

Research Article

A Multiobjective Optimization Algorithm Based on Discrete Bacterial Colony Chemotaxis

Zhigang Lu, Tao Feng, and Zhaozheng Liu

Key Lab of Power Electronics for Energy Conservation and Motor Drive of Hebei Province, Yanshan University, Qinhuangdao, Hebei 066004, China

Correspondence should be addressed to Zhigang Lu; zhglu@ysu.edu.cn

Received 18 February 2014; Accepted 26 June 2014; Published 14 July 2014

Academic Editor: Yang Xu

Copyright © 2014 Zhigang Lu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Bacterial colony chemotaxis algorithm was originally developed for optimal problem with continuous space. In this paper the discrete bacterial colony chemotaxis (DBCC) algorithm is developed to solve multiobjective optimization problems. The basic DBCC algorithm has the disadvantage of being trapped into the local minimum. Therefore, some improvements are adopted in the new algorithm, such as adding chaos transfer mechanism when the bacterium choose their next locations and the crowding distance operation to maintain the population diversity in the Pareto Front. The definition of chaos transfer mechanism is used to retain the elite solution produced during the operation, and the definition of crowding distance is used to guide the bacteria for determinate variation, thus enabling the algorithm obtain well-distributed solution in the Pareto optimal set. The convergence properties of the DBCC strategy are tested on some test functions. At last, some numerical results are given to demonstrate the effectiveness of the results obtained by the new algorithm.

1. Introduction

In the field of optimization, many researchers have been inspired by biological processes such as evolution [1, 2] or the food-searching behavior of ants [3] to develop new optimal methods such as evolutionary algorithms or ant codes. These techniques have been found to perform better than the classical heuristic or gradient-based optimization methods, especially when faced with the problem of optimizing multimodal, nondifferentiable, or discontinuous functions. Examples of applying these biologically inspired strategies in the field of engineering range from aerodynamic design [4] to job-shop scheduling problems [5]. Another biologically inspired optimization method is the chemotaxis algorithm, pioneered by Bremermann [6] and his coworkers [7, 8], that is proposed by analogy to the way bacteria react to chemoattractants in concentration gradients. This algorithm was analyzed for chemotaxis in a three-dimensional (3D) gradient by Bremermann [6] and employed for the training of neural networks [7, 8]. This strategy is evaluated on a number of test functions for local and global optimization, compared with other optimization techniques, and applied to the

problem of inverse airfoil design [9]. A similar algorithm is the guided accelerated random search technique [10], which was applied to optimize parameters in flight control systems, [11] and to optimize perceptrons [12]. The BCC algorithm is a new colony intelligence optimization algorithm which was introduced in [13]. This novel algorithm considers not only the chemotactical strategy but also the communication between the colony members, and the performance has improved greatly.

Most real-world optimization problems require making decisions involving two or more goals. When these goals are the minimization or maximization of the objective functions they are typically referred to as multiobjective optimization (MO). From the 1950s, a variety of methods known as classical have been developed for the solution of multiobjective optimal problems (MOP). These methods are based on formal logic or mathematical programming [14]. Another interesting biological process that has been already implemented as an optimization technique is the BCC algorithm. This novel technique exposed the potential of implementing bacterial chemotaxis as a distributed optimization process, recognizing that in natural colonies,

it is the interaction and communication between bacteria the mechanism that enables them to develop biologically advantageous patterns. Many real-world binary-discrete optimization problems require making decisions involving two or more goals that typically are in contradiction with each other. So in this paper the DBCC algorithm is developed to solve MOP. The basic DBCC algorithm has the disadvantage of being trapped into the local minimum and also cannot maintain the population diversity in the Pareto optimal set. Therefore, some improvements are adopted in the new algorithm, as adding arithmetic operator when the bacteria choose their next locations and taking into account the chaos transfer mechanism.

This paper is organized as follows. Section 2 describes the basic theorem of MOP. The model is shown in Section 3. The convergence properties of the developed DBCC strategy which is applied on test functions are given by comparing with other algorithms in Section 4. Finally, some conclusions are drawn in Section 5.

2. Preparative Theorem of MOP

A MOP [15] is defined as the problem of finding a vector of decision variables that satisfies some restrictions and optimizes a vector function, whose elements represent the values of the functions. A MOP may be formulated as follows [16]:

$$\begin{aligned} \text{Min: } & f_m(x) \quad m = 1, 2, \dots, M \\ \text{Subject to: } & g_j(x) \geq 0 \quad j = 1, 2, \dots, J \\ & h_k(x) = 0 \quad k = 1, 2, \dots, K \\ & x_i \in \{0, 1\} \quad i = 1, 2, \dots, n, \end{aligned} \quad (1)$$

where x is the vector of discrete decision variables $x = (x_1, x_2, \dots, x_n)^T$ and $f_m(x)$ are the m objective functions. The J inequalities g_j and the K equalities h_k are known as constraint functions.

For MOP, instead of a single optimal, there is a set of optimal solutions known as Pareto optimal front (POF). Any solution of this set represents a balance between the objective functions; therefore, it is not possible to say that there is other solution in the search space which is superior to this one when all objectives are considered. In the minimization MOP, Pareto optimality can be mathematically defined as follows [17].

Definition 1. Pareto dominate: given two candidate solutions x and y from S , where S is the feasible region, the vector $f(x)$ is said to dominate vector $f(y)$ (denoted by $x \prec y$) if and only if,

$$\begin{aligned} f_m(x) \leq f_m(y), \quad & \forall m \in \{1, 2, \dots, M\}, \\ f_m(x) < f_m(y), \quad & \exists m \in \{1, 2, \dots, M\}. \end{aligned} \quad (2)$$

Denoted by $x \succ y$ if and only if,

$$\begin{aligned} f_m(x) \leq f_m(y), \quad & \exists m \in \{1, 2, \dots, M\}, \\ f_j(x) \geq f_j(y), \quad & \exists j \in \{1, 2, \dots, M\}. \end{aligned} \quad (3)$$

Definition 2. Pareto optimal: the candidate solution $x^* \in S$ is Pareto optimal if and only if,

$$x \prec x^*, \quad \neg \exists x \in S. \quad (4)$$

Definition 3. Pareto optimal set

$$F = \{f(x) \mid y \prec x, \neg \exists y \in S\}. \quad (5)$$

The set of solutions that satisfy all constraints is known as the Pareto optimal set and the fitness values corresponding to these solutions form the Pareto front or trade-off surface in objective space. In most cases it is not easy to find analytical expressions for the line or curve that contains the POF; thus, commonly optimal solutions points and the objective functions values in them are calculated. In order to find optimal solutions, there are two goals that any multiobjective optimization algorithm (MOA) seeks to achieve [18]:

- (1) guide the search towards the global Pareto optimal region,
- (2) maintain the population diversity in the Pareto optimal front.

3. Multiobjective Optimal Algorithm Based on DBCC (MOADBCC)

In this section, we present further improvements to the algorithm. MOADBCC can be summarized in the following steps:

The strategy parameters T_0 , b , and τ_c are relevant to the calculation precision ε

$$\begin{aligned} T_0 &= \varepsilon^{0.30} \cdot 10^{-1.73}, \\ b &= T_0 \cdot (T_0^{-1.54} \cdot 10^{0.60}), \\ \tau_c &= \left(\frac{b}{T_0} \right)^{0.30} \cdot 10^{1.16}. \end{aligned} \quad (6)$$

Step 1. Initialize variables.

Generate N bacteria randomly by using binary-discrete code and make it turn to real code. Compute the velocity v . In this model, the velocity is assumed to be a scalar constant value $v = 1$. Evaluate the fitness of the bacteria and store the nondominated solutions in the Pareto optimal set.

Step 2. Individual bacterial optimization and find the new position \vec{x}_1 .

(a) Compute the duration of the trajectory τ from the distribution of a random variable with an exponential probability density function

$$P(X = \tau) = \frac{1}{T} e^{-\tau/T}. \quad (7)$$

The time T is given by

$$T = \begin{cases} T_0 & \text{for } \vec{x}_{\text{pre}} \prec \vec{x}_{\text{cur}} \\ T_0 \left(1 + b * \min \left\{ \left| \frac{f_{m_{\text{pr}}}}{l_{\text{pr}}} \right| \right\} \right) & \text{for } \vec{x}_{\text{pre}} \succ \vec{x}_{\text{cur}}, \end{cases} \quad (8)$$

where T_0 is minimal mean time, \vec{x}_{pre} and \vec{x}_{cur} are the position of a bacterium in the previous and current step, respectively, $f_{m \text{pr}} m = 1, 2, \dots, M$ is the difference between the actual and the previous function value, \vec{l}_{pr} is vector connecting the previous and the actual position in the parameter space, and b is dimensionless parameter.

(b) The position and motion of a bacterium are defined by x , with a radius γ and $n-1$ angles $\Phi = \{\phi_1, \phi_2, \dots, \phi_{n-1}\}$, $x_1 = \gamma \prod_{s=1}^{n-1} \cos(\phi_s)$, $x_i = \gamma \sin(\phi_{i-1}) \prod_{s=i}^{n-1} \cos(\phi_s)$, $i = 2, 3, \dots, n$, $x_n = \gamma \sin(\cos(\phi_{n-1}))$.

The angle ϕ_i between the previous and the new direction obeys Gaussian distribution, for turning left or right, respectively,

$$\begin{aligned} P(X_i = \phi_i, v_i = \mu_i) &= \frac{1}{\sigma_i \sqrt{2\pi}} \exp \left[-\frac{(\phi_i - \nu_i)^2}{2\sigma_i^2} \right], \\ P(X_i = \phi_i, v_i = -\mu_i) &= \frac{1}{\sigma_i \sqrt{2\pi}} \exp \left[-\frac{(\phi_i + \nu_i)^2}{2\sigma_i^2} \right], \end{aligned} \quad (9)$$

where $\phi_i \in [0^\circ, 180^\circ]$, the expectation value $\mu = E(X_i)$, and the variance $\sigma_i = \sqrt{\text{VAR}(X_i)}$

$$\begin{aligned} \mu_i &= 62^\circ (1 - \cos(\theta)), \\ \sigma_i &= 26^\circ (1 - \cos(\theta)) \end{aligned} \quad (10)$$

with

$$\cos(\theta) = \begin{cases} 0 & \text{for } \vec{x}_{\text{pre}} \prec \vec{x}_{\text{cur}} \\ \exp\left(-\frac{\tau_c}{\tau_{\text{pr}}}\right) & \text{for } \vec{x}_{\text{pre}} \succ \vec{x}_{\text{cur}}, \end{cases} \quad (11)$$

where τ_{pr} is the duration of the previous step.

(c) Compute the position \vec{x}_1 of the bacterium. The length of the path l is given by

$$l = \nu \tau. \quad (12)$$

The normalized new direction vector \vec{n}_u with $|\vec{n}_u| = 1$ is multiplied by l to obtain the displacement vector \vec{x}_1

$$\vec{x} = \vec{n}_u l \quad (13)$$

such that the new location of the bacterium is

$$\vec{x}_1 = \vec{x}_{\text{pre}} + \vec{n}_u l. \quad (14)$$

Step 3. Optimize by the bacterial colony and find the new position \vec{x}_2 .

The individual bacterial acquires information about its environment; compute the best neighbor center \vec{x}_i . We can get the new position by

$$\vec{x}_2 = \vec{x}_{\text{pre}} + \vec{l}, \quad (15)$$

where $\vec{l} = R(0, 2) \cdot \text{dis}(\vec{x}_i, \text{Center}(\vec{x}_i))$, where $R(0, 2)$ is the random number within the interval $(0, 2)$ and $\text{dis}(x, y)$ is the distance connecting the vector x and y .

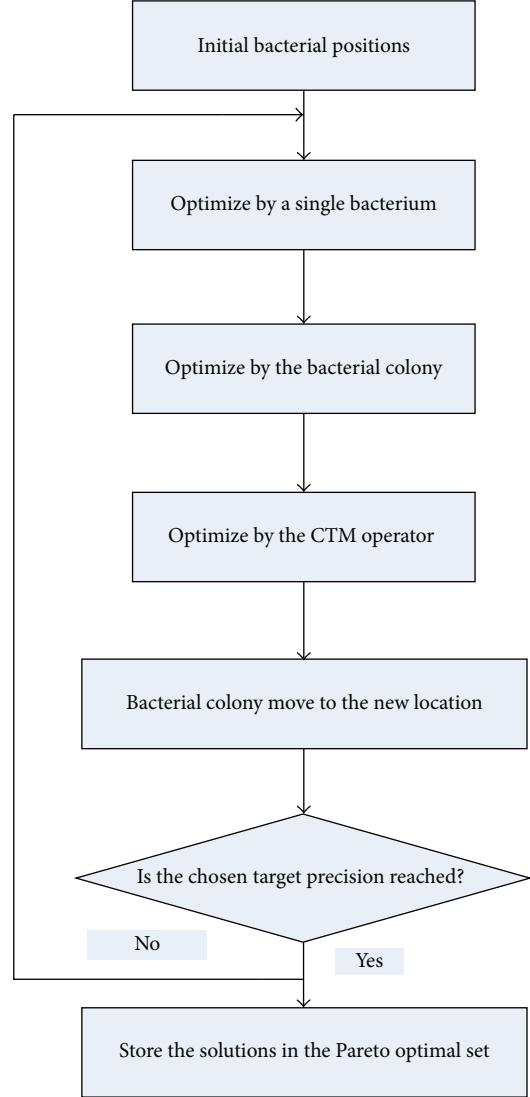


FIGURE 1: Flowchart of the MOADBCC algorithm.

Step 4. Optimize by the crowding distance operator and find the new position \vec{x}_3 .

The new position of the individual bacterial with crowding distance operator is given by

$$\vec{x}_3 = \vec{x}_{\text{pre}} + F \cdot \vec{x}_{R\text{pr}}, \quad (16)$$

where $F \in [-1.2, 1.2]$ is assumed to be a real number and $\vec{x}_{R\text{pr}}$ is the biggest crowding distance position of bacterium in the previous step. The count of \vec{x}_3 is given no more than $N * 5\% + 2$.

Step 5. The reference colony \vec{x}_{ref} is produced and bacteria move to the new location [19].

Step 6. If the chosen target precision ε_{end} is not reached, then go to Step 2; otherwise end.

TABLE 1: Test functions.

Problems	n	Constraint	Objective functions
SCH	1	$[-10^3, 10^3]$	$f_1(x) = x^2$ $f_2(x) = (x - 2)^2$
DEB	2	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = (1 + 10x_2) \times \left[1 - \left(\frac{x_1}{(1 + 10x_2)} \right)^2 - x_1 \times \frac{\sin(8\pi x_1)}{(1 + 10x_2)} \right]$
ZDT1	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - \left(\frac{x_1}{g(x)} \right)^{0.5} \right]$ $g(x) = 1 + \frac{9(\sum_{i=2}^n x_i)}{(n - 1)}$
ZDT2	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - \left(\frac{x_1}{g(x)} \right)^2 \right]$ $g(x) = 1 + \frac{9(\sum_{i=2}^n x_i)}{(n - 1)}$
SRN	2	$[-20, 20]$	$f_1(x) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$ $f_2(x) = 9x_1 - (x_2 - 1)^2$ $g_1(x) = x_1^2 + x_2^2 - 225 \leq 0$ $g_2(x) = x_1 - 3x_2 + 10 \leq 0$

TABLE 2: Comparison table of CM.

Standard test function	SCH	ZDT1	ZDT2
MODBCC (with operators)			
Mean	0.003100	0.000909	0.000816
Stdev	0.000178	0.000458	0.005400
MODBCC (without operators)			
Mean	0.003300	0.003400	0.005600
Stdev	0.000258	0.005400	0.007100
NSGA-II			
Mean	0.003200	0.104700	0.187300
Stdev	0.000150	0.197700	0.229800

In practical optimization, the calculation precision ε can be adjusted adaptively with $\varepsilon_{\text{next}} = \varepsilon_{\text{begin}}/\alpha$, where $\alpha > 1$ is the precision update constant.

As the initial precision is reached, the parameters are adapted to another precision (defined by the number of parameter changes) and the search continues, until this new precision is reached. A given precision ε_{end} is reached if the difference between function values found by the bacterium is smaller than ε_{end}

$$|f_{m \text{ pr}}| < \varepsilon_{\text{end}} \quad m = 1, 2, \dots, M \quad (17)$$

for a given number n_{end} .

When the

$$|f_{m \text{ pr}}| < \varepsilon_i \quad m = 1, 2, \dots, M, \quad (18)$$

the difference between the actual and the previous function value is reached in a row n_i that has assumed before. The bacterial colony has to move to other positions and neglects information that the bacterial colony has held in previous

step. This has been called bacterial migration. We modify the algorithm with chaos transfer mechanism (CTM).

Therefore, one has the following:

$$X_{n+1} = \mu X_n (1 - X_n). \quad (19)$$

So the algorithm can escape from the local optimal region.

In the other side, sometimes the bacterium jumps out of the area of the global optimal and is never able to get back. To prevent the bacteria strategy from leaving the global optimal regions, the algorithm will search each of the bacterial migration positions, compare the function values with that in the POF, and then take the better one into the POF. Flowchart of the MOADBCC algorithm is given as shown in Figure 1.

4. Tests Results of the MOADBCC

MOADBCC with operators, MOADBCC without operators, and NSGA-II are, respectively, applied on simulation, among which NSGA-II applies binary coding to obtain Pareto

TABLE 3: Comparison table of RVI.

Standard test function	SCH	ZDT1	ZDT2
MODBCC (with operators)			
Mean	1.000000	0.970000	0.965000
Stdev	0.007200	0.012900	0.011100
MODBCC (without operators)			
Mean	0.990000	0.720000	0.665000
Stdev	0.029100	0.043600	0.056900
NSGA-II			
Mean	1.000000	1.000000	1.000000
Stdev	0.000000	0.000000	0.122100

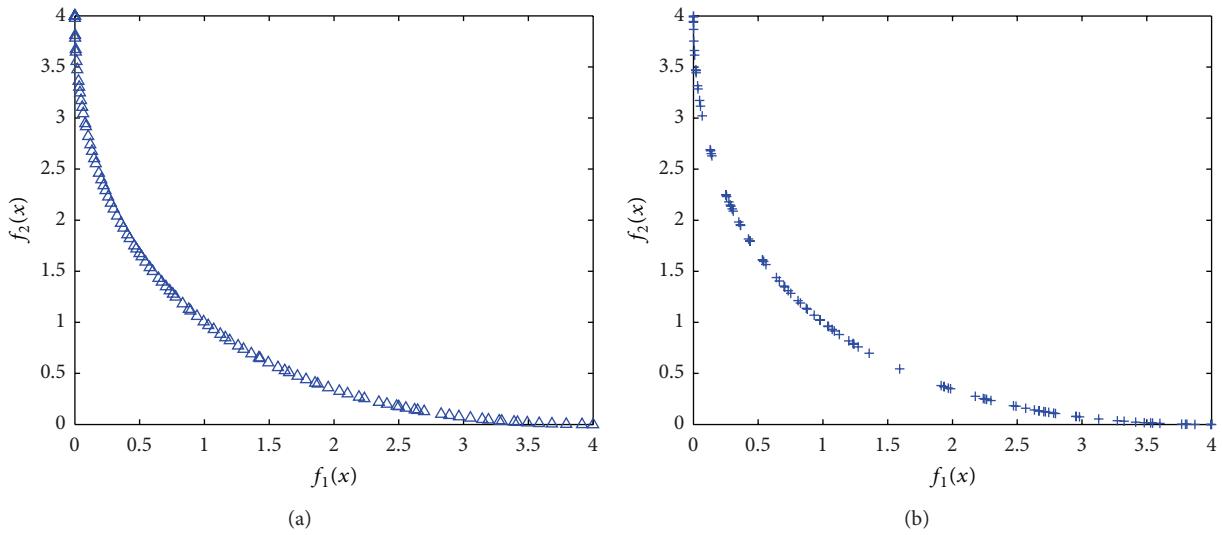


FIGURE 2: The result of MOADBCC with and without operators on SCH.

optimal solution set. The parameter selection of MOADBCC algorithm can refer to [6]. The initial precision is set as 2.0 with a precision update constant $\alpha = 1.05$. The cross probability of NSGA-II is 0.9, and the variation probability is 0.01. The colony scales of all these three algorithms are 100 and the test function iterates 100 times except for the ZDT series function which iterates 200 times. Test condition was AMD Athlon(tm) IIX2 255 Processor, 2 GB Memory, Microsoft Windows XP operation system; the procedure was tested in Matlab 7.1.

This paper applies five test functions to examine the performance of the algorithm, including the research achievement SCH of Schaffer [20], DErB, and ZDT1 and ZDT2 [21–23] of Deb and SRN [24, 25] of test function Srinivas with constraint condition,. Binary coding and discretization treatment are applied by test function while the optimization is conducted in the discrete area. Among them, ZDT1 and ZDT2 and DEB apply 10 coding and SRN and SCH apply 20 coding. Table 1 shows test functions.

In order to evaluate the comprehensive performance of the algorithm, CM, and RVI performance evaluation indexes are applied in this paper to measure the advantages of the algorithm performance.

(1) Convergence index (CM) [26] is used to measure the intimacy level between the solutions obtained by the

algorithm and real solution set. The smaller the CM index value is, the more intimate the obtained solutions are with real POF

$$\text{CM} = \frac{\sqrt{\sum_{i=1}^N d_i^2}}{n}. \quad (20)$$

(2) Effective solution proportion index (RVI) refers to the proportion of nondominant solutions in all solutions, which is used to examine the convergence degree of final solutions. The bigger the RVI index value is, the better performance the algorithm is

$$\text{RVI} = \frac{V}{N}, \quad (21)$$

where V means that the nondominant solutions minus the bacteria amount in same location.

The algorithm operates for 30 times independently for each index to obtain their mean value Mean and standard deviation Stdev. Tables 2 to 3 show the mean and Stdev of CM and RVI index for MOADBCC with operators, MOADBCC without operators, and NSGA-II respectively.

From the table of comparisons, the results show that all indexes of MOADBCC with operators are better than that of MOADBCC without operators. The performance of

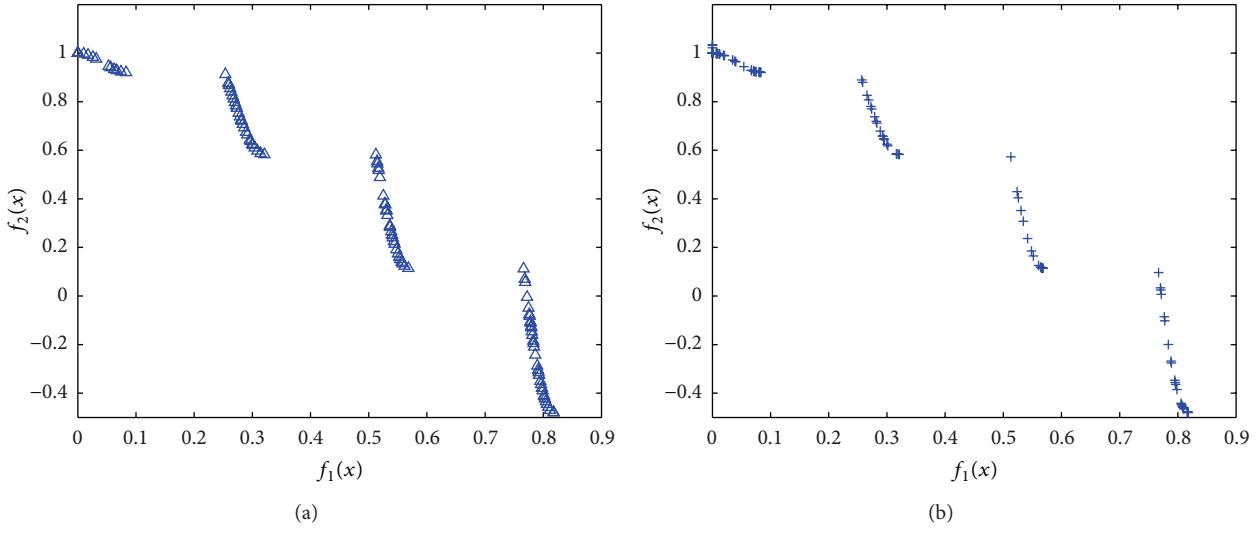


FIGURE 3: The result of MOADBCC with and without operators on DEB.

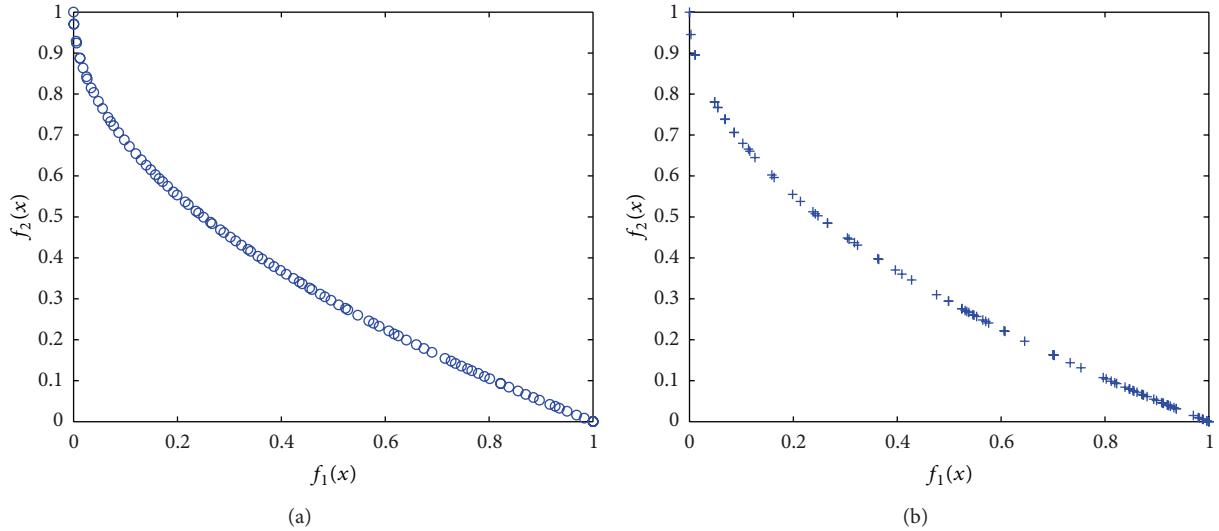


FIGURE 4: The result of MOADBCC with and without operators on ZDT1.

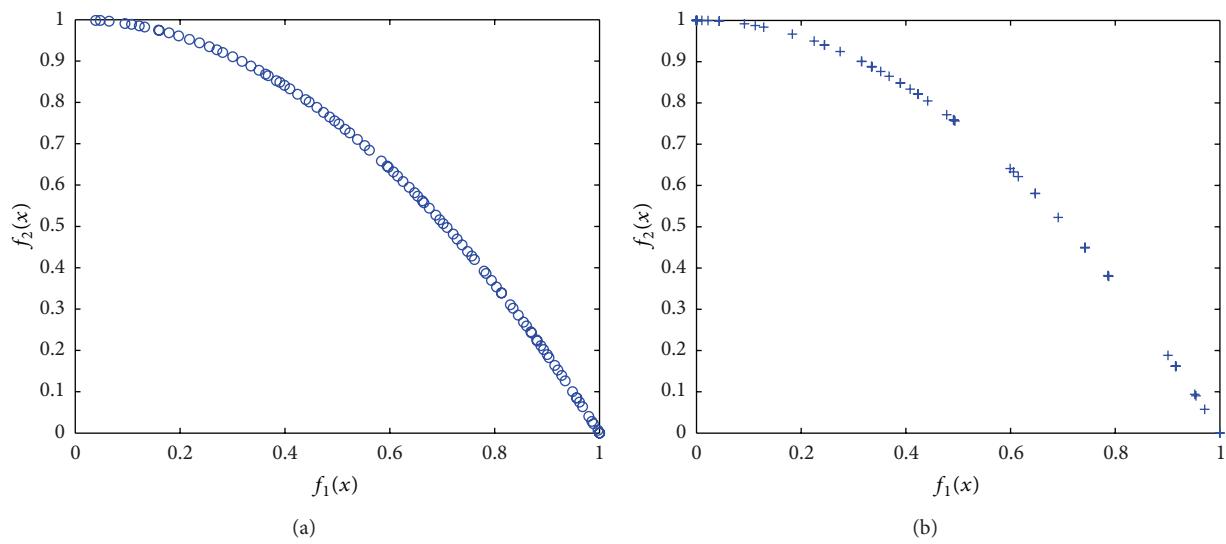


FIGURE 5: The result of MOADBCC with and without operators on ZDT2.

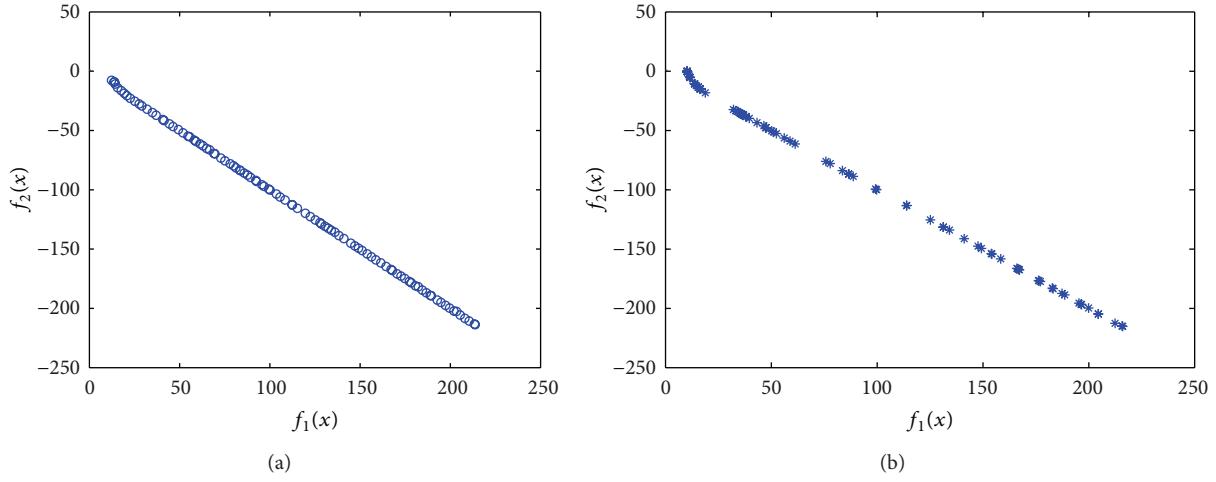


FIGURE 6: The result of MOADBCC with and without operators on SRN.

MODBCC and NSGA-II is equal to each other in terms of RVI index. However, the performance of MOADBCC is better than NSGA-II in terms of CM index, indicating it has better performance than NSGA-II algorithm. It is due to the fact that the operators proposed by this paper make it easier for bacteria to jump out of local convergence and accelerate the convergence speed of the algorithm, thus enabling it to obtain more effective solutions with same iterations and increasing the value of RVI.

In order to further compare the performances among these three algorithms, this paper listed the Figures of simulation results of three algorithms corresponding to SCH, DEB, and FON test functions, shown in Figures 2, 3, 4, 5, 6, and 7. From the simulation Figures, those results show that MOADBCC with operators is better than the other two algorithms in view of test problems in the Pareto Front, as it can converge to the Pareto Front more effectively and obtain more well-distributed solutions.

5. Conclusions

In this paper, a discrete bacterial colony chemotaxis (DBCC) algorithm is developed by chaos transfer mechanism and crowding distance strategy and used to solve multiobjective optimization problems. An additional term containing the chaos transfer mechanism draws the search away from local optima. Incorporation of crowding distance strategy also improves performance. The performance on tests is given to demonstrate the validity of the proposed algorithm. In fact, the results given in this paper not only show the effectiveness of the DBCC for the test functions, but also generally support the use of DBCC for other practical problems. Hence, there are future works including parameter settings and more applications of the proposed algorithmic framework to other optimization problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

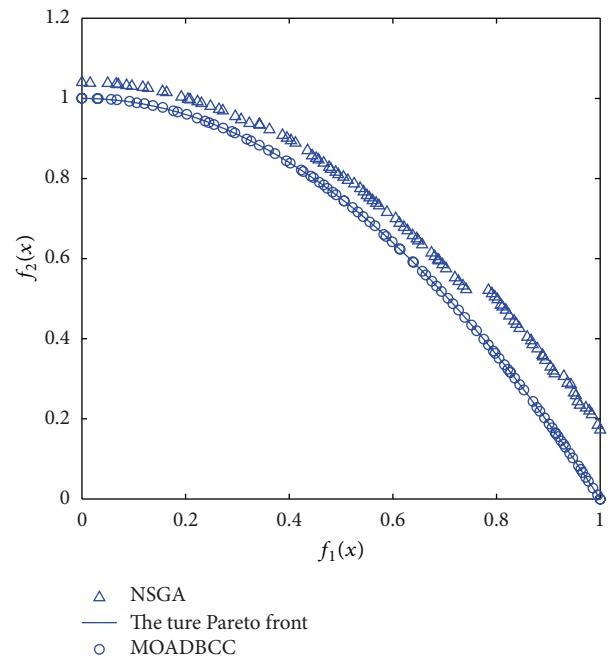


FIGURE 7: The result of MOADBCC and NSGA-II on ZDT2.

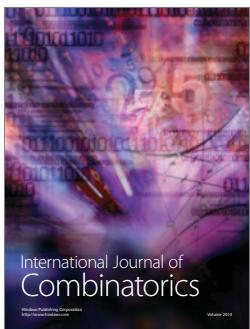
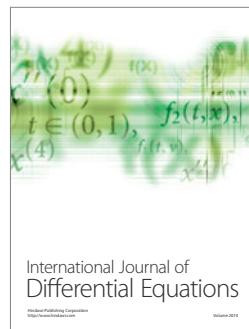
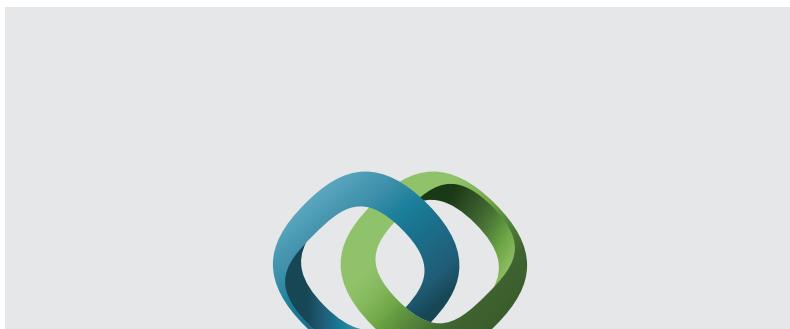
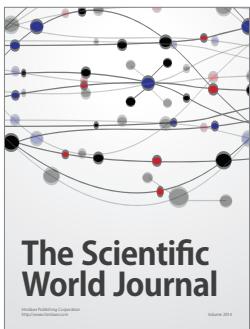
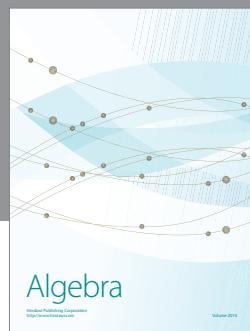
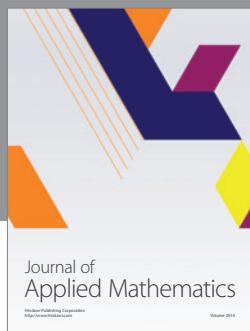
Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. Project 61374098 was supported by National Natural Science Foundation of China.

References

- [1] H. P. Schwefel, *Evolution and Optimum Seeking*, Wiley, New York, NY, USA, 1995.
 - [2] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.
 - [3] M. Dorigo, *Learning and natural algorithms [Ph.D. dissertation]*, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy, 1992.

- [4] S. Obayashi, "Pareto genetic algorithm for aerodynamic design using the Navier-Stokes equations," in *Genetic Algorithms in Engineering and Computer Science*, Wiley, New York, NY, USA, 1997.
- [5] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian, "Ant system for job-shop scheduling," *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 34, no. 1, pp. 39–53, 1994.
- [6] H. J. Bremermann, "Chemotaxis and optimization," *Journal of the Franklin Institute*, vol. 297, no. 5, pp. 397–404, 1974.
- [7] H. J. Bremermann and R. W. Anderson, "How the brain adjusts synapses—maybe," in *Automated Reasoning: Essays in Honor of Woody Bledsoe*, R. S. Boyer, Ed., pp. 119–147, Kluwer, Norwell, Mass, USA, 1991.
- [8] R. W. Anderson, "Biased random-walk learning: a neurobiological correlate to trial-and-error," in *Neural Networks and Pattern Recognition*, O. M. Omnidvar and J. Dayhoff, Eds., pp. 221–244, Academic Press, New York, NY, USA, 1998.
- [9] S. Müller, S. Airaghi, and J. Marchetto, "Optimization based on bacterial chemotaxis," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 16–29, 2002.
- [10] R. L. Barron, "Self-organizing and learning control systems," in *Proceedings of the Cybernetic Problems in Bionics. Bionics Symposium*, pp. 147–203, Gordon and Breach, New York, NY, USA, May 1966.
- [11] "Neuromine nets as the basis for predictive component of robot brains," in *Proceedings of the 4th Annual Symposium American Society of Cybernetics—Cybernetics, Artificial Intelligence, and Ecology*, H. W. Robinson and D. E. Knight, Eds., pp. 159–193, Spartan, Washington, DC, USA, 1972.
- [12] A. N. Mucciardi, "Adaptive flight control systems," in *Proceedings of the Principles and Practice of Bionics—NATO AGARD Bionics Symposium*, pp. 119–167, Brussels, Belgium, September 1968.
- [13] W. W. Li, H. Wang, and Z. J. Zou, "Function optimization method based on bacterial colony chemotaxis," *Chinese Journal of Circuits and Systems*, vol. 10, no. 1, pp. 58–63, 2005 (Chinese).
- [14] G. B. Dantzig and M. N. Thapa, *Linear Programming 1: Introduction*, Springer Series in Operations Research, Springer, New York, NY, USA, 1997.
- [15] M. A. Guzmán, A. Delgado, and J. de Carvalho, "A novel multiobjective optimization algorithm based on bacterial chemotaxis," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 3, pp. 292–301, 2010.
- [16] K. Deb, *Multi-objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester, UK, 2009.
- [17] A. Chinchuluun and P. M. Pardalos, "A survey of recent developments in multiobjective optimization," *Annals of Operations Research*, vol. 154, pp. 29–50, 2007.
- [18] K. Deb, "Evolutionary algorithms for multi-criterion optimization in engineering design," in *Proceedings of the Evolutionary Algorithms in Engineering and Computer Science (EUROGEN '99)*, K. Miettinen, M. Mäkelä, P. Neittaanmäki, and J. Périaux, Eds., pp. 135–161, Jyväskylä, Finland, 1999.
- [19] C. Huilin, L. Zhigang, and S. Songqiang, "Multiobjective optimization using bacterial colony chemotaxis," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems*, 2011.
- [20] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 93–100, 1985.
- [21] K. Deb, "Multi-objective genetic algorithms: problem difficulties and construction of test problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.
- [22] K. Deb, *Scalable Test Problems for Evolutionary Multi-Objective Optimization*, Springer, Berlin, Germany, 2005.
- [23] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 253–274, 2003.
- [24] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evolutionary Computation*, vol. 2, pp. 221–248, 1994.
- [25] T. T. Binh and U. Korn, "MOBES: a multiobjective evolution strategy for constrained optimization problems," in *Proceedings of the IMACS World Congress on Scientific Computation*, pp. 357–362, Berlin, Germany, 1997.
- [26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.



Submit your manuscripts at
<http://www.hindawi.com>

