Sensors & Transducers
© 2013 by IFSA
http://www.sensorsportal.com

An Efficient Optimization Algorithm for Super High Dimensional Numerical Function Inspired by Cellular Differentiation

Yanjiang Wang, Chengna Yuan, Hui Li, Yujuan Qi
College of Information and Control Engineering, China University of Petroleum, No. 66,
Changjiang West Road, Economic and Technological Development Zone, Qingdao, 266580, China
Tel.: (86) 532-86983470
E-mail: yjwang@upc.edu.cn

Received: 1 August 2013 /Accepted: 25 October 2013 /Published: 30 November 2013

Abstract: Inspired by the cellular differentiation behaviors, a new biomimetic optimization algorithm, cellular differentiation optimization algorithm (CDOA), is proposed. First, a certain number of cells are randomly distributed in the search space in which each cell represents a solution. Then, several cellular differentiation behaviors such as division, growth, migration, adhesion and apoptosis are exhibited for finding the optimal solution according to the activity value of a cell. The proposed algorithm is applied to several benchmark complex functions optimization with 20-1000 dimensions. Experimental results show that the proposed cellular differentiation optimization algorithm can converge to the optimum of complex numerical functions with super high dimensions rapidly in spite of its simple procedure and effortless implementation. *Copyright © 2013 IFSA.*

Keywords: Cellular differentiation optimization algorithm (CDOA), Bio-inspired, Complex function with super high dimensions, Numerical function optimization.

1. Introduction

In practical projects, a large number of engineering problems can be represented as the function optimization problems. Usually these functions are very complex and have large scale, high dimensions, nonlinearity, non-convex and non-differentiable characteristics. In addition some functions have a large number of local minima. So efficient and reliable optimization strategies of high-dimension complex functions are needed urgently, and will become a hot research issue in optimization design field. Since the 1980s, the biomimetic swarm intelligence optimization algorithms have got extremely rapid development. Inspired by the natural biological system, many biomimetic swarm intelligence optimization algorithms are proposed to solve the complex optimization problems. Some typical algorithms include the genetic algorithms (GAs) [1] for simulating the genetic evolutionary process of organisms, particle swarm optimization (PSO) [2] coming of the research on bird flock preying behaviors, ant colony optimization (ACO) [3] algorithm for imitating the ant colony foraging, and the multi-agent optimization algorithm [4-7], etc. These algorithms could overcome some shortcomings of the traditional numerical methods to some extent, such as continuity, differentiability, and unimodality. Therefore, they have been applied widely and successfully in solving optimization problems in different fields such as constraint satisfaction problems and numerical function optimization [7-9], image segmentation [10, 11], image feature extraction [12], moving object tracking [13], information retrieval [14], and vehicle scheduling [15, 16], etc. However, there are also some disadvantages among the algorithms in practical applications, such as premature convergence, slow convergence speed and poor convergence performance, etc. Especially for the complex functions with high or super high dimensions, it is very difficult to achieve efficient optimization. In 2004, Zhong *et al.* [17-19] proposed a multiagent genetic algorithm (MAGA) which can deal with high dimensional function optimization problems with 20-1000 dimensions. However, the algorithm usually needs thousands of iterations to get the optimal solution even with 20 dimensional functions.

Inspired by the cellular differentiation mechanism at the micro level, this paper proposes a new optimization algorithm, the cellular differentiation optimization algorithm (CDOA) to improve the slow convergence speed and poor convergence performance of some existed biomimetic algorithms. Through the simulation of cellular differentiation behaviors, the algorithm searches the solution space in parallel to find the optimal solution. The experimental results show that the proposed CDOA is more effective in solving high even super high dimensional numerical function optimization problems.

The remainder of the paper is organized as follows. In Section 2, a brief introduction of CDOA is provided. Section 3 describes the proposed CDOA in detail, including the definitions of the cellular differentiation behaviors and the implementation of CDOA. Section 4 demonstrates the performance of the proposed algorithm compared with the multi-agent genetic algorithm (MAGA). Section 5 draws the conclusions.

2. The Main Idea of CDOA

All living things, whether they are animals or plants, are made up of cells. Cell is the basic unit of organism structure and functions. It is the basis of all life phenomenon including growth, propagation and death. All of the cells are from cellular differentiation. Cellular differentiation is a transition of a cell from one cell type to another and it involves a switch from one pattern of gene expression to another. Differentiation dramatically changes a cell's size and shape. Different cells can have very different physical characteristics despite they have the same genome. Cellular differentiation is the result of a Darwinian selective process occurring among cells. It goes through the whole life in multi-cellular organisms.

Inspired by the cellular differentiation phenomenon, this paper presents CDOA for simulating cellular differentiation process. CDOA first generates a certain number of cells in search space randomly, and each cell represents one solution. Then all cells form into an initial solution space. We make use of objective function to evaluate the activity values of cells. Cells with higher activity values are closer to the optimal solution. According to the activity values of cells, some behaviors such as division, growth, migration, adhesion and apoptosis are exhibited for generating a new generation of cells. The cells with higher activity values will survive while those with smaller activity values will be weeded out. Finally, the cell with the largest activity value will be selected as the optimal solution when the algorithm terminates. The main procedure of CDOA is illustrated in Fig. 1.

From Fig. 1 it can be seen that the parent cell 1 (with higher activity value) first pre-divide into two sub-cells and each sub-cell will perform growth, migration, and adherence behaviors to form into a new cell while cell 2 can not pre-divide due to its lower activity value and only migration and adherence behaviors are performed. After division judges can be made to select or refresh the cell swarm.

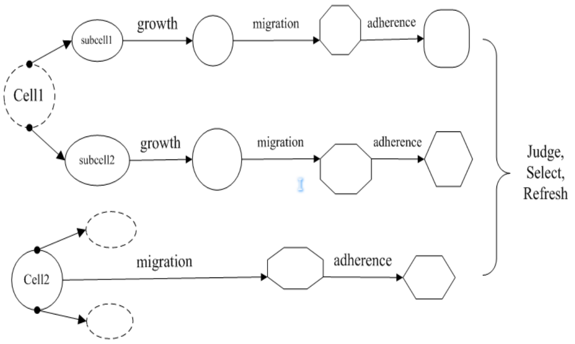


Fig. 1. The main procedure of CDOA.

3. The Modeling of Cellular Differentiation Optimization Algorithm

In this paper, the proposed CDOA is built based on function optimization. Without loss of generality, the function optimization can be described as:

$$\min \ f(X), \ X = (x_1, x_2, \cdots, x_n),$$

where $X \in S$, $S \subseteq R^n$ represents the *n*-dimensional search space $\left[\underline{s}, \overline{s}\right]^n$, $f(X)$ is the objective function and each element x_i satisfies $\underline{s} \le x_i \le \overline{s}$.

3.1. Cells

Cells are the basic functional entities in CDOA which can simulate the real cells in organisms.

Specifically, an optimization cell can be defined by a 5-tuple set:

$$cell ::= \langle l, d, age, act, beh \rangle ,$$

where l represents the location of a cell in environment, d denotes its current differentiation direction, *age* refers to its living time-steps before division, *act* stands for its activity value which can be computed by energy function, *beh* symbolizes the cell's behaviors such as division, growth, migration, adhesion, and apoptosis, etc.

3.2. Energy Function and Activity Value

Suppose each cell $c_i = [x_1, x_2, \cdots, x_n]$ represents a candidate solution for the function to be optimized, S is the search space, then the energy of cell c_i can be defined as:

$$E(c_i) = f(c_i), c_i \in S \tag{1}$$

In order to evaluate each cell in the solution space, the energy function value can be used to calculate the activity value of a cell, that is,

$$\begin{cases} act(c_i) = -E(c_i), search\ for\ f_{\min} \\ act(c_i) = E(c_i), search\ for\ f_{\max} \end{cases}, \quad c_i \in S \tag{2}$$

As will be explained, the final optimal solution can be decided by activating some differentiation behaviors such as division, growth, migration, adhesion and apoptosis according to the activity value of each cell. The definitions of these differentiation behaviors are as follows.

3.3. Cellular Differentiation Behaviors

Definition 1: Division Behavior

When the age of one cell is smaller than its lifespan, it starts to pre-divide. The parent cell pre-divides into two child cells (sub-cells). If the activity values of the two child cells are both smaller than that of the parent cell, the parent cell is performed a negative feedback and its age will be increased by one. If the parent cell's activity value is smaller than one of the child cells, it will receive a positive feedback and be replaced by the two child cells, forming into a new group of cells. In chromosome level, it means dividing the genome into two parts. In order to keep the dimension of the child cells, we set the other part of the chromosome to zero by default. Thus, the division of a cell can be defined as,

$$cell_i = \begin{cases} subcell_1\ and\ subcell_2, if\ age_i < lifespan \\ 0, \qquad\qquad\qquad\qquad if\ age_i > lifespan \end{cases}, \tag{3}$$

where age_i is the age of the ith cell, *lifespan* is the maximum age of a cell. The two sub-cells are initialized as follows after division behavior performed:

$$\begin{aligned} subcell_1 &= [x_1, \cdots, x_{\frac{n}{2}}, 0, \cdots, 0] \\ subcell_2 &= [0, \cdots, 0, x_{\frac{n}{2}+1}, \cdots, x_n] \end{aligned} \tag{4}$$

Definition 2: Growth Behavior

The growth is an important stage of cell differentiation. It is worth noting that after the cell division, the child cells are smaller and instability. Therefore, the selective expression of the gene is not mature and the growth behavior should be performed for the child cells to increase their size and promote the gene expression. In this paper we define the growth behavior by adding $n/2$ random values within a given range into the zero-part of the child cells. After growth, the child cells are updated as,

$$\begin{aligned} subcell_1 &= [x_1, \cdots, x_{\frac{n}{2}}, rand_{\frac{n}{2}+1}(), \cdots, rand_n()] \\ subcell_2 &= [rand_1(), \cdots, rand_{\frac{n}{2}}(), x_{\frac{n}{2}+1}, \cdots, x_n] \end{aligned} \tag{5}$$

Definition 3: Migration Behavior

The so-called migration behavior is the movement of cells in the search space, which makes the group of cells gather towards the cell with higher activity value. It is irreversible, namely cells can only migrate towards the optimal cell in each evolution to increase their activity values. The distance a cell moves is determined by its activity value, the bigger the activity value is, the greater probability of the cell moves to the optimal cell. Accordingly, we define the cell migration behavior as follows,

$$\begin{aligned} cell_i &= cell_i + (cell_{best} - cell_i) * step(i) * steplength \\ step(i) &= act(i) / \sum_{j=1}^{n} act(j) \end{aligned} , \tag{6}$$

where *act*(i) stands for the activity value of the ith cell, *steplength* represents the unit size of migration, $cell_{\text{best}}$ is the current optimal cell with the highest activity value.

Definition 4: Adhesion Behavior

Cell adhesion is a form of communication between cells. For any cell in a group of cells, if the activity value of the cell is smaller than that of the cells surrounding it, the cell will exchange information with the optimal cell among the neighbors to increase its activity value. The adhesion behavior embodies the idea of exchanging information and can be defined by,

$$cell_{i\_new} = cell_i + cell_{i\_best} , \tag{7}$$

$$cell_{i\_new} = [x_{i1} + x_{i\_best1}, x_{i2} + x_{i\_best2}, \cdots, x_{in} + x_{i\_bestn}], \quad (8)$$

where $cell_{i\_best}$ is the cell with the highest activity value among its neighbors, while $cell_{i\_new}$ is the new cell after adhesion. If the new cell is beyond the search space, then update the $cell_{i\_new}$ by subtracting its absolute value from the upper limit of the search space.

Definition 5: Apoptosis Behavior

The process that cells whose age is greater than the life span are removed from the system is called apoptosis behavior. Apoptosis means that a cell dies. Cell apoptosis occurs in the whole stage of cell differentiation. The division of parent cells means the apoptosis of the parent cells. Sometimes in order to improve the search efficiency and reduce computational cost, some poor performers will be eliminated in advance. For example, a cell will be eliminated if its activity value is less than sixty percent of the average activity value. That is,

$$\begin{gathered} if \;\; act(cell_i) < 0.6 \cdot \frac{\sum_{i=1}^{N} act(cell_i)}{N}, \\ cell_i \rightarrow \Phi \end{gathered} \quad (9)$$

where N is the number of cells, Φ stands for the empty set meaning that the cell is removed from the cell swarm.

The pre-division mechanism in CDOA can not only retain the cells with higher activity value, but also guarantee the diversity of the cells to some extent and reduce the probability of premature convergence. Meanwhile, it can reduce the computational cost and improve search efficiency by eliminating cells whose age is greater than the lifespan. The design of CDOA can avoid invalid search and have the ability to jump out of local optimum at a greater probability.

3.4. The Implementation of CDOA

According to the above analysis, the proposed CDOA can be implemented by the following steps:

Step 1: Initialize the cell swarm by randomly distributing N cells over the solution space. Each cell represents a solution and can be specified by an n-dimension vector;

Step 2: Compute the activity value of each cell then decide whether it satisfies the termination conditions or not. If it satisfies then end the algorithm;

Step 3: For each living cell perform division operator first, then perform growth, migration, and adhesion operators for the successfully divided cells and increase the age by one for the failed cells and then perform migration and adhesion operators;

Step 4: For each cell, decide whether it satisfies the apoptosis conditions, obsolete cells that meet the apoptosis conditions will be removed from the swarm;

Step 5: Update the cell swarm and find the one with the highest activity value (optimal cell);

Step 6: Decide whether it satisfies the termination conditions or not. If it satisfies then end the algorithm and output related parameters and results, otherwise return to Step 3.

The specific flowchart of the proposed CDOA is shown in Fig. 2.

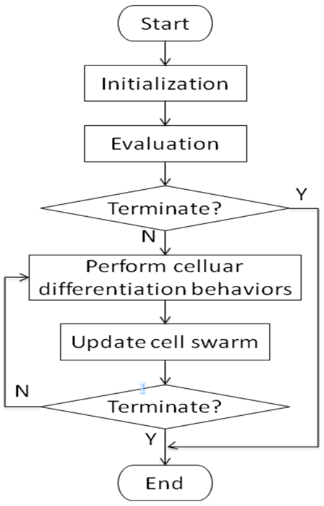


Fig. 2. The flowchart of CDOA.

4. Numerical Emulation and Analysis

To verify the feasibility and validity of the algorithm, five benchmark functions with 20-1000 dimensions are tested and the results are compared with the well-known MAGA algorithm. The termination criterion of CDOA is as follows:

$$|f_{best}| < \varepsilon, f_{\min} = 0, \quad (10)$$

where f_{best} is the best solution found in the current generation, $f_{\min}$ is the global optimal solution, and ε is the accuracy of the solution, $\varepsilon = 10^{-4}$.

The five benchmark functions to be used for performance testing are described by Eq. (11)-Eq. (15), named Rastrigin function, Ackley function, Griewank function, Sphere function, and Schwefel function, respectively. It is noteworthy that these functions are either multimodal with multiple local minima or unimodal with global minima.

Rastrigin function:

$$\begin{gathered} f_1(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10], \\ S = [-5.12, 5.12]^n; \end{gathered} \quad (11)$$

Ackley function:

$$f_2(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \\ -\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e, \\ S = [-32,32]^n; \tag{12}$$

Griewank function:

$$f_3(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(x_i/\sqrt{i}) + 1, \\ S = [-600,600]^n; \tag{13}$$

Sphere function:

$$f_4(x) = \sum_{i=1}^{n} x_i^2,\ S = [-100,100]^n; \tag{14}$$

Schwefel function:

$$f_5(x) = \max\{|x_i|, 1 \le i \le n\}, \\ S = [-100,100]^n; \tag{15}$$

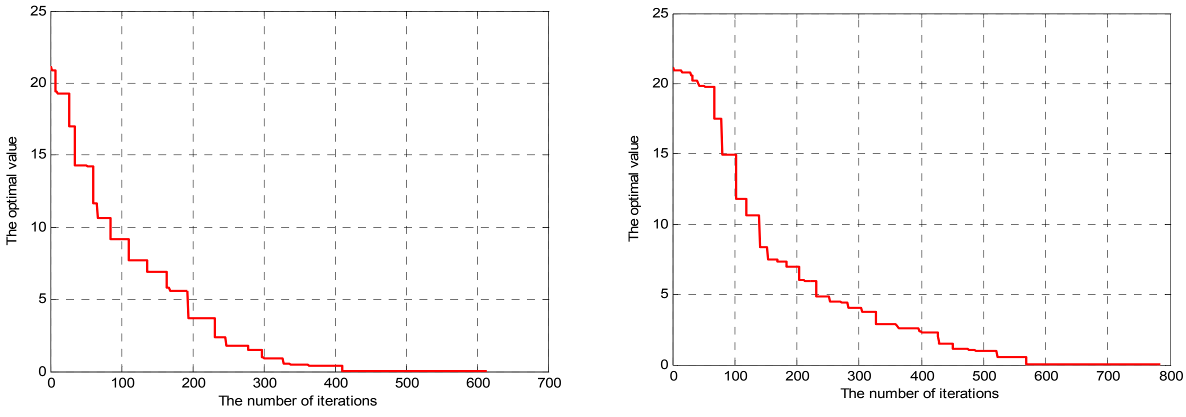
20, 100, 200, 400, 800, and 1000 dimension of the above functions are respectively chosen for testing. Some parameters of the CDOA are selected by experiments: the number of cells N = 45, the life span of a cell *lifespan* = 25, the migration step *steplength* = 145, and the accuracy of the optimal solution $\varepsilon = 10^{-4}$. Besides, the termination criterion is decided by Eq. (10).

On account of the randomness of the algorithm, the mean numbers of function evaluations averaged over 50 trials are compared between CDOA and MAGA. The results are shown in Table 1. For function $f_1(x)$, MAGA needs more than 20000 iterations to obtain the optimal solution ($\varepsilon = 10^{-4}$), while CDOA only requires a little more than 1000 iterations to reach the same accuracy. Whereas for function $f_2(x)$~ $f_5(x)$, CDOA only needs hundreds of evaluations at all listed dimensions to get the high quality solution while MAGA requires thousands of evaluations to get the same accuracy. It is obvious that CDOA obtains better solutions at a lower computational cost than MAGA, and exhibits a good performance in solving high dimensional functions.

Additionally, CDOA can also solve super high dimensional function optimization problems. Take Ackley function for example, only more than 600 iterations are needed to reach the accuracy for the case of 5000 dimension and more than 700 iterations for 10000 dimension. The convergence curves are shown in Fig. 3.

Table 1. Comparison between CDOA and MAGA in the mean number of function evaluations.

Dimension	$f_1(x)$		$f_2(x)$		$f_3(x)$		$f_4(x)$		$f_5(x)$	
	CDOA	MAGA	CDOA	MAGA	CDOA	MAGA	CDOA	MAGA	CDOA	MAGA
20	256	4301	184	3583	190	2566	110	2420	189	6823
100	576	10265	271	5410	227	4447	426	4199	200	8920
200	597	14867	334	6051	386	5483	475	4966	215	9307
400	658	17939	376	6615	423	6249	511	5576	235	9662
800	986	20306	481	7069	754	6883	653	6079	220	9823
1000	1189	20083	533	7188	963	7358	747	6273	351	9945



(a) 5000 dimension.

(b) 10000 dimension.

Fig. 3. The convergence curve for optimizing super high dimensional Ackley function.

5. Conclusions

Inspired by the cellular differentiation mechanism at the micro level, this paper proposes a new optimization algorithm, the cellular differentiation optimization algorithm (CDOA) to solve the super dimensional function optimization problems. Through the simulation of cellular differentiation behaviors, the algorithm searches the solution space in parallel to find the optimal solution. Experimental results on five benchmark complex functions with high dimensions show that the CDOA can rapidly converge to high quality solutions and has a good global optimization performance. The CDOA proposed in this paper is a new algorithm in solving high dimensional complex optimization problems. Additionally, it is simple and easy to implement. Nevertheless, there still exist some problems with the algorithm such as lack of theoretical analysis of the convergence, many other cellular differentiation behaviors not considered, and the rationale for parameter selection, etc. We will improve the algorithm and embed it into a theoretical framework as well as apply the algorithm to other optimization fields such as combinatorial optimization and constraint satisfaction problems in the future.

Acknowledgements

The paper is funded by the National Natural Science Foundation of P.R. China (No. 60873163, No. 61271407) and the Fundamental Research Funds for the Central Universities (No.R1405008A).

References

[1]. J. H. Holland, Adaptation in nature and artificial system: an introductory analysis with application to biology. Control and Artificial Intelligence, 2nd edition, *MIT Press*, Cambridge, MA, 1992.

[2]. J. Kennedy, R. C. Eberhart, Particle swarm optimization, in *Proceedings of the IEEE International Conference on Neural Networks*, Perth, 1995, pp. 1942-1948.

[3]. Marco Dorigo, Vittorio Maniezzo, Alberto Colorni, The ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man and Cybernetics*, Part B, l.26, 1, 1996, pp. 1-13.

[4]. Jiming Liu, Xiaolong Jin, Kwok Ching Tsui, Autonomy oriented computing: formulating computational systems with autonomous components, *IEEE Transactions on Systems, Man and Cybernetics*, Part A: Systems and Humans, 35, 6, 2005, pp. 879-902.

[5]. Kwok Ching Tsui and Jiming Liu, Evolutionary multi-agent diffusion approach to optimization, *International Journal of Pattern Recognition and Artificial Intelligence*, 16, 6, 2002, pp. 715-733.

[6]. J. Liu, Autonomous agents and multi-agent systems: explorations in learning self-organization, and adaptive computation, *World Scientific*, Singapore, 2002.

[7]. J. Liu, H. Jing and Y. Y. Tang, Multi-agent oriented constraint satisfaction, *Artificial Intelligence*, 136, 1, 2002, pp. 101-144.

[8]. E. K. Burke, D. B. Varley, A genetic algorithms tutorial tool for numerical function optimization, in *Proceedings of the 2nd Conference on Integrating Technology into Computer Science Education,* 29, 3, 1997, pp. 27- 30.

[9]. Yi-Tung Kaoa, Erwie Zahara, A hybrid genetic algorithm and particle swarm optimization for multimodal functions, *Applied Soft Computing*, 8, 2, 2008, pp. 849–857.

[10]. Fengjie Sun, Ye Tian, Transmission line image segmentation based on GA and PSO hybrid algorithm, in *Proceedings of the International Conference on Computational and Information Sciences*, 2010, pp. 677-680.