Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080617

An Optimization Algorithm for Solving Systems of Singular Boundary Value Problems

Zaer Abo-Hammour[1], *Omar Abu Arqub*[2], *Othman Alsmadi*[3], *Shaher Momani*[4,5,*] *and Ahmed Alsaedi*[5]

[1] Department of Mechatronics Engineering, Faculty of Engineering, The University of Jordan, Amman 11942, Jordan
[2] Department of Mathematics, Faculty of Science, Al Balqa Applied University, Salt 19117, Jordan
[3] Department of Electrical Engineering, Faculty of Engineering, The University of Jordan, Amman 11942, Jordan
[4] Department of Mathematics, Faculty of Science, The University of Jordan, Amman 11942, Jordan
[5] Nonlinear Analysis and Applied Mathematics (NAAM) Research Group, Faculty of Science, King Abdulaziz University (KAU), Jeddah 21589, Kingdom of Saudi Arabia

Received: 18 Oct. 2013, Revised: 16 Jan. 2014, Accepted: 17 Jan. 2014
Published online: 1 Nov. 2014

Abstract: In this paper, an optimization algorithm is presented for solving systems of singular boundary value problems. In this technique, the system is formulated as an optimization problem by the direct minimization of the overall individual residual error subject to the given constraints boundary conditions, and is then solved using continuous genetic algorithm in the sense that each of the derivatives is replaced by an appropriate difference quotient approximation. Two numerical experiments are carried out to verify the mathematical results, and the theoretical statements for the solutions are supported by the results of numerical experiments. Meanwhile, the statistical analysis is provided in order to capture the behavior of the solutions and to discover the effect of system parameters on the convergence speed of the algorithm. The numerical results demonstrate that the algorithm is quite accurate and efficient for solving systems of singular boundary value problems.

Keywords: Genetic algorithm; Singular problem; Optimization problem; Optimization technique

1 Introduction

[1]In mathematics, in the field of differential equations, a system of singular boundary value problems (BVPs) is a system of singular differential equations together with a set of additional restraints, called the boundary conditions. A solution to such systems is a solution to the given system of singular differential equations which also satisfies the given boundary conditions. Systems of singularly BVPs appear in several branches of applied mathematics, theoretical physics, engineering, and control and optimization theory [1,2,3,4,5]. Generally speaking, most systems of singular BVPs cannot be solved exactly by using the well-known analytical methods and relatively few of those can be solved in closed form by the standard mathematical tricks. So, it's natural to begin thinking about what can be said about such systems in the absence of solutions. In such situations, one has to resort to numerical methods to obtain approximate solutions to some number of decimal points by a computer by computer algorithmic approximation techniques.

Informally, an algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into the output. We can also view an algorithm as a tool for solving a well-specified computational problem. The statement of the problem specifies in general terms the desired input/output relationship. The algorithm describes a specific computational procedure for achieving that input/output relationship.

Numerical optimization is an important tool in decision science and in the analysis of physical and engineering systems. It is well known that optimization and nonlinear analysis are two branches of modern mathematics much developed lately. An important step in

[1] This paper is dedicated to the great Iraqi scientist Samir Hadid on the occasion of his 65th birthday.

* Corresponding author e-mail: s.momani@ju.edu.jo

© 2014 NSP
Natural Sciences Publishing Cor.

optimization is the identification of some objective, i.e., a quantitative measure of the performance of the system. This objective can be any quantity or combination of quantities that can be represented by a single number. The objective depends on certain characteristics of the system, called parameters, which are often restricted or constrained in some way. Furthermore, the parameters can have either continuous or discrete values [6].

In this paper, we utilize a methodical way based on the use of continuous genetic algorithm (GA) for numerically approximating a solution of systems of singular BVPs in which the given boundary conditions can be involved. The new method has the following characteristics; first, it should not require any modification while switching from the linear to the nonlinear case; as a result, it is of versatile nature. Second, it should not resort to more advanced mathematical tools; that is, the algorithm should be simple to understand, implement, and should be thus easily accepted in the mathematical and engineering application's fields. Third, the algorithm is of global nature in terms of the solutions obtained as well as its ability to solve other mathematical and engineering problems. Fourth, the present algorithm can avoid any singularities in the computations process with less effort and less computation cost.

The rest of the paper is synthesized as follows. In Section 2, a short introduction to optimization problems and their applications is presented. In Section 3, we formulate the system of singular BVPs as an optimization problem. Section 4 shows a brief preface to optimization techniques. Section 5 covers the description of GA in detail. Section 6 utilizes and evaluates the continuous GA as an efficient evolutionary algorithm for solving system of singular BVPs. Software libraries and numerical inputs are given in Section 7 in order to verify the mathematical simulation of the proposed algorithm. In Section 8, we report our numerical finding and demonstrate the accuracy of the proposed scheme by considering two numerical examples. Statistical analysis is supported by the results of numerical experiments in Section 9. Finally, in Section 10 some concluding remarks are presented.

2 Optimization problems

Optimization problems are common in many disciplines and various domains [7,8,9,10,11,12,13,14]. In optimization problems, we have to find solutions which are optimal or near-optimal with respect to some goals. Usually, we are not able to solve problems in one step, but we follow some process which guides us through problem solving. Often, the solution process is separated into different steps which are executed one after the other. Commonly used steps are recognizing and defining problems, constructing and solving models, and evaluating and implementing solutions [15].

In general, optimization problems have the following characteristics [15]; first, different decision alternatives are available. Second, additional constraints limit the number of available decision alternatives. Third, each decision alternative can have a different effect on the evaluation criteria. Fourth, an evaluation function defined on the decision alternatives describes the effect of the different decision alternatives. On the other aspect as well, optimization problems can be divided into several categories depending on whether they are continuous or discrete, constrained or unconstrained, single-objective or multi-objective, static or dynamic [16]. In order to find satisfactory solutions for these problems, metaheuristics can be used. A metaheuristic is an algorithm designed to solve approximately a wide range of hard optimization problems without having to deeply adapt to each problem. Almost all metaheuristics share the following characteristics [16]: first, they are nature-inspired (based on some principles from physics, biology, or ethology). Second, they make use of stochastic components (involving random variables). Third, they do not use the gradient or Hessian matrix of the objective function. Fourth, they have several parameters that need to be fitted to the problem at hand.

In mathematics, information science, and decision theory, optimization is the selection of the best solution within certain given domain, which can minimize or maximize a function. As yet, different approaches are developed to deal with these optimization problems. Classification of these approaches can be accomplished in many ways. However, in most cases, according to their nature, these approaches can primarily be classified as two groups: classical methods and stochastic algorithms. Classical methods have a fixed move. For the same initial input values, they follow the same path and eventually find the same final solutions. However, stochastic algorithms are based on randomization, and the final solutions will be dissimilar each time even starting from the identical initial values. Despite these differences, in most of the cases, though slightly dissimilar, finally these two sorts of algorithms will find the similar optimal values. The immersion of nature-inspired metaheuristic algorithms as a benediction from the statistical and artificial intelligence theory has opened up a novel aspect with the aim of fulfilling function optimization. To a degree, all stochastic methods attempt to make trade-off between exploitation and exploration [17].

3 Problem formulation

Mathematically speaking, optimization is minimization or maximization of a function subject to constraints on its variables. In this section, system of singular BVPs is first transformed into discrete version in order to formulate it as an optimization problem based on the minimization of the cumulative residual of all unknown interior nodes.

© 2014 NSP
Natural Sciences Publishing Cor.

The aim of this paper is to apply the continuous GA for solving systems of singular ordinary differential equations subject to given boundary conditions. More specifically, we consider the following system:

$$\begin{aligned} &y_1''(x)+\frac{a_1(x)}{p_1(x)}G_1(x,y_1'(x),y_2'(x)) \\ &\quad+\frac{a_2(x)}{p_2(x)}G_2(x,y_1(x),y_2(x))+\frac{a_3(x)}{p_3(x)}=0, \\ &y_2''(x)+\frac{b_1(x)}{q_1(x)}H_1(x,y_1'(x),y_2'(x)) \\ &\quad+\frac{b_2(x)}{q_2(x)}H_2(x,y_1(x),y_2(x))+\frac{b_3(x)}{q_3(x)}=0, \end{aligned} \tag{1}$$

subject to the boundary conditions

$$\begin{aligned} y_1(0)=\alpha_1, y_1(1)=\beta_1, \\ y_2(0)=\alpha_2, y_2(1)=\beta_2, \end{aligned} \tag{2}$$

where $x\in(0,1)$, α_k,β_k are real finite constants, and G_k,H_k are linear or nonlinear real-valued functions, where $k=1,2$.

Remark 1: The two functions $p_i(x),q_i(x)$ may take the values $p_i(0)=q_i(0)=0$ or $p_i(1)=q_i(1)=0$ which make Eq. (1) to be singular at $x=0$ or $x=1$, while $a_i(x),a_i(x)$ are continuous real-valued functions on $[0,1]$, where $i=1,2,3$.

Remark 2: Throughout this work, we assume that Eq. (1) subject to boundary conditions (2) has a unique two solutions on $[0,1]$ (one solution for each dependent variable).

Remark 3: The term "continuous" is used to emphasize that the continuous nature of the optimization problem and the continuity of the resulting solution curves.

For the first step of formulation, the independent interval $[0,1]$ is partitioned into N subintervals of equal length h given as $h=1/N$. The mesh points, nodes, are obtained using the equation $x_i=ih$, $i=0,1,\ldots,N$. Thus, at the interior mesh points, x_i, $i=1,2,\ldots,N-1$, the system to be approximated is given as:

$$\begin{aligned} &y_1''(x_i)+\frac{a_1(x_i)}{p_1(x_i)}G_1(x_i,y'(x_i)) \\ &\quad+\frac{a_2(x_i)}{p_2(x_i)}G_2(x_i,y(x_i))+\frac{a_3(x_i)}{p_3(x_i)}=0, \\ &y_2''(x_i)+\frac{b_1(x_i)}{q_1(x_i)}H_1(x_i,y'(x_i)) \\ &\quad+\frac{b_2(x_i)}{q_2(x_i)}H_2(x_i,y(x_i))+\frac{b_3(x_i)}{q_3(x_i)}=0, \end{aligned} \tag{3}$$

subject to the boundary conditions

$$y(x_0)=\alpha, y(x_N)=\beta,$$

where $y=(y_1,y_2)$, $\alpha=(\alpha_1,\alpha_2)$, $\beta=(\beta_1,\beta_2)$, and $x_1\le x_i\le x_{N-1}$, where $i=1,2,\ldots,N-1$.

The finite difference approximation for derivatives is one of the very effective methods used for solving the differential equations numerically. It involves replacing the derivatives appearing in the differential equation by suitable finite difference approximations. In fact, the accuracy of the solution depends upon the number of mesh points chosen. However, by increasing the number of mesh points one can increase the accuracy of the solution to a desire degree. In this paper, we will employ this technique to approximate the solutions of Eqs. (1) and (2) numerically using continuous GA. Anyhow, the difference approximation formulas, which closely approximate $y_k'(x_i)$ and $y_k''(x_i)$, where $k=1,2$ and $i=1,2,\ldots,N-1$ using $(n+1)$-point at the interior mesh points with error of order $O\left(h^{n-m+1}\right)$, where $n=2,3,\ldots,N$ and $m=1,2$ is the order of the derivative can be easily obtained by using Algorithm (6.1) in [18]. We mention here that the number n is starting from 2 and gradually increases up to N.

To complete the formulation substituting the approximate formulas of $y_k'(x_i)$ and $y_k''(x_i)$ in Eq. (3), a discretized form of Eqs. (1) and (2) is obtained. The resulting algebraic equations will be a discrete function of $x_i, y_k\left(x_{i-(n-1)}\right), y_k\left(x_{i-(n-2)}\right)$, ..., and $y_k\left(x_{i+(n-1)}\right)$, where $k=1,2$. After that, it is necessary to rewrite the discretized Eq. (3) in the form of the following:

$$\begin{aligned} &F_1\left(x_i,y\left(x_{i-(n-1)}\right),y\left(x_{i-(n-2)}\right),\ldots,y\left(x_{i+(n-1)}\right)\right) \\ &\quad+\frac{a_2(x_i)}{p_2(x_i)}G_2(x_i,y(x_i))+\frac{a_3(x_i)}{p_3(x_i)}\approx 0, \\ &F_2\left(x_i,y\left(x_{i-(n-1)}\right),y\left(x_{i-(n-2)}\right),\ldots,y\left(x_{i+(n-1)}\right)\right) \\ &\quad+\frac{b_2(x_i)}{q_2(x_i)}H_2(x_i,y(x_i))+\frac{b_3(x_i)}{q_3(x_i)}\approx 0, \end{aligned}$$

where F_1,F_2 are given as

$$\begin{aligned} F_1(x,y'(x),y_1''(x))&=y_1''(x)+\frac{a_1(x_i)}{p_1(x_i)}G_1(x,y'(x)), \\ F_2(x,y'(x),y_2''(x))&=y_2''(x)+\frac{b_1(x_i)}{q_1(x_i)}H_1(x,y'(x)). \end{aligned}$$

The conventional design procedures aim at finding an acceptable or adequate design that merely satisfies the functional and other requirements of the problem. In general, there will be more than one acceptable design, and the purpose of optimization is to choose the best one of the many acceptable designs available. Thus a criterion has to be chosen for comparing the different alternative acceptable designs and for selecting the best one. The criterion with respect to which the design is optimized, when expressed as a function of the design variables, is known as the fitness function. The choice of fitness function is governed by the nature of problem and the selection of this function can be one of the most

© 2014 NSP
Natural Sciences Publishing Cor.

important decisions in the whole optimum design process. Next, we define the fitness function related to Eqs. (1) and (2). But firstly, we present the following definitions.

Definition 1: The residual of the general interior node, denoted by Res, is defined as:

$$\begin{aligned}&\mathrm{Res}_1(i)\\&= F_1(x_i, y(x_{i-(n-1)}), y(x_{i-(n-2)}), \ldots, y(x_{i+(n-1)}))\\&\quad + \frac{a_2(x_i)}{p_2(x_i)} G_2(x_i, y(x_i)) + \frac{a_3(x_i)}{p_3(x_i)},\\&\mathrm{Res}_2(i)\\&= F_2(x_i, y(x_{i-(n-1)}), y(x_{i-(n-2)}), \ldots, y(x_{i+(n-1)}))\\&\quad + \frac{b_2(x_i)}{q_2(x_i)} H_2(x_i, y(x_i)) + \frac{b_3(x_i)}{q_3(x_i)}.\end{aligned} \tag{4}$$

Definition 2: The overall individual residual, Oir, is a function of the residuals of all interior nodes. It may be stated as

$$\mathrm{Oir} = \sqrt{\sum_{i=1}^{N-1} \left(\mathrm{Res}_1^2(i) + \mathrm{Res}_2^2(i)\right)}.$$

Definition 3: The fitness function, denoted by Fit, is defined as:

$$\mathrm{Fit} = \frac{1}{1+\mathrm{Oir}}.$$

The fitness function plays a fundamental rule in optimization techniques (continuous and discrete) and their applications. This function is required in our work in order to convert the minimization problem of Oir into a maximization problem of Fit. In fact, the value of individual fitness is improved if a decrease in the value of the Oir is achieved. On the other hand, the optimal solution of the problem, nodal values, will be achieved when Oir approaches zero and thus Fit approaches unity.

4 Techniques for optimization

There are a few common techniques which are common to both single-objective and multi-objective optimization problems. However there are some advanced techniques which are applied to multi-objective optimization problems as these problems contain multi-dimensional objectives to be satisfied.

In general, the different optimization techniques can be broadly classified into the following three categories [19]: first, calculus-based techniques or numerical methods. Second, enumerative techniques. Third, random techniques.

Calculus methods, also known as numerical methods use a set of necessary and sufficient conditions which must be satisfied by the solution of the optimization problem. Numerical methods further divided into direct and indirect methods. Direct search methods deals with hill climbing in the function space by moving in local gradient direction, while in indirect methods the gradient of the objective function is set to zero and thus solution is get by solving these set of equations. All the calculus based methods assume strictly the existence of derivatives and are local in scope too. These constrains limit their application in real-world problems; however in small class of unimodal problems these can be efficiently used.

Enumerative techniques tends to evaluate each and every point of the finite, or discrete infinite, search space to sought optimal solution. A well-known example of enumerative search technique is dynamic programming. Thus, in order to search each and every point enumerative needs to break down the problems even of moderate size and complexity into smaller divisions.

Guided random search techniques are based on the concept of enumerative methods only but with the use of additional information about the search space in order to seek the potential regions faster. Guided is further categorized into single-point and multi-point search, means whether it is searching just with one point or with several points at a given time. For single-point search technique, simulated annealing is widely used. It uses thermodynamic evolution in order to find states of minimum energy. For multi-point search, where random choice is used as a tool to guide through a highly explorative search space, GA is in trend. They are basically used assuming that a near-optimal solution will be accepted; given the search space is huge, noisy, multi-modal as well as discontinuous.

5 Overview of genetic algorithm

Modern metaheuristics like GA is easy-to-apply optimization concepts that are widely used for fully-automated decision making and problem solving. In this section, a summary of GA approach and its characteristics to the design optimization problem are presented by a brief introduction.

GA is efficient, self-adaptable, self-repairable, and robust, nature-inspired search and optimization tool. GA performs well in large, complex and multi-modal search space. GA are modelled based upon the natural genetic principles where the potential solution is encoded in structures known as chromosomes. These make use of problem or domain dependent knowledge to search potential and promising areas in search space. Each individual or chromosome has a fitness value associated with it, which describes its goodness compared to other individuals in the current population with respect to the solution. The genetic operators such as selection, crossover, and mutation are also inspired by the nature and are applied to chromosomes in order to yield better and potential solutions. GA is adaptive computational tools modelled on the mechanics of nature. These efficiently exploit historical information to guess newly upcoming offspring with improved performance. GA is

© 2014 NSP
Natural Sciences Publishing Cor.

metaheuristics search methods means it estimates the solution, which can be used for both solving problems and modelling evolutionary systems. GA is preferred when the search space is huge, discontinuous, multi-dimensional, multi-modal and noisy [19,20,21,22,23].

The major features of GA like direct use of coding, search from a population, blindness to auxiliary information and randomized operators contribute to its robustness and resulting advantage over other more commonly used techniques. In particular, GA has many advantages over the traditional numerical optimization approaches, including the following facts [24]; first, it can optimize with continuous or discrete parameters. Second, it does not require derivative information. Third, it simultaneously searches from a wide sampling of the cost surface. Fourth, it can work with large number of variables. Fifth, it provides a list of optimum parameters, not just a single solution. Sixth, it may encode the parameters and the optimization is done with the encoded parameters. Seventh, it works with numerically generated data, experimental data, or analytical functions.

6 Continuous genetic algorithm

This section utilizes and evaluates the continuous GA as an efficient evolutionary algorithm including its history and its characteristics. In this summary, it is intended to present the most representative works in a continuous GA.

Continuous GA depends on the evolution of curves in one-dimensional space, surfaces in two-dimensional space, and volumes in three-dimensional space. Generally, continuous GA uses smooth operators and avoids sharp jumps in the parameter values. The algorithm begins with a population of randomly generated candidates and evolves towards better solution by applying genetic operators. This novel approach is a relatively new class of optimization technique, which generates a growing interest in the mathematics and engineering community.

Continuous GA has recently emerged as a powerful framework in the numerical analysis field. Continuous GA was developed in [25] as an efficient method for the solution of optimization problems in which the parameters to be optimized are correlated with each other or the smoothness of the solution curve must be achieved. It has been successfully applied in the motion planning of robot manipulators, which is a highly nonlinear, coupled problem [26,27], in the solution of collision-free path planning problem for robot manipulators [28], in the numerical solution of second-order, two-point regular BVPs [29], in the solution of optimal control problems [30], in the solution of second-order, two-point singular BVPs [31], and in the solution of systems of second-order regular BVPs [32]. Their novel development has opened the doors for wide applications of the algorithm in the fields of mathematics, physics, and engineering. It has been also applied in the solution of fuzzy differential equations [33]. On the other hand, the numerical solvability of other version of differential equations and other related equations can be found in [34,35,36,37,38,39,40] and references therein.

The reader is asked to refer to [25,26,27,28,29,30,31,32,33] in order to know more details about continuous GA, including their justification for use, conditions on smoothness of the functions used in the algorithm, several advantages of continuous GA over conventional GA (discrete version) when it is applied to problems with coupled parameters and(or) smooth solution curves, etc.

7 Software libraries and numerical inputs

The solution of most practical optimization problems requires the use of computers. Several commercial software systems are available to solve optimization problems that arise in different mathematical and engineering areas. All the simulations in the optimization runs presented in this paper were performed using Visual Basic Platform.

The continuous GA proposed in this paper is used to solve the given system of singular BVPs. The input data to the algorithm is divided into two parts; the continuous GA related parameters and the system of singular BVPs related parameters. The continuous GA related parameters include the population size, N_p, the individual crossover probability, p_{ci}, the curve crossover probability, p_{cc}, the individual mutation probability, p_{mi}, the curve mutation probability, p_{mc}, the rank-based ratio, R_{br}, the initialization method, the selection scheme used, the replacement method, the immigration threshold value and the corresponding number of generations, and finally the termination criterion. The system of singular BVPs related parameters include the governing singular differential system, the independent interval $[0,1]$, the boundary values, α, β, and finally the number of nodes, N. The reader is kindly requested to go through [41,42,43,44,45,46,47,48,49,50] for more details about the selection scheme, the replacement method, the control parameters, and others.

Optimization algorithms are iterative. They begin with an initial guess of the variable and generate a sequence of improved estimates (called “iterates or generations”) until they terminate, hopefully at a solution. What is the termination condition? As the algorithm runs until it reach a certain extent, the structure of individual between every two ones is very similar. It will be difficult to find a better individual if the algorithm till runs. Under this condition, we think this algorithm is convergent. In this paper, we think the population is convergent when one of the following conditions is met. First, the fitness of the best individual of the population reaches a value of 0.9999999. Second, the maximum nodal residual of the best individual of the population is less than or equal to 0.00000001. Third, a maximum number of 3000

© 2014 NSP
Natural Sciences Publishing Cor.

generations is reached. Fourth, the improvement in the fitness value of the best individual in the population over 1000 generations is less than 0.001. It is to be noted that the first two conditions indicate to a successful termination process (optimal solution is found), while the last two conditions point to a partially successful end depending on the fitness of the best individual in the population (near-optimal solution is reached) [25,26,27,28,29,30,31,32,33].

The sequence of steps taken in a continuous GA to solve Eqs. (1) and (2) numerically is shown in the next algorithm.

Algorithm 1: To approximate the solutions of system of singular BVPs (1) and (2) at $N-1$ equally spaced number in $(0,1)$:

Input: System of singular BVPs parameters: endpoints of $[0,1]$; integer N; and boundary conditions α,β. Continuous GA parameters: population size N_p; values of probabilities p_{ci}, p_{cc}, p_{mi}, p_{mc}; and rank-based ratio R_{br}.

Output: Approximation Φ_1 and Φ_2 of y_1 and y_2, respectively, at $N+1$ values of x.

Step 1: Set $h=\frac{1}{N}$;

Step 2: For $i=0,1,...,N$ set $x_i=ih$;

Step 3: Do steps $(1',2',3',4',5',6',7',8',9')$:

Step $1'$: Initialization process;

Step $2'$: Fitness evaluation process;

Step $3'$: Selection process;

Step $4'$: Crossover process;

Step $5'$: Mutation process;

Step $6'$: Fitness evaluation process;

Step $7'$: Replacement process;

Step $8'$: If termination process doesn't hold then go to Step $3'$, else go to $9'$;

Step $9'$: Output $(x_i,\Phi_1(x_i))$ and $(x_i,\Phi_2(x_i))$;

Step 4: Stop.

To summarize the evolution process in continuous GA an individual is a candidate solution that consists of two curves each of $N-1$ nodal values. The population of individuals undergoes the selection process, which results in a mating pool among which pairs of individuals are crossed over with probability p_{ci} within that pair of parents, individual solution curves are crossed with probability p_{cc}. This process results in an offspring generation where every child undergoes mutation with probability p_{mi}, within that child individual solution curves are mutated with probability p_{mc}. After that, the next generation is produced according to the replacement strategy applied. The complete process is repeated till the convergence criterion is met where the two curves of the best individual are the required solution curves. The final goal of discovering the required nodal values is translated into finding the fittest individual in genetic terms.

8 Numerical experiments

Numerical experiments are performed to measure the relative accuracy and computational efficiency. Accuracy refers to how closely a computed or measured value agrees with the true value, since, a computer has a finite word length and so only a fixed number of digits are stored and used during computation. Thus, we offer two types of error, the absolute error which is the absolute numerical difference between the exact solution and the approximate solution, and the absolute residual error which is a positive number that found by Eq. (4). In fact, the residual error will be zero at the two endpoints of $[0,1]$.

Next, two examples are studied to demonstrate the accuracy and applicability of the present algorithm. Results obtained are compared with the exact solution and are found to be in good agreement with each other.

Example 1: Consider the following linear singular differential system with singulariteis at left-right endpoints:

$$y_1''(x)+\frac{2x-1}{x^2\sqrt{x}}\left[y_1'(x)+x^2y_2'(x)\right]-\frac{x^3}{\cos\left(\frac{\pi}{2}x\right)}\left[y_1(x)-e^xy_2(x)\right]+f_1(x)=0,$$

$$y_2''(x)+\frac{x}{(x-1)^2}y_1'(x)+\frac{x}{\sin(\pi x)}\left[y_1(x)-xy_2(x)\right]+f_2(x)=0,$$

subject to the boundary conditions

$$y_1(0)=0,y_1(1)=0,$$
$$y_2(0)=0,y_2(1)=0,$$

where $x\in(0,1)$ and $f_1(x),f_2(x)$ are chosen such that the exact solutions are $y_1(x)=\sin(\pi x)$ and $y_2(x)=x^2-x$.

Using continuous GA, taking $x_i=\frac{i}{N}$, $i=0,1,...,N$ with the fitness function Fit, the Algorithm 1, and the termination conditions listed in the previous section, the numerical results of $\Phi_1(x_i)$ and $\Phi_2(x_i)$ for approximating $y_1(x_i)$ and $y_2(x_i)$ at some selected grid points for $N=10$, $N_p=500$, $p_{ci}=0.9$, $p_{cc}=0.5$, $p_{mi}=0.9$, $p_{mc}=0.5$, and $R_{br}=0.1$ are given in Tables 1 and 2, respectively.

Example 2: Consider the following nonlinear singular differential system with singulariteis at left-right endpoints:

$$y_1''(x)+\frac{20}{x(x-1)}\left[y_1'(x)-(y_2'(x))^2\right]-\frac{\cos(x)}{\sinh^2(x)}\times\left[y_1^2(x)+x\sin(y_1(x)y_2(x))\right]+f_1(x)=0,$$

$$y_2''(x)+\frac{5e^x}{x\sin x}(y_2'(x))^3-\frac{x}{\sqrt{1-x}}[\sinh(x)(y_2(x))^3+\sin(x)y_2(x)\cos(y_1(x))]+f_2(x)=0,$$

© 2014 NSP
Natural Sciences Publishing Cor.

Table 1: Numerical results of $y_1(x)$ for Example 1.

x_i	$y_1(x_i)$	$\Phi_1(x_i)$	$\lvert y_1(x_i)-\Phi_1(x_i)\rvert$	$\lvert \mathrm{Res}_1(x_i)\rvert$
0	0	0	0	0
0.1	0.3090169944	0.3090169709	2.34377574×10^{-8}	2.46906269×10^{-7}
0.2	0.5877852523	0.5877852301	2.21766321×10^{-8}	7.33246741×10^{-7}
0.3	0.8090169944	0.8090169716	2.27738666×10^{-8}	1.31747821×10^{-7}
0.4	0.9510565163	0.9510564898	2.64554492×10^{-8}	1.57196727×10^{-7}
0.5	1	0.9999999884	1.16274315×10^{-8}	1.43210365×10^{-7}
0.6	0.9510565163	0.9510565048	1.15035971×10^{-8}	1.16714975×10^{-7}
0.7	0.8090169944	0.8090169852	9.20499440×10^{-9}	9.08216851×10^{-7}
0.8	0.5877852523	0.5877852479	4.43163132×10^{-9}	6.09962319×10^{-7}
0.9	0.3090169944	0.3090169837	1.07166703×10^{-8}	3.19497157×10^{-7}
1	0	0	0	0

Table 2: Numerical results of $y_2(x)$ for Example 1.

x_i	$y_2(x_i)$	$\Phi_2(x_i)$	$\lvert y_2(x_i)-\Phi_2(x_i)\rvert$	$\lvert \mathrm{Res}_2(x_i)\rvert$
0	0	0	0	0
0.1	−0.09	−0.0900000036	3.55472876×10^{-9}	2.82238980×10^{-8}
0.2	−0.16	−0.1600000087	8.68203299×10^{-9}	3.31277996×10^{-8}
0.3	−0.21	−0.2100000323	3.22981050×10^{-8}	3.94634569×10^{-7}
0.4	−0.24	−0.2400000186	1.85934107×10^{-8}	4.50275413×10^{-7}
0.5	−0.25	−0.2500000991	9.91166535×10^{-8}	4.94881138×10^{-7}
0.6	−0.24	−0.2400000641	6.41332590×10^{-8}	5.20706622×10^{-7}
0.7	−0.21	−0.2100000110	1.10134028×10^{-8}	5.45125595×10^{-7}
0.8	−0.16	−0.1600000024	2.36580521×10^{-9}	5.35900147×10^{-8}
0.9	−0.09	−0.0900000074	7.42388421×10^{-9}	5.38354922×10^{-8}
1	0	0	0	0

subject to the boundary conditions

$$y_1(0)=1, y_1(1)=e,$$
$$y_2(0)=0, y_2(1)=\sinh(1),$$

where $x\in(0,1)$ and $f_1(x), f_2(x)$ are chosen such that the exact solutions are $y_1(x)=e^x$ and $y_2(x)=\sinh(x)$.

Using continuous GA, taking $x_i=\frac{i}{N}$, $i=0,1,...,N$ with the fitness function Fit, the Algorithm 1, and the termination conditions listed in the previous section, the numerical results of $\Phi_1(x_i)$ and $\Phi_2(x_i)$ for approximating $y_1(x_i)$ and $y_2(x_i)$ at some selected grid points for $N=10$, $N_p=500$, $p_{ci}=0.9$, $p_{cc}=0.5$, $p_{mi}=0.9$, $p_{mc}=0.5$, and $R_{br}=0.1$ are given in Tables 3 and 4, respectively.

It is to be noted that the accuracy of a certain node is in advanced, since it has a truncation error of the order $O(h^{10})$. On the other aspect as well, from the last mentioned tables, we see that we can achieve a good approximations with the exact solutions.

9 Statistical analysis

In this section, the effects of various continuous GA operators and control parameters on the convergence speed of the proposed algorithm are investigated in order to capture the behavior of solutions. The analysis includes the evolutionary progress plots, of the best-fitness individual, the evolution of nodal values, the effect of the step size in addition to an analysis of the population size, the curve crossover and the curve mutation probabilities, and the maximum nodal residual effect.

Definition 4: The convergence speed of the algorithm, whenever used, means the average number of generations required for convergence.

Remark 4: Throughout this paper, we will try to give the results of the two examples; however, in some cases we will switch between the results obtained for the examples in order not to increase the length of the paper without the loss of generality for the remaining results.

Due to the stochastic nature of continuous GA, twelve different runs were made for every result obtained in this work using a different random number generator seed; results are the average values of these runs. This means that each run of the continuous GA will result in a slight different result from the other runs.

The convergence data of the two examples is given in Table 5. It is clear from the table that the examples take 1623 generations, on average, to converge to a fitness value of about 0.999999789 with an average absolute nodal residual of the value 1.75502448×10^{-7} and an

© 2014 NSP
Natural Sciences Publishing Cor.

Table 3: Numerical results of $y_1(x)$ for Example 2.

x_i	$y_1(x_i)$	$\Phi_1(x_i)$	$\lvert y_1(x_i)-\Phi_1(x_i)\rvert$	$\lvert \mathrm{Res}_1(x_i)\rvert$
0	1	1	0	0
0.1	1.1051709181	1.1051709176	4.52360174×10^{-10}	6.13045170×10^{-9}
0.2	1.2214027582	1.2214027580	1.11405697×10^{-10}	4.18143742×10^{-9}
0.3	1.3498588076	1.3498588074	1.66059423×10^{-10}	3.81930265×10^{-9}
0.4	1.4918246976	1.4918246975	1.87788104×10^{-10}	3.52955665×10^{-9}
0.5	1.6487212707	1.6487212704	3.38139288×10^{-10}	2.24281038×10^{-9}
0.6	1.8221188004	1.8221188003	1.35799902×10^{-10}	2.11427986×10^{-9}
0.7	2.0137527075	2.0137527073	1.57605628×10^{-10}	1.63476344×10^{-9}
0.8	2.2255409285	2.2255409278	7.36103139×10^{-10}	1.07448717×10^{-9}
0.9	2.4596031112	2.4596031111	7.00001307×10^{-11}	3.15012016×10^{-9}
1	2.7182818285	2.7182818285	0	0

Table 4: Numerical results of $y_2(x)$ for Example 2.

x_i	$y_2(x_i)$	$\Phi_2(x_i)$	$\lvert y_2(x_i)-\Phi_2(x_i)\rvert$	$\lvert \mathrm{Res}_2(x_i)\rvert$
0	0	0	0	0
0.1	0.1001667500	0.1001667490	1.06187525×10^{-9}	8.02476012×10^{-8}
0.2	0.2013360025	0.2013360002	2.37455608×10^{-9}	3.96756814×10^{-8}
0.3	0.3045202934	0.3045202917	1.78401083×10^{-9}	4.46209980×10^{-8}
0.4	0.4107523258	0.4107523239	1.93059550×10^{-9}	5.46656354×10^{-8}
0.5	0.5210953055	0.5210953042	1.28153001×10^{-9}	3.33396211×10^{-8}
0.6	0.6366535821	0.6366535804	1.76460065×10^{-9}	3.02547742×10^{-8}
0.7	0.7585837018	0.7585836993	2.54339745×10^{-9}	3.84077958×10^{-8}
0.8	0.8881059822	0.8881059811	1.11469865×10^{-9}	2.51190269×10^{-8}
0.9	1.0265167257	1.0265167250	6.74489897×10^{-10}	2.78001644×10^{-9}
1	1.1752011936	1.1752011936	0	0

Table 5: Convergence data of Examples 1 and 2.

Example	Average generations	Average fitness	Average error	Average absolute residual
1	1597	0.999999836	2.16394062×10^{-8}	3.30061098×10^{-7}
2	1649	0.999999741	9.38056433×10^{-10}	2.09437978×10^{-8}

average absolute nodal error of the value 1.12887313×10^{-8}.

The evolutionary progress plots, of the best-fitness individual of Examples 1 and 2 are shown in Figure 1. It is clear from the figure that, in the first 30% of generations the best-fitness approaches to one very fast, after that, it approaches to one slower. That means the approximate of continuous GA converge to the actual solution very fast in the first 30% of the generations.

The way in which the nodal values evolve for Example 2 is studied next. Figure 2 shows the evolution of the first, x_1, and middle, x_5, nodal values of Φ_1, while Figure 3 shows the evolution of the middle, x_5, and ninth, x_9, nodal values of Φ_2.

It is observed that from the evolutionary plots that the convergence process is divided into two stages: the coarse-tuning stage and the fine-tuning stage, where the coarse-tuning stage is the initial stage in which oscillations in the evolutionary plots occur, while the fine-tuning stage is the final stage in which the evolutionary plots reach steady-state values and do not have oscillations by usual inspection. In other words, evolution has initial oscillatory nature for all nodes, in the same example. As a result, all nodes, in the same example, reach the near optimal solution together.

The effect of the step size on the convergence speed and the corresponding errors is explored next. Tables 6 and 7 give the relevant data for Example 1, where the number of nodes covers the range from 10 to 80. It is observed that the reduction in the step size results in a reduction in the error and correspondingly an improvement in the accuracy of the obtained solutions. This goes in agreement with the known fact about finite difference schemes where more accurate solutions are achieved using a reduction in the step size. On the other hand, the cost to be paid while going in this direction is the rapid increase in the number of generations required for convergence. For instance, while reducing the step

© 2014 NSP
Natural Sciences Publishing Cor.

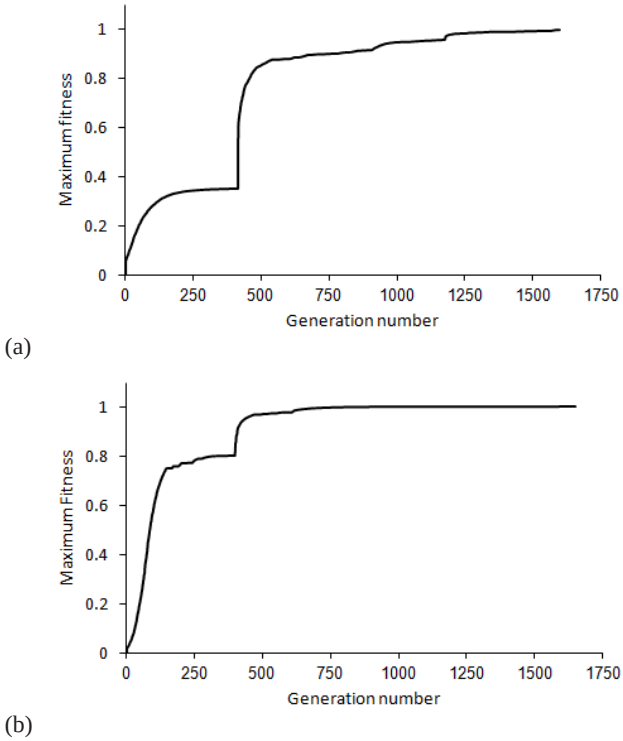


Fig. 1: Evolutionary progress plots of fitness function across all generations for: (a) Example 1; (b) Example 2.

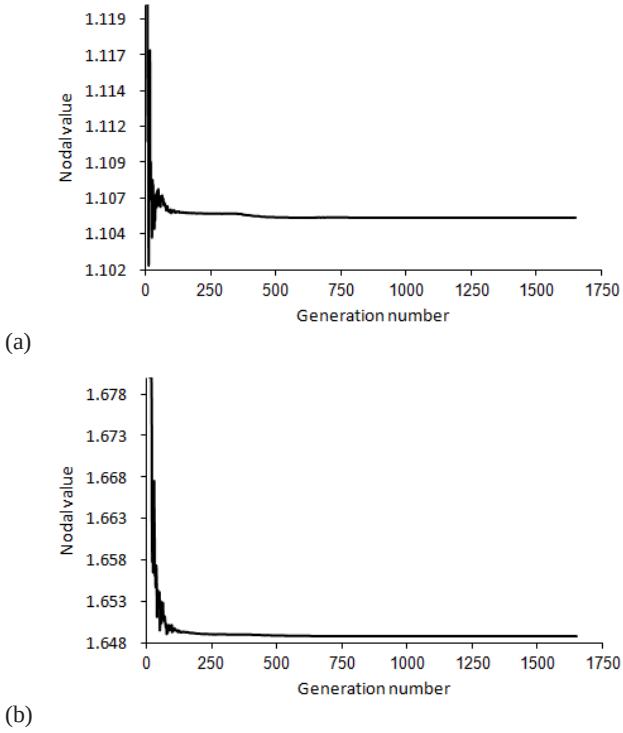


Fig. 2: Evolution of the nodal values of Φ_1 for Example 2 across all generations at: (a) the first nodal; (b) the fifth nodal.

size from 0.1 to 0.05, the required number of generations for convergence jumps from almost 1600 to 2000, i.e. 1.25 multiplication factor.

The influence of the population size on the convergence speed of CGA, the average fitness, and the corresponding errors is studied next for Example 2 as shown in Table 8. The population size is increased in steps of 100 starting with 100 and ending with 1000. Small population sizes suffer from larger number of generations required for convergence and the probability of being trapped in local minima, while large population size suffer from larger number of fitness evaluations that means larger execution time. However, it is noted that the improvement in the convergence speed becomes almost negligible after a population size of 700.

The particular settings of several continuous GA tuning parameters including the probabilities of applying crossover operator and mutation operator are investigated here. These tuning parameters are typically problem-dependent and have to be determined experimentally. They play a non-negligible role in the improvement of the efficiency of the algorithm. Table 9 shows the effect of the crossover probability, p_c, and the mutation probability, p_m, on the convergence speed of the algorithm for Example 1. The probability value is increased in steps of 0.2 starting with 0.1 and ending with 0.9 for both p_c and p_m. It is clear from the tables that when the probabilities values p_c and p_m are increasing gradually, the average number of generation required for convergence is decreasing as well. It is noted that the best performance of the algorithm is achieved for $p_c = 0.9$ and $p_m = 0.9$. As a result, these values are set as the algorithm default values.

Finally, the influence of the maximum nodal residual of the best individual on the convergence speed, the average execution time, and the corresponding fitness is investigated. This is the second termination condition of the algorithm and its value is set between 0.1 and 0.0000000001. Table 10 gives the relevant data for Example 2. Regarding the convergence speed, it is obvious that as the maximum nodal residual decreases, the number of generations required for convergence increases rapidly since the searching process will be dominated by the fine-tuning stage. The difference between the exact and the continuous GA nodal values decreases initially till a maximum nodal residual of the value 0.0000000001 is reached. After that, there will be no improvement in the accuracy of the solutions obtained for further reduction in the maximum nodal residual. The proposed approach is a variant of the finite difference scheme with a truncation error of order $O(h^{10})$. As a result, the accuracy of the solutions obtained is dependent

© 2014 NSP
Natural Sciences Publishing Cor.

Table 6: The influence of the step size on the convergence speed and the corresponding errors of $y_1(x)$ for Example 1.

Step size	Average generations	Maximum absolute error	Maximum absolute residual
0.1	1597	$2.64554492 \times 10^{-8}$	$9.08216851 \times 10^{-7}$
0.05	2034	$3.08997577 \times 10^{-10}$	$1.51010432 \times 10^{-8}$
0.025	2469	$8.87751763 \times 10^{-11}$	$8.87771443 \times 10^{-9}$
0.0125	3000	$1.08971653 \times 10^{-11}$	$4.08996140 \times 10^{-9}$

Table 7: The influence of the step size on the convergence speed and the corresponding errors of $y_2(x)$ for Example 1.

Step size	Average generations	Maximum absolute error	Maximum absolute residual
0.1	1597	$9.91166535 \times 10^{-8}$	$5.45125595 \times 10^{-7}$
0.05	2034	$1.59980346 \times 10^{-10}$	$4.99476087 \times 10^{-8}$
0.025	2469	$6.09975746 \times 10^{-11}$	$6.53925227 \times 10^{-9}$
0.0125	3000	$7.16424350 \times 10^{-12}$	$8.09066997 \times 10^{-10}$

Table 8: The effect of the population size on the convergence speed, the average fitness, and the corresponding errors for Example 2.

N_p	Average generations	Average fitness	Average error	Average residual
100	2751	0.9990485702	$2.47489011 \times 10^{-7}$	$7.28242506 \times 10^{-6}$
200	2316	0.9999370096	$8.01924967 \times 10^{-8}$	$1.00274567 \times 10^{-6}$
300	2042	0.9999910414	$1.06727973 \times 10^{-8}$	$8.21219268 \times 10^{-7}$
400	1852	0.9999963452	$4.00331248 \times 10^{-9}$	$3.46816826 \times 10^{-7}$
500	1649	0.9999997407	$9.38056433 \times 10^{-10}$	$2.09437978 \times 10^{-8}$
600	1598	0.9999997956	$3.15343296 \times 10^{-10}$	$9.23276812 \times 10^{-9}$
700	1507	0.9999998259	$8.24209177 \times 10^{-11}$	$6.52598677 \times 10^{-9}$
800	1479	0.9999998474	$6.33182201 \times 10^{-11}$	$9.37609028 \times 10^{-10}$
900	1423	0.9999999000	$3.15861153 \times 10^{-11}$	$5.01801005 \times 10^{-10}$
1000	1399	0.9999999000	$1.91056846 \times 10^{-11}$	$2.19434710 \times 10^{-10}$

on the step size used, and for a certain step size there will be initial improvement while decreasing the maximum nodal residual till the step size limit is reached where further reduction will be of no use.

10 Concluding remarks

The aim of present work is to develop an efficient and accurate method for solving systems of singular BVPs. We can conclude that the continuous GA approach is powerful and efficient technique in finding approximate solutions for linear and nonlinear systems of singular BVPs with singularity at one or both endpoints. In the proposed algorithm, each of the derivatives is replaced by an appropriate difference quotient approximation, where two smooth solution curves are used for representing the required nodal values. There is an important point to make here; the results obtained by the continuous GA approach are very effective and convenient in linear and nonlinear cases with less computational work and time. This confirms our belief that the efficiency of our technique gives it much wider applicability in the future for general classes of linear and nonlinear BVPs of different orders and types. On the other aspect as well, the influence of different parameters, including the evolution of nodal values, the maximum nodal residual, the population size, the curve's probabilities, and the step size is also studied.

The evolutionary progress investigations showed that approximate of continuous GA converge to the actual solution very fast in the first 30% of the generations, and the problems spent about 20% of generations, on average, in the coarse-tuning stage, while the remaining 80% is spent in the fine-tuning stage. The accuracy obtained using continuous GA is moderate since it has a truncation error of the order $O(h^{10})$. The accuracy of the solution obtained is dependent on the step size used, and for a certain step size there will be initial improvement while decreasing the maximum nodal residual till the step size limit is reached where further reduction will be of no use. Saturation population is reached at a population size of 700.

Acknowledgments

The authors would like to express their thanks to unknown referees for their careful reading and helpful comments.

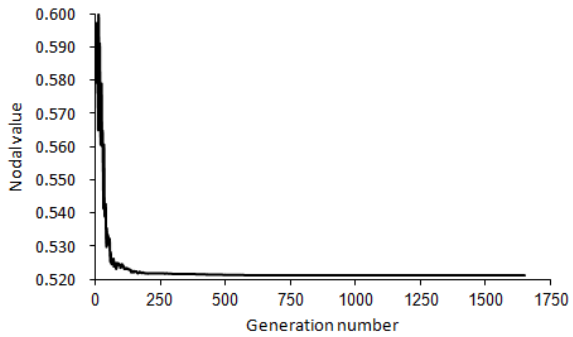
© 2014 NSP
Natural Sciences Publishing Cor.

Table 9: The effect of the crossover probability and the mutation probability on the convergence speed for Example 1.

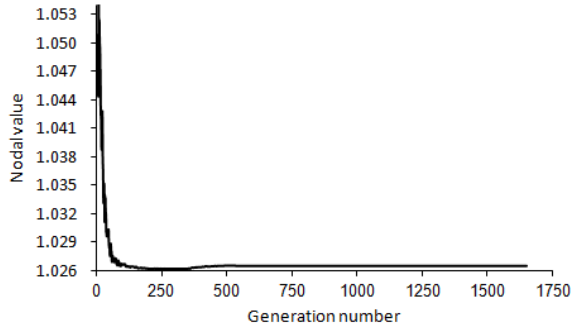
(p_m, p_c)	0.1	0.3	0.5	0.7	0.9
0.1	3000	2916	2681	2422	2139
0.3	2915	2774	2592	2269	1955
0.5	2722	2540	2377	2117	1838
0.7	2354	2264	2122	1922	1619
0.9	2266	2106	1934	1703	1597

Table 10: The influence of the maximum nodal residual on the convergence speed, the average execution time, and the average fitness for Example 2.

Maximum nodal residual	Average execution time (seconds)	Average generations	Average fitness
0.1	239.540	580	0.766103194
0.01	253.180	607	0.939420902
0.001	358.939	871	0.990364603
0.0001	433.396	1003	0.999389592
0.00001	486.255	1155	0.999787974
0.000001	576.378	1353	0.999987891
0.0000001	589.056	1416	0.999997543
0.00000001	685.984	1649	0.999999741
0.000000001	747.077	1783	0.999999846
0.0000000001	787.691	1871	0.999999900



(a)



(b)

Fig. 3: Evolution of the nodal values of Φ_2 for Example 2 across all generations at: (a) the fifth nodal; (b) the ninth nodal.

References

[1] H. Lü, H. Yu, Y. Liu, Positive solution for singular boundary value problems of a coupled system of differential equations, Journal of Mathematical Analysis and Applications, **302**, 14-29 (2005).

[2] S. Bellew, E. O'Riordan, A parameter robust numerical method for a system of two singularly perturbed convection-diffusion equations, Applied Numerical Mathematics, **51**, 171-186 (2004).

[3] C. Xenophontos, L. Oberbroeckling, A numerical study on the finite element solution of singularly perturbed systems of reaction-diffusion problems, Applied Mathematics and Computation, **187**, 1351-1367 (2007).

[4] J. Baštinec, J. Diblík, Z. Šmarda, Multipoint singular boundary-value problem for systems of nonlinear differential equations, Boundary Value Problems, **2009**, Article ID 137451, 20 pages, (2009).

[5] H. Yao, Y. Lin, M. Cui, Representation of exact solution for singular boundary value problems of system of equations, Applied Mathematical Sciences, **1**, 1387-1397 (2007).

[6] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, New York, Berlin, Heidelberg, (1999).

[7] S.T. Cheng, Topological optimization of a reliable communication network, IEEE Transactions on Reliability, **47**, 225-233 (1998).

[8] I.S. Misra, A. Raychowdhury, K.K. Mallik, M.N. Roy, Design and optimization of a nonplanar multiple array using genetic algorithms for mobile communications, Microwave and Optical Technology Letters, **32**, 301-304 (2002).

[9] J. Burm, Optimization of high-speed metal semiconductor metal photodetectors, IEEE Photonics Technology Letters, **6**, 722-724 (1994).

[10] A. Vossinis, Shape optimization of aerodynamics using nonlinear generalized minimal residual algorithm, Optimal Control Applications & Methods, **16**, 229-249 (1995).

[11] R. Fondacci, Combinatorial issues in air traffic optimization, Transportation Science, **32**, 256-267 (1998).

[12] E. deKlerk, C. Roos, T. Terlaky, T. Illés, A.J. deJong, J. Valkó, J.E. Hoogenboom, Optimization of nuclear reactor

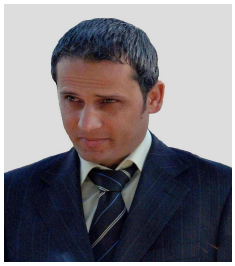
© 2014 NSP
Natural Sciences Publishing Cor.

reloading patterns, Annals of Operations Research, **69**, 65-84 (1997).
[13] Y. Cherruault, Global optimization in biology and medicine, Mathematical and Computer Modelling, **20**, 119-132 (1994).
[14] J.G. Rowse, On the solution of optimal tax models and other optimization models in economics, Economics Letters, **18**, 217-222 (1985).
[15] F. Rothlauf, Design of Modern Heuristics: Principles and Application, Natural Computing Series, Springer-Verlag Berlin Heidelberg, (2011).
[16] I. Boussaïd, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Information Sciences, **237**, 82-117 (2013).
[17] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, United Kingdom, Luniver Press, (2010).
[18] J. Li, General explicit difference formulas for numerical differentiation, Journal of Computational and Applied Mathematics, **183**, 29-52 (2005).
[19] S. Bandyopadhyay, S. Saha, Unsupervised Classification: Similarity Measures, Classical and Metaheuristic Approaches, and Applications, Springer-Verlag Berlin Heidelberg, (2013).
[20] P. Punia, M. Kaur, Various genetic approaches for solving single and multi-objective optimization problems: a review, International Journal of Advanced Research in Computer Science and Software Engineering, **3**, 1014-1020 (2013).
[21] K. Deb, Single and multi-objective optimization using evolutionary computation, Wiley, (2008).
[22] X. Zou, Y. Chen, M. Liu, L. Kang, A new evolutionary approach for solving many-objective optimization problems, IEEE Transactions on Systems, Man, and Cybernetics, Part B, **38**, 1402-1412 (2008).
[23] K. Kristinsson, G.A. Dumont, System-identification and control using genetic algorithms, IEEE Transactions on Systems, Man, and Cybernetics, **22**, 1033-1046 (1992).
[24] R. Jain, G.S. Mani, Solving "antenna array thinning problem" using genetic algorithm, Applied Computational Intelligence and Soft Computing, **2012**, Article ID 946398, 14 pages. doi:10.1155/2012/946398.
[25] Z.S. Abo-Hammour, Advanced Continuous Genetic Algorithms and their Applications in the Motion Planning of Robotic Manipulators and the Numerical Solution of Boundary Value Problems (Ph.D. Thesis, Quiad-Azam University, Pakistan,) (2002).
[26] Z.S. Abo-Hammour, A novel continuous genetic algorithms for the solution of the cartesian path generation problem of robot manipulators, in: J.X. Lui (Eds.), Robot Manipulators: New Research, Nova Science Publishers Inc., New York, 133-190 (2005).
[27] Z.S. Abo-Hammour, N. Mirza, S. Mirza, M. Arif, Cartesian path planning of robot manipulators using continuous genetic algorithms, Robotics and Autonomous Systems, **41**, 179-223 (2002).
[28] Z.S. Abo-Hammour, O. Alsmadi, S.I. Bataineh, M.A. Al-Omari, N. Affach, Continuous genetic algorithms for collision-free cartesian path planning of robot manipulators, International Journal of Advanced Robotic Systems, **8**, 14-36 (2011).
[29] Z.S. Abo-Hammour, M. Yusuf, N Mirza, S. Mirza, M. Arif, J. Khurshid, Numerical solution of second-order, two-point boundary value problems using continuous genetic algorithms, International Journal for Numerical Methods in Engineering, **61**, 1219-1242 (2004).
[30] Z.S. Abo-Hammour, A. Al-Asasfeh, A. Al-Smadi, O. Alsmadi, A novel continuous genetic algorithm for the solution of optimal control problems, Optimal Control Applications and Methods, **32**, 414-432 (2010).
[31] O. Abu Arqub, Z.S. Abo-Hammour, S. Momani, N. Shawagfeh, Solving singular two-point boundary value problems using continuous genetic algorithm, Abstract and Applied Analysis, **2012**, Article ID 205391, 25 page, (2012). doi.10.1155/2012/205391.
[32] O. Abu Arqub, Z.S. Abo-Hammour, S. Momani, Application of continuous genetic algorithm for nonlinear system of second-order boundary value problems, Applied Mathematics and Information Sciences, **8**, 235-248 (2014).
[33] O. Abu Arqub, Numerical Solution of Fuzzy Differential Equation using Continuous Genetic Algorithms (Ph.D. Thesis, University of Jordan, Jordan,) (2008).
[34] A. El-Ajou, O. Abu Arqub, S. Momani, Homotopy analysis method for second-order boundary value problems of integrodifferential equations, Discrete Dynamics in Nature and Society, **2012**, Article ID 365792, 18 pages, (2012). doi:10.1155/2012/365792.
[35] O. Abu Arqub, M. Al-Smadi, S. Momani, Application of reproducing kernel method for solving nonlinear Fredholm-Volterra integrodifferential equations, Abstract and Applied Analysis, **2012**, Article ID 839836, 16 pages, 2012. doi:10.1155/2012/839836.
[36] M. Al-Smadi, O. Abu Arqub, S. Momani, A computational method for two-point boundary value problems of fourth-order mixed integrodifferential equations, Mathematical Problems in Engineering, **2013**, Article ID 832074, 10 pages, 2013. dx.doi.org/10.1155/2013/832074.
[37] KF Man, KS Tang, S Kwong, Genetic algorithms: concepts and applications, IEEE Transactions on Industrial, (1996).
[38] O. Abu Arqub, M. Al-Smadi, N. Shawagfeh, Solving Fredholm integro-differential equations using reproducing kernel Hilbert space method, Applied Mathematics and Computation, **219**, 8938-8948 (2013).
[39] O. Abu Arqub, A El-Ajou, A. Bataineh, I. Hashim, A representation of the exact solution of generalized Lane-Emden equations using a new analytical method, Abstract and Applied Analysis, **2013**, Article ID 378593, 10 pages, (2013). dx.doi.org/10.1155/2013/378593.
[40] O. Abu Arqub, Z. Abo-Hammour, R. Al-badarneh, S. Momani, A reliable analytical method for solving higher-order initial value problems, Discrete Dynamics in Nature and Society, **2013**, Article ID 673829, 12 pages. dx.doi.org/10.1155/2013/673829.
[41] J.H. Jiang, J.H. Wang, X. Chu, R.Q. Yu, Clustering data using a modified integer genetic algorithm, Analytica Chimica Acta, **354**, 263-274 (1997).
[42] E.M. Rudnick, J.H. Patel, G.S. Greenstein, T.M. Niermann, A genetic algorithm framework for test generation, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, **16**, 1034-1044 (1997).
[43] J.R. Vanzandt, A Genetic Algorithm for Search Route Planning, ESD-TR-92-262, United States Air Force, (1992).
[44] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley: Reading, MA, (1989).

© 2014 NSP
Natural Sciences Publishing Cor.

[45] U. Horchner, J.H. Kalivas, Further investigation on a comparative study of simulated annealing and genetic algorithm for wavelength selection, Analytica Chimica Acta, **311**, 1-13 (1995).
[46] Y. Davidor, Genetic Algorithms and Robotics: A Heuristic Strategy for Optimization, World Scientific: Singapore, (1991).
[47] K.D.M. Harris, R.L. Johnston, B.M. Kariuki, The genetic algorithms: foundation and applications in structure solution from powder diffraction data, Acta Crystallographica A, **54**, 632-645 (1998).
[48] Z.S. Abo-Hammour, O. Alsmadi, A. Al-Smadi, M. Zaqout, M. Saraireh, ARMA model order and parameter estimation using genetic algorithms, Mathematical and Computer Modelling of Dynamical Systems, **18**, 201-221 (2012).
[49] Z. Abo-Hammour, O. Alsmadi, S. Momani, O. Abu Arqub, A genetic algorithm approach for prediction of linear dynamical systems, Mathematical Problems in Engineering, **2013**, Article ID 831657, 12 pages, (2012). dx.doi.org/10.1155/2013/831657.
[50] P. Cong P, T. Li, Numeric genetic algorithm Part I. Theory, algorithm and simulated experiments, Analytica Chimica Acta, **293**, 191-203 (1994).

Zaer Abo-Hammour received his Ph.D. degree from the Quiad-Azam university (Pakistan) in 2002. He is an associate professor at the mechatronics engineering department, the university of Jordan. His research interests are focused on control systems, robotics, GA, and numerical fields.



Omar Abu Arqub received his Ph.D. from the university of Jordan (Jordan) in 2008. He then began work at Al Balqa applied university in 2008 as an assistant professor of applied mathematics and promoted to associate professor in 2013. His research interests focus on numerical analysis, optimization techniques, optimal control, fractional calculus theory, and fuzzy calculus theory.

Othman Alsmadi received his Ph.D. degree in 1999 with emphasis in control systems from Wichita State university (USA). In the year 2000, he worked as a senior research engineer at the Caterpillar Inc., Kansas, USA. Currently, Dr. Alsmadi is an associate professor and chairman of the electrical engineering department, university of Jordan. His research interest is focused in the area of control systems, neural networks, optimal control, model order reduction, system identification, and numerical analysis.

Shaher Momani received his Ph.D. from the university of Wales (UK) in 1991. He then began work at Mutah university in 1991 as an assistant professor of applied mathematics and promoted to full Professor in 2006. He left Mutah university to the university of Jordan in 2009 until now. Professor Momani has been at the forefront of research in the field of fractional calculus in two decades. His research interests focus on the numerical solution of fractional differential equations in fluid mechanics, non-Newtonian fluid mechanics, and numerical analysis.

Ahmed Alsaedi obtained his Ph.D. degree from Swansea university (UK) in 2002. He has a broad experience of research in applied mathematics. His fields of interest include dynamical systems, nonlinear analysis involving ordinary differential equations, fractional differential equations, boundary value problems, mathematical modeling, biomathematics, Newtonian and Non-Newtonian fluid mechanics. He has published several articles in peer-reviewed journals. He has supervised several M.S. students and executed many research projects successfully. He is reviewer of several international journals. He served as the chairman of the mathematics department at KAU and presently he is serving as director of the research program at KAU. Under his great leadership, this program is running quite successfully and it has attracted a large number of ISI highly cited researchers and distinguished professors from all over the world. He is also the head of NAAM international research group at KAU.

© 2014 NSP
Natural Sciences Publishing Cor.