GA that selects with distance has better performance than what selects randomly.

6 Conclusion

In this paper, a hybrid ALife system for optimum design problems such as function minimization that combines Artificial Life colonization with GA is proposed. The optimization procedure of the algorithm is described and tested with Banana Function. Selecting the first population with distance from the emergent ecology from the ALife system, GA's efficiency and accuracy is greatly improved.

In the experiment, Artorgs, which are initially, placed at random locations in the solution space, gradually move to the closest optimum location from their present locations and find the optimums in the near regions. Therefore, Artorgs can find all global optimums and produce the emergent colonization for the optima that will be searched by GA in the following phase. At the same time, the proposed algorithm requires no additional assumptions such as continuity and differentiability;

it can be applied to very wide class of optimization problems.

References:

[1] C. Adami, *Introduction to Artificial Life*, Published by TELOS, The Electronic Library of Science, Santa Clara, California.1998, pp:1~15

[2] A. M. Assad, N. H. Packard, “Emergent Colonization in an Artificial Ecology”, in *Toward a Practice of Autonomous Systems*, MIT Press, 1992,pp.143-152

[3] Hayashi D et al, Distributed Optimization by Using Artificial Life. T. IEE Japan 1996; 116(CS): 584─30 [in Japanese].

[4] Yang BS, Lee YH, Artificial Life Algorithm for Function Optimization. In: *2000 ASME Design Engineering Technical Conferences & Computer and Information in Engineering*, 2000, DETC2000/DAC─14524.

[6] A. Colorni, M. Dorigo, V. Maniezzo, “Distributed Optimization by Ant Colonies”, in *Toward A Practice of Autonomous Systems*, MIT Press, pp.134-452, 1992.

[5] C. Yang, H. Ye, A Distributed Algorithm for Function Optimization Based on Artificial Life, *ICCA’ 02*, 2002.

[7] Michalewicz. Z., *Genetic Algorithm + Data Structure = Evolution Programs*, AI Series, Springer-Verlag, New York, 1994.

[8] Grefenstette J. J., Baker J. E., How Genetic Algorithms work: a Critical Look at Implicit parallelism, *Proc 3rd nt Conf Genetic Algorithm[C]*, 1989.

[9] Tong Z, Hua Z, Zi-cai W, Float Encoding Genetic Algorithm and its application, *Journal of Harbin Institute of Technology*, Vol.32, NO.4, 2000, pp. 59-61.