Numerical Approach to the Inverse Convection-Diffusion Problem

Xiao-hua Yang¹, Dun-xian She¹, Jian-qiang Li²

1. State Key Laboratory of Water Environment Simulation, School of Environment, Beijing Normal University, NO.19, Xinjiekouwai St., Beijing, 100875, China
Email: xiaohuayang@bnu.edu.cn

2. Water Resources and Hydropower Planning and Design General Institute, MWR, Beijing 100011, China

Abstract: In this paper, the inverse problem on source term identification in convection-diffusion equation is transformed into an optimization problem. To reduce the computational cost and improve computational accuracy for the optimization problem, a new algorithm, chaos real-coded hybrid-accelerating evolution algorithm (CRHAEA), is proposed, in which an initial population is generated by chaos mapping, and new chaos mutation and simplex evolution operation are used. With the shrinking of searching range, CRHAEA gradually directs to an optimal result with the excellent individuals obtained by real-coded evolution algorithm. Its convergence is analyzed. Its efficiency is demonstrated by 15 test functions. Numerical simulation shows that CRHAEA has some advantages over the real-coded accelerated evolution algorithm, the chaos algorithm and the pure random search algorithm.

1. Introduction

This paper deals with a class of inverse problems related to the source term identification in convection-diffusion equations. There is a growing interest in the inverse problems in water quality modeling[1-3]. The source identification problem is transformed into a nonlinear optimization problem here. These kinds of nonlinear parameter optimization models are very intractable[4]. Firstly the objective function is very complicated, sometimes is non-derivative. Secondly the objective function has many local extreme points. The genetic algorithm (GA) based on the genetic evolution of species was proposed by Holland[5]. GA is a probability of global optimization algorithm, which is independent of the form of the objective function. GA and other evolution algorithms have been applied to many fields[6-10]. However, the computational cost is very large. And GA is very difficult to maintain the population diversity. To overcome these difficulties, standard binary-coded GA (SGA), was improved with real-coded evolution algorithm (REA)[6]. But REA still needs a large cost of computation.

Nelder and Mead have proposed a simplex search method, which is a simple direct search technique that has been widely used in unconstrained optimization scenarios [11,12]. One of the reasons for its popularity is that this method is easy to use and does not need the derivatives of the function under exploration. However, one has to be very careful when using this method since it is very sensitive to the choice of initial points and not guaranteed to attain the global optimum.

Chaos is the phenomenon which occurs in a deterministic nonlinear dynamic system [13]. The chaotic mapping has special characters such as the ergodic property, stochastic property and sensitivity depending on initial conditions. Taking advantage of this characteristic feature of the chaotic system,

¹ To whom any correspondence should be addressed.
one can design an efficient approach for maintaining the population diversity in their interest fields[14].

In this paper, a new algorithm, chaos real-coded hybrid-accelerating evolution algorithm (CRHAEA) is proposed for nonlinear optimizations and source term identifications in convection-diffusion equation in which initial population is generated by chaos mapping. It gradually directs to an optimal result with the excellent individuals obtained by real-coded evolution algorithm and simplex searching operation. Its efficiency is demonstrated by application of 15 nonlinear test functions and one source term identification problem.

2. Chaos mapping

The chaotic system can be produced by the following well-known one-dimensional logistic mapping defined by formula (1)[13].

$$x(i + 1) = \mu \cdot x(i) \cdot (1 - x(i)), x(i) \in [0,1], \quad i = 1 \sim n$$

where $x(i)$ is the value of the variable $x$ at the $i$th iteration, $x(i) \in [0,1]$; $\mu$ is a so-called control parameter of the system. For different parameter $\mu$, figure 1 can be obtained according to Eq. (1).

![Figure 1. Bifurcation for logistic mapping (for each value of $\mu \in [0, 4]$, the initial value of $x(0) = 0.29$, and the ordinate is $x(i + 1)$ for $i = 1, 2, ..., 300$).](image)

Figure 1 shows the values of $x(i + 1)$ obtained with increasing $i$ from 0 to 300 and plotted against various values of the parameter $\mu$. Although the above equation is deterministic, it exhibits chaotic dynamics when $\mu = 4$ and $x(0) \not\in \{0.25, 0.5, 0.75\}$. That is, it shows the sensitive dependence on initial conditions, which is the basic characteristic of chaos. A minute difference in the initial value of the chaotic variable will result in a considerable difference in its long time behavior. The track of chaotic variable can travel ergodically over the whole search space. In general, the above chaotic variable has special characters, i.e. ergodicity, pseudo-randomness and irregularity.

Figures 2(a)-(f) show the distribution of generated points given by logistic mappings with different values of $\mu$ at $x(0) = 0.29$. One notices that when the value of $\mu$ gets to 3.60, a relatively even distribution of the generated points occurs. Unexpectedly, a period-3 pattern appears when $\mu$ reaches 3.85, and figure 2(e) exhibits chaotic dynamics when $\mu = 4$. The value of parameter $\mu$ in the logistic mapping is selected as 4.00 in this paper.
Figure 2. (a) The distribution of generated points given by logistic mappings with $\mu = 3.60$; (b) The distribution of generated points given by logistic mappings with $\mu = 3.70$; (c) The distribution of generated points given by logistic mappings with $\mu = 3.85$; (d) The distribution of generated points given by logistic mappings with $\mu = 3.90$; (e) The distribution of generated points given by logistic mappings with $\mu = 4.0$; (f) The distribution of generated points given by logistic mappings with $\mu = 4.01$ (the $x$-axis denotes variable $n$ and $y$-axis denotes variable $x$).

Figure 3(a) shows its chaotic dynamics, where $n=300, \mu = 4.0, x(0) = 0.29$. Figure 3(b) shows its chaotic dynamics, where $n=500, \mu = 4.0, x(0) = 0.29$.

Figure 3. (a) Dynamics of logistic mapping at $\mu = 4.0, x(0) = 0.29, n=300$; (b) Dynamics of logistic mapping at $\mu = 4.0, x(0) = 0.29, n=500$ (the $x$-axis denotes variable $n$ and $y$-axis denotes variable $x$).

From figure 2 to figure 3, it can be seen that the track of chaotic variable can travel ergodically over the whole search space especially on the boundary at $\mu = 4.0$ when the values of $x(i+1)$ are taken with any real number between 0.0 and 1.0 and never repeats a value having turned up already.
Figure 4 gives the random variable in uniformity distribution system. Figure 4(a) shows uniformity distribution system, where \( n=300 \). Figure 4(b) shows uniformity distribution system, where \( n=500 \).

**Figure 4.** (a) The random variable in uniformity distribution system with \( n=300 \); (b) The random variable in uniformity distribution system with \( n=500 \) (the x-axis denotes variable \( n \) and y-axis denotes variable \( x \)).

From figure 3 to figure 4, it can be concluded that chaotic system is different from the uniformity random system. Chaotic variable can well travel ergodically in the whole search space especially on the boundary.

3. CRHAEA

Consider the general nonlinear optimization problem as follows:

\[
\min f(X) \quad (2)
\]

\[\text{s.t.} \quad a(j) \leq x(j) \leq b(j), \quad j = 1 \sim p\]

where \( X = \{x(j), \, j = 1 \sim p\} \), \( x(j) \) is a parameter to be optimized, \( f \) is an objective function, the range of the \( j \)th parameter \( x(j) \) is the interval \([a(j), b(j)]\). The procedure of CRHAEA is shown as follows.

**Step 1.** Real-valued encoding. Consider the following linear mapping

\[x(j) = a(j) + y(j) \cdot (b(j) - a(j))\]  

(3)

Suppose the \( j \)th parameter range is the interval \([a(j), b(j)]\). The real-valued code array of the \( j \)th parameter is denoted by the logistic chaotic variable \( y(j) \).

CRHAEA’s process operates on a population of individuals (also called real-valued code array, strings or chromosomes). Each individual represents a potential solution to the problem.

**Step 2.** Creating chaos initial population. To cover homogeneously the whole solution space and to avoid individuals entering into the same region, large uniformity random population is created in this algorithm. The initial population of \( n \) chromosomes is generated by chaos algorithm by use of logistic mapping at \( \mu=4.0, \, y(0)=-0.29 \) (here let \( n=300 \)). The \( n \)-corresponding chromosomes (individuals) are \( y(j, i), \quad (j = 1 \sim p; \, i = 1 \sim n). \)

**Step 3.** Evaluating fitness value of each individual (chromosome). Substitution of \( x(j) \) into Eq.(2) produces the objective function \( f(i) \). The smaller the value \( f(i) \) is, the higher the fitness of its corresponding \( i \)th chromosome is. So the fitness function of \( i \)th chromosome is defined as follows:

\[F(i) = \frac{1}{[f(i)]^2 + 0.1}\]

**Step 4.** Selection. Select chromosome pairs randomly depending on their fitness value (using
roulette wheel method) from the initial population. And \( n \) -chromosomes \( y_i(j,i) \), 
\( (j = 1 \sim p; i = 1 \sim n) \) are gotten.

**Step 5.** Crossover. Perform crossover on each chromosome pair \( y(j,i_1), y(j,i_2) \) according to probability \( p_c \), to generate one offspring \( y_2(j,i) \), 
\( (j = 1 \sim p; i = 1 \sim n) \) by linearly combining them randomly.

**Step 6.** Chaotic mutation. A new offspring \( y_3(j,i) \), 
\( (j = 1 \sim p; i = 1 \sim n) \) can be computed by a mutating probability \( p_m(i) = \{1- p(i)\} , \) where \( p(i) \) is the selection probability. The operator of chaotic mutation is as follows:

\[
y_3(j,i) = u(j) , \quad u_m < p_m(i) ; \quad y_3(j,i) = y(j,i) , \quad u_m \geq p_m(i)
\]

where \( u(j), (j = 1 \sim p) \) is chaotic variable, \( u_m \) is the uniformity random variable within the interval \( [0, 1] \).

**Step 7.** Nelder and Mead simplex search evolution. Nelder and Mead simplex algorithm is a useful, local descent algorithm, which does not make use of the objective function derivatives[11]. The best point in the previous phase becomes a new initial solution in the simplex search algorithm(SSA), and then a new best point is obtained by \( m \) times search with SSA algorithm. The new best point inside the offspring will be inserted to replace the worst one in the previous phase. Repeat step 3 to step 7 until the evolution times \( Q \), or the termination condition is satisfied.

**Step 8.** Accelerating cycle. The parameter ranges of \( n_s \) excellent individuals (the \( n_s \) individuals nearest to the optimum solution in the evolution) obtained by \( Q \)-times of the SSA evolution alternating are regarded as the new ranges, and then the whole process back to the real valued-encoding[15]. The CRHAEA computation is over until the algorithm running time gets to the design \( T \) times or there exists a chromosome \( c_{fi} \) whose fitness satisfies a given criterion. In the former case the \( c_{fi} \) is the fittest chromosome or the most excellent chromosome in the population. The chromosome \( c_{fi} \) represents the solution.

The parameter design of CRHAEA is given as follows: \( n = 300, n_s = 10, p_c = 0.5, m = 300, Q = 2 \).

**Convergence theorem:** Suppose the initial ranges of the variables in Eq.(2) are a closed \( p \)-dimensional space \( B_0 \):

\[
B_0 = \{(x_1, x_2, \cdots , x_p) | a^0_j \leq x_j \leq b^0_j, j = 1, 2, \cdots , p \}
\]

and the new ranges of the variables after a moment \( t \) in the accelerating evolution are

\[
B_t = \{(x_1, x_2, \cdots , x_p) | a^t_j \leq x_j \leq b^t_j, j \in \{1, 2, \cdots , p \}\}
\]

and \( 0 \leq b^t_j - a^t_j \leq b^{t-1}_j - a^{t-1}_j \). For a given constant \( a < 1 \), the CRHAEA is convergent if the shrinking interval ratio \( k^t_j \) has the following relation

\[
k^t_j = \frac{b^t_j - a^t_j}{b^{t-1}_j - a^{t-1}_j} \leq a < 1
\]

The proof is similarity with the reference [9].

4. **Numerical simulation**

In this paper, the 15 nonlinear test functions [6] are used in order to test the CRHAEA.

Three main criteria (relative error criteria, absolute error criteria, total computation number of function values criteria) [9] appeared to be very important when trying to determine the performances
of an algorithm: convergence, speed and robustness. To clarify the efficiency of CRHAEA we have done below.

The aim of this study is the search for global optimum of nonlinear functions. So it is very important that CRHAEA effectively converges to this optimum. The global optimization of the 15 test functions is accomplished by use of the following methods: standard binary-coded GA (Named SGA)[6], improved GA (Named IGA)[6] and CRHAEA. To compare with the global optimization ability of the three above algorithms objectively, the less than or equal 18,000 computations of the objective functions are done, and one of the three termination criteria is used for ensuring the optimization precision and avoiding algorithm invalidation.

For each of these criteria, the verifications are carried out in a statistic method and 100 tests for each of test functions are completed, which implies the obtained result may be reliable and an algorithm is robustness.

The computational results, convergent speed and computational accuracy of the 15 nonlinear test functions on global optimization with the methods of SGA, IGA and CRHAEA are given in tables 1~2, respectively.

### Table 1. Results with the SGA for 15 nonlinear test functions.

| Name of the functions | Number of variables | Theoretical minimum | Minimum found | Number of evaluation of the functions | Success rate % |
|------------------------|---------------------|----------------------|---------------|----------------------------------------|----------------|
| F1                     | 1                   | -1.12323             | -1.12323      | 5566                                   | 100            |
| Branin                 | 2                   | 0.39789              | 0.39789       | 8125                                   | 81             |
| Goldprice              | 2                   | 3.00000              | 3.00000       | 8185                                   | 59             |
| Quartic                | 2                   | -0.35239             | -0.35239      | 8181                                   | 83             |
| Shubert                | 2                   | -186.73091           | -186.73100    | 6976                                   | 93             |
| Hartman1               | 3                   | -3.86278             | -3.86249      | 1993                                   | 94             |
| Shekel1                | 4                   | -10.15320            | -10.13490     | 7495                                   | 1              |
| Shekel2                | 4                   | -10.40294            | -10.16770     | 8452                                   | 0              |
| Shekel3                | 4                   | -10.53641            | -10.40340     | 8521                                   | 0              |
| Hartman2               | 6                   | -3.32237             | -3.30652      | 19452                                  | 23             |
| Brown1                 | 20                  | 2                    | 43.62810      | 6844                                   | 0              |
| Brown3                 | 20                  | 0                    | 1.30600       | 8410                                   | 5              |
| F5n                    | 20                  | 0                    | 0.47353       | 8725                                   | 0              |
| F10n                   | 20                  | 0                    | 7.83515       | 9298                                   | 0              |
| F15n                   | 20                  | 0                    | 0.52117       | 9541                                   | 0              |

### Table 2. Results with IGA and CRHAEA for 15 nonlinear test functions.

| Name of the functions | Number of evaluation of the functions | Success rate % |
|------------------------|----------------------------------------|----------------|
| IGA                    | CRHAEA                                 | IGA CRHAEA     |
| F1                     | -1.12323                               | -1.12321       | 784 530 | 100 100 |
| Branin                 | 0.39791                                | 0.39789        | 2040 530 | 100 100 |
| Goldprice              | 3.00028                                | 3.00083        | 4632 530 | 100 100 |
| Quartic                | -0.35238                               | -0.35239       | 3168 530 | 100 100 |
| Shubert                | -186.72802                             | -186.73090     | 2364 550 | 100 100 |
| Hartman1               | -3.86114                               | -3.86277       | 1680 550 | 100 100 |
| Shekel1                | -10.14866                              | -10.15211      | 36388 550 | 97 100 |
| Shekel2                | -10.38253                              | -10.38739      | 36774 510 | 98 100 |
| Shekel3                | -10.51404                              | -10.52089      | 36772 510 | 100 100 |
| Hartman2               | -3.31383                               | -3.31955       | 53792 1100 | 82 100 |
| Brown1                 | 8.55162                                | 1.99879        | 128644 1200 | 0 100 |
| Brown3                 | 0.67464                                | 0.06951        | 106859 70300 | 5 85 |
| F5n                    | 0.00221                                | 0.00000        | 99945 1100 | 100 100 |
| F10n                   | 0.04960                                | 0.00000        | 113929 2030 | 49 100 |
| F15n                   | 0.00342                                | 0.00000        | 102413 1050 | 100 100 |
From Table 1 to Table 2, it can be concluded that the global convergent rate of CRHAEA is 93.3%, the global convergent rate of IGA is 60.0%, and the global convergent rate of SGA is 6.7%. So the CRHAEA is the best one both in the global convergence and the average evaluation number of the functions for the 15 nonlinear test functions.

We have performed a comparison of CRHAEA with six other methods of iterative improvement listed in Table 3: pure random search (PRS)[16], multistart (MS)[17], simulated diffusion (SD)[18], simulated annealing (SA)[19], tabu search (TS)[20] and binary-coded evolution algorithm (GA)[6]. The efficiency is qualitative in terms of the number of function evaluations necessary to find the global optimum. Each program is stopped as soon as the relative error between the best point found and the known global optimum is less than 1%. The numbers of function evaluations used by the various algorithms to optimize four test functions are listed in Table 4. It can be pointed out that we do not program ourselves the competitive algorithms used for the comparison, but we report the results published by Cvijovic (1995) and Andre (2001). We can see that results achieved with CRHAEA are satisfactory in global optimum and convergent speed (see the numbers of function evaluations in Table 4. In addition our results are the average outcome of 100 independent runs.

### Table 3. Global optimization methods used for performance analysis.

| Method | Name                  | Reference   |
|--------|-----------------------|-------------|
| PRS    | Pure random search    | Anderssen, 1972 |
| MS     | Multistart            | Rinnoy, 1987 |
| SD     | Simulated diffusion   | Aluffi-Pentini, 1985 |
| SA     | Simulated annealing   | Dekkers, 1991 |
| TS     | Tabu search           | Cvijovic, 1995 |
| GA     | Binary-coded algorithm| Andre, 2001  |
| CRHAEA | Chaos real-coded      | This work    |

### Table 4. Number of function evaluations for global optimization with seven different methods.

| Method | Function | Goldprice | Branin | Hartman1 | Hartman2 |
|--------|----------|-----------|--------|----------|----------|
| PRS    |          | 5125      | 4850   | 5280     | 18 090   |
| MS     |          | 4400      | 1600   | 2500     | 6000     |
| SD     |          | 5439      | 2700   | 3416     | 3975     |
| SA     |          | 563       | 505    | 1459     | 4648     |
| TS     |          | 486       | 492    | 508      | 2845     |
| GA     |          | 4632      | 2040   | 1680     | 53792    |
| CRHAEA |          | 530       | 530    | 550      | 1100     |

Results in Table 4 for test functions indicate that CRHAEA is reliable and efficient compared with SA, SD, TS, GA(binary encoding evolution algorithm), PRS and MS. CRHAEA significantly reduces the searching times of earlier techniques such as PRS, MS and SD, and it is efficient for the given nonlinear test functions. Numerical simulation shows that real encoding is available in CRHAEA compared with binary encoding evolution algorithm.

### 5. Source term identification in convection-diffusion equation

A convection-diffusion equation is given as follows:

$$\frac{\partial C}{\partial t} + U \frac{\partial C}{\partial x} = E \frac{\partial^2 C}{\partial x^2} - KC + \sum_{i=1}^{q} S_i \delta(x - x_i), \quad 0 < x < L, t > 0$$

$$C(0, t) = C_0, C(L, t) = C_L$$

$$C(x, 0) = C_1$$

where $C$ is the mean concentration of substance dispersed in the general cross section of the flow, $U$ is the mean flow velocity, $S_i$ is the intensity of source, $\delta$ is a Dirac function ( $x_i$ is source coordinate), $E$ is the dispersion coefficient, and $K$ is the degradable rate of pollution.
Suppose the distribution of $C(x,t)$ in $x = x_0 (0 < x_0 < L)$ and $t = T$ is known, an inverse problem of source term identification for convection-diffusion equation is making certain source term

$$
\sum_{i=1}^{q} S_i \delta(x - x_i), \quad \text{namely the position of pollution source } x_i \text{ and the intensity of source } S_i, \quad (i = 1, 2, \cdots, q),
$$

where $q$ is the total number of source term.

The inverse problem is investigated assuming that a source set on the abscissa $x$ introduces passive substances into the current with an intensity described by an unknown time law $S_i$. Having available an auxiliary measurement of the concentration $C(x,t)$ in a point set downstream of the source and the time behavior of the concentration at the domain boundaries, we wish to decide the source term $S_i$ or $x_i$. The inverse problems also have another explanation: we hope pollutant consistence is $C(x,T)$ at $T$ moment, or pollutant concentration is $C(x_0,t)$ at position $x_0$, where $C(x,T)$ or $C(x_0,t)$ is the concentration distribution. And the question is how to select the position $x_i$ and the intensity $S_i$ $(i = 1, 2, \cdots, q)$.

The problem investigated is of great interest both in hydraulic engineering and in the broader realm of environmental engineering [2]. Often, in fact the need occurs to discover hidden pollutant sources or to reconstruct the evolution of accidental discharges of pollutants starting from information recorded in established measuring stations along the water courses.

Let $C_0 = 0, C_1 = 0, C_L = 0$, the solution of equation (7) is given as follows:

$$
C(x,t) = \sum_{n=1}^{\infty} \sum_{j=1}^{q} S_i e^{\frac{Ux}{2E} \sin \frac{n\pi x}{L}} \cdot \frac{U^2}{4E + E(n\pi/L)^2 + K} \cdot e^{-\frac{Ux}{2E} - e^{-\frac{Ux}{2E} - \frac{U^2}{4E} (n\pi/L)^2 + K}} \sin \frac{n\pi x}{L} e^{Ux/2E - U^2/2E + \delta t/4E} \sin \frac{n\pi x}{L} e^{Ux/2E - U^2/2E + \delta t/4E}
$$

(8)

In order to decide the position $x_i$ of pollutant disposal and the intensity $S_i$ of pollutants disposal, some additional conditions should be given. In this paper, we give the conditions $C(x,T)$.

From equation (8), it can be seen that $C(x,T)$ depends on $x_i$ and $S_i, i = 1, 2, \cdots, q$, where $q$ is the total number of source term. Here we let $C(x,T) = C(x,T, x_1, x_2, \cdots, x_q, S_1, S_2, \cdots, S_q)$, where $x_i$ or $S_i$ are estimated parameters, $i = 1, 2, \cdots, q$.

The problem can be formalized as follows.

$$
\min f = \sum_{j=1}^{n} |C_j - C_{\bar{j}}|
$$

(9)

where $C_j$ is observed value, $\bar{x}_j$ is observed point and $C_{\bar{j}}$ is calculated value.

This is a complicated nonlinear parameter optimization problem. This kind of parameter optimization model is very intractable mathematically with traditional optimization methods. So CRHAEA is used to solve it.

**Case:** The case concerns the following values:

$$
U = 1, \quad E = 1.5, \quad K = 2.0, \quad T = 1.5, \quad L = 20.
$$

Because of the total number of the position of pollutants disposal in a small river is less than 5 in our practical work, we let $q = 5$, and $S_i, x_i$ are all unknown. Figure 5 gives the observed value $C_j, j = 1, 2, \cdots, 30$ for difference $\bar{x}_j$, which depends on $S_1, x_i, i = 1, 2, \cdots, q$, where $q$ is the total number of source terms.
Figure 5. 30 experiment points for \( \left( x_j, C \left( x_j, 1.5 \right) \right) \).

Here let \( C(x,T) = C(x,T,x_1,x_2,...,x_q,S_1,S_2,...,S_q) \). The parameters \( S_i, x_i \) are required in this model. Ten parameters are estimated with respect to one criterion, namely the sum of least residual absolute value. The form of the objective function is described as Eq.(9).

The least residual absolute value sum \( f \) is 0.000 007 523 with CRHAEA. For CRHAEA, the evaluation number of the objective function is 3 300. Mean relative error is 0.061866 and mean absolute error is 0.000454 with CRHAEA for 30 experiment points.

The parameters \( S_i, x_i \) are given in table 5 for the intensity and the position of pollution source.

| Table 5. Parameters estimated for source term identification with four methods. |
|---------------------------|----------------|-----------------|-----------------|-----------------|
| Objective function        | Parameters     | True value      | Optimal parameters with different methods |
|                          |                |                 | CRHAEA          | RAGA            | PRS             | CA              |
|                          |                | 1.00            | 1.0040          | 0.9532          | 0.6745          | 0.0913          |
|                          | \( S_1 \)      | 2.00            | 1.9983          | 2.0554          | 1.4723          | 1.9162          |
|                          | \( S_2 \)      | 3.00            | 3.0002          | 3.0764          | 2.1214          | 3.9999          |
|                          | \( S_3 \)      | 4.00            | 4.0000          | 3.8920          | 4.3561          | 3.3631          |
|                          | \( S_4 \)      | 5.00            | 5.0041          | 5.0230          | 5.4753          | 5.1887          |
|                          | \( x_1 \)      | 0.50            | 0.4987          | 0.4899          | 0.9717          | 0.9744          |
|                          | \( x_2 \)      | 1.00            | 0.9992          | 0.9916          | 0.7139          | 0.6374          |
|                          | \( x_3 \)      | 1.50            | 1.4996          | 1.5091          | 1.4109          | 1.4741          |
|                          | \( x_4 \)      | 2.00            | 2.0006          | 2.0039          | 1.9045          | 2.4973          |
|                          | \( x_5 \)      | 2.50            | 2.5000          | 2.4983          | 2.5132          | 2.1096          |
| \( f \)                  | 0.000007        | 0.000192        | 0.042370        | 0.068107        |

The evaluation number of the objective function is 3300 and the least residual absolute value sum \( f \) is 0.000 192 356 with RAEC(real-coded accelerated evolution algorithm)[9].

The evaluation number of the objective function is 30000 and the least residual absolute value sum \( f \) is 0.042 370 460 with PRS (pure random search)[16].

The evaluation number of the objective function is 30000 and the least residual absolute value sum \( f \) is 0.068 107 840 with CA(chaos algorithm)[14].

From the above results, it can be concluded that the evaluation number of the objective function of CRHAEA is the least among the methods of RAEC, PRS, CA and CRHAEA.

The comparison of the above optimization methods is given in table 5.

From table 5, it can be seen that the position of pollution source and the intensity of the source are well found by CRHAEA. CRHAEA is better than PRS, CA and RAEC both in the convergent speed...
6. Conclusions
A new algorithm, CRHAEA is proposed for nonlinear optimization and inverse convection-diffusion problem. Chaos mapping, new chaos mutation and simplex evolution operation are used to maintain the population diversity, reduce the computational cost and improve computational accuracy. Numerical simulation and case application show that CRHAEA is available.

CRHAEA’s convergence is analyzed. Numerical simulation results show that CRHAEA is faster and more accurate than previously reported global optimization techniques SGA and IGA.

The position and intensity of pollution source are well found by CRHAEA. Compared with RAEA, CA and PRS, CRHAEA has faster convergent speed and higher computational accuracy. It is good for the global optimization of the inverse problem in convection-diffusion equation.

Acknowledgements
The research was supported by the Opening Fund of State Key Laboratory (Grant No. 2006411411), the National Science and Technology Support Program (Grant No.2006BAB14B02) and the National Basic Research Program of P.R. China (Grant No. 2003CB415204).

References
[1] Gryazin A Y, Klibanov V M and Lucas R T 2003 Journal of Computational Physics. 184 122
[2] Zou R, Lung W S and Wu J 2007 Water Resources Research. 43 27
[3] Ridolfi L and Macis M 1997 Math. Comput. Modeling. 26 69
[4] Jacob B, Vladimir P and David R 1997 Science. 276 1094
[5] Holland J H 1975 Adaptation in natural and artificial systems University of Michigan
[6] Andre J, Siarry P and Dognon T 2001 Advances in Engineering Software. 32 49
[7] Yau H T and Yan J J 2007 International Journal of Nonlinear Sciences and Numerical Simulation. 8 165
[8] Ge X S, Chen L Q and Liu Y Z 2005 International Journal of Nonlinear Sciences and Numerical Simulation. 6 113
[9] Yang X H, Yang Z F and Li J Q 2005 Communications in Nonlinear Science and Numerical Simulation. 10 355
[10] Yang X H, Yang Z F, Shen Z Y and Li J Q 2005 Progress in Natural Science. 15 448
[11] Nelder J A and Mead R 1965 Computer Journal. 7 308
[12] Olsson D M and Nelson L S 1975 Technometrics. 17 45
[13] Lorenz E N 1963 Jat Mos Sci. 20 130
[14] Liu B, Wang L and Jin Y H 2005 Solitons and Fractals. 25 1261
[15] Yang X H, Yang Z F and Shen Z Y 2005 Lecture Notes in Computer Science: to LNCS/LNAI Proceedings. 3612 129
[16] Anderssen R S 1972 Optimization (Australia:University of Queensland)
[17] Rinnoy K and Timmer G T 1987 Mathematical Programming. 39 57
[18] Aluffi P F, Parisi V and Zililli F 1985 J. Opt. Theor. Appl. 47 1
[19] Dekkers A and Aarts E 1991 Mathematical Programming. 50 367
[20] Cvijovic D and Klinowski J 1995 Science. 267 664