Iranian Journal of Numerical Analysis and Optimization
Vol. 8, No. 2, (2018), pp 119–140
DOI:10.22067/ijnao.v8i2.64202

An efficient hybrid algorithm based on genetic algorithm (GA) and Nelder–Mead (NM) for solving nonlinear inverse parabolic problems

H. Dana Mazraeh* and R. Pourgholi

Abstract

In this paper a hybrid algorithm based on genetic algorithm (GA) and Nelder–Mead (NM) simplex search method is combined with least squares method for the determination of temperature in some nonlinear inverse parabolic problems (NIPP). The performance of hybrid algorithm is established with some examples of NIPP. Results show that hybrid algorithm is better than GA and NM separately. Numerical results are obtained by implementation expressed algorithms on 2.20GHz clock speed CPU.

Keywords: Hybrid; NIPP; Nonlinear inverse parabolic problem; Genetic algorithm; Nelder–Mead; The least squares method.

1 Introduction

Most phenomena in real world are described through nonlinear equations, and these type of equations have attracted lots of attention among scientists. For example, consider that the sun shines in a room, and it begins to warm up. Apparently, the physical property here is the temperature, which is a function of the time and space, and the governing equations that represent the temperature variations, are a nonlinear parabolic problem (NPP) [1].

*Corresponding author

Received 5 May 2017; revised 25 November 2017; accepted 13 June 2018

H. Dana Mazraeh
School of Mathematics and Computer Science, Damghan University, P.O.Box 36715-364, Damghan, Iran. e-mail:dana@du.ac.ir

R. Pourgholi
School of Mathematics and Computer Science, Damghan University, P.O.Box 36715-364, Damghan, Iran. e-mail:pourgholi@du.ac.ir

In linear and nonlinear parabolic problems, we are usually facing a problem, where problem conditions, initial conditions, and boundary conditions are identified and in the main equation only the equation main function is unknown. In fact, there is just one unknown factor at the problem. These problems are called direct problems.

On the contrary there is another category of problems. In this category, in addition to the main equation of the problem, there are another unknown parts such as boundary conditions. This type of problems are called inverse problems [12].

In the present paper, we consider nonlinear inverse parabolic problems as the following form:

$$U_t = \phi(x,t,U,U_x,U_{xx}), \quad (x,t) \in (0,1) \times (0,t_m), \tag{1a}$$

with the initial condition

$$U(x,0) = f(x), \quad x \in [0,1], \tag{1b}$$

and the boundary conditions

$$U(0,t) = p(t), \quad t \in [0,t_m], \tag{1c}$$
$$U(1,t) = q(t), \quad t \in [0,t_m], \tag{1d}$$

and the overspecified condition

$$U(\alpha,t) = s(t), \quad t \in [0,t_m], \quad 0 < \alpha < 1, \tag{1e}$$

where ϕ is some nonlinear expression in terms of U, U_x, U_{xx}, and $f(x)$ is a continuous known function. $p(t)$ and $q(t)$ are infinitely differentiable known functions, and t_M represents the final existence time for the time evolution of the problem, while function $q(t)$ is unknown, which remains to be determined from some interior temperature measurements (overspecified condition).

In this paper we present a hybrid algorithm based on Nelder–Mead simplex search method and genetic algorithm for solving NIPPs. Genetic algorithm is one of the basic heuristics to solve various types of problems. In recent years, this algorithm has been applied on some linear and nonlinear partial differential equation to find unknown term in these equations. On the other hand, there are some hybrid algorithms, which use local search method as an inner subroutine to reduce the likelihood of the premature convergence and to expedite convergence rate of genetic algorithms. These algorithms are called "*Memetic Algorithms*" or "*Hybrid Algorithms*". In this study, to expiate convergence rate of genetic algorithm and increase accuracy of solutions, we add Nelder–Mead method to the algorithm as a local search method. The Nelder–Mead method is a commonly applied numerical method used to find the minimum or maximum of an objective function in a multidimensional space. It is successfully applied to nonlinear optimization problems for which derivatives may not be known. However, the Nelder–Mead technique is a heuristic search method that can converge to nonstationary points [16] on problems that can be solved by alternative methods.

Our main purpose is to find an unknown boundary condition in NIPPs using our proposed hybrid method. Because by having this unknown boundary condition, problem becomes a direct NPP and fortunately, many methods have been reported to solve direct NPPs. In this paper we use finite difference method for solving direct NPPs.

2 Some nonlinear inverse parabolic problems

In this section, we consider three NIPPs. The first equation is formulated as follows [1]:

$$
\begin{aligned}
&U_t(x,t)+U(x,t)U_x(x,t)=U_{xx}(x,t), && 0<x<1, \quad 0<t<t_M, && \text{(2a)}\\
&U(x,0)=f(x), && 0\le x\le 1, && \text{(2b)}\\
&U(0,t)=p(t), && 0\le t\le t_M, && \text{(2c)}\\
&U(1,t)=q(t), && 0\le t\le t_M, && \text{(2d)}
\end{aligned}
$$

and the overspecified condition

$$
U(\alpha,t)=s(t), \qquad 0\le t\le t_M, \quad 0<\alpha<1, \qquad \text{(2e)}
$$

The second equation is formulated as follows:

$$
\begin{aligned}
&U_t(x,t)-U(x,t)(1-U(x,t))=U_{xx}(x,t), && 0<x<1, \quad 0<t<t_M, && \text{(3a)}\\
&U(x,0)=f(x), && 0\le x\le 1, && \text{(3b)}\\
&U(0,t)=p(t), && 0\le t\le t_M, && \text{(3c)}\\
&U(1,t)=q(t), && 0\le t\le t_M, && \text{(3d)}
\end{aligned}
$$

and the overspecified condition

$$
U(\alpha,t)=s(t), \qquad 0\le t\le t_M, \quad 0<\alpha<1, \qquad \text{(3e)}
$$

and the third equation is formulated as follows:

$$
\begin{aligned}
&U_t(x,t)=a(t)U_{xx}(x,t), && 0<x<1, \quad 0<t<t_M, && \text{(4a)}\\
&U(x,0)=f(x), && 0\le x\le 1, && \text{(4b)}\\
&U(0,t)=p(t), && 0\le t\le t_M, && \text{(4c)}\\
&U(1,t)=q(t), && 0\le t\le t_M, && \text{(4d)}
\end{aligned}
$$

and the overspecified condition

$$U(\alpha,t)=s(t), \qquad 0\le t\le t_M, \quad 0<\alpha<1, \tag{4e}$$

where $f(x)$ is a continuous known function, $p(t)$, $a(t)$, and $s(t)$ are infinitely differentiable known functions and t_M represents the final existence time for the time evolution of the problem. In these equations, if $q(t)$ is unknown, then we are facing nonlinear inverse parabolic problems. In this case, unknown $q(t)$ must be determined from some interior temperature measurements(Overspecified condition).

Remark 1. In this study we use implicit finite difference approximation (Crank–Nicolson method) for discretizing above equations.

Therefore we have the following discretization for the first equation:

$$\begin{aligned} -r_1U_{i-1,j+1}+(2+2r_1)U_{i,j+1}-r_1U_{i+1,j+1} \\ =& r_1U_{i-1,j}+(2-2r_1)U_{i,j}+r_1U_{i+1,j} \\ &+2r_2({U_{i,j}}^2-U_{i,j}U_{i+1,j}),\ i=1,\dots,N-1,\ j=0,\dots,N-1, \end{aligned} \tag{5}$$

$$\begin{aligned} U_{i,0}&=f(ih), \quad j=0,\ i=1,\dots,N-1,\\ U_{0,j}&=p(jk), \quad i=0,\ j=0,1,\dots,N-1,\\ U_{N,j}&=q(jk), \quad i=N,\ j=0,1,\dots,N-1, \end{aligned}$$

where $x_i=ih$, $t_j=jk$, $r_1=k/h^2$, and $r_2=k/h$.

Using equation (5), we obtain the following linear algebraic system of equations:

$$\begin{pmatrix} 2+2r_1 & -r_1 & 0 & 0 & 0 & 0 & 0\\ -r_1 & 2+2r_1 & -r_1 & 0 & 0 & 0 & 0\\ . & . & . & . & . & . & .\\ . & . & . & . & . & . & .\\ . & . & . & . & . & . & .\\ 0 & 0 & 0 & 0 & -r_1 & 2+2r_1 & -r_1\\ 0 & 0 & 0 & 0 & 0 & -r_1 & 2+2r_1 \end{pmatrix} \begin{pmatrix} U_{1,j+1}\\ U_{2,j+1}\\ .\\ .\\ .\\ U_{N-2,j+1}\\ U_{N-1,j+1} \end{pmatrix}$$

$$= \begin{pmatrix} 2-2r_1 & r_1 & 0 & 0 & 0 & 0 & 0\\ r_1 & 2-2r_1 & r_1 & 0 & 0 & 0 & 0\\ . & . & . & . & . & . & .\\ . & . & . & . & . & . & .\\ . & . & . & . & . & . & .\\ 0 & 0 & 0 & 0 & r_1 & 2-2r_1 & r_1\\ 0 & 0 & 0 & 0 & 0 & r_1 & 2-2r_1 \end{pmatrix} \begin{pmatrix} U_{1,j}\\ U_{2,j}\\ .\\ .\\ .\\ U_{N-2,j}\\ U_{N-1,j} \end{pmatrix}$$

$$+r_1\begin{pmatrix} U_{0,j}+U_{0,j+1} \\ 0 \\ . \\ . \\ . \\ 0 \\ U_{N,j}+U_{N,j+1} \end{pmatrix} + 2r_2\begin{pmatrix} {U_{1,j}}^2 - U_{1,j}U_{2,i} \\ {U_{2,j}}^2 - U_{2,j}U_{3,i} \\ . \\ . \\ . \\ {U_{N-2,j}}^2 - U_{N-2,j}U_{N-1,i} \\ {U_{N-1,j}}^2 - U_{N-1,j}U_{N,i} \end{pmatrix} \tag{6}$$

Linear system (6) gives $(N-1)$ unknown pivotal values along the boundary $x=1$.
We have the following discretization for the second equation:

$$\begin{aligned} -rU_{i-1,j+1} &+ (2+2r)U_{i,j+1} - rU_{i+1,j+1} \\ &= rU_{i-1,j} + (2-2r)U_{i,j} \\ &\quad + rU_{i+1,j} + 2k(U_{i,j} - {U_{i,j}}^2),\ i=1,\dots,N-1,\ j=0,\dots,N-1, \end{aligned} \tag{7a}$$

$$U_{i,0} = f(ih), \qquad j=0,\ i=1,\dots,N-1, \tag{7b}$$

$$U_{0,j} = p(jk), \qquad i=0,\ j=0,1,\dots,N-1, \tag{7c}$$

$$U_{N,j} = q(jk), \qquad i=N,\ j=0,1,\dots,N-1, \tag{7d}$$

where $x_i = ih$, $t_j = jk$ and $r = k/h^2$.

Using equation (7), we obtain the following linear algebraic system of equations:

$$\begin{pmatrix} 2+2r & -r & 0 & 0 & 0 & 0 & 0 \\ -r & 2+2r & -r & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & -r & 2+2r & -r \\ 0 & 0 & 0 & 0 & 0 & -r & 2+2r \end{pmatrix}\begin{pmatrix} U_{1,j+1} \\ U_{2,j+1} \\ . \\ . \\ . \\ U_{N-2,j+1} \\ U_{N-1,j+1} \end{pmatrix}$$

$$= \begin{pmatrix} 2-2r & r & 0 & 0 & 0 & 0 & 0 \\ r & 2-2r & r & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & r & 2-2r & r \\ 0 & 0 & 0 & 0 & 0 & r & 2-2r \end{pmatrix}\begin{pmatrix} U_{1,j} \\ U_{2,j} \\ . \\ . \\ . \\ U_{N-2,j} \\ U_{N-1,j} \end{pmatrix}$$

$$+r\begin{pmatrix} U_{0,j}+U_{0,j+1} \\ 0 \\ . \\ . \\ . \\ 0 \\ U_{N,j}+U_{N,j+1} \end{pmatrix} + 2k\begin{pmatrix} U_{1,j}-{U_{1,j}}^2 \\ U_{2,j}-{U_{2,j}}^2 \\ . \\ . \\ . \\ U_{N-2,j}-{U_{N-2,j}}^2 \\ U_{N-1,j}-U^2_{N-1,j} \end{pmatrix} \tag{8}$$

Linear system (8) gives $(N-1)$ unknown pivotal values along the boundary $x=1$.
Finally, following discretization is described for the third equation:

$$\begin{aligned} -ra_jU_{i-1,j+1} + (2+2ra_j)U_{i,j+1} - ra_jU_{i+1,j+1} & \\ = ra_jU_{i-1,j} + (2-2ra_j)U_{i,j} & \\ + ra_jU_{i+1,j},\ i=1,\ldots,N-1,\ j=0,\ldots,N-1, & \qquad \text{(9a)} \\ U_{i,0} = f(ih), \qquad j=0,\ i=1,\ldots,N-1, & \qquad \text{(9b)} \\ U_{0,j} = p(jk), \qquad i=0,\ j=0,1,\ldots,N-1, & \qquad \text{(9c)} \\ U_{N,j} = q(jk), \qquad i=N,\ j=0,1,\ldots,N-1, & \qquad \text{(9d)} \\ a_j = a(jk), & \qquad \text{(9e)} \end{aligned}$$

where $x_i = ih$, $t_j = jk$ and $r = k/h^2$.

Using equation (9), we obtain the following linear algebraic system of equations:

$$\begin{pmatrix} 2+2ra_j & -ra_j & 0 & 0 & 0 & 0 & 0 \\ -ra_j & 2+2ra_j & -ra_j & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & -ra_j & 2+2ra_j & -ra_j \\ 0 & 0 & 0 & 0 & 0 & -ra_j & 2+2ra_j \end{pmatrix}\begin{pmatrix} U_{1,j+1} \\ U_{2,j+1} \\ . \\ . \\ . \\ U_{N-2,j+1} \\ U_{N-1,j+1} \end{pmatrix}$$

$$= \begin{pmatrix} 2-2ra_j & ra_j & 0 & 0 & 0 & 0 & 0 \\ ra_j & 2-2ra_j & ra_j & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & ra_j & 2-2ra_j & ra_j \\ 0 & 0 & 0 & 0 & 0 & ra_j & 2-2ra_j \end{pmatrix}\begin{pmatrix} U_{1,j} \\ U_{2,j} \\ . \\ . \\ . \\ U_{N-2,j} \\ U_{N-1,j} \end{pmatrix}$$

$$+ra_j \begin{pmatrix} U_{0,j}+U_{0,j+1} \\ 0 \\ . \\ . \\ . \\ 0 \\ U_{N,j}+U_{N,j+1} \end{pmatrix} \tag{10}$$

Linear system (10) gives $(N-1)$ unknown pivotal values along the boundary $x=1$.

Problems (2), (3), and (4) can be solved in least-square sense, and a cost function can be defined as a sum of squared differences between measured temperatures and calculated values of $U(x,t)$ by considering guesses estimated values of $q(t)$.

$$f(Guesses\quad estimated\quad values\quad of\quad q(t)) = \sum_{j=1}^{N}(U(a,t_j)-s_j)^2, \tag{11}$$

where $U(a,t_j)$ are calculated by solving the direct parabolic problem. To do this, we consider prior guess for $q(t)$. Also $s_j = s(t_j)$ are measured temperatures at $x=\alpha$. To find optimal solution of $q(t)$, the equation (11) must be minimum.

3 Genetic algorithm (GA) for solving NIPP

Genetic algorithms, primarily developed by Holland [5], 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 predefined fitness value has been reached [8].

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 [8].

In this study a GA is considered for solving IPP. Consider chromosomes $G_i = \{g_{i,1}, g_{i,2}, g_{i,3}, \ldots, g_{i,m}\},\ i = 1, 2, 3, \ldots, n$ and each $g_{i,j} \in [-M, M]$ (In this work M is one). Each chromosome estimates values of $q(t)$ at $t_j, j = 1, 2, 3, \ldots, n$. Solve then parabolic problem by expressed discretization in previous section. In this work, equation (11) is considered as fitness function than must be minimum. Finally for determining unknown $q(t)$, we find interpolation of m-points of the best chromosome at the end of algorithm. In this study an additional step is added to algorithm after step 4. We named this step repair operator. After applying crossover and mutation, some entries of chromosomes may exceed from $[-M, M]$; so repair operator returns those entries to the interval. The steps of GA for determining $q(t)$ can be divided into the following steps:

1. Generate randomly an initial population of chromosomes.
2. Evaluate the fitness of each chromosome in the population.
3. Choose by tournament selection pairs of chromosome for combination. By applying N-point crossover create offspring of this selected parents.
4. Apply bitwise mutation on offspring.
5. Apply repair operator.
6. Evaluate fitness of offspring.
7. Update population and copy the offspring by α probability.
8. Repeat step 3 to step 7, until finding acceptable fitness.

4 Nelder–Mead simplex search method for solving NIPP

This simplex search method, first proposed by Spendley, Hext, and Himsworth [14] and later refined by Nelder and Mead [9]. Their methods is one of the most efficient pattern search method currently available. This method is a derivative-free line search method that was particularly designed for traditional unconstrained minimization scenarios, such as the problems of nonlinear least squares, nonlinear simultaneous equations, and other types of function minimization [10]. In this method for N vertices of an initial simplex, evaluate cost function for each vertex at the first. Then the worth vertex replace by newly reflected and better point, which can be approximately located in the negative gradient direction. In the minimization problem with three initial simplex vertices, the method can be mention as follows [6, 17]:
x_h: Vertex with highest cost function value.
x_s: Vertex with the second highest cost function value.
x_l: Vertex with lowest cost function value.
x_c: The centroid of vertices except x_h.

1. Reflection. Reflect x_h (Figure1) and find x_0 such that

$$x_0 = 2x_c - x_h.$$

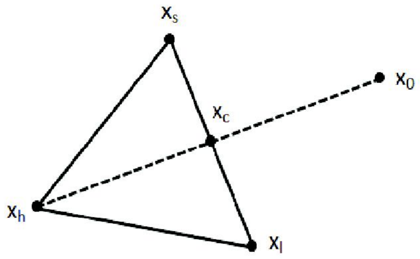

Figure 1: Reflection x_h toward x_0

2. If $f(x_l) < f(x_0) < f(x_s)$, replace x_h by x_0 and return to step 1.

3. Expansion. If $f(x_0) < f(x_l)$, then expansion operation makes x_{00} (Figure2). We replace x_h by x_0 or x_{00} depending on which function value is lower and return to step 1.

$$x_{00} = 2x_0 - x_c.$$

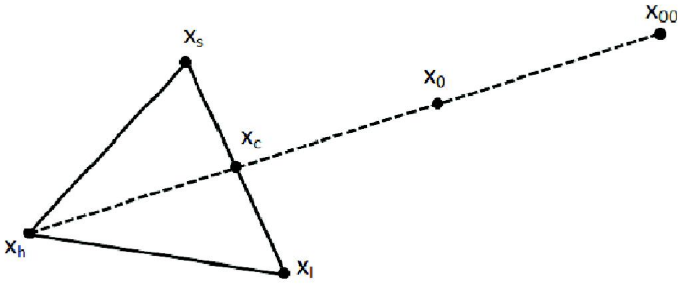

Figure 2: Expansion

4. Contraction. If $f(x_0) > f(x_s)$, then contraction operation makes x_{00} by consider two cases:

(a) If $f(x_0) < f(x_h)$, find x_{00} such that (Figure 3)

$$x_{00} = \frac{1}{2}x_0 - \frac{1}{2}x_c.$$

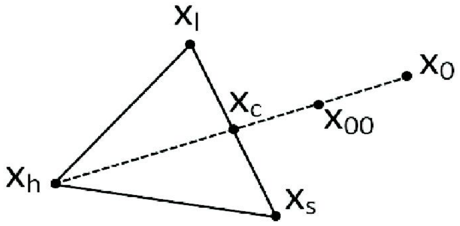

Figure 3: Contraction operator when $f(x_0) < f(x_h)$

(b) If $f(x_0) \geq f(x_h)$, find x_{00} such that (Figure 4)

$$x_{00} = \frac{1}{2}x_h - \frac{1}{2}x_c.$$

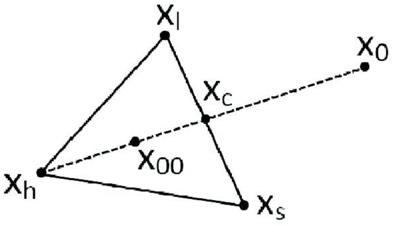

Figure 4: Contraction operator when $f(x_0) \geq f(x_h)$

(c) If $f(x_{00}) < f(x_h)$ and $f(x_{00}) < f(x_0)$, then replace x_h by x_{00} and return to step 1.

(d) If $f(x_{00}) \geq f(x_h)$ or $f(x_{00}) > f(x_0)$, then reduce size of simplex by halving distances from x_l and return to step 1.

The process terminates when either the number of iterations has exceeded a preset amount or the simplex size is smaller than a given value.

In this work, we consider the vectors, which are estimated $q(t)$, as the vertices of the simplex and equation (11) as the cost function. Initial vertices generate randomly. To find unknown $q(t)$, we interpolate the final vertex at the end of NM simplex search method.

5 hybrid algorithm for solving NIPP

GA and NM discussed separately; we now present a hybrid algorithm based on GA and NM. In this algorithm the population size is considered 10. In each iteration we sort the population by fitness of particles. Then top three particles are considered as vertices of NM and lead to NM subroutine, and other seven particles lead to GA subroutine. After applying these subroutines, all of particles are considered as entire of next iteration. The process terminates when pre certain number of iteration done [6]. In this algorithm each particle estimates unknown $q(t)$; also the initial population is generated randomly, and the cost function is considered equation (11). Finally for determining $q(t)$ we find interpolation of m-points of the best particle at the end of algorithm. The steps of hybrid algorithm for determining unknown $q(t)$ can be divided into the following steps:

1. Generate randomly a 10 dimension initial population of particles.

2. Sort population by fitness of particles.

3. Top three particles are lead to NM subroutine.

4. Next seven particles is lead to GA subroutine.

5. Repeat step 2 to step 4, until terminate criteria is not satisfied.

6 Convergence study

The convergence of the Genetic Algorithm and the Nelder–Mead simplex search method have been studied in [13, 15]. Since our presented hybrid method in Section 5 uses the Nelder–Mead simplex search method as a inner subroutine in the Genetic Algorithm, the general convergence is depend on the convergence of the genetic algorithm. Therefore, The hybrid method used in this work converges to optimal solution.

7 Numerical results

In this section we are going to demonstrate numerically, some results for the unknown function $q(t)$ in the NIPPs of Section 2. The aim of this section is to show the applicability of the presented hybrid algorithm for solving NIPPs. To show utility of this algorithm, we solved three examples using GA, NM, and hybrid algorithm separately and compared them with each other. Now, we give the following examples in $0 < x < 1, 0 < t < 1$.

Example 1.

$$\begin{aligned}
&U_t(x,t) + U(x,t)U_x(x,t) = U_{xx}(x,t), \quad 0 < x < 1, \quad 0 < t < t_M, && \text{(12a)}\\
&U(x,0) = \frac{1}{2} - \frac{1}{2}\tanh(\frac{x}{4}), \quad 0 \le x \le 1, && \text{(12b)}\\
&U(0,t) = \frac{1}{2} - \frac{1}{2}\tanh(-\frac{t}{8}), \quad 0 \le t \le t_M, &&\\
&U(1,t) = q(t), \quad 0 \le t \le t_M, &&
\end{aligned}$$

and the overspecified condition

$$s(t_j) = U(0.5, t_j), \quad t_j = 0.05 \times j, \quad j = 0, 1, 2, \ldots, 20.$$

Here the exact $U(x,t)$ and $q(t)$ are $(\frac{1}{2} - \frac{1}{2}\tanh(\frac{1}{4}(x - \frac{t}{2})))$ and $(\frac{1}{2} - \frac{1}{2}\tanh(\frac{1}{4}(1 - \frac{t}{2})))$, respectively.

Example 2.

$$U_t(x,t) - U(x,t)(1 - U(x,t)) = U_{xx}(x,t), \quad 0 < x < 1, \quad 0 < t < t_M, \tag{13a}$$

$$\begin{aligned}
&U(x,0) = \frac{1}{4}(1 - \tanh(\frac{1}{2\sqrt{6}}x))^2, \quad 0 \le x \le 1,\\
&U(0,t) = \frac{1}{4}(1 - \tanh(-\frac{5t}{24}))^2, \quad 0 \le t \le t_M,\\
&U(1,t) = q(t), \quad 0 \le t \le t_M,
\end{aligned}$$

and the overspecified condition

$$s(t_j) = U(0.5, t_j), \quad t_j = 0.05 \times j, \quad j = 0, 1, 2, \ldots, 20.$$

Here the exact $U(x,t)$ and $q(t)$ are $(\frac{1}{4}(1 - \tanh(\frac{1}{2\sqrt{6}}(x - \frac{5t}{2\sqrt{6}})))^2)$ and $(\frac{1}{4}(1 - \tanh(\frac{1}{2\sqrt{6}}(1 - \frac{5t}{2\sqrt{6}})))^2)$, respectively.

Example 3.

$$\begin{aligned}
&U_t(x,t) = \frac{2t}{t^2+1}U_{xx}(x,t), \quad 0 < x < 1, \quad 0 < t < t_M, && \text{(14a)}\\
&U(x,0) = \frac{1}{3}e^{-x}, \quad 0 \le x \le 1, && \text{(14b)}\\
&U(0,t) = \frac{t^2+1}{3}, \quad 0 \le t \le t_M, && \text{(14c)}\\
&U(1,t) = q(t), \quad 0 \le t \le t_M, && \text{(14d)}
\end{aligned}$$

and the overspecified condition

$$s(t_j) = U(0.9, t_j) + \sigma R, \quad j = 1, 2, 3, \cdots, 9, \tag{14e}$$

where t_j's are the sinc times nodes.

$$\tag{14f}$$

Here the exact $U(x,t)$ and $q(t)$ are $e^{-x}(\frac{t^2+1}{3})$ and $e^{-1}(\frac{t^2+1}{3})$, respectively.

Remark 2. In a NIPP there are two sources of error in the estimation. The first source is the unavoidable bias deviation (or deterministic error). The second source of error is the variance due to the amplification of measurement errors (stochastic error). The global effect of deterministic and stochastic errors are considered in terms of the mean squared error or total error, [2].

$$S = \Big[\frac{1}{N-1}\sum_{j=1}^{N}(\widehat{q_j} - q_j)^2\Big]^{\frac{1}{2}}, \tag{15}$$

where N is the total number of estimated values, $\widehat{q_i}$ are calculated values from interpolated equation, and q_i are exact values of $q(t)$. We use from value of total error to compare methods.

7.1 Solving examples by genetic algorithm

In this subsection, at first, we use genetic algorithm for solving examples. Table 1 presents parameters of the proposed genetic algorithm.

In this algorithm, a population of 10 chromosomes of 20 genes is used as the initial guess for numerical results. The gens estimate 20 values of $q(t_j)$ at $t_j = j \times 0.05$, $j = 1, 2, 3, \ldots, 20$. To determine unknown $q(t)$, the best chromosome is interpolated at the end of algorithm. Tables 2–4 present results of implementation of the proposed genetic algorithm to determine unknown $q(t)$ at Examples 1–3, respectively.

Table 1: Parameters of the proposed genetic algorithm

Representation	Real valued vectors
Length of chromosomes	20
Recombination	N point crossover
Recombination probability	100%
Mutation	Swap
Mutation probability	$1/n$
Parent selection	Best of 2 out of random 4
Survivor selection	Replace random
Population size	10
Number of offspring	1
Initialization	Random.
Termination condition	Number of generation

Table 2: The results of 100 to 1000000 generations for determining $q(t)$ at Example 1 by implementing proposed genetic algorithm for a population of 10 chromosomes of 20 genes.

Generation	Best fitness	Time (s)	S
100	0.027745	0.16	0.03165
1000	0.003935	1.01	0.00946
10000	0.000901	5.76	0.00685
100000	0.000097	58.42	0.00279
1000000	0.000051	563.14	0.00047

Table 3: The results of 100 to 1000000 generations for determining $q(t)$ at Example 2 by implementing proposed genetic algorithm for a population of 10 chromosomes of 20 genes.

Generation	Best fitness	Time (s)	S
100	0.014947	0.51	0.29677
1000	0.009230	1.75	0.05083
10000	0.003019	7.19	0.00917
100000	0.000223	63.41	0.00231
1000000	0.000076	688.60	0.00048

Table 4: The results of 100 to 1000000 generations for determining $q(t)$ at Example 3 by implementing proposed genetic algorithm for a population of 10 chromosomes of 20 genes

Generation	Best fitness	Time (s)	S
100	0.010579	0.62	0.01337
1000	0.002323	1.36	0.00915
10000	0.000530	8.78	0.00411
100000	0.000021	59.46	0.00175
1000000	0.000018	480.67	0.00014

7.2 Solving examples by Nelder–Mead simplex search method

To solve presented examples by NM simplex search method, three initial vertices were generated randomly. Each vertex is considered a real valued vector by 20 entries. So each vertex estimates unknown $q(t)$, and each entry of vertices estimates $q(t_j)$ at $t_j = j \times 0.05$,$j = 1, 2, 3, \ldots, 20$. To determine unknown $q(t)$, the best vertex is interpolated at the end of algorithm. Tables 6–7 present results of implementation of NM for determine unknown $q(t)$ at Examples 1–3, respectively.

Table 5: The results of 100 to 1000000 iteration for determining $q(t)$ at Example 1 by implementing NM for three vertices

Iteration	Best fitness	Time (s)	S
100	0.545731	1.07	0.25919
1000	0.526974	7.32	0.24156
10000	0.073689	23.87	0.08513
100000	0.041615	184.09	0.06720
1000000	0.036581	1942.50	0.06669

Table 6: The results of 100 to 1000000 iteration for determining $q(t)$ at Example 2 by implementing NM for three vertices

Iteration	Best fitness	Time (s)	S
100	0.156634	0.89	0.20997
1000	0.147336	2.87	0.13720
10000	0.099725	17.90	0.08711
100000	0.054095	153.17	0.08124
1000000	0.017139	1626.01	0.02814

Table 7: The results of 100 to 1000000 iteration for determining $q(t)$ at Example 3 by implementing NM for three vertices

Iteration	Best fitness	Time (s)	S
100	0.115466	0.73	0.19575
1000	0.109927	1.98	0.12761
10000	0.080606	19.36	0.07540
100000	0.044853	113.71	0.07341
1000000	0.012802	1094.75	0.01937

7.3 Solving examples by hybrid algorithm

In this subsection, Examples 1–3 are solved by proposed hybrid algorithm in Section 5. In this algorithm, a population of 10 vectors of 20 entries is used as the initial guess for numerical results. Therefore, each vector estimates unknown $q(t)$ and each entry of vectors estimates $q(t_j)$ at $t_j = j \times 0.05$,$j = 1, 2, 3, \ldots, 20$. At the each iteration of hybrid algorithm, population is sorted by fitness. Then NM-subroutine is run for 10 iteration on top of three individuals of population and GA-subroutine applies genetic operations (recombination and mutation) on next seven individuals. To determine unknown $q(t)$, the best vertex is interpolated at the end of algorithm. Tables 8–10 present results of implementation of hybrid algorithm for determine unknown $q(t)$ at Examples 1–3, respectively. Figures 5–7 present exact and numeric $q(t)$ for 1000 iterations at Examples 1–3, respectively.

Table 8: The results of 100 to 1000 iteration for determining $q(t)$ at Example 1 by implementing hybrid algorithm for ten vertices

hybrid iterations	Best fitness	Time (s)	S
100	0.000077	49.51	0.00126
1000	0.000039	461.29	0.00041

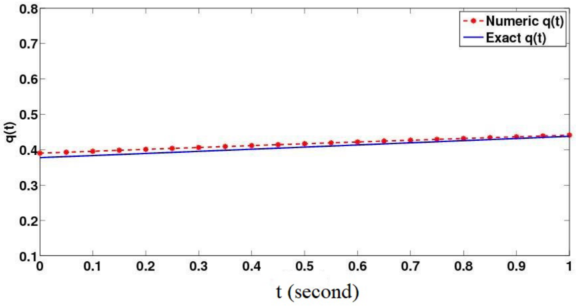

Figure 5: Exact and numeric $q(t)$ for 1000 iterations by implementing hybrid algorithm at Example 1

Table 9: The results of 100 to 1000 iteration for determining $q(t)$ at Example 2 by implementing hybrid algorithm for ten vertices

Hybrid iterations	Best fitness	Time (s)	S
100	0.000105	41.78	0.00212
1000	0.000028	436.22	0.00038

Table 10: The results of 100 to 1000 iteration for determining $q(t)$ at Example 3 by implementing hybrid algorithm for ten vertices

Hybrid iterations	Best fitness	Time (s)	S
100	0.000063	37.19	0.00092
1000	0.000016	389.83	0.00038

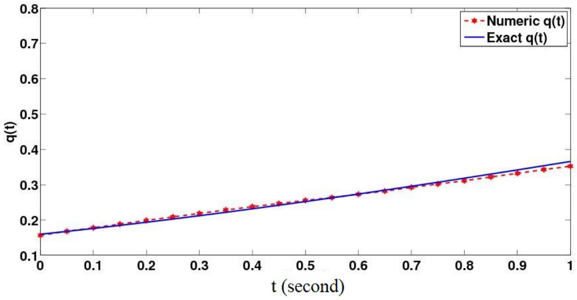

Figure 6: Exact and numeric $q(t)$ for 1000 iterations by implementing hybrid algorithm at Example 2

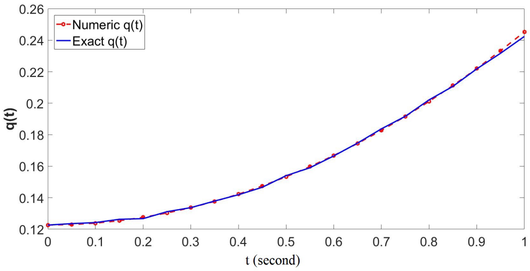

Figure 7: Exact and numeric $q(t)$ for 1000 iterations by implementing hybrid algorithm at Example 3

7.4 Comparison

In this paper, the genetic algorithms, Nelder–Mead simplex search method and hybrid algorithm have explained for solving the nonlinear inverse parabolic problem by some numerical examples. To examine the utility of solutions, we compare the values of total error(s) and their execute times in the estimated solutions. The unknown $q(t)$ has estimated by the some different numbers of iterations for each method. Results show that the accuracy of GA is better than NM for 100 to 1000000 iterations. Figures 8–10 show the convergency study for different iterations for GA and NM at first and second example, respectively. Also execute time of GA is less than NM at both examples in all iterations. But when GA and NM have combined and hybrid algorithm has created, exploitation of NM and exploration of GA caused that accuracy improve. As total error of hybrid algorithm for 100 and 1000 iterations became better in comparison with the NM and GA performance with almost the same execute time.

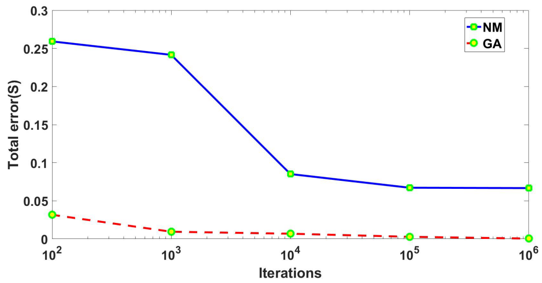

Figure 8: Values of total error(S) for different numbers of iterations by implementing NM and GA at Example 1

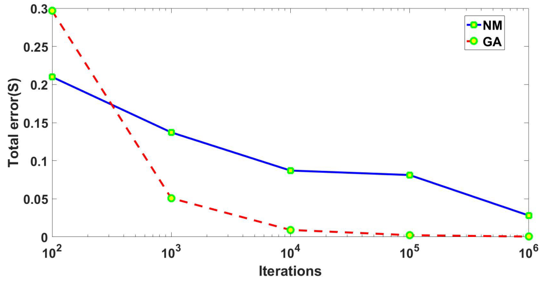

Figure 9: Values of total error(s) for different numbers of iterations by implementing NM and GA at Example 2

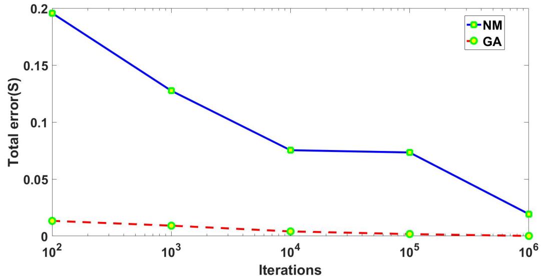

Figure 10: Values of total error(s) for different numbers of iterations by implementing NM and GA at Example3

8 Conclusion

A numerical method to estimate unknown boundary condition is proposed for these kinds of NIPPs, and the following results are obtained:

1. The present study successfully applies the numerical method to NIPPs.
2. To solve the NIPPs by GA, NM, and hybrid algorithm, the unknown function will be guessed and we do not need the regularization. This will improve the execution time.
3. This hybrid algorithm is able to combine whit every direct solution methods.
4. This method does not need to powerful mathematic base.
5. Acceptable accuracy and execute time at the hybrid algorithm.

References

1. Babolian, E. and Saeidian, J. *Analytic approximate solutions to Burgers, Fisher, Huxley equations and two combined forms of these equations*, Commun. Nonlinear Sci. Numer. Simulat., 14 (2009), 1984–1992.

2. Cabeza, J.M.G., Garcia, J.A.M., and Rodriguez, A.C. *A Sequential Algorithm of Inverse Heat Conduction Problems Using Singular Value Decomposition*, Int. J. Thermal Sci., 44 (2005), 235–244.

3. Cantú-Paz, E. *A Summary of Research on Parallel Genetic Algorithms*, IlliGAL Report No 95007, University of Illinois, 1995.

4. Chiwiacowsky, L.D., de Campos Velho, H.F., Preto, A.J., and Stephany, S. *Identifying initial condition in heat conduction transfer by a genetic algorithm: a parallel approach*, in: 24th Iberian Latin American Congress on Computational Methods in Engineering, Ouro Preto, Brazil, 2003.

5. Holland, J.H. *Adaptation in Natural and Artificial System*, University of Michigan Press, Ann Arbor, 1975.

6. Katari, V., Malireddi, S., and Srujan Kollisetty, V.N.K., *An efficient hybrid algorithm for data clustering using improved genetic algorithm and Nelder Mead simplex search*, International Conference on Computational Intelligence and Multimedia Applications, 2007.

7. Levine, D. *A parallel genetic algorithm for the set partitioning problems*, Mathematics and Computer Science Division, Argone National Laboratory Rept., University of Illinois, 1994.

8. Liu, F.-B. *A modified genetic algorithm for solving the inverse heat transfer problem of estimating plan heat source*, Int. J. Heat and Mass Transfer, 51 (2008), 3745–3752.

9. Nelder, J. A. and Mead, R. *A simplex method for function minimization*, Comput. J. 7 (1965), no. 4, 308–313.

10. Olsson, D. M. and Nelson, L. S. *The Nelder–Mead simplex procedure for function minimization*, Technometrics, 7 (1965) 308–313.

11. Pourgholi, R., Tavallaei, N., and Foadian, S. *application of Haar basis method for solving some ill-posed inverse problems*, J. Math. Chem., 8(50) (2012), 2317–2337.

12. Shahrezaee, A. M. and Hoseini nia, M. *Solution of Some Parabolic Inverse Problems by Adomian Decomposition Method*, Appl. Math. Sci., 5(50) (2011), 3949–3958.

13. Sharapov R. R. and Lapshin A. V. *Convergence of Genetic Algorithms*, Pattern Recognition and Image Analysis, 16(3) (2006), 392–397.

14. Spendley, W., Hext, G. R., and Himsworth, F. R. *Sequential application of simplex designs in optimization and evolutionary operation.* Technometrics, 4(4) (1962), 441–461.

15. Mckinnon K. I. M. *Convergence of the Nelder–Mead simplex method to a nonstationary point*, SIAM J. Optim., 9(1) (1998), 148–158.

16. Powell, M.J.D. *On Search Directions for Minimization Algorithms*, Mathematical Programming, 4(2) (1973), 193–201.

17. Walsh, G. R. *An Introduction to Linear Programming*, ISBN 0-471-90719-7 (Wiley), 2nd edition, 1985.

یک الگوریتم ترکیبی کارا بر اساس الگوریتم ژنتیک و روش جستجوی نلدر- مید برای حل مسائل معکوس سهموی غیرخطی

حسن دانا مزرعه و رضا پورقلی

دانشگاه دامغان، دانشکده ریاضی و علوم کامپیوتر

دریافت مقاله ۱۵ اردیبهشت ۱۳۹۶، دریافت مقاله اصلاح شده ۴ آذر ۱۳۹۶، پذیرش مقاله ۲۳ خرداد ۱۳۹۷

چکیده : در این مقاله، یک الگوریتم ترکیبی بر اساس الگوریتم ژنتیک و روش جستجوی سیمپلکس نلدر- مید با روش کمترین مربعات برای تعیین درجه حرارت در مسایل سهموی معکوس غیر خطی ترکیب می‌شود. کارایی الگوریتم ترکیبی با چند مثال از مسائل معکوس غیرخطی سهموی مورد تایید قرار می گیرد. نتایج نشان می‌دهد که این روش ترکیبی، بهتر از الگوریتم ژنتیک و روش جستجوی سیمپلکس نلدر- مید به طور جداگانه است. نتایج عددی با پیاده سازی الگوریتم های مطرح شده در یک پردازنده تک هسته ای با سرعت ۲,۲۰ GHz بدست آمده است.

کلمات کلیدی : ترکیبی؛ NIPP ؛ مسائل معکوس غیرخطی سهموی؛ الگوریتم ژنتیک؛ نلدر-مید؛ روش کمترین مربعات.