International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:8, No:9, 2014

Inverse Heat Conduction Analysis of Cooling on Run Out Tables

M. S. Gadala, Khaled Ahmed, Elasadig Mahdi

Abstract**—**In this paper, we introduced a gradient-based inverse solver to obtain the missing boundary conditions based on the readings of internal thermocouples. The results show that the method is very sensitive to measurement errors, and becomes unstable when small time steps are used. The artificial neural networks are shown to be capable of capturing the whole thermal history on the run-out table, but are not very effective in restoring the detailed behavior of the boundary conditions. Also, they behave poorly in nonlinear cases and where the boundary condition profile is different.

GA and PSO are more effective in finding a detailed representation of the time-varying boundary conditions, as well as in nonlinear cases. However, their convergence takes longer. A variation of the basic PSO, called CRPSO, showed the best performance among the three versions. Also, PSO proved to be effective in handling noisy data, especially when its performance parameters were tuned. An increase in the self-confidence parameter was also found to be effective, as it increased the global search capabilities of the algorithm. RPSO was the most effective variation in dealing with noise, closely followed by CRPSO. The latter variation is recommended for inverse heat conduction problems, as it combines the efficiency and effectiveness required by these problems.

Keywords**—**Inverse Analysis, Function Specification, Neural Net Works, Particle Swarm, Run Out Table.

I. Introduction

IN an inverse heat conduction problem (IHCP), the boundary conditions, initial conditions, or thermo-physical properties of material are not fully specified. These properties are determined from measured internal temperature profiles. The challenge is that the effect of changes in boundary conditions are normally damped or lagged, that would be a typically ill posed and sensitive to measurement errors. Also, in the uniqueness and stability of the solution are not generally guaranteed [1], [2].

Inverse heat conduction problems may be reformulated as an optimization problem with an objective function that would be normally highly nonlinear and would involve the squared difference between measured (vector Y) and estimated unknown variables (vector T), i.e.:

$$U = [Y - T]^T [Y - T] \tag{1}$$

Normally a regularization term should be added to Eqn. 1 in order to eliminate oscillations and enhance solution stability. The above equation is only valid, if the measured temperatures and the associated errors have certain statistical characteristics stated in [3].

While classical methods, such as the least square regularization method [4], the sequential function specification method [1]-[4], the space marching method [5], conjugate gradient method [6], steepest descent method [7], and the model reduction algorithm [8] are vastly studied in the literature, and applied to the problems in thermal engineering [9]-[11], there are still some unsolved problems. The main problems are solution stability, damping peak heat fluxes, measurement errors sensitivity, time step limit and temporal resolutions. More recent optimization techniques may be used in the solution of the IHCPs to aid in solving such problems. Some of these techniques are briefly outlined in the following section.

The Genetic Algorithm technique has been widely adopted to solve inverse problems [12], [13]. Genetic algorithms (GAs) belong to the family of computational techniques originally inspired by the living nature. They perform random search optimization algorithms to find the global optimum to a given problem. The main advantage of GAs may not necessarily be their computational efficiency, but their robustness and ability to reach a global optimum. Luckily, they are inherently parallel algorithms, and can be easily implemented on parallel structures.

Artificial neural networks can be successfully applied in the solution of inverse heat conduction problems [14]-[15]. They are capable of dealing with significant non-linearities and are known to be effective in damping the measurement errors.

The method of Self-learning finite elements (FE) combines neural network with nonlinear FE in an algorithm, which uses basic conductivity measurements to produce a constitutive model of the material under study [12]. It is also shown to exhibit a great stability when dealing with noisy data.

The method of Maximum entropy seeks the solution that maximizes the entropy functional under given temperature measurements. It converts the inverse problem to a non-linear constrained optimization problem. The constraint is the statistical consistency between the measured and estimated temperatures. It can guarantee the uniqueness of the solution [16].

In the technique of Proper Orthogonal Decomposition, the idea is to expand the direct problem solution into a sequence

M. S. Gadala is with the Mechanical Engineering Department, University of British Columbia, Vancouver, BC, V6T-1Z4, Canada (e-mail: gadala@mail.ubc.ca).

Khaled Ahmed, corresponding author, was with the Mechanical Engineering Department, Assiut University, Assiut, Egypt. He is now with the Department of Mechanical & Industrial Engineering, Qatar University, Doha 2713, Qatar (phone: +974-4403-6659; fax: +974-4403-4201; e-mail: kh.ahmed@qu.edu.qa.)

El Sadig Mahdi is with the Mechanical & Industrial Engineering Department, Qatar University, Doha 2713, (e-mail: elsadigms@qu.edu.qa).

of orthonormal basis vectors, describing the most essential features of spatial and temporal variation of the temperature field. This can result in the filtration of the noise in the field under study [17].

The Particle Swarm Optimization (PSO) is a population based stochastic optimization technique, inspired by social behavior of bird flocking or fish schooling. Unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Compared to GA, the advantages of PSO are ease of implementation and that there are few parameters to adjust. Some researchers showed that it requires less computational expense when compared to GA for the same level of accuracy in finding the global minimum [18].

II. Assessed Techniques

A. Function Specification Methods

As mentioned above it is very common to include more variables in the objective function to stabilize the solution. A common choice in inverse heat transfer problems is to use a scalar quantity based on the boundary heat fluxes, with a weighting parameter α, which is normally called the regularization parameter. In [11], it is shown that using the heat flux values (zeroth-order regularization) is the most suitable choice. The objective function then will be

$$F(q) = \sum_{i=1}^{N}\left(\mathrm{T}_m^i - \mathrm{T}_c^i\right)^T\left(\mathrm{T}_m^i - \mathrm{T}_c^i\right) + \alpha\sum_{i=1}^{N}\mathrm{q}^{iT}\mathrm{q}^i \qquad (2)$$

where $\mathbf{T}_m^i$ and $\mathbf{T}_c^i$ are the vectors of expected (measured) and calculated temperatures at the i^{th} time step, respectively, each having J spatial components; α is the regularization coefficient; and q^i is the boundary heat flux. Because inverse problems are generally ill posed, the solution may not be unique and would be in general sensitive to measurement errors. To decrease such sensitivity and improve the simulation, a number of future time steps (n_{FTS}) are utilized in the analysis of each time step. Details of this algorithm may be found in [11].

B. Genetic Algorithm

Genetic algorithm is widely used in stochastic optimization method and in heat transfer applications, including inverse heat transfer analysis [19]. GA starts its search from a randomly generated population. This population evolves over successive generations (iterations) by applying three major operations. The first operation is “Selection”, which mimics the principle of “Survival of the Fittest” in nature. It finds the members of the population with the best performance, and assigns them to generate the new members for future generations. The second operator is called “Reproduction” or “Crossover”, which imitates mating and reproduction in biological populations. It propagates the good features of the parent generation into the offspring population. The last operator is “Mutation”, which allows for global search of the best features, by applying random changes in random members of the generation. This operation is crucial in avoiding the local minima traps [20]. Among the many variations of GAs, in this study, we use a real encoded GA with roulette selection, intermediate crossover, and uniform high-rate mutation.

C. Particle Swarm Optimization

Particle swarm optimization (PSO) is a high-performance stochastical search algorithm that can also be used to solve inverse problems. The method is based on the social behavior of species in nature, e.g., a swarm of birds or a school of fish [22]. In the original PSO algorithm [21] the velocity of a particle i in a given iteration m is updated with a velocity vector according to the following relation:

$$v_i^{m+1} = v_i^m + c_1r_1(p_i^m - x_i^m) + c_2r_2(g^m - x_i^m) \qquad (3)$$

where x_i^m and v_i^m are the position and velocity of particle i at the m-th iteration, respectively; p_i^m and g^m are the best positions found up to now by this particle (local memory) and by the whole swarm (global memory) so far in the iterations, respectively; c_0 is called the inertia coefficient or the self-confidence parameter and is usually between zero and one; c_1 and c_2 are the acceleration coefficients that pull the particles toward the local and global best positions; and r_1 and r_2 are random vectors in the range of (0,1). The ratio between these three parameters controls the effect of the previous velocities and the trade-off between the global and local exploration capabilities.

One popular way of preventing divergence in PSO is a technique called “constriction”, which dynamically scales the velocity update [21]. The first method was used in the previous research by [23]. However, further investigation showed that a better performance is obtained when combining the constriction technique with limiting the maximum velocity. In this work, the velocity updates are done using constriction and can be written as:

$$v_i^{m+1} = K\left(v_i^m + c_1r_1(p_i^m - x_i^m) + c_2r_2(g^m - x_i^m)\right) \qquad (4)$$

where K is the constriction factor, and is calculated as [21]:

$$K = \frac{2}{2-\emptyset-\sqrt{\emptyset^2-4\emptyset}} \qquad (5)$$

where $\varphi = c_1 + c_2$. Here, following the recommendations in [21], the initial values for c_1 and c_2 are set to 2.8 and 1.3, respectively. These values will be modified in subsequent iterations, as discussed below.

Several variants of PSO have been developed to improve the performance of the basic algorithm [22]-[24]. One of these variants is called the Repulsive Particle Swarm Optimization (RPSO), and is based on the idea that repulsion between the particles can be effective in improving the global search capabilities and finding the global minimum [25]. The velocity update equation for RPSO is;

$$v_i^{m+1} = c_o v_i^m + c_1r_1(p_i^m - x_i^m) + c_2r_2\left(p_j^m - x_i^m\right) + c_3r_3v_r \quad (6)$$

where p_i^m is the best position of a randomly chosen other particle among the swarm, c_3 is an acceleration coefficient, r_3 is a random vector in the range of (0,1), and v_r is a random velocity component. The modified technique does not benefit from the global best position found. A modification to RPSO that also uses the tendency towards the best global point is called the "Complete Repulsive Particle Swarm Optimization" or CRPSO [23]. The velocity update equation for CPRSO will be:

$$v_i^{m+1} = c_o v_i^m + c_1 r_1 (p_i^m - x_i^m) + c_2 r_2 (g^m - x_i^m) + c_3 r_3 (p_j^m - x_i^m) + c_4 r_4 v_r \qquad (7)$$

In CRPSO, by having both an attraction toward the particle's best performance, and a repulsion from the best performance of a random particle, we are trying to create a balance between the local and global search operations. In [23], it was concluded then that the CRPSO variation is the suitable choice for IHCPs and several other modifications were presented.

D. Artificial Neural Networks

Artificial Neural Networks (ANN) are motivated by the efficiency of brain in performing computations. These networks are made of a large number of processing units (neurons) that are interconnected through weighted connections, similar to synapses in brain. In order for the network to perform the expected tasks, it should first go through a "learning" process. There are two main categories of learning: supervised, or unsupervised. The supervised learning is useful in function fitting and prediction, while unsupervised learning is more applicable to pattern recognition and data clustering. Since the learning process in our application is a supervised one, we focus on this type of learning process. While there are several major classes of neural networks, in this paper we will test only two of types; Feedforward Multilayer Perceptrons (FMLP) and Radial Basis Function Networks (RBFN) as shown schematically in Fig. 1.

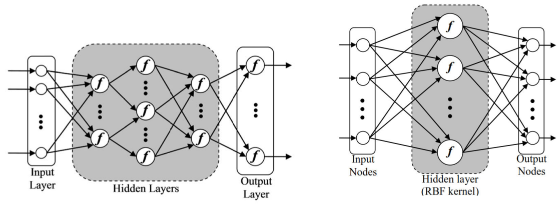


Fig. 1 (a) Feed-forward and (b) RBF network topology

In order to use the artificial neural networks in the inverse heat conduction problem, we first started with direct heat conduction FE code, and applied several sets of heat fluxes on the boundary. The resulting temperatures in locations inside the domain, which correspond to the thermocouple locations in the experiments, were obtained and used to train the neural network to reproduce the heat fluxes. In addition, the problem was reformulated by using the change in temperature in each time step as the input, which showed much better performance to the neural network.

III. Test Cases

A block containing nine thermocouples is modeled for each pass of water jet cooling of a steel strip. The length of the block is 114.3 mm (9 sections of each 12.7 mm). The width and thickness are 12.7 mm and 6.65 mm, respectively. To model the thermocouple hole, a cylinder of radius 0.5 mm and height of 5.65 mm is taken out of the block. Isoparametric eight-node brick elements are used to discretize the domain. Fig. 2 (a) shows the whole domain, and Fig. 2 (b) is a close-up view of one of the TC holes.

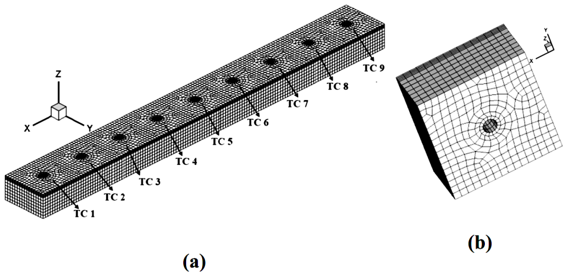


Fig. 2 (a) The whole block consisting of nine thermocouples, (b) A close-up view of the TC hole from bottom

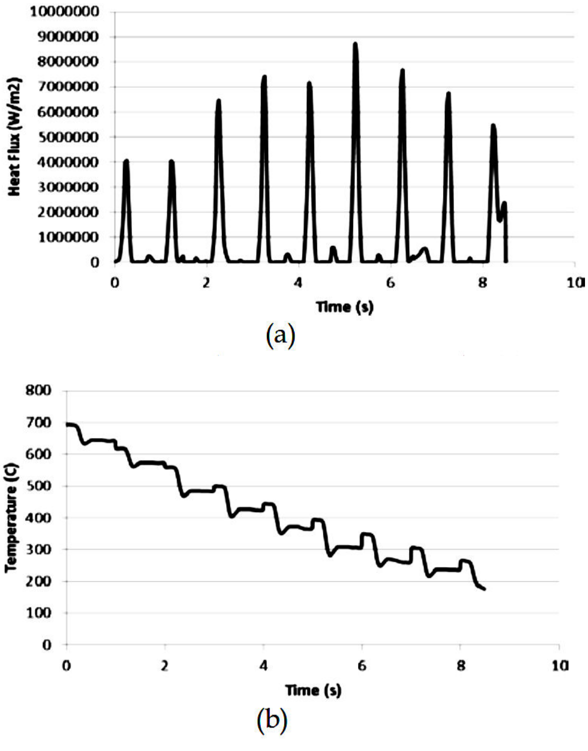


Fig. 3 Cooling on a run-out table; (a) Surface heat fluxes; (b) Internal temperatures

The boundary condition on the top surface is prescribed heat flux, which is chosen to resemble the one in water-cooling of steel strips. Fig. 3 (a) shows the applied heat fluxes on top of one of the thermocouple locations for the whole cooling process, while Fig. 3 (b) shows the history of the temperature drop at the corresponding thermocouple location. Fig. 4 shows a close-up of the applied heat flux and the FE results of the temperature history at five of the nine thermocouples locations, which resembles an actual heat flux values on a run-out table with two rows of staggered circular jets, impinging on the third and seventh locations [23]. The physical properties of the steel strips used in our experiment are: density, *ρ*, is *7850 kg/m^3*, C_p is *475 J kgK*, and the thermal conductivity, *k*, *40 W/m.°C* (later changed to be temperature dependent). Results are obtained at the top of the cylindrical hole, which is the assumed position of a thermocouple. Inverse analysis is conducted to obtain the transient heat flux profile at the top surface of the plate.

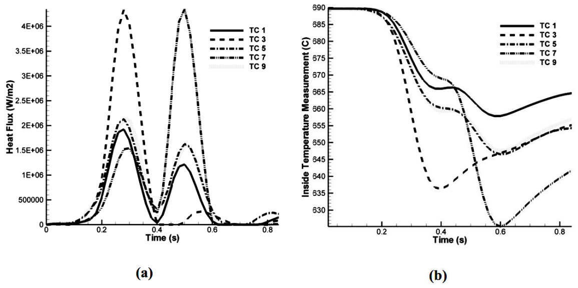


Fig. 4 (a) The applied heat flux on the top surface; (b) The thermocouple readings used for inverse analysis

IV. Results and Discussion

We start by applying the artificial neural networks to the inverse heat conduction problem. Fig. 5 shows the heat flux vs. time result of the application of the radial basis function neural networks for the whole history of the heat fluxes on the runout table. Temperatures start at 700 ºC and go down to 176 ºC. As can be seen from this figure, neural networks are generally capable of dealing with the whole range of the cooling history.

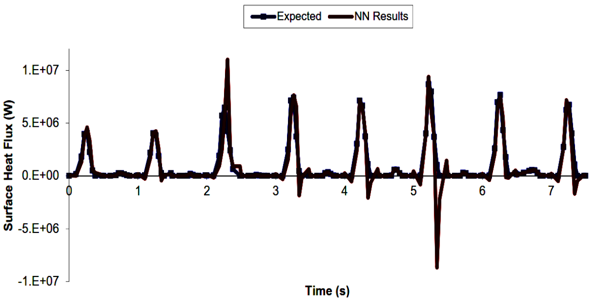


Fig. 5 Time History of Heat Fluxes in a Typical Run-Out Table Application; Expected Results (Squares) vs. the RBF Network Results (Line)

However, going through the individual peaks of heat fluxes, it is apparent that the success or failure of NNs is not that much related to the temperature range, or the magnitude of heat fluxes, but on the actual shape of the heat flux profile. On the other hand, GA and PSO algorithms show reasonably good predictions of the details of the missing boundary conditions. The figures for the results of GA and PSO are not presented here for the sake of brevity, but can be found in [23]. They will be used, however, for comparisons in the next sections.

E. Time Step Size and Efficiency

Unlike direct problems where the stability requirement gives the upper limit of the time step size, in inverse problems the time step is bounded from below. Fig. 6 (a) [23] shows the oscillation in the results obtained by the function specification method and a time step size of 0.01 (s), which corresponds to the onset of instability. PSO, GA, and NNs successfully produce, however, the results for the same time step size as presented in Fig. 6 (b) for PSO. Note that the oscillations here are not due to the instability caused by the time step size, and can be improved by performing more iterations. It is, however, important to mention that the time requirements for these techniques are much higher than those of the classical function specification approaches.

Now we compare the solution time required for GA, the three variations of PSO, and feed forward and radial basis function neural networks. We assume that there is no noise in the solution. Table I compares the solution time for different inverse analysis algorithms. The fastest solution technique is the gradient-based function specification method. The stochastical methods such as GA and PSO variants suffer a high computational cost. RBF neural networks perform much faster than GA and PSO, but they are still slower than the gradient-based methods, such as function specification.

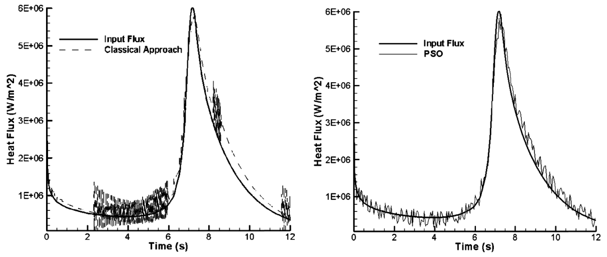
F. Noisy Domain Solution

To investigate the behavior of different algorithms with noisy data, random errors are imposed onto the calculated exact internal temperatures with the following equation:

$$T_m = T_{exact} + \sigma \cdot r \tag{8}$$

where T_m is the virtual internal temperature that is used in the inverse calculations instead of the exact temperature, T_{exact}; r is a normally distributed random variable with zero mean and unit standard deviation; and σ is the standard deviation. Virtual errors of 0.1% and 1% of the temperature magnitude are investigated here. A more detailed comparison between the efficiency of GA and PSO variations can be found in [23].

Fig. 7 shows the results of the RBF network (red pluses) versus the expected results (blue circles) for sample individual heat flux peaks during the cooling history of the plate. The amount of added noise in the figure is ±1%. There are several ways to make an inverse algorithm more stable when dealing with noisy data. For example, [11] have shown that increasing the number of "future time steps" in their sequential function specification algorithm resulted in greater stability. They have also demonstrated that increasing the regularization parameter, α, improves the ability of the algorithm but on the expense of requiring more iterations and the possibility of diverging. Their results are reproduced in Fig. 8.



(a)

(b)

Fig. 6 Heat flux vs. time: (a) classical approach, (b) PSO [23]

TABLE I
COMPARISON OF THE SOLUTION TIME FOR DIFFERENT INVERSE ANALYSIS ALGORITHMS

Method	Solution Time (sec)
Function Specification Method	1406
Genetic Algorithm, GA	8430
Particle Swarm Optimization, PSO	6189
Repulsive Particle Swarm Optimization, RPSO	5907
Complete Repulsive Particle Swarm Optimization, CRPSO	6136
Feed-forward Multilayer Perceptron FMLP	7321
Radial Basis Function Networks, RBFN	2316

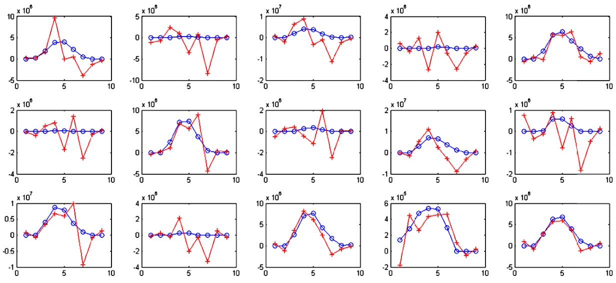


Fig. 7 Heat flux peaks vs. time from a typical run-out table application; Expected results (blue circles) vs. the RBF NN results (red pluses); Artificial noise added: c = ±1%

Another factor that can affect the performance of a PSO inverse approach in dealing with noisy data is the value of the self-confidence parameter, c0, or the ratio between this parameter and the acceleration coefficients. The acceleration coefficients are set to the default value of 1.42. The initial value of the self-confidence parameter, c0, is changed from the default value of 0.7. As can be seen in Fig. 8 (for α = 10^{-10}), increasing the value of the self-confidence parameter results in better handling of the noisy data. As can be seen in Table II, the best effectiveness is normally obtained by RPSO, closely followed by CRPSO.

G. Effect of Non-Linearity

To investigate the effect of temperature dependent thermal properties, the following expression is assumed for the behavior of thermal conductivity:

$$k = 60.571 - 0.03849 \times T \ W/m.°C \qquad (9)$$

As expected, the nonlinearity weakens the performance of both NN algorithms. The effect is seen as the training of the network stalls after a number of epochs. In order to deal with this, increasing the number of hidden layers, increasing the number of neurons in each layer, and choosing different types of transfer function were investigated. However, none of these methods showed a significant improvement in the behavior of the network. The other methods of solving the inverse problem are much less sensitive to the effect of nonlinearity.

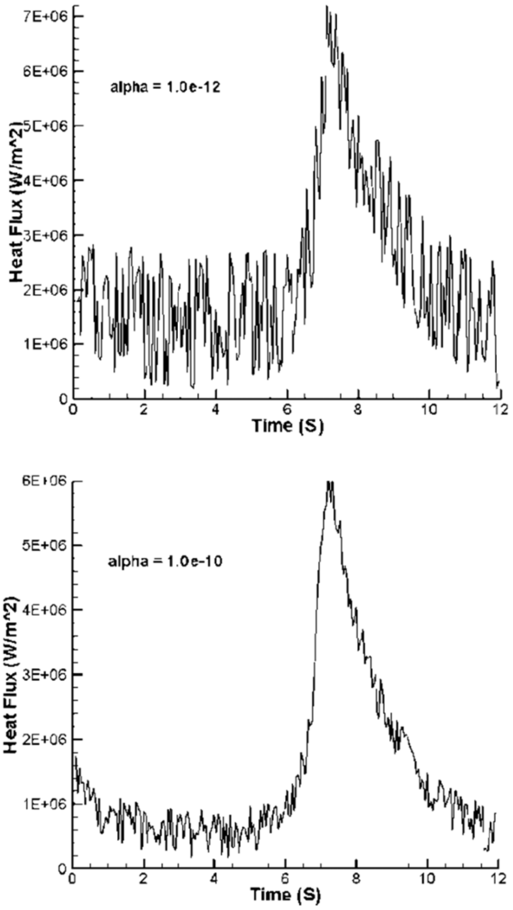


Fig. 8 Effect of Regularization Parameter α

TABLE II
EFFECT OF THE SELF-CONFIDENCE PARAMETER ON THE L2 NORM OF ERROR IN THE SOLUTION

C_0	PSO	RPSO	CRPSO
0.7	8.105 E+4	7.577 E+4	7.611 E+4
0.8	7.532 E+4	7.079 E+4	6.823 E+4
0.95	6.257 E+4	7.064 E+4	6.685 E+4
1.1	6.346 E+4	5.816 E+4	6.739 E+4
1.2	6.117 E+4	5.999 E+4	5.822 E+4

TABLE III
THE L2 NORM OF ERROR IN THE SOLUTION IN AN EXACT DOMAIN FOR DIFFERENT ALGORITHMS

Method	Linear	Non-Linear
Function Specification Method	1.81 E+2	2.14 E+2
GA	7.62 E+2	7.71 E+2
PSO	3.85 E+2	4.46 E+2
RPSO	3.42 E+2	5.12 E+2
CRPSO	3.17 E+2	4.26 E+2
FMLP	9.90 E+2	3.57 E+4
RBFN	5.35 E+2	2.76 E+4

V. CONCLUSION

In this paper, we introduced a gradient-based inverse solver to obtain the missing boundary conditions based on the readings of internal thermocouples. The results show that the method is very sensitive to measurement errors, and becomes unstable when small time steps are used. The artificial neural networks are shown to be capable of capturing the whole thermal history on the run-out table, but are not very effective in restoring the detailed behavior of the boundary conditions. Also, they behave poorly in nonlinear cases and where the boundary condition profile is different.

GA and PSO are more effective in finding a detailed representation of the time-varying boundary conditions, as well as in nonlinear cases. However, their convergence takes longer. A variation of the basic PSO, called CRPSO, showed the best performance among the three versions. Also, PSO proved to be effective in handling noisy data, especially when its performance parameters were tuned. An increase in the self-confidence parameter was also found to be effective, as it increased the global search capabilities of the algorithm. RPSO was the most effective variation in dealing with noise, closely followed by CRPSO. The latter variation is recommended for inverse heat conduction problems, as it combines the efficiency and effectiveness required by these problems.

ACKNOWLEDGMENT

This publication was made possible by NPRP grant No. 09507-2-191 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] Alifanov, Oleg M., A. V. Nenarokomov, S. A. Budnik, V. V. Michailov, and V. M. Ydin. "Identification of thermal properties of materials with applications for spacecraft structures." Inverse Problems in Science and Engineering 12, no. 5 (2004): 579-594.

[2] Beck, James V. Inverse heat conduction: Ill-posed problems. James Beck, 1985.

[3] Beck, James Vere, and Kenneth J. Arnold. Parameter estimation in engineering and science. James Beck, 1977.

[4] Beck, J. V., B. Blackwell, and A. Haji-Sheikh. "Comparison of some inverse heat conduction methods using experimental data." International Journal of Heat and Mass Transfer 39, no. 17 (1996): 3649-3657.

[5] Al-Khalidy, Nehad. "A general space marching algorithm for the solution of two-dimensional boundary inverse heat conduction problems." Numerical Heat Transfer, Part B 34, no. 3 (1998): 339-360.

[6] Jarny, Yvon. "Determination of heat sources and heat transfer coefficient for two-dimensional heat flow–numerical and experimental study." International Journal of Heat and Mass Transfer 44, no. 7 (2001): 1309-1322.

[7] Huang, Cheng-Hung, I-Cha Yuan, and Herchang Ay. "A three-dimensional inverse problem in imaging the local heat transfer coefficients for plate finned-tube heat exchangers." International Journal of Heat and Mass Transfer 46, no. 19 (2003): 3629-3638.

[8] Girault, Manuel, Daniel Petit, and Etienne Videcoq. "The use of model reduction and function decomposition for identifying boundary conditions of a linear thermal system." Inverse Problems in Science and Engineering 11, no. 5 (2003): 425-455.

[9] Kim, H. K., and S. I. Oh. "Evaluation of heat transfer coefficient during heat treatment by inverse analysis." Journal of Materials Processing Technology 112, no. 2 (2001): 157-165.

[10] Louahlia-Gualous, H., P. K. Panday, and E. A. Artioukhine. "Inverse determination of the local heat transfer coefficients for nucleate boiling on a horizontal cylinder." Journal of heat transfer 125, no. 6 (2003): 1087-1095.

[11] Gadala, Mohamed S., and Fuchang Xu. "An FE-based sequential inverse algorithm for heat flux calculation during impingement water cooling." International Journal of Numerical Methods for Heat & Fluid Flow 16, no. 3 (2006): 356-385.

[12] Roudbari, Shawhin. "Self-Adaptive Finite Element Analysis." PhD diss., Cornell University, 2006.

[13] Silieti, M., E. Divo, and A. J. Kassab. "An inverse boundary element method/genetic algorithm based approach for retrieval of multi-dimensional heat transfer coefficients within film cooling holes/slots." Inverse Problems in Science and Engineering 13, no. 1 (2005): 79-98.

[14] Shiguemori, Elcio H., José Dem?Sio S. Da Silva, and Haroldo F. de Campos Velho. "Estimation of initial condition in heat conduction by neural network." Inverse Problems in Science and Engineering 12, no. 3 (2004): 317-328.

[15] Lecoeuche, S., G. Mercere, and S. Lalot. "Evaluating time-dependent heat fluxes using artificial neural networks." Inverse Problems in Science and Engineering 14, no. 2 (2006): 97-109.

[16] Ostrowski, Z., R. A. Bialstrokecki, and A. J. Kassab. "Solving inverse heat conduction problems using trained POD-RBF network inverse method." Inverse Problems in Science and Engineering 16, no. 1 (2008): 39-54.

[17] Hassan, Rania, Babak Cohanim, Olivier De Weck, and Gerhard Venter. "A comparison of particle swarm optimization and the genetic algorithm." In Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference, pp. 18-21. 2005.

[18] Gosselin, Louis, Maxime Tye-Gingras, and François Mathieu-Potvin. "Review of utilization of genetic algorithms in heat transfer problems." International Journal of Heat and Mass Transfer 52, no. 9 (2009): 2169-2188.

[19] Davis, Lawrence, ed. Handbook of genetic algorithms. Vol. 115. New York: Van Nostrand Reinhold, 1991.

[20] Clerc, Maurice. Particle swarm optimization. Vol. 93. John Wiley & Sons, 2010.

[21] Kennedy, James, James F. Kennedy, and Russell C. Eberhart. Swarm intelligence. Morgan Kaufmann, 2001.

[22] Vakili, S., and M. S. Gadala. "Effectiveness and efficiency of particle swarm optimization technique in inverse heat conduction analysis." Numerical Heat Transfer, Part B: Fundamentals 56, no. 2 (2009): 119-141.

[23] Alrasheed, M. R., C. W. de Silva, and M. S. Gadala. "Evolutionary optimization in the design of a heat sink." Mechatronic Systems: Devices, Design, Control, Operation and Monitoring (2008).

[24] De Silva, Clarence W. Mechatronic Systems. Taylor and Francis, 2007.

[25] Urfalioglu, Onay. "Robust estimation of camera rotation, translation and focal length at high outlier rates." In Computer and Robot Vision, 2004. Proceedings. First Canadian Conference on, pp. 464-471. IEEE, 2004.