Available online at www.sciencedirect.com
ScienceDirect
International Journal of Heat and Mass Transfer 51 (2008) 3745–3752
International Journal of HEAT and MASS TRANSFER
www.elsevier.com/locate/ijhmt

A modified genetic algorithm for solving the inverse heat transfer problem of estimating plan heat source

Fung-Bao Liu *

Department of Mechanical and Automation Engineering, I-Shou University, No. 1, Sec. 1, Syuecheng Road, Dashu Township, Kaohsiung 84001, Taiwan

Received 4 October 2007; received in revised form 10 January 2008
Available online 4 March 2008

Abstract

This work considers a new approach for solving the inverse heat conduction problem of estimating unknown plan heat source. It is shown that the physical heat transfer problem can be formulated as an optimization problem with differential equation constraints. A modified genetic algorithm is developed for solving the resulting optimization problem. The proposed algorithm provides a global optimum instead of a local optimum of the inverse heat transfer problem with highly-improved convergence performance. Some numerical results are presented to demonstrate the accuracy and efficiency of the proposed method.

Keywords: Inverse problem; Genetic algorithm; Fitness function

1. Introduction

The direct heat transfer problems consist of determining the temperature distribution inside the medium when the boundary and initial conditions, heat generation rate, thermophysical properties and geometric parameters are known. In contrast, the inverse heat transfer problems consider the identification of boundary and initial conditions, heat generation rate, thermophysical properties and geometric parameters by using the known temperature measurements at certain locations during the process.

An enormous amount of work has been dedicated to the inverse heat transfer problems since 1960s. A variety of numerical and analytical techniques for solving the inverse problem have been proposed in the literature. The least square method developed by the addition of regularization term was introduced by Beck et al. [1] and latter modified by Tikhonov and Arsenin [2]. Blackwell [3] applied the sequential estimation approach to solve the one-dimensional inverse heat conduction problem. The conjugate gradient methods have been widely used in inverse heat conduction and convection problems by Huang and his colleagues [4–7]. Development of these methods is based on the progress in mathematical theory and the advanced computer technology. To improve the efficiency of these methods and the accuracy of solutions, modified algorithms have been documented recently.

Genetic algorithms are stochastic search methods that lead a population towards an optimum using the principles of evolution and natural genetics. With the proper encoding, they can manipulate integer or continuously variables and they can handle linear and non-linear constraints. Genetic algorithms require little information of the problem itself. In this case, computations based on the algorithms are attractive to users without optimal control background. Moreover, it is well known that genetic algorithm have been successfully applied to many optimization problems. Extensive research has been performed exploiting the robust properties of genetic algorithms and demonstrating their capabilities across a broad range of problems [8–15]. Compared with the traditional gradient methods

* Tel.: +886 7 6577711x3228; fax: +886 7 6578853.
E-mail address: fliu@isu.edu.tw

0017-9310/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijheatmasstransfer.2008.01.002

which go from one initial guess solution in the search domain to another at every instant, genetic algorithms on the other hand search many possible solutions simultaneously and thus have the potential to give unbiased estimation. This provides more chance to find the global optimum in the search space.

In recent years, genetic algorithms have become a popular optimization tool for many areas of research. However, little work has been done in the inverse heat transfer problems. An inverse analysis based on an improved genetic algorithm was presented by Liu et al. [16]. The algorithm has been applied to heat transfer coefficients identification in electronic system cooling simulation through temperature distribution. Since the slow convergence performance of the micro-genetic algorithm, a search domain narrowing operation has been proposed in [16] to improve the performance of convergence so that the whole computational procedure can be carried out within an acceptable timeframe. The numerical result shows that a 33% computational efficiency has been improved. Moreover, Raudensky et al. [17] studied the one-dimensional inverse heat conduction problem for estimation of unknown material properties. Two artificial intelligence mechanisms, neural network and genetic algorithm, were applied in doing the inverse task. Both approaches can lead to a solution without stability problem. The genetic algorithm approach gives relatively precise results but is quite computer-time consuming. An inverse heat transfer problem for identifying heat transfer coefficient in microelectronic package system was presented by Liu et al. [18]. A reduced basis approach has been formulated and coded for the problem to significantly reduce the time for each forward analysis. In the inverse procedure, an intergeneration-projection genetic algorithm was introduced to speed up the process of finding the desired global optimum that leads to the identification of heat transfer coefficients.

In this work, an inverse heat transfer problem to estimate an unknown transient heat source using genetic algorithm is studied. The problem is analyzed and recast as an optimum problem. A real-valued genetic algorithm is applied to solve the discrete-time optimal control problem. A modified genetic algorithm is then developed to improve the performance of the computation and the accuracy of solutions. Numerical results obtained for the real-valued genetic algorithm and the modified genetic algorithm are compared and discussed. The remainder of this paper is organized as follows. In Section 2 the heat transfer problem is described and formulated. It shows that the inverse problem can be converted into an optimization problem. Section 3 provides an overview of the genetic algorithm. The real-valued genetic algorithm for solving the inverse heat transfer problem is described in Section 4. A modified genetic algorithm is proposed in Section 5. Some numerical results of the implementation of these algorithms for solving the inverse problem are discussed in Section 6. Section 7 concludes this paper by making some remarks.

2. The inverse heat transfer problem

Consider a plate heated by a plane surface heat source of strength $G(\tau)$ located at a specified position $x = x^*$. Both boundaries of the plate are insulated. In this one-dimensional transient heat conduction, the function $G(\tau)$ is the heat source distribution which will be used to control the temperature throughout the media. The mathematical formulation of the problem is defined in the dimensionless form as follows:

$$\frac{\partial^2 T(x,\tau)}{\partial x^2} + G(\tau)\delta(x - x^*) = \frac{\partial T(x,\tau)}{\partial \tau}, \quad 0 < x < 1, \quad \tau > 0, \tag{1a}$$

and subjected to the following boundary and initial condition.

$$\frac{\partial T(0,\tau)}{\partial x} = 0, \quad \text{at } x = 0 \quad \text{for } \tau > 0 \tag{1b}$$

$$\frac{\partial T(1,\tau)}{\partial x} = 0, \quad \text{at } x = 1 \quad \text{for } \tau > 0 \tag{1c}$$

$$T(x,0) = 0, \quad \text{at } 0 \leqslant x \leqslant 1 \quad \text{for } \tau = 0 \tag{1d}$$

where $\delta(.)$ is the Dirac delta function. The direct heat transfer problem is a forward computation for finding the temperature field $T(x, \tau)$, i.e. the solutions of the above partial differential equation, Eq. (1a), with heat source of strength $G(\tau)$, boundary and initial conditions, Eqs. (1b)–(1d), are pre-assigned. Eqs. (1a)–(1d) are discrete and constructed a system of equations either by finite difference method, finite element method, or boundary element method for the numerical analysis. The nature of the problem changes when the temperatures in certain location are known from experiments and the heat source of strength $G(\tau)$ has to be found. In this case, the direct heat transfer problem becomes an inverse heat transfer problem.

The heat transfer problem defined by Eqs. (1a), (1b), (1c) and (1d) becomes an inverse problem when the heat source is unknown and to be estimated. The problem is to find such unknown plan heat source, for which temperature histories computed from the mathematic model are close to measured histories at the sensor location. The problem is solved in the least-square sense and a cost function can be defined as a sum of squared differences between the measured temperatures and those from the computational results using a set of guess or estimated solutions.

$$f(G) = \sum_{j=1}^{m} (T_j - \theta_j)^2, \tag{2}$$

where T_j, $j = 1, 2, \ldots, m$, is the temperature calculated from direct problem by using an estimated function, G, and θ_j, $j = 1 \;\; 2, \ldots, m$, is the measured temperature at the same location and time step. The inverse problem can be recast as the following discrete-time optimization problem, where the cost function is defined as Eq. (2), and the desired system parameters, $G(\tau)$, are determined by minimizing the cost function:

$$\min \quad f(G) = \sum_{j=1}^{m} (T_j - \theta_j)^2 \tag{3}$$

subject to T_j is the solution of Eqs. (1a)–(1d)

There are many numerical methods and algorithms available for solving problem (3). In this work, a genetic algorithm based approach is considered to solve the discrete-time optimal control problem.

3. Genetic algorithms

Genetic algorithms, primarily developed by Holland [19], have been successfully applied to various optimization problems. It is essentially a searching method based on the Darwinian principles of biological evolution. Genetic algorithm is a stochastic optimization algorithm which employs a population of chromosomes, each of them represents a possible solution. By applying genetic operators, each successive incremental improvement in a chromosome becomes the basis for the next generation. The process continues until the desired number of generations has been completed or the pre-defined fitness value has been reached.

The genetic algorithms differ from other methods of search and optimization in a number of ways. (a) Genetic algorithms search from a population of possible solutions instead of a single one. (b) The fitness or cost function used to resolve the redundancy has no requirement for continuity in the derivatives, so virtually "any" fitness function can be selected for optimizing. (c) Genetic algorithms use random operators throughout the process including reproduction, crossover, and mutation. (d) Genetic algorithms are blind since no specified information about the intended problem is needed to obtain the final solution.

Basically, a genetic algorithm is a randomized search technique to simulate the nature evolution. It operates on a finite population of chromosomes. The chromosome is formed from genes. The fundamental characteristics of the chromosome are the values and the positions of the genes. Each chromosome has its own fitness measure based on the value and position. The new offspring of chromosomes is provided through genetic operations, i.e. selection, crossover and mutation, which provide a powerful global search mechanism.

Typically binary coding is used in classic genetic algorithm, where each solution is encoded as a chromosome of binary digits. Each member of the population represents an encoded solution in the classic genetic algorithm. For many problems, this kind of coding is not nature. The genetic algorithm used in this work is not a classic genetic algorithm. Instead, the application of genetic algorithm to this discrete-time optimal control problem is called a real-valued genetic algorithm. The continuous function is discrete for numerical computation and simulated by a chromosome. The value of each gene is a real number and indicates the heat generation at each time step.

Generally, a genetic algorithm consists of the following steps that are repeated until the optimum solution is found.

(1) *Initialization* means the creation of the initial population. Many adaptive search methods work from point to point, using local information to decide which point to explore next. By contrast, the genetic algorithms use a set of population in which consist many points to start the search simultaneously. The initial population is generated randomly and the population size is kept constant throughout the process.
(2) *Evaluation* of the fitness functions of chromosomes in the population. The fitness function is in most cases the objective function that should be maximized (or minimized) in the optimization problems. The fitness function is defined as the sum squared differences between the randomly generated guess values and the exact solutions.
(3) *Selection* is the survival of the fittest within the genetic algorithm and it is based on the fitness functions of chromosomes to produce a new pool of population for the subsequent genetic operations. There are many ways to achieve effective selection. One of them is proportionate selection with ranking. The fitness function is normalized with the average value, so the chromosomes with above average fitness will be kept for the next steps. That is at each generation relatively good chromosomes are reproduced, the relatively bad chromosomes die out. This step directs the search towards the best.
(4) *Crossover* is a mating operator to allow production of new offspring through combination of parts of chromosomes with the purpose of constructing a better solution. The simplest way is called the one-point crossover. It is done by choosing randomly a pair of the chromosomes and swapping parts of these chromosomes to form a new pair of chromosomes. A random cross-site between 1 and $i-1$ is chosen along the parent chromosomes, where i is the length of the chromosome. Then, the chromosomes are cut at the selected cross-site, and their end parts are exchanged. The search is emphasized towards the best and new solutions explored by using information about things that have worked well in the previous step. If one-point crossover is performed, it is possible to obtain a chromosome that cannot be matched. In this case, a two-point or multi-point crossover can be introduced to overcome this problem. As a result, the performance of generating offspring is greatly improved.
(5) *Mutation* is the occasional (with small probability) random alteration of the genes of chromosome. The mutation operator arbitrarily alters one or more genes of a selected chromosome, which increase the variability of the population and against the loss of important genetic genes at a particular position. Each bit position of each chromosome in the new popula-

tion undergoes a random change with equal probability. After mutation, the offspring are including into the pool of population to repeat the process.

The computational procedure of a genetic algorithm is as follows:

Step 1. Generate at random an initial population of chromosomes.
Step 2. Evaluate the fitness of each chromosome in the population.
Step 3. Select the best chromosomes, based on the fitness function, for reproduction.
Step 4. Choose at random pairs of chromosomes for mating. Exchange bit genes with crossover to create new pairs of chromosomes.
Step 5. Process the new chromosomes by the mutation operator. Put the resulting chromosomes into the population.
Step 6. Repeat Step 2 to Step 5, until the fitness function is convergent or less than a predefined value.

The algorithm only requires fitness function for each of the chromosome, without the need for assumption such as differentiability, which makes it very useful for a discontinuous function.

4. A real-valued genetic algorithm for solving the inverse heat transfer problem

In this section, a real-valued genetic algorithm is considered for solving the inverse heat transfer problem of estimating transient heat source of strength. The continuous function of heat source, G, is discretized for numerical computation and simulated by a chromosome. The value of each gene is a real number and indicates the heat source function at each time step. The problem is to find such unknown heat source, for which temperature histories computed from the mathematic model at the sensor location are close to measured histories. In our problem measured temperatures, θ, are simulated by solving the direct problem with exact heat source. Then the inverse problem becomes an optimization problem as defined in Eq. (3).

Consider the problem of finding a minimum of the function $f(G)$. For this optimization problem, at each generation k of the genetic algorithm we will maintain a population of chromosomes

$$D^k = \{G_1^k, G_2^k, G_3^k \ldots G_i^k \ldots G_n^k\} \quad i = 1, 2, \ldots, n, \tag{4}$$

where G_i^k is a chromosome which represents a feasible solution, k is a generation number and n is an arbitrarily chosen length of the population. The chromosome G_i^k is defined as follows:

$$G_i^k = (g_{i1}, g_{i2}, g_{i3}, \ldots, g_{ij}, \ldots, g_{im}), \quad j = 1, 2, \ldots, m, \tag{5}$$

where g_{ij} is an individual gene and m is the number of genes which represents the number of time steps for computation. Substituting G_i^k into the constraints Eqs. (1a)–(1c) and (1d), the computed temperatures, T_j, can be calculated. Each feasible solution G_i^k is evaluated by computing the fitness function $f(G_i^k)$, which provides us the measurement of fitness of the solution,

$$f(G_i^k) = \sum_{j=1}^{m} (T_j - \theta_j)^2. \tag{6}$$

Some of the feasible solutions are selected based on the ranking of the fitness of solutions. Then the selected solutions are recombined using crossover and mutation to form a new set population at the $(k+1)$th generation (the population size n remains fixed for the computation).

Crossover is a process of exchanging members between two solutions. Two solutions are randomly selected as parent structures from the population. Then arbitrary positions on both solutions are chosen for crossing location, where the exchanges of members take place. For example, if parent structures are represented by six members, say,

$$G_1 = (g_{11}, g_{12}, g_{13}, g_{14}, g_{15}, g_{16}) \quad \text{and}$$
$$G_2 = (g_{21}, g_{22}, g_{23}, g_{24}, g_{25}, g_{26}),$$

then the crossing the solutions between the second and the sixth members would produce the offspring

$$(g_{11}, g_{12}, g_{23}, g_{24}, g_{25}, g_{16}) \quad \text{and}$$
$$(g_{21}, g_{22}, g_{13}, g_{14}, g_{15}, g_{26}).$$

A mutation operator arbitrary alters one or more members of a selected solution. This increases the variability of the population. Each member position of each solution in the new population undergoes a random change with the probability equal to the mutation rate.

After the step of mutation, the new set of population is formed for the $(k+1)$th generation. The process is repeated until a pre-defined number of generations is reached or the solutions converge.

5. A modified genetic algorithm

For the real-valued genetic algorithm, one major difficulty lies in solving the inverse heat transfer problem to a satisfactory degree of precision. Although the real-valued genetic algorithm has been proved highly successful as an optimization, the slowly convergence performance is a bottleneck as an iterative solver. Raudensky et al. [20] studied the application of genetic algorithm for searching the solution of an inverse heat conduction problem. In their work the heat transfer coefficient h at the convective boundary is to be identified. The numerical solutions perform well by using the genetic algorithm. The accuracy of the results improved significantly before certain number of genetic steps. However, the improvement is discouraging for further computations. To improve the search performance in their work, a regularization term is added to the fitness function to penalize the oscillations of h. Different techniques for improving the accuracy and performance of the genetic algorithm have been reported. Among them, local search techniques are often recommended to combine with the genetic algorithms. In this work an additional cost function, J, is added to the real-valued genetic algorithm to improve the computational efficiency, and it can be defined as follows:

$$J^k(g_{ij}) = |T_j - \theta_j|, \quad i = 1, 2, \ldots, n, \; j = 1, 2, \ldots, m. \tag{7}$$

The cost function J introduced here is to select the best individual genes to form new solutions for the next steps. After the step of selection, the best solutions are selected according to the fitness function $f(G_i)$. Instead of going to the step of crossover and mutation in the real-valued genetic algorithm, the cost function J, is added for evaluating the individual genes before going to the step of crossover in the modified genetic algorithm. Among the best solutions after selection, the best genes are selected in each individual gene positions according to the cost function J to form other new solutions. That is, the individual genes g_{i1}, $i = 1, 2, \ldots, n$, are ranking and selected according to the function J for the first gene position of all solutions. For example, if possible solutions are represented by six members, say,

$$G_1 = (g_{11}, g_{12}, g_{13}, g_{14}, g_{15}, g_{16}),$$
$$G_2 = (g_{21}, g_{22}, g_{23}, g_{24}, g_{25}, g_{26}), \quad \text{and}$$
$$G_3 = (g_{31}, g_{32}, g_{33}, g_{34}, g_{35}, g_{36}).$$

The computations of the function J are

$$J(g_{11}) = 3, \quad J(g_{21}) = 5, \quad \text{and} \quad J(g_{31}) = 1.$$

Then the rearrangement of genes for the first gene position according to the function J would produce the following solutions:

$$(g_{31}, g_{12}, g_{13}, g_{14}, g_{15}, g_{16}),$$
$$(g_{11}, g_{22}, g_{23}, g_{24}, g_{25}, g_{26}), \quad \text{and}$$
$$(g_{21}, g_{32}, g_{33}, g_{34}, g_{35}, g_{36}).$$

Then the same ranking and selected system are implemented on the second gene position of all solutions. The procedure is completed until the best genes are selected at all positions of the solutions. New solutions are composite by combining these highly ranking individual genes at different positions. The modified genetic algorithm does not only select the best solutions, but also bring the best genes into the next steps and the next generation. A flowchart of the modified genetic algorithm is shown in Fig. 1.

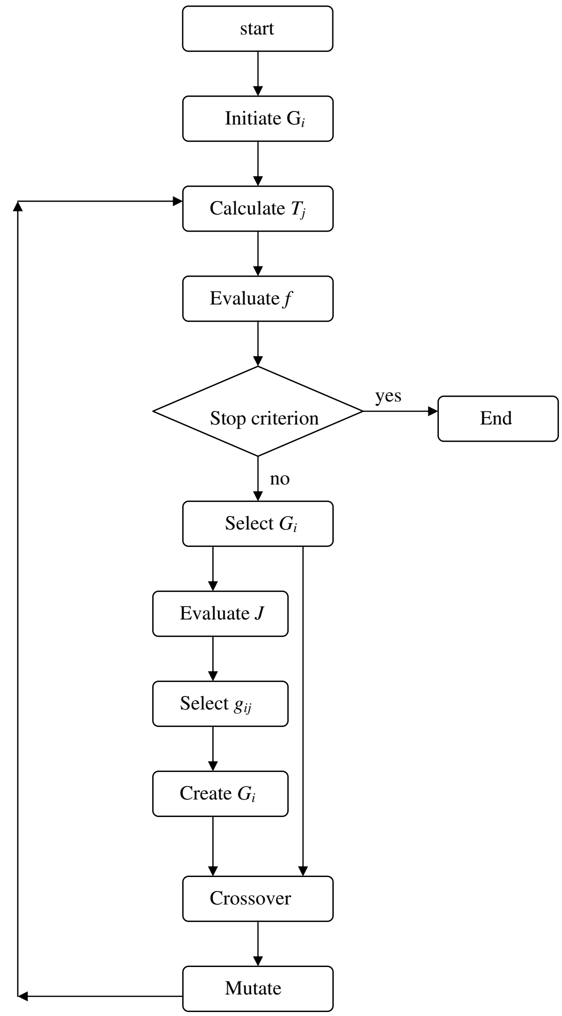

Fig. 1. Flowchart of the modified genetic algorithm.

6. Numerical examples and discussion

In this section, the genetic algorithms for solving the inverse heat transfer problem of estimating transient heat source of strength is illustrated by some numerical examples. Consider the continuous function of the heat source as follows:

$$G(\tau) = \begin{cases} 1.8\tau, & 0 \leqslant \tau < 0.5, \\ 1.9 - 2\tau, & 0.5 \leqslant \tau < 0.8, \\ 0.3, & 0.8 \leqslant \tau \leqslant 1, \end{cases} \tag{8}$$

where G is a triangular time variation and discrete for numerical computation.

Genetic algorithms are based on the premise that most chromosomes contain at least some useful information and that, by sharing it, an improved solution can be created. For this reason the initial population should be reasonably as large and diverse as possible. In our problem a population of 24 chromosomes of 60 genes is used as the initial guess solutions for numerical experiments. These initial values of chromosomes are created by using the random number generator.

At first the test case is carried out for the real-valued genetic algorithm to check the stability of the approach. Fig. 2 shows the fitness functions of three different runs for 500 generations. The initial population of 24 feasible solutions of 60 genes is generated by using the random number generator. Therefore, different runs generate differ-

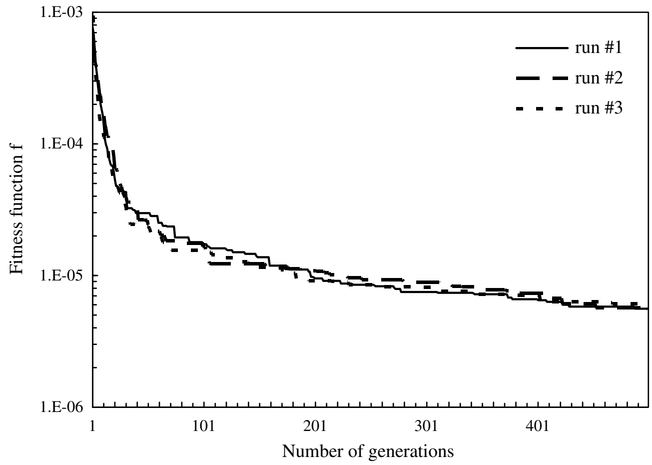

Fig. 2. Convergence study for different initial guess solutions.

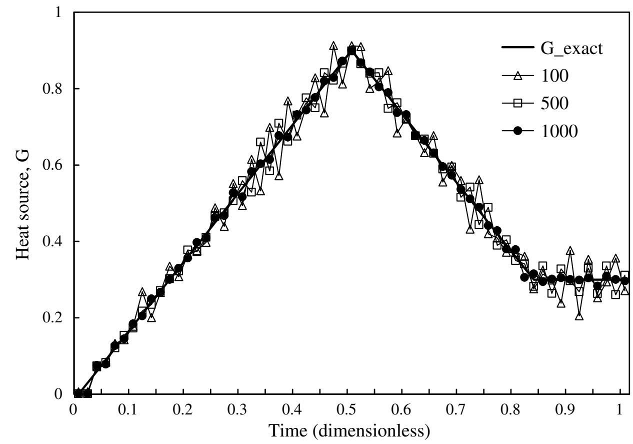

Fig. 4. The estimated results for different number of generations using modified genetic algorithm.

ent initial guess solution for starting the search procedure. The trends of the fitness functions perform similarly. Moreover, the fitness functions are close to one another in the generations computed in all three runs. In this case the solution of the real-valued genetic algorithm is not highly dependent on the initial guess solutions.

To examine the accuracy of solutions, the estimated solutions obtained by the genetic algorithm are compared to the exact solutions as shown in Fig. 3. The heat source function estimates by the three different numbers of generations, namely 1000, 10,000 and 100,000, are plotted with the exact heat source. The functional form is well estimated, but not good enough, by all three cases even though for the case of 100,000 generations. The more number of generations, the better accuracy the solution is.

To solve the current inverse problem to a satisfactory degree of precision, an additional cost function is added to the real-valued algorithm to enhance its performance. Results of inverse analysis using the proposed modified genetic algorithm are shown in Fig. 4 for three different numbers of generations, 100, 500, and 1000. The performance of the solutions obtained by the modified genetic algorithm is much improved than that of the real-valued genetic algorithm for 1000 generations as shown in Fig. 5. Eventually, the numerical results for the case of 100 generations using the modified genetic algorithm are much better than that of 100,000 generations using the real-valued genetic algorithm. The convergence results of the fitness functions for both the real-valued and the modified algorithms for 1000 generations are shown in Fig. 6. The modified genetic algorithm is efficiently performed since the number of generations is much less to reach the same fitness function for the real-valued genetic algorithm.

In [20], a regularization term is added to improve the accuracy of solutions for the genetic algorithm. The regularization parameter must be selected based on some knowledge about the expected variation of the unknown function. Compared to the method of Raudensky et al. [20] the present work used no prior information on the function to be predicted.

For inverse heat conduction problems the temperature measurements may contain some noise. To illustrate the

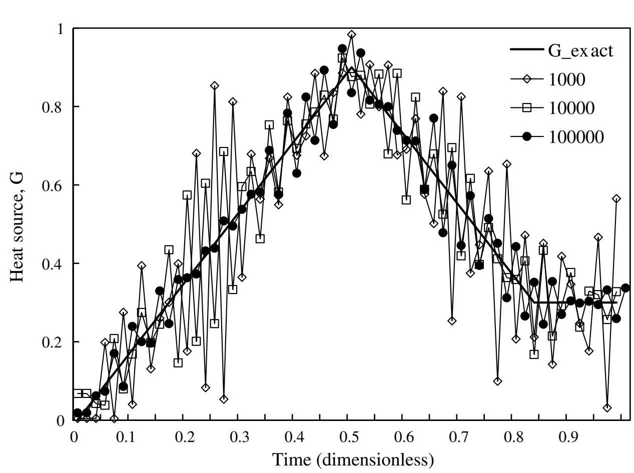

Fig. 3. The estimated results for different number of generations using the real-valued genetic algorithm.

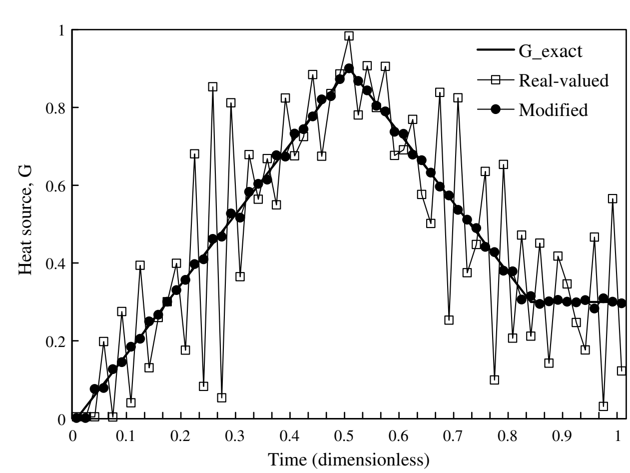

Fig. 5. Numerical results of the real-valued and the modified genetic algorithms in the case of 1000 generations.

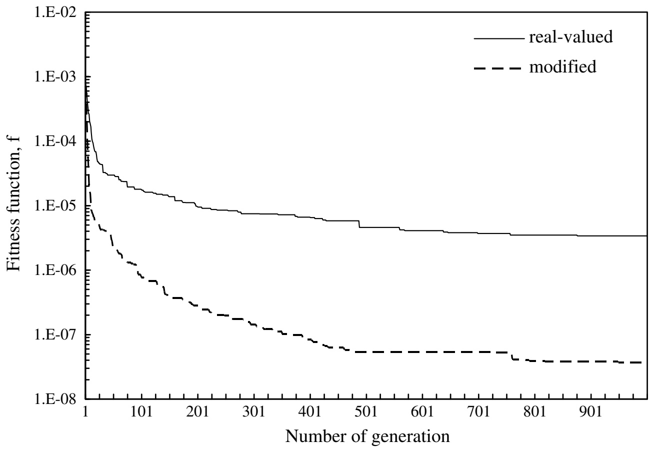

Fig. 6. The convergence study for the real-valued and modified genetic algorithm.

accuracy of the modified genetic algorithm in predicting the unknown heat source, the measured temperatures involving random measurement errors are also investigated. Fig. 7 presents the results of estimation for 1000 generations with 0%, 3% and 5% errors using the proposed modified genetic algorithm. The random errors are normally distributed with zero mean in the measured temperatures (θ_j). The results reveal that the accuracy of estimation decreasing while the measurement error is increasing.

All numerical experiments were implemented on a personal computer with AMD Athlon 64 2.01 GHz CPU and 1.0 GB RAM. The CPU time for carrying out the real-valued genetic algorithm is about 132, 1294 and 12,689 s for 1000, 10,000, and 100,000 generations, respectively. The computer code of the modified genetic algorithm is larger than that of real-valued genetic algorithm because of the additional cost function for evaluation. The CPU time for carrying out the modified genetic algorithm is about 138 s for 1000 generations. For calculating the same number of generations, the modified genetic algorithm spends more CPU time than the real-valued genetic algorithm, while the solution is highly improved. In our implementation, it takes less than 7 s and 40 generations for the modified genetic algorithm compared to 132 s and 1000 generations for the real-valued genetic algorithm to achieve the same solution accuracy (fitness).

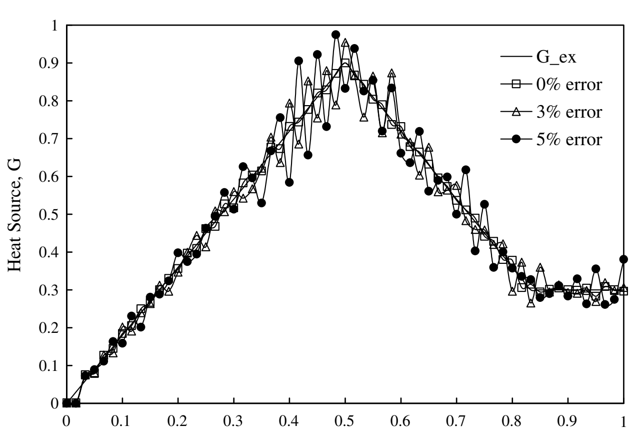

Fig. 7. The estimated results for measurements with noise using modified genetic algorithm.

7. Conclusion

An inverse analysis for the unknown heat source function by using the genetic algorithms is studied. A modified genetic algorithm is developed for solving the inverse heat transfer problem to a satisfactory degree of precision. Some empirical results are provided to illustrate the proposed algorithms. Compared to the real-valued genetic algorithm, the modified genetic algorithm essentially reduces the computational time for convergence with highly qualitative correct results.

References

[1] J.V. Beck, B. Blackwell, C.R. St Clair Jr., Inverse Heat Conduction, Wiley, New York, 1985.

[2] A.N. Tikhonov, V.Y. Arsenin, Solution of Ill-posed Problem, Winston, Washington, DC, 1977.

[3] B. Blackwell, Efficient technique for the numerical solution of one-dimensional inverse problem of heat conduction, Numer. Heat Transfer (1981) 229–238.

[4] C.-H. Huang, C.-W. Chen, A boundary element-based inverse problem in estimating transient boundary conditions with conjugate gradient method, Int. J. Numer. Meth. Eng. 42 (1998) 943–965.

[5] C.-H. Huang, C.-W. Chen, A three-dimensional inverse forced convection problem in estimating surface heat flux by conjugate gradient method, Int. J. Heat Mass Transfer 43 (2000) 3171–3181.

[6] C.-H. Huang, J.-H. Hsiao, A non-linear fin design problem in determining the optimum shape of spine and longitudinal fins, Commun. Numer. Meth. Eng. 19 (2003) 111–124.

[7] C.-H. Huang, C.-Y. Huang, An inverse biotechnology problem in estimating the optical diffusion and absorption coefficients of tissue, Int. J. Heat Mass Transfer 47 (2003) 447–457.

[8] J.T. Alander, Population size, building blocks, fitness landscape and genetic algorithm search efficiency in combinatorial optimization: an empirical study, in: L.D. Chambers (Ed.), Practical Handbook of Genetic Algorithms: Complex Coding Systems, vol. III, CRC Press, Boca Raton, 1999.

[9] N. Ansari, E. Hon, Computational Intelligence for Optimization, Kluwer, Norwell, 1997.

[10] C. Bierwith, D.C. Malfeld, Production scheduling and rescheduling with genetic algorithms, Evol. Comput. 7 (1999) 1–17.

[11] E. Falkenauer, Genetic Algorithms and Grouping Problem, Wiley, Chichester, 1998.

[12] J.R. Koza, Hierarchical automatic function definition in genetic programming, in: L.D. Whitley (Ed.), Foundations of Genetic Algorithms 2, Morgan Kauffman, San Mateo, 1993.

[13] S. Rasmussen, C.L. Barrettt, Elements of a Theory of Simulation, in: Proceedings of the European Conference on Artificial Life, Springer, Berlin, 1995.

[14] S.D. Scott, S. Seth, A. Samal, A synthesizable VHDL coding of a genetic algorithm, in: L.D. Chambers (Ed.), Practical Handbook of Genetic Algorithms: Complex Coding Systems, vol. III, CRC Press, Boca Raton, 1999.

[15] M. Vainio, T. Schonberg, A. Halme, P. Jakubik, Optimization in performance of a robot society in structured environments through

genetic algorithms, in: Proceedings of the European Conference on Artificial Life, Springer, Berlin, 1995.
[16] G.R. Liu, J.-J. Zhou, J.G. Wang, Coefficient identification in electronic system cooling simulation through genetic algorithm, Comp. Struct. 80 (2002) 23–30.
[17] M. Raudensky, J. Horsky, J. Krejsa, L. Slama, Usage of artificial intelligence methods in inverse problems for estimation of material parameters, Int. J. Num. Meth. Heat Fluid Flow 6 (8) (1996) 19–29.
[18] G.R. Liu, J.-H. Lee, A.T. Patera, Z.L. Yang, K.Y. Lam, Inverse identification of thermal parameters using reduced-basis method, Comput. Meth. Appl. Mech. Eng. 194 (2005) 3090–3107.
[19] J.H. Holland, Adaptation in Natural and Artificial System, University of Michigan Press, Ann Arbor, 1975.
[20] M. Raudensky, K.A. Woodbury, J. Kral, T. Brezina, Genetic algorithm in solution of inverse heat conduction problems, Num. Heat Transfer, Part B 28 (1995) 293–306.