EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
Vol. 4, No. 2, 2011, 117-128
ISSN 1307-5543 – www.ejpam.com

Hybridization of Neural Nets and Genetic Algorithms to Compute the Boundary Control for Controlled Heat Equation

N. K. Tomar[1*], N. Sukavanam[2], K. P. Singh[3]

[1] *Department of Mathematics, Indian Institute of Technology Patna-800013, India*
[2] *Department of Mathematics, Indian Institute of Technology Roorkee-247667, India*
[3] *Indian Institute of Information Technology Allahabad-211012, India*

Abstract. This paper presents a computational method for boundary control of controlled heat equation. The proposed method relies upon the function approximation capability of feedforward neural network and global optimization power of genetic algorithm.

2000 Mathematics Subject Classifications: 93C20, 92B20, 90C90

Key Words and Phrases: Control system governed by heat equation, Artificial Neural Networks, Genetic Algorithms

1. Introduction

Let Ω be a nonempty, bounded, open and connected domain in R^n with a smooth boundary Γ. Consider the time interval $(0,\tau)$ and the sets $Q = \Omega \times (0,\tau)$ and $\sum = \Gamma \times (0,\tau)$. The controlled semilinear heat equation, with boundary control, in Q is

$$\begin{aligned} \frac{\partial \omega(x,t)}{\partial t} &= c^2 \Delta \omega(x,t) + \phi(t, \omega(x,t)); \quad \text{in } Q \\ \omega(x,0) &= \omega_0; \quad \text{in } \Omega \\ \omega &= C(\sigma,t); \quad \text{on } \Gamma \end{aligned} \tag{1}$$

where, $\omega(x,t)$ denotes the temperature at $x \in \Omega$ and $t \in [0,\tau]$, Δ denotes the Laplace operator, ϕ is a linear or nonlinear operator and c is a real constant. $C(\sigma,t)$ represents boundary control function, σ being boundary variable.
Now, the problem is to find a boundary control function C on Γ such that the solution ω satisfies $\omega(x,\tau) = \omega_\tau$ on Ω, where ω_τ is any final required temperature.

*Corresponding author.

Email addresses: nktomar@iitp.ac.in (N. Tomar), nsukvfma@iitr.ernet.in (N. Sukavanam), kpsinghiitr@gmail.com (K. Singh)

Studies on the controlled heat equation have revealed many theoretical concepts for different types of controllability problems, see for instance [19, 2, 9, 4, 14, 6, 17, 8, 16]. But there is a gap in the literature for numerical solution for boundary control, which is one of the important issues in the context of controlled heat equation model. The numerical computation of boundary control of a two dimensional linear heat equation has been shown in [2] using finite difference, finite element and conjugate gradient method. In this paper, a method for computation of boundary control for mathematical model (1), under the given initial, final and boundary conditions, is proposed using developing less conventional Artificial Neural Network (ANN) and Genetic Algorithm (GA) techniques.
In [11, 12] Neural Networks have been used for solving ordinary and partial differential equations with known initial and boundary conditions. Sukavanam and Panwar [15] presented a numerical solution of boundary control for controlled heat equation using ANN. In all these thee papers [11, 12, 15], the back propagation algorithm is applied to train a feed forward neural network, which is based on steepest descent method.
In this paper a similar technique as that of [11, 12, 15] is used to approximate the state and control terms, that is possible due to the capability of a feed forward neural network. After the approximation, above problem is converted into an optimization problem, and then for resulting problem, a real coded GA is applied. Use of evolutionary techniques in control problems is not an new idea. For example, Michalewicz *et. at.* [13] gave applications of genetic algorithm to study some discrete-time optimal control problems. In the same year, Krishnakumar and Goldberg [10] explored aerospace-related control system optimization problems using genetic algorithms.
Chang [3] applied a real-coded genetic algorithm to the system identification and control for a class of nonlinear systems. Some more applications of evolutionary algorithms in control systems can be seen in [18, 1, 7].
The organization of this paper is as follows. Next section is concerned with some preliminaries of a feedforward neural network and its derivatives formulations, which will be useful in Section 3. Using ANN and GA, the proposed hybrid procedure is described in the Section 3. Numerical example, for two dimensional linear heat equation, illustrates the proposed method in the Section 4. Finally, in the last, some concluding remarks are given.

2. Preliminaries

Consider a multilayer feedforward neural network, with n input units, one hidden layer with H sigmoid units and a linear output unit. For a given input vector x, where $x = (x_1, x_2, \dots, x_n)$, the output of the network is $N(x) = \sum_{i=1}^{H} v_i \sigma(z_i)$, where $z_i = \sum_{j=1}^{n}(w_{ij}x_j + b_i)$. Here, w_{ij} denotes the weight from the input unit j to the hidden unit i , v_i denotes the weight from the hidden unit i to the output, b_i denotes the bias of hidden unit i. $\sigma(z) = 1/(1 + e^{-z})$ is the sigmoid transfer function. It is straightforward to show that

$$\frac{\partial^k N}{\partial x_j^k} = \sum_{i=1}^{H} v_i w_{ij}^k \sigma_i^k$$

where $\sigma_i = \sigma(z_i)$ and $\sigma^{(k)}$ denote the k^{th}-order derivative of the sigmoid function. Moreover,

$$\frac{\partial^{\lambda_1}}{\partial x_1^{\lambda_1}} \cdot \frac{\partial^{\lambda_2}}{\partial x_2^{\lambda_2}} \cdots \frac{\partial^{\lambda_n}}{\partial x_n^{\lambda_n}} N = \sum_{i=1}^{H} v_i \rho_i \sigma_i^{\alpha} \tag{2}$$

where $\rho_i = \prod_{k=1}^{n} w_{ik}^{\lambda_k}$ and $\alpha = \sum_{i=1}^{n} \lambda_i$. Equation (2) indicates that the derivative of the network N with respect to any of its inputs is equivalent to a feedforward neural network with one hidden layer, having the same values for the weights, as N, w_{ij} and thresholds b_i and each v_i being replaced with $v_i\rho_i$. The transfer function of each hidden unit is replaced with the α^{th} order derivative of the sigmoid.

One of the main uses of the feed forward neural network is the approximation of any function. In the current effort, a feed forward neural network with three input units, one hidden layer having six nodes (with bias at each node) and one output unit is used to approximate both the temperature and the control functions separately.

3. Method of Computation of Boundary Control

First, we assume the discretization of domain $Q = \Omega \times (0, \tau)$ and its boundary $\sum = \Gamma \times (0, \tau)$ into a set of points $\hat{Q} = \overline{\Omega} \times (t_0 = 0,\ t_1,\ t_2,\ \dots,\ t_k = \tau)$ and $\hat{\sum} = \overline{\Gamma} \times (t_0 = 0,\ t_1,\ t_2,\ \dots,\ t_k = \tau)$ where $\overline{\Omega}$ and $\overline{\Gamma}$ denote the set of grid points in Ω and Γ respectively and $t_0,\ t_1,\ t_2,\ \dots,\ t_k$ are discrete points of the interval $[0,\ \tau]$. Now the problem (1) is transformed into the solution of following system of equations

$$\frac{\partial \omega(x,t)}{\partial t} - c^2 \Delta \omega(x,t) - \phi\,(t, \omega(x,t)) = 0;\ \forall (x,t) \in \hat{Q} \tag{3}$$

subject to the given initial and final conditions.

In the proposed approach, we construct a trial solution with adjustable parameters. This trial solution is written as a sum of two parts. The first part is constructed to satisfy, the known initial and final conditions, and the unknown boundary conditions in the form of a feedforward neural network. The second part is also a feedforward neural network with adjustable weights. It is constructed in a way so as not to affect the given conditions. If $\omega_{\mathrm{tr}}(x,t)$ denotes the trial solution with adjustable weight vector p then it is written as

$$\omega_{\mathrm{tr}}(x,t,p) = A(x,t,N_1(x,t,p_1)) + B(x,t,N_2(x,t,p_2)) \tag{4}$$

where $(x,t) \in \hat{Q}$ and $p = [p_1, p_2]$.

The control term is approximated by an another feedforward neural network. A general description of the network approximating the boundary control function is not easy. It is clear from the example given below that the form of network depends on the shape of boundary.

Now the problem of solving the system of equation (3) is transformed to the following minimization problem

$$E = \min_{p} \sum_{(x,t)\in\hat{Q}} \{\frac{\partial \omega_{\mathrm{tr}}(x,t,p)}{\partial t} - c^2 \Delta \omega_{\mathrm{tr}}(x,t,p) - \phi(t, \omega_{\mathrm{tr}}(x,t,p))\}^2 \tag{5}$$

The weight parameters are found by minimizing $E[(p_1, p_2)]$ using GA.
In this study, the real coded genetic algorithm, MI-LXPM, for details see [5], has been used to solve above single objective optimization problems. Computational steps of MI-LXPM algorithm are as follows:

Step-1: Generate a suitably large initial set of random points within the domain, it may be 5 to 10 times of number of variables. Evaluate their fitness values.

Step-2: Check the stopping criteria? If satisfied stop else goto step 3.

Step-3: Apply tournament selection operator to decide which of these individuals are to be in mating pool.

Step-4: Apply Laplace crossover operator to all individuals in mating pool with probability of crossover P_c. Crossover operator produces new solution points in the search space.

Step-5: Apply Power mutation to all individuals in mating pool with probability of mutation P_m. Mutation operator maintains diversity in the population, so it prevents the algorithm to converge to local optima.

Step-6: Increase generation; goto step 2.

Parameter setting is one of the important aspect of genetic algorithms. In this study parameters value have been taken as $a = 0$, $b = 0.35$, $p = 4$, $P_c = 0.8$, $P_m = 0.005$ with tournament size three (details of the parameters a, b, and p can be seen in [5]). Moreover, 40,000 generation has been used as stopping criteria for a run of GA.

4. Numerical Result

Consider the controlled linear two-dimensional heat equation

$$\frac{\partial \omega(x,t)}{\partial t} = \nu \frac{\partial^2 \omega(x,t)}{\partial x^2} + \frac{\partial^2 \omega(x,t)}{\partial y^2} + 3\pi^3 \nu \exp(2\pi^2 \nu t)(\sin(\pi x) + \sin(\pi y)) \tag{6}$$

with initial condition

$$\omega(x,y,0) = \pi(\sin(\pi x) + \sin(\pi y)) \tag{7}$$

where $t \in (0,1)$, $\nu = \frac{1}{2\pi^2}$ and $(x,y) \in \Omega = (0,1) \times (0,1)$.
on unit square boundary

$$\Gamma = \{y = 0, 0 \le x \le 1; x = 1, 0 \le y \le 1; y = 1, 0 \le x \le 1; x = 0, 0 \le y \le 1\}.$$

Now the problem is to find the unknown control functions, g, h, m, and n, defined on the given boundary as

$$\begin{aligned} \omega(0,y,t) &= g(y,t) \\ \omega(x,0,t) &= h(x,t) \end{aligned}$$

$$\begin{aligned} \omega(1,y,t) &= m(y,t) \\ \omega(x,1,t) &= n(x,t) \end{aligned} \tag{8}$$

such that at the final time $t=1$ the solution $\omega(x,y,t)$ satisfies

$$\omega(x,y,1)=\pi\exp(2\pi^2\nu)(\sin(\pi x)+\sin(\pi y)) \tag{9}$$

From equations (6)-(9) the following compatibility conditions on g, h, m and n can be obtained

$$\begin{aligned} g(y,0) &= \pi\sin(\pi y);\ \ g(y,1)=\pi\exp(2\pi^2\nu)\sin(\pi y) \\ h(x,0) &= \pi\sin(\pi x);\ \ h(x,1)=\pi\exp(2\pi^2\nu)\sin(\pi x) \\ m(y,0) &= \pi\sin(\pi y);\ \ m(y,1)=\pi\exp(2\pi^2\nu)\sin(\pi y) \\ n(x,0) &= \pi\sin(\pi x);\ \ n(x,1)=\pi\exp(2\pi^2\nu)\sin(\pi x) \end{aligned} \tag{10}$$

Also

$$g(0,t)=h(0,t);g(1,t)=n(0,t);h(1,t)=m(0,t);m(1,t)=n(1,t) \tag{11}$$

Hence, we consider the neural network approximation of g, h, m and n as

$$\begin{aligned} g(y,t) &= [1-t+t\exp(2\pi^2\nu)]\pi\sin(\pi y)+t(1-t)[y(1-y)N_g+(1-y)N_1+yN_2] \\ h(x,t) &= [1-t+t\exp(2\pi^2\nu)]\pi\sin(\pi x)+t(1-t)[x(1-x)N_h+(1-x)N_1+xN_3] \\ m(y,t) &= [1-t+t\exp(2\pi^2\nu)]\pi\sin(\pi y)+t(1-t)[y(1-y)N_m+(1-y)N_3+yN_4] \\ n(x,t) &= [1-t+t\exp(2\pi^2\nu)]\pi\sin(\pi x)+t(1-t)[x(1-x)N_n+(1-x)N_2+xN_4] \end{aligned} \tag{12}$$

where,

$$\begin{aligned} N_g &= N_g(y,t,p_g)=\sum v_g\sigma(b_g y+c_g t+d_g) \\ N_h &= N_g(x,t,p_h)=\sum v_h\sigma(a_h x+c_h t+d_h) \\ N_m &= N_m(y,t,p_m)=\sum v_m\sigma(b_m y+c_m t+d_m) \\ N_n &= N_n(x,t,p_n)=\sum v_n\sigma(a_n x+c_n t+d_n) \\ N_1 &= N_1(t,p_1)=\sum v_1\sigma(c_1 t+d_1) \\ N_2 &= N_2(t,p_2)=\sum v_2\sigma(c_2 t+d_2) \\ N_3 &= N_3(t,p_3)=\sum v_3\sigma(c_3 t+d_3) \\ N_4 &= N_4(t,p_4)=\sum v_4\sigma(c_4 t+d_4) \end{aligned} \tag{13}$$

where $p_\alpha=[v_\alpha,c_\alpha,d_\alpha]^T,\ \alpha=1,2,3,4;\ p_\beta=[v_\beta,a_\beta,c_\beta,d_\beta]^T,\ \beta=h,n$ and
$p_\gamma=[v_\gamma,b_\gamma,c_\gamma,d_\gamma]^T,\ \gamma=g,m$ are the weight vectors, and σ is sigmoid function. The trial solution of the system (6) with given conditions is

$$\begin{aligned}
&\omega_{\mathrm{tr}}(x,y,t) = (1-x)g(y,t) + xm(y,t) + (1-y)h(x,t) + yn(x,t)\\
&-(1-y)(1-x)h(0,t) - x(1-y)h(1,t) - y(1-x)n(0,t) - xyn(1,t)\\
&+(1-t)[I(x,y) - \{(1-x)g(y,0) + xm(y,0) + (1-y)h(x,0) + yn(x,0)\\
&-(1-y)(1-x)h(0,0) - x(1-y)h(1,0) - y(1-x)n(0,0) - xyn(1,0)\}]\\
&+t[F(x,y) - \{(1-x)g(y,1) + xm(y,1) + (1-y)h(x,1) + yn(x,1)\\
&-(1-y)(1-x)h(0,1) - x(1-y)h(1,1) - y(1-x)n(0,1) - xyn(1,1)\}]\\
&+t(1-t)x(1-x)y(1-y)N_\omega(x,y,t,p) \qquad (14)
\end{aligned}$$

where $N_\omega(x,y,t,p) = \sum v_w \sigma(a_w x + b_w y + c_w t + d_w)$; $p = [v_w, a_w, b_w, c_w, d_w]^T$ is the weight vector; $I(x,y) = \omega(x,y,0)$ is the given initial condition and $F(x,y) = \omega(x,y,1)$ is the final required condition.
Using (10)-(11) the trial solution becomes

$$\begin{aligned}
&\omega_{\mathrm{tr}}(x,y,t) = [1 - t + t\exp(2\pi^2\nu)]\pi(\sin(\pi x) + \sin(\pi y)) + t(1-t)[(1-x)y(1-y)N_g\\
&+(1-y)x(1-x)N_h + xy(1-y)N_m + yx(1-x)N_n + (1-x)(1-y)N_1 + y(1-x)N_2\\
&+x(1-y)N_3 + xyN_4] + t(1-t)x(1-x)y(1-y)N_\omega. \qquad (15)
\end{aligned}$$

It can be easily verified that the trial solution satisfies all initial and boundary conditions.
Let us consider the grid points in x, y and t as

$$x_i = (0.1)i;\quad i = 0,1,2,\ldots,9,10$$

$$y_j = (0.1)j;\quad j = 0,1,2,\ldots,9,10 \text{ and}$$

$$t_k = (0.1)k;\quad k = 0,1,2,\ldots,9,10.$$

Then the error function is given by

$$\begin{aligned}
E = \sum_i \sum_j \sum_k \quad &[\tfrac{\partial \omega_{\mathrm{tr}}(x_i,y_j,t_k)}{\partial t} - \nu\{\tfrac{\partial^2 \omega_{\mathrm{tr}}(x_i,y_j,t_k)}{\partial x^2} + \tfrac{\partial^2 \omega_{\mathrm{tr}}(x_i,y_j,t_k)}{\partial y^2}\}\\
&-3\pi^3\nu\exp(2\pi^2\nu t_k)(\sin(\pi x_i) + \sin(\pi y_j))]^2 \qquad (16)
\end{aligned}$$

which is to be minimized with respect to all weight parameters p's. Now, the real coded GA, MI-LXPM [5], is used to find out the optimal weight parameters.
The exact solution of the above problem is given in [2] as follows: The (optimal) control function is

$$\pi\exp(2\pi^2\nu t)\sin(\pi x);\quad \text{for } 0 < x < 1;\ y = 0,1$$

and

$$\pi\exp(2\pi^2\nu t)\sin(\pi y);\quad \text{for } 0 < y < 1;\ x = 0,1$$

and the temperature ω being defined by

$$\omega(x,y,t) = \pi\exp(2\pi^2\nu t)(\sin(\pi x) + \sin(\pi y))$$

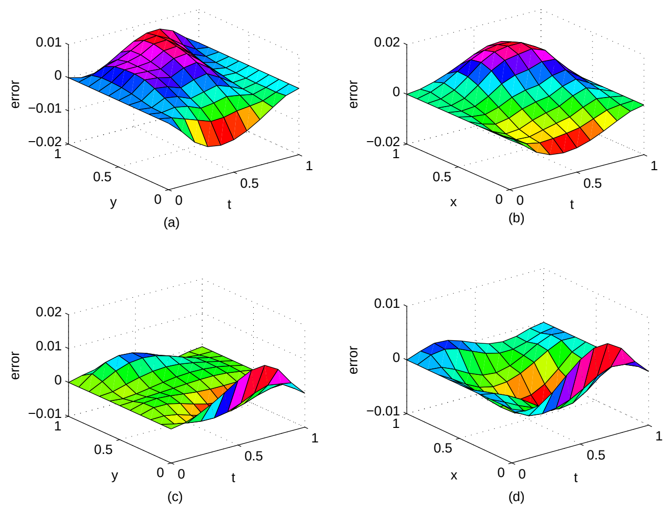

Figure 1: Difference between the exact and computed values of control at boundary (a)$x = 0$ (b)$y = 0$ (c)$x = 1$ (d)$y = 1$

The figure 1 represents the error between exact and computed control functions at the boundary. The figure 2 and tables 1-6 describe the pointwise error between exact and calculated temperature distribution on the region Ω at some intermediate time points between 0 and 1.

Table 1: Difference between the Exact and Computed values of Temperature Distribution at Time $t = 0.1$

y x	0.0	0.2	0.4	0.6	0.8	1.0
0.0	-0.00355	-0.00059	0.000221	0.001451	0.001674	-0.001
0.1	-0.00299	-0.00065	-0.00016	0.000301	0.000372	-0.00217
0.2	-0.00342	-0.00112	-0.00024	0.000177	0.000635	-0.0012
0.3	-0.00345	-0.0014	-0.00019	0.000189	0.000911	-0.00049
0.4	-0.00233	-0.00109	-4.6E-05	-6E-05	0.000532	-0.00109
0.5	-0.00124	-0.00085	-1.6E-05	-0.00038	0.000199	-0.00174
0.6	-0.0008	-0.00097	-0.00011	-0.00055	0.000404	-0.00155
0.7	-0.00043	-0.0011	-0.00029	-0.00089	0.00044	-0.00146
0.8	0.000385	-0.00051	0.000356	-0.00039	0.001404	-0.00023
0.9	0.001301	0.000415	0.001326	0.000316	0.002464	0.001163
1.0	0.000683	8.59E-05	0.001065	-0.00036	0.001959	0.000962

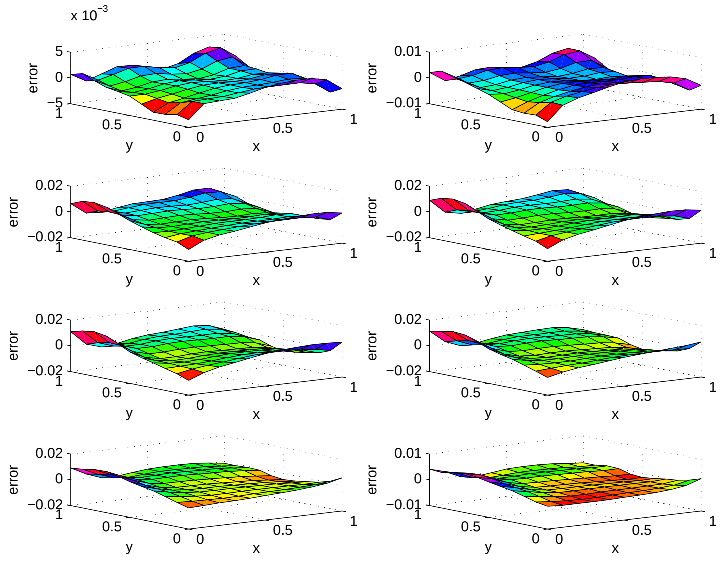

Figure 2: Difference between the exact and computed values of temperature distribution at time (a)$t = 0.1$ (b)$t = 0.2$ (c)$t = 0.4$ (d)$t = 0.5$ (e)$t = 0.6$ (f)$t = 0.7$ (g)$t = 0.8$ (h)$t = 0.9$

Table 2: Difference between the Exact and Computed values of Temperature Distribution at Time $t = 0.2$

y x	0.0	0.2	0.4	0.6	0.8	1.0
0.0	-0.00782	-1.6E-05	0.002893	0.00455	0.003965	-0.00073
0.1	-0.00617	-0.00022	0.001731	0.002088	0.001277	-0.00356
0.2	-0.00636	-0.00119	0.001062	0.0013	0.001247	-0.00251
0.3	-0.0059	-0.00178	0.000751	0.000895	0.00134	-0.00162
0.4	-0.00384	-0.00156	0.000579	0.000132	0.000401	-0.00274
0.5	-0.00179	-0.00144	0.000231	-0.0007	-0.00047	-0.00407
0.6	-0.00034	-0.00139	0.000213	-0.00069	0.000155	-0.00355
0.7	0.001082	-0.00125	0.000164	-0.00085	0.000678	-0.00313
0.8	0.002572	-0.00052	0.000891	-0.00018	0.002219	-0.00134
0.9	0.003587	0.000455	0.002139	0.001059	0.004434	0.001527
1.0	0.002104	-0.00057	0.001414	2.73E-05	0.004024	0.00172

Table 3: Difference between the Exact and Computed values of Temperature Distribution at Time $t = 0.4$

y x	0.0	0.2	0.4	0.6	0.8	1.0
0.0	-0.01087	-0.00241	0.000972	0.00394	0.00545	0.003469
0.1	-0.00685	-0.00213	-0.00056	0.000409	0.000656	-0.00354
0.2	-0.00526	-0.00284	-0.00147	-0.00095	-0.00031	-0.00467
0.3	-0.00326	-0.00314	-0.00179	-0.00165	-0.00062	-0.00474
0.4	-0.0005	-0.00317	-0.00218	-0.0028	-0.00202	-0.00631
0.5	0.002364	-0.00313	-0.00248	-0.00369	-0.00333	-0.00805
0.6	0.005879	-0.00184	-0.00105	-0.00222	-0.0017	-0.00675
0.7	0.009317	-0.00028	0.000641	-0.00034	0.000586	-0.00502
0.8	0.010905	8.36E-05	0.001309	0.000589	0.002272	-0.00379
0.9	0.010141	-0.00042	0.00198	0.002111	0.005338	-0.00027
1.0	0.006458	-0.00311	0.000629	0.001301	0.005781	0.000724

Table 4: Difference between the Exact and Computed values of Temperature Distribution at Time $t = 0.6$

y x	0.0	0.2	0.4	0.6	0.8	1.0
0.0	-0.00862	-0.00099	0.003168	0.006495	0.007819	0.007146
0.1	-0.00523	-0.00147	0.001031	0.002723	0.002478	-0.00131
0.2	-0.00287	-0.00213	-0.00024	0.000856	0.00028	-0.0049
0.3	-0.0004	-0.00258	-0.00103	-0.00026	-0.00075	-0.00647
0.4	0.001772	-0.00339	-0.00223	-0.00184	-0.00223	-0.0081
0.5	0.004502	-0.00357	-0.00277	-0.00281	-0.00322	-0.00903
0.6	0.008888	-0.00166	-0.00092	-0.00111	-0.00148	-0.00706
0.7	0.013116	0.000631	0.001627	0.00164	0.001465	-0.00395
0.8	0.014614	0.000564	0.00184	0.002137	0.002189	-0.00346
0.9	0.013394	-0.00113	0.001187	0.002427	0.003489	-0.00202
1.0	0.010967	-0.00373	-5.8E-05	0.002111	0.004252	-0.00118

Table 5: Difference between the Exact and Computed values of Temperature Distribution at Time $t = 0.8$

y x	0.0	0.2	0.4	0.6	0.8	1.0
0.0	-0.00359	-0.00246	-0.00166	-0.00034	0.001379	0.00572
0.1	-0.00107	-0.00189	-0.00185	-0.00114	-0.00046	0.001339
0.2	0.001704	-0.00088	-0.00148	-0.00129	-0.00133	-0.00129
0.3	0.004206	-9.6E-07	-0.001	-0.00117	-0.00164	-0.00268
0.4	0.0059	0.000105	-0.00118	-0.00161	-0.00235	-0.00404
0.5	0.007745	0.000439	-0.00109	-0.00172	-0.00266	-0.00473
0.6	0.010212	0.001763	0.000188	-0.00046	-0.00142	-0.00356
0.7	0.01196	0.003004	0.001789	0.001495	0.00079	-0.00111
0.8	0.011675	0.002412	0.001633	0.001702	0.001107	-0.0008
0.9	0.010012	0.000539	0.000527	0.001222	0.000816	-0.00134
1.0	0.009157	-0.00121	-1.4E-05	0.001639	0.001541	-0.00121

Table 6: Difference between the Exact and Computed values of Temperature Distribution at Time $t = 0.9$

y x	0.0	0.2	0.4	0.6	0.8	1.0
0.0	-0.00116	-0.00142	-0.00136	-0.00099	-0.0003	0.00255
0.1	-0.00037	-0.00128	-0.0014	-0.00101	-0.0006	0.001097
0.2	0.000944	-0.00064	-0.00109	-0.00085	-0.00072	0.000112
0.3	0.002166	5.63E-06	-0.00071	-0.00061	-0.00064	-0.00034
0.4	0.002943	0.000227	-0.00072	-0.00078	-0.00097	-0.00102
0.5	0.003754	0.000505	-0.00065	-0.00088	-0.00125	-0.00154
0.6	0.004704	0.001148	-0.0001	-0.00036	-0.0008	-0.00116
0.7	0.005135	0.001674	0.000597	0.00056	0.000276	6.22E-05
0.8	0.004602	0.001318	0.000459	0.000643	0.00043	0.000228
0.9	0.003754	0.000394	-0.00013	0.000315	5.77E-05	-0.00041
1.0	0.004129	-0.0002	-8.6E-05	0.000899	0.000611	-0.00043

Conclusions

A numerical method has been presented for computation of boundary control of controlled heat equation. It is hybridization of the function approximation capabilities of feed forward neural network and global optimization capabilities of genetic algorithm. The solutions of controlled heat equation problems obtained through this approach compare very well with the exact solutions. Computational experience in using proposed ANN-GA algorithm in solving this heat equation problem has shown that the total time requirement of this technique is less then to compare with other local search techniques like gradient descent method, because the time has been compared to those taken by conventional method used by Sukavanam and Panwar [15]. Conventional search techniques depend on an initial guess, so these techniques generally give local optimal solutions. However, initial guess is not required in proposed ANN-GA technique. Since, GA works on population of almost all possible solutions, it gives, mostly, global optimal solution.
The methodology discussed here can be applied to both linear and nonlinear state equation and this technique may be further hybridized with local search techniques to find more precise solutions. The operational range of the proposed, ANN-GA, technique can also be extended for the optimal control problems by converting them into constrained optimization problems via neural networks or any other discretization method.

References

[1] Y Arfiadi and M N S Hadi. Optimal Direct (Static) Output Feedback Controller using Real Coded Genetic Algorithms. *Computers and Structures*, 79:1625–1634, 2001.

[2] C Carthel, R Glowinski, and J L Lions. On Exact and Approximate Boundary Controllabilities for the Heat Equation: A Numerical Approach. *Journal of Optimization Theory and Applications*, 82(2):429–484, 1994.

[3] W D Chang. Nonlinear System Identification and Control using a Real-Coded Genetic Algorithm. *Applied Mathematical Modelling*, 31:541–550, 2007.

[4] J M Coron and E Trelat. Global Steady-state Controllability of One-dimensional Semilinear Heat Equation. *SIAM J. Control*, 43(2):549–569, 2004.

[5] K Deep, K P Singh, M L Kansal, and C Mohan. A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation*, 212(2):505–518, 2009.

[6] C Fabre, J P Puel, and E Zuazua. Approximate Controllability of the Semilinear Heat Equation. *Proceedings of the Royal Society of Edinburgh*, 125A:31–61, 1995.

[7] P J Fleming and R C Purshouse. Evolutionary Algorithms in Control Systems Engineering: A Survey. *Control Engineering Practice*, 10:1233–1241, 2002.

[8] A Friedman and L S Jiang. Nonlinear Optimal Control in Heat Conduction. *SIAM Journal on Control and Optimization*, 21(6):940–951, 1983.

[9] R Glowinski. Boundary controllability problems for wave and heat equations. In J P Zelensio, editor, *Boundary Control and Boundary Variation, Lecture Notes in Control and Information Sciences.*, volume 178, pages 221–237. Springer Verlag, Berlin, Germany, 1992.

[10] K Krishnakumar and D E Goldberg. Control System Optimization using Genetic Algorithms. *J Guidance, Control and Dynamics*, 15(3):735–742, 1992.

[11] I E Lagaris, A C Likas, and D I Fotiadis. Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

[12] I E Lagaris, A C Likas, and D G Papageorgiou. Neural Network Methods for Boundary Value Problems with Irregular Boundaries. *IEEE Transactions on Neural Networks*, 11(5):1041–1049, 2000.

[13] Z Michalewicz, C Z Janikow, and J B Krawczyk. A Modified Genetic Algorithm for Optimal Control Problems. *Computers Math. Applic.*, 23(12):83–94, 1992.

[14] M Sirbu. Feedback Null Controllability of the Semilinear Heat Equation. *Differential and Integral Equation*, 15(1):115–128, 2002.

[15] N Sukavanam and V Panwar. Computation of boundary control of controlled heat equation using artificial neural networks. *Int. Comm. Heat Mass Transfer*, 30(8):1137–1146, 2003.

[16] N Sukavanam and N K Tomar. Approximate Controllability of Semilinear Delay Control Systems. *Nonlinear Functional Analysis and Applications*, 12(1):53–59, 2007.

[17] L D Teresa. Approximate Controllability of a Semilinear Heat Equation. *SIAM Journal on Control and Optimization*, 36(6):2128–2147, 1998.

[18] Y Yamashita and M Shima. Numerical Computational Method using Genetic Algorithm for the Optimal Control Problem with Terminal Constraints and Free Parameters. *Nonlinear Analysis, Theory, Method and Applications*, 30(4):2285–2290, 1997.

[19] H X Zhou. A Note on Approximate Controllability for Semilinear One-dimensional Heat Equation. *Appl. Math. Optim.*, 8:275–285, 1982.