An Evolutionary Algorithm for Parameters Identification in Parabolic Systems*

Zhijian Wu [1], Zhilong Tang [1], Jun Zou [2], Lishan Kang [1], and Mingbiao Li[3]

[1]The State Key Lab. of Software Eng., Wuhan Uni., Wuhan, China
zjwu@public.wh.hb.cn
[2]Dept. of Math., The Chinese Univ. of Hong Kong, Hong Kang, China
[3]School of Computer Sci. and Eng., Wenzhou Normal College, Zhejiang, China

Abstract. In this paper we construct a novel evolutionary algorithm. It yields good performance on a collection of parabolic parameter identification problems. The algorithm has a good tolerability for the noise in the observed data. Even when the noise level is up to 10% we can also get such a good result.

1 Description of Problem

Let's consider the following parabolic problem:

$$\begin{cases} L(q)u = \dfrac{\partial u}{\partial t} - \dfrac{\partial u}{\partial x}(q(x)\dfrac{\partial u}{\partial x}) = f(x,t), & (x,t)\in(0,1)\times(0,T) \\ u(x,0)=u_0(x), & x\in(0,1) \\ u(0,t)=u(1,t)=0, & t\in(0,T) \end{cases}$$

Parameters identification is the process to find the potential solution $q^*(x)$ that makes $u_{q^*}(x,t)$ match the observed data of $u(x, t)$ as optimally as possible.

The interval [0,1] is divided equally into n parts, the step size $h= 1/n$, and mesh point $x_i = ih$ (i=0, 1,2,...,n). Suppose we have the observed values $u_i^{(o)}$ at points (x_i, T), $\vec{u}_{ob} = (u_1^{(o)}, u_2^{(o)}, \ldots, u_{n-1}^{(o)})$. We consider the case in which $q(x)$ is continuous and smooth. We adopt Hat Functions $\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x)$ as the basis of $q(x)$, then $q(x)$ can be expressed as $q_h(x) = \sum_{i=0}^{n} q_i \varphi_i(x)$. .Denote $\vec{q} = (q_0, q_1, \ldots, q_n)$. The parameters identification problem can be transferred to the optimization problem: $\min_{q_u} fitness(\vec{q}) = \min_{q}(h\|\vec{u}_q - \vec{u}_{ob}\|_2^2 + \frac{\beta}{h}\sum_{i=1}^{n}(q_i - q_{i-1})^2)$.

* This work was supported by Open Fund of State Key Laboratory of Software Engineering, Wuhan University, National Natural Science Foundation of China (Nos. 60133010, 60073043, 70071042)

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 1336–1337, 2004.

2 Description of Algorithm

Step 1. N individuals $\vec{q}^{(1)}, \vec{q}^{(2)}, \ldots, \vec{q}^{(N)}$ are randomly and uniformly produced in the search space to form initial population $P(0)$, evaluate $P(0)$. Set t=0;

Step 2. While t> *Max-Generation* go to Step 5.

Step 3. Select M individuals $\vec{q}^{(1)}, \vec{q}^{(2)}, \ldots, \vec{q}^{(M)}$ randomly from $P(t)$ to form a sub space $V = \left\{ \vec{q} / \vec{q} \in D^{n+1}, \vec{q} = \sum_{i=1}^{M} a_i \vec{q}^{(i)} \right\}$, where a_i satisfies the condition $\sum_{i=1}^{M} a_i = 1,\ a_i \in [-0.5, 1.5]$, produce a new individual $\vec{q}^{(new)}$ randomly in V, for i=1 to n-1 do $q_i^{(new)} = (q_{i-1}^{(new)} + q_i^{(new)} + q_{i+1}^{(new)})/3$, if $\vec{q}^{(new)}$ is better than $\vec{q}^{(worst)}$, then substitute $\vec{q}^{(new)}$ for $\vec{q}^{(worst)}$, where $\vec{q}^{(worst)}$ is the worst individual in $P(t)$;

Step 4. Select an individual $\vec{q}^*$ randomly from the population and an element q_i^* randomly from $\vec{q}^*$, conduct $q_i^* = \alpha_l + (\alpha_u - \alpha_l) \bullet rand()$, where $rand()$ is a random number in [0,1], get a new individual $\vec{q}^{(mutatiom)}$. If $\vec{q}^{(mutatiom)}$ is better than $\vec{q}^{(worst)}$, then substitute $\vec{q}^{(mutatiom)}$ for $\vec{q}^{(worst)}$; t=t+1;

Step 5. Output the best individual.

3 Numerical Experiments

In the experiments, we get observed values of $u(x)$ by adding some noise to the function $u(x)$, $\vec{u}_{ob} = (1 + \delta \bullet rand(x)) \bullet \vec{u}(x)$, where $rand(x)$ is an uniformly distributed random function in [-1,1], and δ is the noise level parameter. Parameters set: N=200, M=10, *Max-Gen*=100000, $\beta = 10^{-6}$, $\alpha_1 = 0.001$, $\alpha_u = 10.0$

Test Problem: $u(x) = \sin(2\pi x), q(x) = 3 + 2x^2 - 2\sin(2\pi x)$.

We conducted three experiments for noise level parameter δ=1%,5% and 10%. The result of the experiments is given as follows.

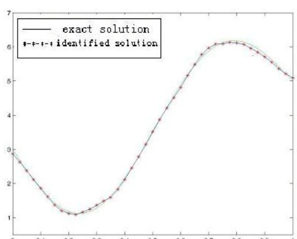

Fig. 1. δ =1%

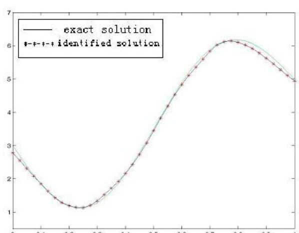

Fig. 2. δ =5%

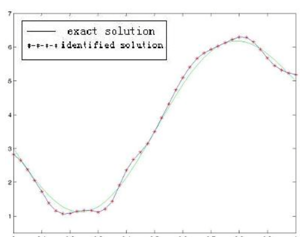

Fig. 3. δ =10%