Research Article

Optimization Solution of Troesch's and Bratu's Problems of Ordinary Type Using Novel Continuous Genetic Algorithm

Zaer Abo-Hammour,[1] Omar Abu Arqub,[2] Shaher Momani,[3,4] and Nabil Shawagfeh[2]

[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 Mathematics, Faculty of Science, The University of Jordan, Amman 11942, Jordan*

[4] *Nonlinear Analysis and Applied Mathematics (NAAM) Research Group, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia*

Correspondence should be addressed to Omar Abu Arqub; o.abuarqub@bau.edu.jo

A new kind of optimization technique, namely, continuous genetic algorithm, is presented in this paper for numerically approximating the solutions of Troesch's and Bratu's problems. The underlying idea of the method is to convert the two differential problems into discrete versions by replacing each of the second derivatives by an appropriate difference quotient approximation. The new method has the following characteristics. First, it should not resort to more advanced mathematical tools; that is, the algorithm should be simple to understand and implement and should be thus easily accepted in the mathematical and physical application fields. Second, the algorithm is of global nature in terms of the solutions obtained as well as its ability to solve other mathematical and physical problems. Third, the proposed methodology has an implicit parallel nature which points to its implementation on parallel machines. The algorithm is tested on different versions of Troesch's and Bratu's problems. Experimental results show that the proposed algorithm is effective, straightforward, and simple.

1. Introduction

Nonlinear phenomena are of fundamental importance in various fields of science and engineering and other disciplines, since most phenomena in our world are essentially nonlinear and are described by nonlinear equations. On the other hand, most models of real-life problems are still very difficult to solve analytically. Therefore, approximate and numerical solutions were introduced.

Numerical methods are methods for solving problems on computers by numerical calculations, often giving a table of numbers and/or graphical representations or figures. Numerical methods tend to emphasize the implementation of algorithms. The aim of numerical methods is therefore to provide systematic methods for solving problems in a numerical form. The process of solving problems generally involves starting from an initial data, using high precision digital computers, following the steps in the algorithms, and finally obtaining the results. Often the numerical data and the methods used are approximate ones.

The goal of this paper is to give an effective optimization approach for solving two certain nonlinear second-order, two-point boundary value problems (BVPs) based on the use of the continuous genetic algorithm (GA) as an alternative to existing methods. The present technique can avoid any sensitivity of the problems and can be applied without any limitation on the number of mesh points. On the other aspect as well, the new technique is accurate, needs less effort to achieve the results, and is developed especially for nonlinear case. More specifically, we consider the following two problems.

Problem 1. Troesch's problem, which is covered by the ordinary differential equation:

$$y''(x) - \lambda \sinh(\lambda y(x)) = 0, \quad \lambda > 0, \ x \in [0, 1], \quad (1)$$

subject to the boundary conditions

$$y(0) = 0, \qquad y(1) = 1. \tag{2}$$

Problem 2. Bratu's problem, which is covered by the ordinary differential equation:

$$y''(x) + \mu \exp(y(x)) = 0, \quad \mu > 0, \ x \in [0, 1], \tag{3}$$

subject to the boundary conditions

$$y(0) = 0, \qquad y(1) = 0. \tag{4}$$

The closed form solution of Troesch's problem in terms of the Jacobian elliptic function has been given in [1] as $y(x) = (2/\lambda)\sinh^{-1}[(1/2)y'(0)\text{sc}(\lambda x \mid 1 - (1/4)(y'(0))^2)]$, where $y'(0) = 2\sqrt{1-m}$ in which the constant m being the solution of the transcendental equation $\sinh(\lambda/2) = \text{sc}(\lambda \mid m)\sqrt{1-m}$ such that $\text{sc}(\lambda \mid m)$ is the Jacobi function defined by $\text{sc}(\lambda \mid m) = \sin\phi/\cos\phi$, where ϕ, λ, and m are related by the integral $\lambda = \int_0^\phi (1/\sqrt{1 - m\sin^2\theta})d\theta$. It was noticed that a pole of $y(x)$ occurs at a pole of $\text{sc}(\lambda x \mid 1 - (1/4)(y'(0))^2)$. It was also noticed in [1] that the pole occurs at $x_s \approx (1/2\lambda)\ln(16/(1-m))$, which implies that the singularity lies within the integration range if $y'(0) > 8e^{-\lambda}$. The closed form solution of Bratu's problem has been given in [2] as $y(x) = -2\ln[\cosh(0.5(x - 0.5)\theta)/\cosh(\theta/4)]$, where θ is the solution of the transcendental equation $\theta = \sqrt{2\lambda}\cosh(\theta/4)$. In fact, this equation has two, one, or no solution when $\lambda < \lambda_c$, $\lambda = \lambda_c$, or $\lambda > \lambda_c$, respectively, in which the critical value λ_c satisfies the equation $4 = \sqrt{2\lambda_c}\sinh(\theta/4)$. It was evaluated in [2] that the critical value of λ_c equals 3.513830719.

Troesch's problem was first described and solved by Weibel [3]. Troesch's equation comes from the investigation of the confinement of a plasma column under radiation pressure and also in the theory of gas porous electrodes [4–6]. Bratu's problem was first studied and solved by Bratu [7]. Bratu's equation is used in a large variety of applications such as the model of thermal reaction process, the Chandrasekhar model of the expansion of the universe, chemical reaction theory, nanotechnology, and radiative heat transfer [8–10].

In general, nonlinear BVPs do not always have solutions which we can obtain using analytical methods. In fact, many of real physical phenomena encountered are almost impossible to solve by this technique. Due to this, some authors have proposed numerical methods to approximate the solutions of Troesch's and Bratu's problems. To mention a few, the Laplace transform decomposition method has been applied to solve Troesch's and Bratu's problems as described in [11, 12]. In [13] also, the author has provided the decomposition method to further investigate to Troesch's and Bratu's equations. Furthermore, the homotopy perturbation method is carried out in [14] for solving (1) and (2). The variational iteration method has been used to solve (1) and (2) as presented in [15]. In [16], the authors have developed the B-spline method to solve Problem 2. Recently, Lie-group shooting method for solving Problem 2 is proposed in [17].

The organization of the remainder of this paper is as follows. In Section 2, we present a short preface to optimization technique. In Section 3, we formulate the fitness function in order to solve Problems 1 and 2 using continuous GA. Section 4 covers the description of continuous GA in detail. Simulation results are given in Section 5 in order to verify the mathematical results of the proposed method. Statistical and convergence analysis are provided by the results of numerical simulation in Section 6. Finally, in Section 7 a brief summery is presented.

2. Overview of Optimization Technique

Optimization is the process of making something better. An engineer or scientist conjures up a new idea and optimization improves on that idea. Optimization consists in trying variations on an initial concept and using the information gained to improve on the idea. A computer is the perfect tool for optimization as long as the idea or variable influencing the idea can be input in electronic format. Feed the computer some data and out comes the solution.

The terminology "best" solution implies that there is more than one solution and the solutions are not of equal value. The definition of best is relative to the problem at hand, its method of solution, and the tolerances allowed. Thus the optimal solution depends on the person formulating the problem. Education, opinions, bribes, and amount of sleep are factors influencing the definition of best. Optimization plays a crucial role in various disciplines in sciences, industry, engineering, and almost in every aspect of the daily life. Optimization problems are encountered, for example, in communication systems [18], antenna design [19], semiconductors manufacturing [20], aerodynamics [21], transportation and traffic [22], nuclear reactor design [23], medicine [24], and economics [25].

Optimization occupies a fundamental position in engineering design and applications, since the classical function of the engineer is to design new, better, or more efficient and less expensive systems as well as to advise planes and procedures for the improved operation of exiting systems [26]. The application of optimization methods to engineering problems requires the selection of the problem decision variables that are adequate to characterize the possible candidate designs or operating conditions of the system, the definition of the objective function on the basis of which candidates will be ranked to determine the best solution, and the definition of a model that describes the manner in which the problem variables are related and the way in which the performance criterion is influenced by the variables. The problem's model normally includes a set of equality constraints, inequality constrains, and some bounds for the variables. In its most general case, the optimization problem involves the determination of the optimal set decision variables of a given objective function in the presence of some constraints. In the context of optimization, the "best" will always mean the candidate system with either the minimum or maximum value of the objective function.

Optimization problems can be divided into two categories depending on whether the solution is continuous or discrete. An optimization problem with discrete solution is known as a conventional optimization problem, while the continuous version is known as a continuous optimization problem. Numerical continuous optimization is the study of how to get the global numerical solution (value or point) for continuous mathematical and physical problems. The importance of numerical continuous optimization has been arisen in lots of research fields, such as science, engineering, and business [27–29].

3. Formulation of the Fitness Function

In this section, Troesch's and Bratu's problems of ordinary types are first formulated as optimization problems by the direct minimization of the overall individual residual error subject to the given constraints boundary conditions and are then solved using continuous GA.

To approximate the solutions of Troesch's and Bratu's problems, we make the stipulation that the mesh points are equally distributed through the interval $[0, 1]$. This condition is ensured by setting $x_i = ih$, $i = 0, 1, \dots, N$, where $h = 1/N$. Thus, at the interior mesh points, x_i, $i = 1, 2, \dots, N - 1$, the equation to be approximated is given as

$$y''(x_i) = F(y(x_i)), \quad x_1 \le x_i \le x_{N-1}, \quad (5)$$

subject to the boundary conditions

$$y(x_0) = 0, \qquad y(x_N) = \beta, \quad (6)$$

where $F(y(x_i)) = \lambda \sinh(\lambda y(x_i))$, $\beta = 1$, for Troesch's problem and $F(y(x_i)) = -\mu \exp(y(x_i))$, $\beta = 0$, for Bratu's problem.

The finite difference approximations for derivatives are one of the simplest and of the oldest methods to solve differential equations. This method consists of approximating the differential operator by replacing the derivatives in the equation using difference quotients. In this work, we will employ this technique to approximate the solutions of Troesch's and Bratu's problems numerically using continuous GA. Anyhow, the difference quotients approximation formulas, which closely approximate $y''(x_i)$, $i = 1, 2, \dots, N - 1$, when h is small using 5 points at the interior mesh points with error of order $O(h^3)$, are given as follows:

$$y''(x_1) \approx \frac{1}{h^2} \Delta(y(x_0), y(x_4)),$$

$$y''(x_i) \approx \frac{1}{h^2} \Pi(y(x_{i-2}), y(x_{i+2})), \quad i = 2, 3, \dots N - 2,$$

$$y''(x_{N-1}) \approx \frac{1}{h^2} \nabla(y(x_{N-4}), y(x_N)),$$

$$(7)$$

where Δ, Π, and ∇ are forward, central, and backward differences, respectively, and are given as

$$\Delta(y(x_0), y(x_4)) = \frac{11}{12} y(x_0) - \frac{5}{3} y(x_1) + \frac{1}{2} y(x_2)$$
$$+ \frac{1}{3} y(x_3) - \frac{1}{12} y(x_4),$$

$$\Pi(y(x_{i-2}), y(x_{i+2})) = -\frac{1}{12} y(x_{i-2}) + \frac{4}{3} y(x_{i-1}) - \frac{5}{2} y(x_i)$$
$$+ \frac{4}{3} y(x_{i+1}) - \frac{1}{12} y(x_{i+2}),$$

$$\nabla(y(x_{N-4}), y(x_N)) = -\frac{1}{12} y(x_{N-4}) + \frac{1}{3} y(x_{N-3})$$
$$+ \frac{1}{2} y(x_{N-2}) - \frac{5}{3} y(x_{N-1})$$
$$+ \frac{11}{12} y(x_N).$$

$$(8)$$

To complete the formulation substituting the difference approximation formulas of $y''(x_i)$, $i = 1, 2, \dots, N - 1$, into (5), the discretized form of this equation is obtained. The resulting algebraic equations will be a function of $y(x_0)$, $y(x_1)$, ..., and $y(x_N)$. On the other hand, using formulas in (7), the discrete equations to be optimized will take the following form:

$$\frac{1}{h^2} \Delta(y(x_0), y(x_4)) - F(y(x_1)) \approx 0,$$

$$\frac{1}{h^2} \Pi(y(x_{i-2}), y(x_{i+2})) - F(y(x_i)) \approx 0,$$

$$i = 2, 3, \dots N - 2, \quad (9)$$

$$\frac{1}{h^2} \nabla(y(x_{N-4}), y(x_N)) - F(y(x_{N-1})) \approx 0.$$

In order to construct the fitness function, we first define the residual of the general interior node, Res, as

$$\text{Res}(1) = \frac{1}{h^2} \Delta(y(x_0), y(x_4)) - F(y(x_1)),$$

$$\text{Res}(i) = \frac{1}{h^2} \Pi(y(x_{i-2}), y(x_{i+2})) - F(y(x_i)),$$

$$i = 2, 3, \dots N - 2,$$

$$\text{Res}(N - 1) = \frac{1}{h^2} \nabla(y(x_{N-4}), y(x_N)) - F(y(x_{N-1})).$$

$$(10)$$

As the second step in the construction, we define the overall individual residual function, Oir, in l_2 norm to be a function of the residuals of all interior nodes. This function may be stated as

$$\text{Oir} = \sqrt{\text{Res}^2(1) + \sum_{i=2}^{N-2} \text{Res}^2(i) + \text{Res}^2(N - 1)}. \quad (11)$$

For the final step, we define the fitness function, Fit, as

$$\text{Fit} = \frac{\delta}{\delta + \text{Oir}}; \quad \delta \text{ is a small positive number.} \quad (12)$$

A mapping of the overall individual residual function into a fitness function is required in the algorithm in order to convert the minimization problem of Oir into a maximization problem of Fit. In fact, we do this to facilitate the calculations and planning graphics. The value of individual fitness is improved if a decrease in the value of the Oir is achieved. The optimal solution of the problem, nodal values, will be achieved when Oir approaches zero and Fit approaches unity.

4. Outline of the Genetic Algorithm

The GA is an optimization and search technique based on the principles of genetics and natural selection. The GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the fitness function. In this section, a general review of the GA is presented. After that, a detailed description of the continuous GA is given. As will be shown later, the efficiency and performance of continuous GA depend on several factors, including the design of the continuous GA operators and the settings of the system parameters.

GA, initiated in [30] as stochastic search technique based on the mechanism of natural selection and natural genetics, has received a great deal of attention regarding its potential as optimization technique for solving discrete optimization problems or other hard optimization problems. GA start with an initial population of individuals generated at random. Each individual in the population represents a potential solution to the problem under consideration. The individuals evolve through successive iterations, called generations. During each generation, each individual in the population is evaluated using some measure of fitness. Then, the population of the next generation is created through genetic operators. The procedure continues until the termination conditions are satisfied. GA is nature-inspired optimization method that can be advantageously used for many optimization problems. GA imitates basic principles of life and applies genetic operators like mutation, crossover, or selection to a sequence of alleles. The sequence of alleles is the equivalent of a chromosome in nature and is constructed by a representation which assigns a string of symbols to every possible solution of the optimization problem.

GA is an evolutionary computation technique, developed for optimization of nonlinear, constrained and unconstrained, nondifferentiable multimodal functions. Here, we introduce some of the advantages of GA [31]; first, it optimizes with continuous or discrete variables; second, simultaneously searches from a wide sampling of the cost surface; third, it deals with a large number of variables; fourth, it is well suited for parallel computers; fifth, it optimizes variables with extremely complex cost surfaces (they can jump out of a local minimum); sixth, it provides a list of optimum variables, not just a single solution; seventh, it may encode the variables so that the optimization is done with the encoded variables; eighth, it works with numerically generated data, experimental data, or analytical functions.

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

Continuous GA was developed 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 [32]. It has been successfully applied in the motion planning of robot manipulators, which is a highly nonlinear, coupled problem [33, 34], in the solution of collision-free path planning problem for robot manipulators [35], in the numerical solution of second-order, two-point BVPs [36], in the solution of optimal control problems [37], in the solution of second-order, two-point singular BVPs [38], and in the solution of systems of second-order regular BVPs [39]. Their novel development has opened the doors for wide applications of the algorithm in the fields of mathematics and physics. It has been also applied in the solution of fuzzy differential equations [40]. On the other hand, the numerical solvability of other versions of differential equations and other related equations can be found in [41–48] and references therein. The reader is asked to refer to [32–40] in order to know more details about continuous GA, including its history, its justification for use, its applications, its characteristics, and its operators and control parameters.

The use of continuous GA with coupled parameters and/or smooth curves needs some justification [32, 36]. First, the discrete initialization version of the initial population means that neighbouring parameters might have opposite extreme values that make the probability of valuable information in this population very limited and correspondingly the fitness will be very low. This problem is overcome by the use of continuous curves that eliminate the possibility of highly oscillating values among the neighbouring parameters and result in a valuable initial population. Second, the traditional crossover operator results in a jump in the value of the parameter in which the crossover point lies while keeping the other parameters the same or exchanged between the two parents. This discontinuity results in a very slow converging process. On the other hand, the continuous GA results in smooth transition in the parameter values during the crossover process. Third, the conventional version of the mutation process changes only the value of the parameter in which the mutation occurs, while it is necessary to make some global mutations which affect a group of neighbouring parameters since either the parameters are coupled with each other or curve should be smooth. To summarize, the operators of the continuous GA are of global nature and applied at the individual level, while the operators of the traditional GA are of local nature and applied at the parameter level. As a result, the operators of the traditional GA result in a step-function-like jump in the parameter values, while those of continuous GA result in smooth transitions.
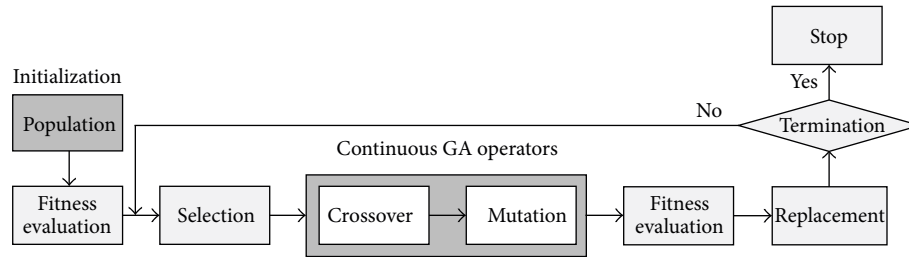
FIGURE 1: The flowchart of the life cycle of continuous GA.

Continuous GA has several advantages over conventional one when it is applied to problems with coupled parameters and/or smooth solution curves [32, 36] as follows.

(1) There are no encoding/decoding processes in continuous GA. This means that the execution time will be smaller in continuous GA case if both GA versions converge in the same number of generations.

(2) The memory requirements of conventional GA are higher than those of continuous GA because the former uses genotype and phenotype representations of the population's individuals, while the later utilizes only the phenotype data. This makes continuous GA more suitable for problems with larger number of parameters.

(3) In conventional GA, the actual range of the parameter values should lie within the fed range during the evolution process; otherwise, the optimal solution will not be reached. This is because the encoding/decoding processes in conventional GA require the range of values within which the solution should lie. To overcome this problem in conventional GA, a wide range of the parameter values should be provided or some adaptive range value scheme should be used. However, this problem is not encountered in continuous GA due to the fact that there are no encoding/decoding processes.

(4) The conventional GA fails to find the optimal solution in such problems. It might reach a near-optimal solution; as a result, it cannot be used in applications where the optimal solution is required. The deficiency of the conventional GA is emphasized when the dimension of the optimization problem increases or the coupling effect is accentuated.

However, when using GA in optimization problems, one should pay attention to two points: first, whether the parameters to be optimized are correlated with each other or not; second, whether there is some restriction on the smoothness of the resulting solution curve or not. In case of uncorrelated parameters or nonsmooth solution curves, the conventional GA will perform well. On the other hand, if the parameters are correlated with each other or smoothness of the solution curve is a must, then the continuous GA is preferable in this case [32–40].

To summarize the evolution process in continuous GA for solving Troesch's and Bratu's problems, an individual is a candidate solution that consists of 1 curve 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_c. This process results in an offspring generation where every child undergoes mutation with probability p_m. 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 $N - 1$ parameters of the best individual are the required nodal values. The final goal of discovering the required nodal values is translated into finding the fittest individual in genetic terms. The flowchart of the algorithm is given in Figure 1.

5. Simulation Results

To validate the integrity of continuous GA and to investigate the errors of the modeling and measurements, we carried out two experiments. In fact, simulation results are carried out in order to verify the mathematical results and the theoretical statements for the optimized solutions. The results obtained by the continuous GA are compared with the analytical solution of each problem. The effects of various continuous GA operators and control parameters on the convergence speed of the proposed algorithm are also investigated in this section. The analysis includes the effect of various initialization methods on the convergence speed of the algorithm in addition to an analysis of the effect of selection schemes, the vector norm used, the crossover and mutation probabilities, the population size, and the step size.

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

The continuous GA produces one population after another. This can be done in an infinite loop. When the user specifies a fitness value to be reached, the procedure can be stopped as soon as at least one individual has a higher fitness than the desired one. Often, the user does not exactly know how big the fitness value of an acceptable solution should be. Therefore, one would like to stop the continuous GA when he cannot expect to obtain much better solutions anymore. From the experience with traditional optimization algorithms one might be tempted do observe convergence and to stop the continuous GA when the maximum fitness remains more or

TABLE 1: Numerical results for Troesch's problem when $\lambda = 0.5$.

x_i	Exact value	Approximate value	$|\text{Err}(x_i)|$	$|\text{Res}(x_i)|$
0	0	0	0	0
0.1	0.0951769020	0.0951768506	$5.14476384 \times 10^{-8}$	$4.33487462 \times 10^{-7}$
0.2	0.1906338691	0.1906337234	$1.45741828 \times 10^{-7}$	$9.77131143 \times 10^{-7}$
0.3	0.2866534030	0.2866533395	$6.34609416 \times 10^{-8}$	$4.75142679 \times 10^{-7}$
0.4	0.3835229288	0.3835228897	$3.90866823 \times 10^{-8}$	$3.31275728 \times 10^{-7}$
0.5	0.4815373854	0.4815373121	$7.33121592 \times 10^{-8}$	$6.41709009 \times 10^{-7}$
0.6	0.5810019749	0.5810019054	$6.95247591 \times 10^{-8}$	$8.39064329 \times 10^{-7}$
0.7	0.6822351326	0.6822350995	$3.31330484 \times 10^{-8}$	$8.13306207 \times 10^{-7}$
0.8	0.7855717867	0.7855717450	$4.16561352 \times 10^{-8}$	$6.07573447 \times 10^{-7}$
0.9	0.8913669875	0.8913669298	$5.77267702 \times 10^{-8}$	$2.24858269 \times 10^{-7}$
1	1	1	0	0

TABLE 2: Numerical results for Troesch's problem when $\lambda = 1$.

x_i	Exact value	Approximate value	$|\text{Err}(x_i)|$	$|\text{Res}(x_i)|$
0	0	0	0	0
0.1	0.0817969966	0.0817961006	$8.95972167 \times 10^{-7}$	$6.02388187 \times 10^{-5}$
0.2	0.1645308709	0.1645279896	$2.88132675 \times 10^{-6}$	$6.70066325 \times 10^{-4}$
0.3	0.2491673608	0.2491660068	$1.35402743 \times 10^{-6}$	$6.78708955 \times 10^{-4}$
0.4	0.3367322092	0.3367315761	$6.33142603 \times 10^{-7}$	$3.82537236 \times 10^{-5}$
0.5	0.4283471610	0.4283456762	$1.48480592 \times 10^{-6}$	$7.09360764 \times 10^{-4}$
0.6	0.5252740296	0.5252726497	$1.37988706 \times 10^{-6}$	$7.37034856 \times 10^{-4}$
0.7	0.6289711434	0.6289706787	$4.64675846 \times 10^{-7}$	$7.35668506 \times 10^{-5}$
0.8	0.7411683782	0.7411675558	$8.22354112 \times 10^{-7}$	$7.16391614 \times 10^{-5}$
0.9	0.8639700206	0.8639681690	$1.85163340 \times 10^{-6}$	$7.21723770 \times 10^{-5}$
1	1	1	0	0

less constant over some populations. However, the continuous GA is stopped when one of the following conditions is met:

(1) the fitness of the best individual of the population reaches a value of 0.999999;

(2) the maximum nodal residual of the best individual of the population is less than or equal to 0.00000001;

(3) a maximum number of 3000 generations is reached;

(4) the improvement in the fitness value of the best individual in the population over 500 generations is less than 0.001.

It is to be noted that the first two conditions indicate 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) [32–40].

The continuous GA proposed in this work is used to solve the given Troesch's and Bratu's problems. The input data to the algorithm is divided into two parts: the continuous GA related parameters and the problems related parameters. The continuous GA related parameters include the population size, N_p, the individual crossover probability, p_c, the individual mutation probability, p_m, 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 criteria. The problem related parameters include the governing Troesch's and Bratu's equations, the independent interval $[0, 1]$, and the number of nodes, N.

Consider Troesch's problems (1) and (2), when $\lambda = 0.5$ and $\lambda = 1$. Using continuous GA technique, taking $x_i = i/N$, $i = 0, 1, 2, \ldots, N$, with the fitness function (12) and the previous termination conditions, the numerical results of approximating $y(x_i)$, when $\lambda = 0.5$ and $\lambda = 1$ at $N = 10$, $N_p = 500$, $p_c = 0.9$, $p_m = 0.9$, $R_{br} = 0.1$, and $\delta = 1$ are tabulated in Tables 1 and 2, respectively.

Consider Bratu's problems (3) and (4), when $\mu = 1$ and $\mu = 2$. Using continuous GA technique, taking $x_i = i/N$, $i = 0, 1, 2, \ldots, N$, with the fitness function (12) and the previous termination conditions, the numerical results of approximating $y(x_i)$, when $\mu = 1$ and $\mu = 2$ at $N = 10$, $N_p = 500$, $p_c = 0.9$, $p_m = 0.9$, $R_{br} = 0.1$, and $\delta = 1$ are tabulated in Tables 3 and 4, respectively.

Numerical comparisons for Troesch's and Bratu's problems are studied next. The conventional numerical methods that are used for comparison of Troesch's problem with continuous GA include the Laplace transform decomposition method [11], decomposition method [13], homotopy perturbation method [14], and variational iteration method [15], while on the other hand, the conventional numerical

TABLE 3: Numerical results for Bratu's problem when $\mu = 1$.

x_i	Exact value	Approximate value	$|Err(x_i)|$	$|Res(x_i)|$
0	0	0	0	0
0.1	0.0498467900	0.0498469017	$1.11747073 \times 10^{-7}$	$1.46008376 \times 10^{-6}$
0.2	0.0891899350	0.0891901414	$2.06424850 \times 10^{-7}$	$3.90633337 \times 10^{-6}$
0.3	0.1176090956	0.1176093611	$2.65491661 \times 10^{-7}$	$2.42396771 \times 10^{-6}$
0.4	0.1347902526	0.1347905498	$2.97185578 \times 10^{-7}$	$1.58574073 \times 10^{-6}$
0.5	0.1405392142	0.1405395202	$3.05996395 \times 10^{-7}$	$2.33841807 \times 10^{-6}$
0.6	0.1347902526	0.1347905431	$2.90489351 \times 10^{-7}$	$2.69406948 \times 10^{-6}$
0.7	0.1176090956	0.1176093394	$2.43813586 \times 10^{-7}$	$1.71196336 \times 10^{-6}$
0.8	0.0891899350	0.0891901126	$1.77595908 \times 10^{-7}$	$1.13445094 \times 10^{-6}$
0.9	0.0498467900	0.0498468894	$9.94223941 \times 10^{-8}$	$1.95960534 \times 10^{-6}$
1	0	0	0	0

TABLE 4: Numerical results for Bratu's problem when $\mu = 2$.

x_i	Exact value	Approximate value	$|Err(x_i)|$	$|Res(x_i)|$
0	0	0	0	0
0.1	0.1144107440	0.1144123667	$1.62268051 \times 10^{-6}$	$3.02045290 \times 10^{-5}$
0.2	0.2064191156	0.2064220793	$2.96373587 \times 10^{-6}$	$5.97375566 \times 10^{-5}$
0.3	0.2738793116	0.2738831613	$3.84970866 \times 10^{-6}$	$3.98775778 \times 10^{-5}$
0.4	0.3150893646	0.3150937394	$4.37480887 \times 10^{-6}$	$3.42529172 \times 10^{-5}$
0.5	0.3289524214	0.3289570153	$4.59390598 \times 10^{-6}$	$4.43231785 \times 10^{-5}$
0.6	0.3150893646	0.3150937944	$4.42984785 \times 10^{-6}$	$4.44801409 \times 10^{-5}$
0.7	0.2738793116	0.2738831212	$3.80957282 \times 10^{-6}$	$2.65423819 \times 10^{-5}$
0.8	0.2064191156	0.2064219434	$2.82777504 \times 10^{-6}$	$2.22101666 \times 10^{-5}$
0.9	0.1144107440	0.1144123281	$1.58406997 \times 10^{-6}$	$3.81609672 \times 10^{-5}$
1	0	0	0	0

TABLE 5: Absolute error results for Troesch's problem when $\lambda = 0.5$.

x_i	Method in [11]	Method in [13]	Method in [14]	Method in [15]	Continuous GA
0	0	0	0	0	0
0.1	7.6745×10^{-4}	7.6145×10^{-4}	7.6266×10^{-4}	4.8651×10^{-3}	5.1448×10^{-8}
0.2	1.4949×10^{-3}	1.4842×10^{-3}	1.4855×10^{-3}	9.7001×10^{-3}	1.4574×10^{-7}
0.3	2.1410×10^{-3}	2.1269×10^{-3}	2.1273×10^{-3}	1.4475×10^{-2}	6.3461×10^{-8}
0.4	2.6619×10^{-3}	2.6458×10^{-3}	2.6446×10^{-3}	1.9154×10^{-2}	3.9087×10^{-8}
0.5	3.0098×10^{-3}	2.9929×10^{-3}	2.9900×10^{-3}	2.3704×10^{-2}	7.3312×10^{-8}
0.6	3.1313×10^{-3}	3.1150×10^{-3}	3.1108×10^{-3}	2.8080×10^{-2}	6.9525×10^{-8}
0.7	2.9660×10^{-3}	2.9517×10^{-3}	2.9471×10^{-3}	3.2235×10^{-2}	3.3133×10^{-8}
0.8	2.4448×10^{-3}	2.4338×10^{-3}	2.4300×10^{-3}	3.6110×10^{-2}	4.1656×10^{-8}
0.9	1.4872×10^{-3}	1.4810×10^{-3}	1.4792×10^{-3}	3.9641×10^{-2}	5.7727×10^{-8}
1	0	1.1000×10^{-9}	0	4.2740×10^{-2}	0

methods that are used for comparison of Bratu's problem with continuous GA include the Laplace transform decomposition method [12], decomposition method [13], B-spline method [16], and Lie-group shooting method [17]. Tables 5 and 6 show a comparison between the absolute errors of our method together with other aforementioned methods for Troesch's problem, while Tables 7 and 8 show a comparison for Bratu's problem.

As it is evident from the comparison results, it was found that our method in comparison with the mentioned methods is much better with a view to accuracy and utilization. Also, from the tables, it can be seen that the continuous GA method is consistent with the accurate approximate solutions.

6. Statistical and Convergence Analysis

Throughout this paper, we will try to give the results of Troesch's and Bratu's problems; however, in some cases we will switch between the results obtained for the problems in

TABLE 6: Absolute error results for Troesch's problem when $\lambda = 1$.

x_i	Method in [11]	Method in [13]	Method in [14]	Method in [15]	Continuous GA
0	0	0	0	0	0
0.1	2.8661×10^{-3}	2.4518×10^{-3}	2.5847×10^{-3}	1.8370×10^{-2}	8.9597×10^{-7}
0.2	5.8663×10^{-3}	4.8998×10^{-3}	5.0899×10^{-3}	3.6808×10^{-2}	2.8813×10^{-6}
0.3	8.2321×10^{-3}	7.2471×10^{-3}	7.4256×10^{-3}	5.5374×10^{-2}	1.3540×10^{-6}
0.4	1.0498×10^{-2}	9.3535×10^{-3}	9.4785×10^{-3}	7.4109×10^{-2}	6.3314×10^{-7}
0.5	1.2262×10^{-2}	1.1055×10^{-2}	1.1095×10^{-2}	9.3026×10^{-2}	1.4848×10^{-6}
0.6	1.3272×10^{-2}	1.2092×10^{-2}	1.2056×10^{-2}	1.1209×10^{-1}	1.3799×10^{-6}
0.7	1.3171×10^{-2}	1.2113×10^{-2}	1.2039×10^{-2}	1.3119×10^{-1}	4.6468×10^{-7}
0.8	1.1454×10^{-2}	1.0620×10^{-2}	1.0565×10^{-2}	1.5012×10^{-1}	8.2235×10^{-7}
0.9	7.4049×10^{-3}	6.9387×10^{-3}	6.9135×10^{-3}	1.6849×10^{-1}	1.8516×10^{-6}
1	0	1.8020×10^{-6}	0	1.8565×10^{-1}	0

TABLE 7: Absolute error results for Bratu's problem when $\mu = 1$.

x_i	Method in [12]	Method in [13]	Method in [16]	Method in [17]	Continuous GA
0	0	0	0	0	0
0.1	1.9788×10^{-6}	2.6851×10^{-3}	2.9797×10^{-6}	7.5085×10^{-7}	1.1175×10^{-7}
0.2	3.9394×10^{-6}	2.0219×10^{-3}	5.4660×10^{-6}	1.0182×10^{-6}	2.0642×10^{-7}
0.3	5.8548×10^{-6}	1.5234×10^{-4}	7.3357×10^{-6}	9.0475×10^{-7}	2.6549×10^{-7}
0.4	7.7038×10^{-6}	2.2017×10^{-3}	8.4967×10^{-6}	5.2393×10^{-7}	2.9719×10^{-7}
0.5	9.4665×10^{-6}	3.0155×10^{-3}	8.8921×10^{-6}	5.0669×10^{-9}	3.0600×10^{-7}
0.6	1.1112×10^{-5}	2.2017×10^{-3}	8.4967×10^{-6}	5.1386×10^{-7}	2.9049×10^{-7}
0.7	1.2572×10^{-5}	1.5234×10^{-4}	7.3357×10^{-6}	8.9485×10^{-7}	2.4381×10^{-7}
0.8	1.3475×10^{-5}	2.0219×10^{-3}	5.4660×10^{-6}	1.0086×10^{-6}	1.7760×10^{-7}
0.9	1.1968×10^{-5}	2.6851×10^{-3}	2.9797×10^{-6}	7.4160×10^{-7}	9.9422×10^{-8}
1	0	0	0	0	0

TABLE 8: Absolute error results for Bratu's problem with $\mu = 2$.

x_i	Method in [12]	Method in [13]	Method in [16]	Method in [17]	Continuous GA
0	0	0	0	0	0
0.1	2.1290×10^{-3}	1.5217×10^{-2}	1.7179×10^{-5}	4.0341×10^{-6}	1.6227×10^{-6}
0.2	4.2097×10^{-3}	1.4675×10^{-2}	3.2597×10^{-5}	5.7027×10^{-6}	2.9637×10^{-6}
0.3	6.1868×10^{-3}	5.8878×10^{-3}	4.4899×10^{-5}	5.2212×10^{-6}	3.8497×10^{-6}
0.4	8.0019×10^{-3}	3.2466×10^{-3}	5.2858×10^{-5}	3.0749×10^{-6}	4.3748×10^{-6}
0.5	9.5992×10^{-3}	6.9851×10^{-3}	5.5614×10^{-5}	1.4554×10^{-8}	4.5939×10^{-6}
0.6	1.0930×10^{-2}	3.2466×10^{-3}	5.2858×10^{-5}	3.0464×10^{-6}	4.4298×10^{-6}
0.7	1.1933×10^{-2}	5.8878×10^{-3}	4.4899×10^{-5}	5.1946×10^{-6}	3.8096×10^{-6}
0.8	1.2378×10^{-2}	1.4675×10^{-2}	3.2597×10^{-5}	5.6787×10^{-6}	2.8278×10^{-6}
0.9	1.0873×10^{-2}	1.5217×10^{-2}	1.7179×10^{-5}	4.0135×10^{-6}	1.5841×10^{-6}
1	0	0	0	0	0

order not to increase the length of the paper without the loss of generality for the remaining cases and 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 [32–40]. However, the convergence data of Troesch's and Bratu's problems are given in Table 9.

The evolutionary progress plots of the best-fitness individual of Troesch's and Bratu's problems are shown in Figures 2 and 3, respectively. It is to be noted from the evolutionary plots that the best-fitness function approaches one very fast in the first stage of computations after that the best-fitness function reaches steady-state values and no further improvement in the fitness value. This means that the approximate of continuous GA converges to the actual solution very fast in the first stage of computations.

TABLE 9: Convergence data of Troesch's and Bratu's problems.

Case	$\overline{Gen}$	$\overline{Fit}$	$\overline{|Err|}$	$\overline{|Res|}$
$\lambda = 0.5$	1051	0.99999503	$6.38988847 \times 10^{-8}$	$5.93727586 \times 10^{-7}$
$\lambda = 1$	1231	0.99992419	$1.30753614 \times 10^{-6}$	$3.45671315 \times 10^{-4}$
$\mu = 1$	1134	0.99999255	$2.22018533 \times 10^{-7}$	$2.13495920 \times 10^{-6}$
$\mu = 2$	1313	0.99990899	$3.33956728 \times 10^{-6}$	$3.47339266 \times 10^{-5}$

(a)

(b)
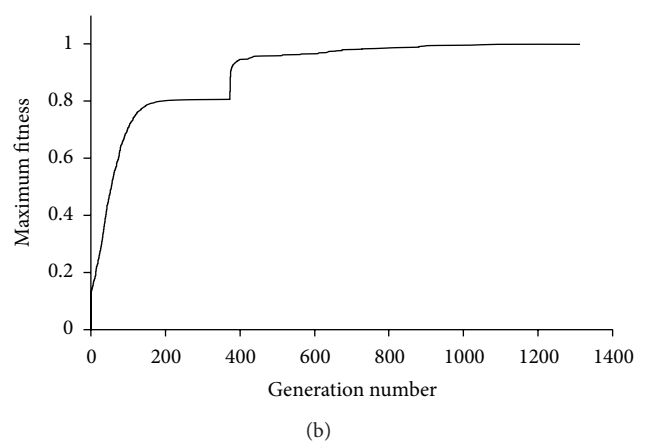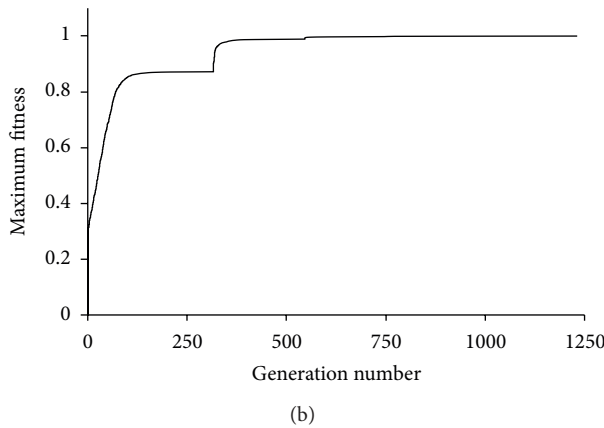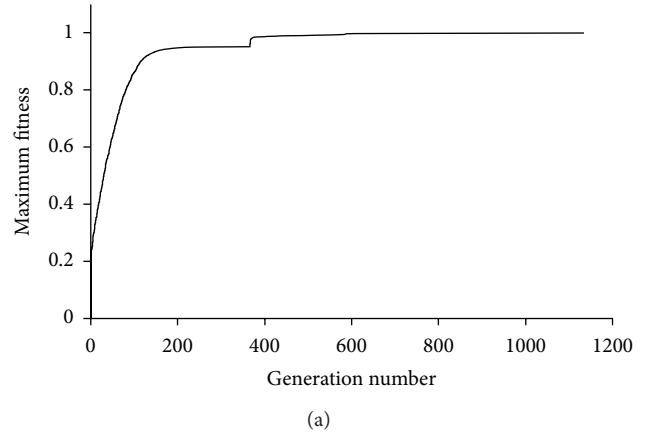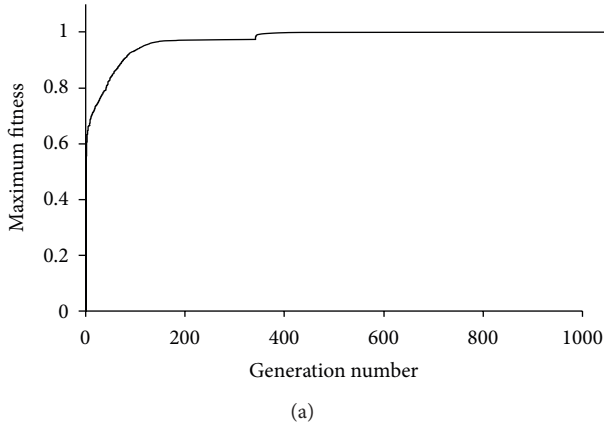
FIGURE 2: Plot of the best-fitness function for Troesch's problem when (a) $\lambda = 0.5$ and (b) $\lambda = 1$.

(a)

(b)

FIGURE 3: Plot of the best-fitness function for Bratu's problem when (a) $\mu = 1$ and (b) $\mu = 2$.

The way in which the nodal values evolve is studied next. Figure 4 shows the evolution of the first, x_1, middle, x_5, and ninth, x_9, nodal values for Troesch's problem, while Figure 5 shows the evolution of the same nodal values for Bratu's problem.

It is observed 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 problem. As a result, all nodes, in the same problem, reach the near-optimal solution together.

The effect of the different types of initialization methods on the convergence speed of the algorithm is discussed next.

Three initialization methods are investigated in this paper; the first method uses the modified normal Gaussian function (MNGF), the second uses the modified tangent hyperbolic function (MTHF), and the third is the mixed-type initialization method that initializes the first half of the population using the MNGF and the second half of the population using the MTHF [32–40]. Table 10 shows that the use of the initialization method has a minor effect on the convergence speed because usually the effect of the initial population dies after few tens of generations and the convergence speed after that is governed by the selection mechanism, crossover, and mutation operators [32–40]. For Troesch's problem, the MTHF results in the fastest convergence speed while for Bratu's problem, the MNGF results in the fastest convergence speed. However, for a specific problem, the initialization method with the highest convergence speed is the one that provides initial solution curves which are close
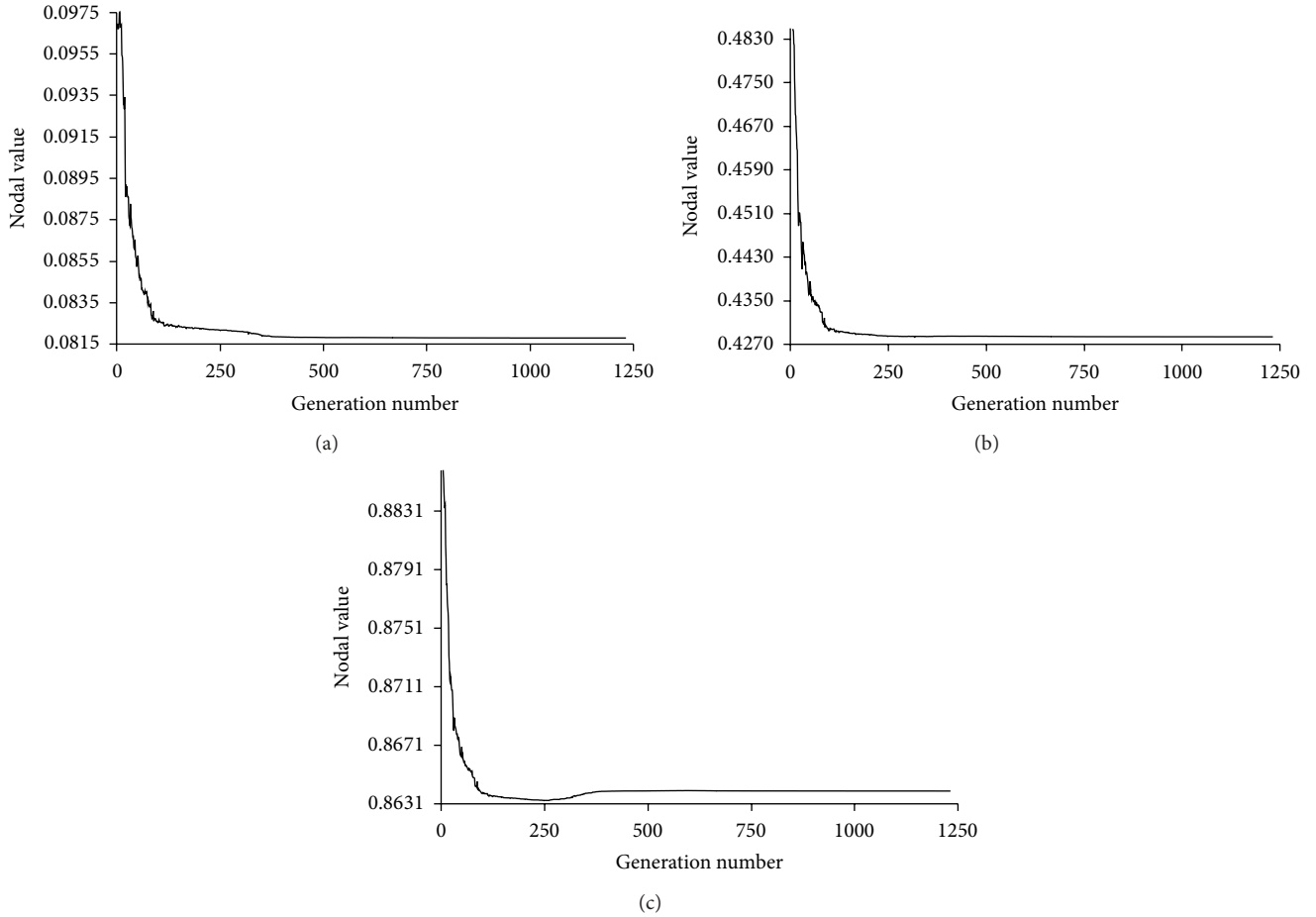
(a)

(b)

(c)

FIGURE 4: Evolution of the nodal values for Troesch's problem when $\lambda = 1$ for (a) $y(x_1)$, (b) $y(x_5)$, and (c) $y(x_9)$.

TABLE 10: The convergence speed of the algorithm using different initialization functions for Troesch's and Bratu's problems.

Troesch's problem	MNGF	MTHF	Mixed	Bratu's problem	MNGF	MTHF	Mixed
$\lambda = 0.5$	1110	917	1051	$\mu = 1$	1019	1220	1134
$\lambda = 1$	1369	1105	1231	$\mu = 2$	1202	1404	1313

TABLE 11: The effect of the vector norm on the convergence speed of the algorithm for Troesch's and Bratu's problems.

Troesch's problem	L_1 norm	L_2 norm	Bratu's problem	L_1 norm	L_2 norm
$\lambda = 0.5$	1107	1051	$\mu = 1$	1298	1134
$\lambda = 1$	1324	1231	$\mu = 2$	1411	1313

to the optimal solution of that problem; that is, the optimal solution of Troesch's problem is close to the MTHF and the optimal solution of Bratu's problem is close to the MNGF. However, since the optimal solution of any given problem is not assumed to be known, it is better to have a diverse initial population by the use of the mixed-type initialization method. As a result, the mixed-type initialization method is used as the algorithm default method [32–40]. The reader is asked to refer to [32–40] in order to know more details about the initialization methods used in continuous GA, including their kinds and types and their justification and conditions for use.

The effect of the vector norm used in the fitness evaluation is studied here. Two vector norms are used: L_1 norm and L_2 norm. The L_1 norm is governed by the equation

$$\text{Oir} = |\text{Res}(1)| + \sum_{i=2}^{N-2} |\text{Res}(i)| + |\text{Res}(N-1)|, \qquad (13)$$

while L_2 norm is governed by (11). Figure 6 shows the evolutionary progress plots for the best-of-generation individual for Troesch's problem when $\lambda = 1$ and Bratu's problem when $\mu = 2$ using L_1 and L_2 norms, while Table 11 gives the convergence speed for Troesch's and Bratu's problems
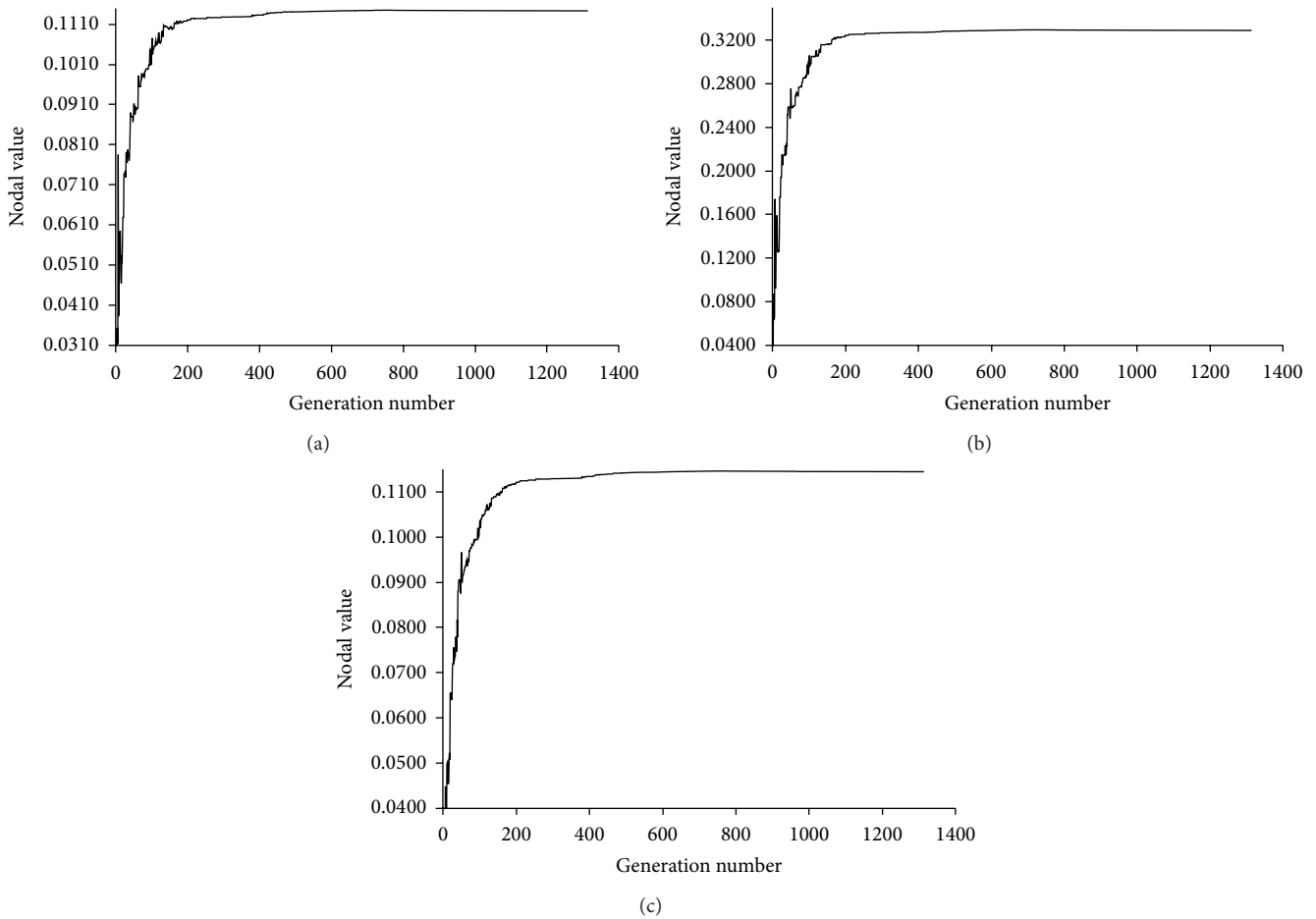
(a)

(b)

(c)

Figure 5: Evolution of the nodal values for Bratu's problem when $\mu = 2$ for (a) $y(x_1)$, (b) $y(x_5)$, and (c) $y(x_9)$.

for two different cases. Two observations are made in this regard. First, the evolutionary progress plots of both norms show that L_2 norm has higher fitness values than those of L_1 norm throughout the evolution process. Second, L_2 norm converges a little bit faster than L_1 norm. The key factor behind these observations is the square power appearing in L_2 norm. Regarding the first observation, it is known that for a given set of nodal residuals with values less than 1, L_1 norm results in a higher value than L_2 norm and correspondingly the fitness value using L_2 norm will be higher than that using L_1 norm. Regarding the second observation, L_2 norm tries to select individual solutions, vectors, with distributed nodal residuals among the nodes rather than lumped nodal residuals where one nodal residual is high and the remaining nodal residuals are relatively small. This distributed selection scheme results in closer solutions to the optimal one than the lumped selection scheme. In addition to that, the crossover operator will be more effective in the former case than in the latter one. These two points result in the faster convergence speed in L_2 norm as compared with L_1 norm. Furthermore, it is observed that L_2 norm is less sensitive to variations in the genetic related parameters and problem related parameters. As a result, L_2 norm is preferred over L_1 norm and it is used as the algorithm's default norm [32–40].

The particular settings of several continuous GA tuning parameters including the probabilities of applying crossover operator and mutation operator are investigated now. 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. Figure 7 shows the effect of the crossover probability, p_c, and the mutation probability, p_m, on the convergence speed of the algorithm for Troesch's problem when $\lambda = 0.5$. It is clear from Figure 7 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. Also, it is noted that the best performance of the algorithm is achieved when $p_c = 0.9$ and $p_m = 0.9$. As a result, these values are set as the algorithm default values [32–40].
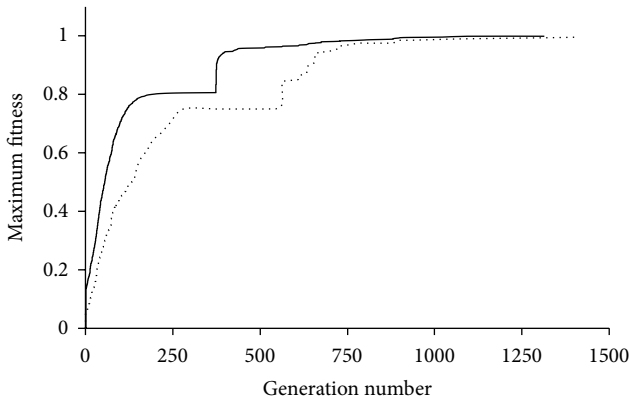
The influence of the population size on the convergence speed, the average fitness, and the corresponding errors of continuous GA is studied next for Troesch's problem when $\lambda = 1$ as shown in Figure 8. Small population sizes suffer from larger number of generations required for convergence and the probability of being trapped in local minima, while large population sizes suffer from larger number of fitness evaluations that means larger execution time. However, it

TABLE 12: The influence of the number of nodes on the convergence speed, the average fitness, and the corresponding errors of the algorithm for Bratu's problems when $\mu = 1$.

N	$\overline{\text{Gen}}$	$\overline{\text{Fit}}$	$\overline{|\text{Err}|}$	$\overline{|\text{Res}|}$
5	601	0.99999077	$1.00934085 \times 10^{-5}$	$8.99016147 \times 10^{-4}$
10	1134	0.99999255	$2.22018533 \times 10^{-7}$	$2.13495920 \times 10^{-5}$
20	2409	0.99999191	$1.16620506 \times 10^{-9}$	$9.58206567 \times 10^{-8}$
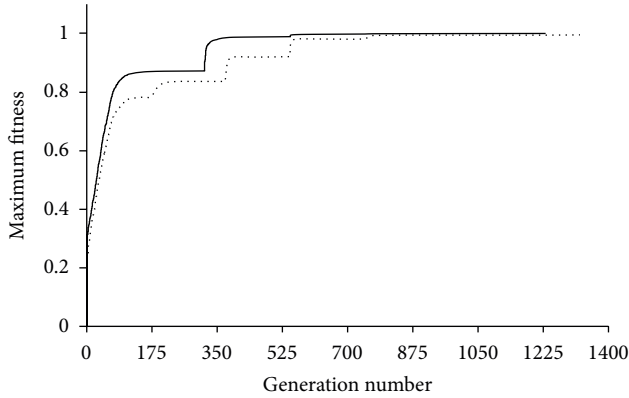
(a)

(b)

FIGURE 6: Evolutionary progress plots of best-fitness function using L_1: ($\ldots$) and L_2: (—) vector norms for (a) Troesch's problem, when $\lambda = 1$, and (b) Bratu's problem, when $\mu = 2$.

FIGURE 7: The effect of the crossover and mutation probabilities on the convergence speed of the algorithm for Troesch's problem when $\lambda = 0.5$.

is noted that the improvement in the convergence speed becomes almost negligible after a population size of 700.

Now, the effect of the number of nodes on the convergence speed, the average fitness, and the corresponding errors is explored. Table 12 gives the relevant data for Bratu's problems when $\mu = 1$. 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 solution. 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 increasing the number of nodes from 5 to 10 to 20, the required number of
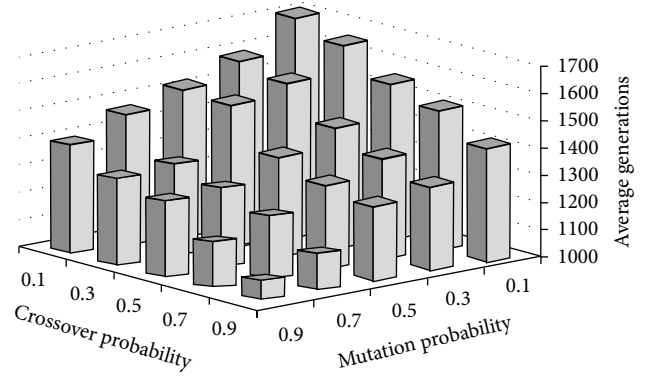
generations for convergence jumps from almost 600 to 1130 to 2400, that is, 1.89 to 2.21 multiplication factor.

Finally, the effect of the most commonly used selection schemes by GA community of the performance on the continuous GA is investigated. Table 13 represents the effect of selection schemes on the convergence speed, the average fitness, and the corresponding errors for Bratu's problem when $\mu = 2$. It is clear from Table 13 that the rank-based selection scheme [49] has the faster convergence speed. The tournament selection with replacement [49] and tournament selection without replacement [50] approaches come in the second place with almost similar convergence speeds. It is obvious that the roulette wheel [30], stochastic universal [50], and half biased selection [51] schemes have slower convergence speed of the rest of the methods. The half biased selection scheme has the slowest convergence speed. The reader is kindly requested to go through [32–40] for more details about the selection scheme used in the algorithm.

7. Summery

This paper has introduced a new optimization technique based on the use of continuous GA where two smooth solution curves are used for representing the required nodal values. The continuous GA was found to be accurate in which the boundary conditions of the problems are satisfied. Simulation results are carried out in order to verify the mathematical results and the theoretical statements for the optimized solutions. The applicability and efficiency of the proposed algorithm for the solution of different cases of Troesch's and Bratu's problems are investigated. On the other aspect as well, the effect of different parameters, including
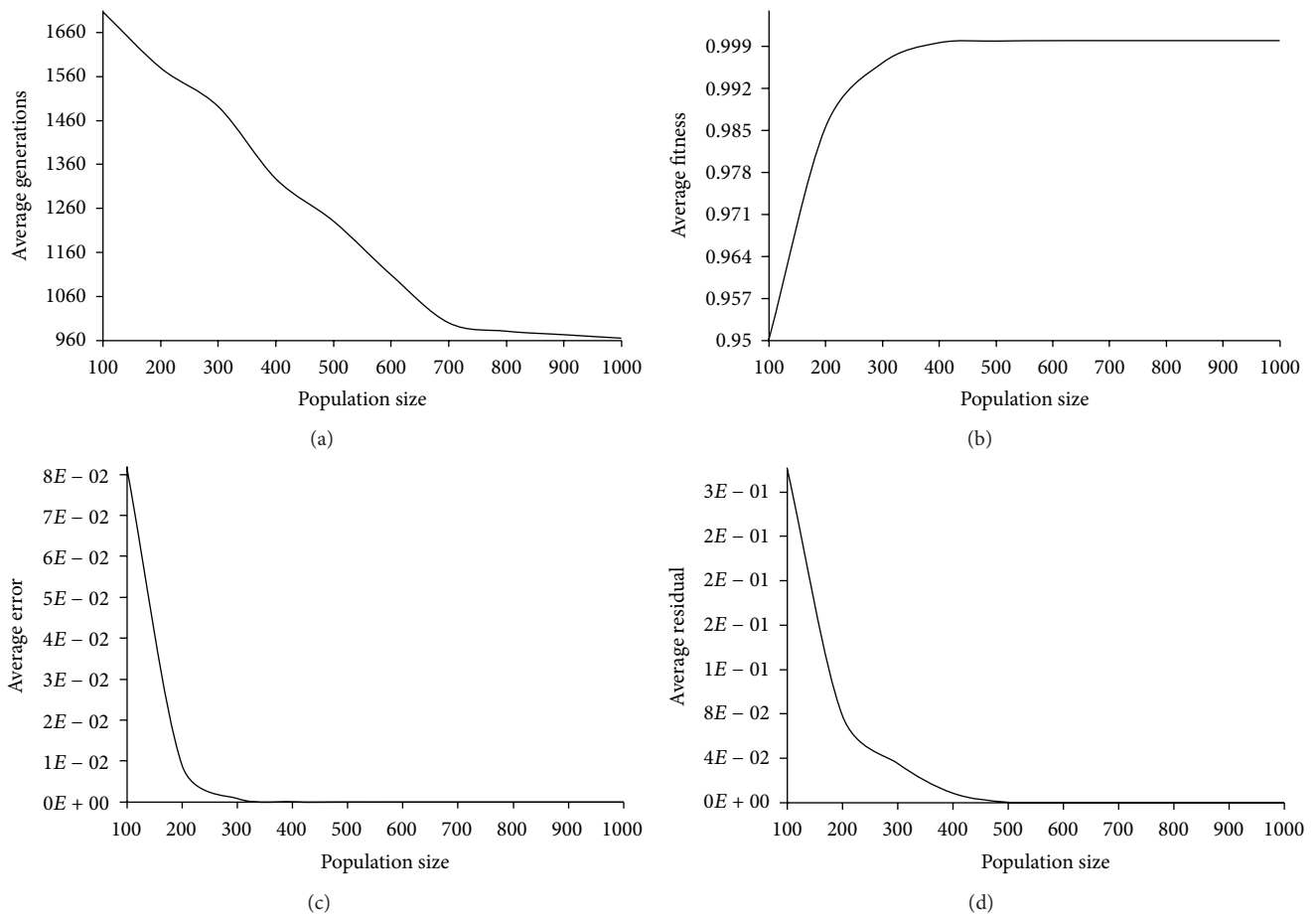
(a)

(b)

(c)

(d)

FIGURE 8: The effect of the population size of the algorithm for Troesch's problem when $\lambda = 1$ on the (a) $\overline{\text{Gen}}$, (b) $\overline{\text{Fit}}$, (c) $\overline{|\text{Err}|}$, and (d) $\overline{|\text{Res}|}$.

TABLE 13: The influence of selection schemes on the convergence speed, the average fitness, and the corresponding errors for Bratu's problems when $\mu = 2$.

Selection method	$\overline{\text{Gen}}$	$\overline{\text{Fit}}$	$\overline{\text{Err}}$	$\overline{\text{Res}}$
Rank-based	1313	0.99990899	$3.33956728 \times 10^{-6}$	$3.47339266 \times 10^{-5}$
Tournament with replacement	1404	0.99998805	$1.32758514 \times 10^{-5}$	$0.10260139 \times 10^{-4}$
Tournament without replacement	1429	0.99999092	$1.00934084 \times 10^{-5}$	$0.11360136 \times 10^{-4}$
Roulette wheel	1699	0.99999184	$9.06686522 \times 10^{-5}$	$0.10602135 \times 10^{-4}$
Stochastic universal	1880	0.99998117	$2.09275009 \times 10^{-4}$	$0.11740260 \times 10^{-3}$
Half biased	2008	0.99998950	$1.16620506 \times 10^{-3}$	$0.13856138 \times 10^{-2}$

the evolution of nodal values, the initialization method, the selection method, the vector norm used, the crossover and mutation probabilities, the population size, and the step size, is studied.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] S. M. Roberts and J. S. Shipman, "On the closed form solution of Troesch's problem," *Journal of Computational Physics*, vol. 21, no. 3, pp. 291–304, 1976.

[2] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 1995.

[3] E. S. Weibel, *The Plasma in Magnetic Field*, Edited by Landshoff R. K. M, Stanford University Press, Stanford, Calif, USA, 1958.

[4] E. S. Weibel, "On the confinement of a plasma by magnetostatic fields," *Physics of Fluids*, vol. 2, pp. 52–56, 1959.

[5] V. S. Markin, A. A. Chernenko, Y. A. Chizmadehev, and Y. G. Chirkov, "Aspects of the theory of gas porous electrodes," in *Fuel Cells: Their Electrochemical Kinetics, Consultants Bureau*, V. S. Bagotskii and Y. B. Vasilev, Eds., pp. 21–33, New York, NY, USA, 1966.

[6] D. Gidaspow and B. S. Baker, "A model for discharge of storage batteries," *Journal of the Electrochemical Society*, vol. 120, pp. 1005–1010, 1973.

[7] G. Bratu, "Sur certaines équations intégrales non linéares," *Comptes Rendus*, vol. 150, pp. 896–899, 1910.

[8] R. Buckmire, "Application of a Mickens finite-difference scheme to the cylindrical Bratu-Gelfand problem," *Numerical Methods for Partial Differential Equations*, vol. 20, no. 3, pp. 327–337, 2004.

[9] J. S. McGough, "Numerical continuation and the Gelfand problem," *Applied Mathematics and Computation*, vol. 89, no. 1–3, pp. 225–239, 1998.

[10] A. S. Mounim and B. M. de Dormale, "From the fitting techniques to accurate schemes for the Liouville-Bratu-Gelfand problem," *Numerical Methods for Partial Differential Equations*, vol. 22, no. 4, pp. 761–775, 2006.

[11] S. A. Khuri, "A numerical algorithm for solving Troesch's problem," *International Journal of Computer Mathematics*, vol. 80, no. 4, pp. 493–498, 2003.

[12] S. A. Khuri, "A new approach to Bratu's problem," *Applied Mathematics and Computation*, vol. 147, no. 1, pp. 131–136, 2004.

[13] E. Deeba, S. A. Khuri, and S. Xie, "An algorithm for solving boundary value problems," *Journal of Computational Physics*, vol. 159, no. 2, pp. 125–138, 2000.

[14] X. Feng, L. Mei, and G. He, "An efficient algorithm for solving Troesch's problem," *Applied Mathematics and Computation*, vol. 189, no. 1, pp. 500–507, 2007.

[15] S. Momani, S. Abuasad, and Z. Odibat, "Variational iteration method for solving nonlinear boundary value problems," *Applied Mathematics and Computation*, vol. 183, no. 2, pp. 1351–1358, 2006.

[16] H. Caglar, N. Caglar, M. Özer, A. Valarıstos, and A. N. Anagnostopoulos, "B-spline method for solving Bratu's problem," *International Journal of Computer Mathematics*, vol. 87, no. 8, pp. 1885–1891, 2010.

[17] S. Abbasbandy, M. S. Hashemi, and C.-S. Liu, "The Lie-group shooting method for solving the Bratu equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 11, pp. 4238–4249, 2011.

[18] S. T. Cheng, "Topological optimization of a reliable communication network," *IEEE Transactions on Reliability*, vol. 47, pp. 225–233, 1998.

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

[20] J. Burm, "Optimization of high-speed metal semiconductor metal photodetectors," *IEEE Photonics Technology Letters*, vol. 6, pp. 722–724, 1994.

[21] A. Vossinis, "Shape optimization of aerodynamics using nonlinear generalized minimal residual algorithm," *Optimal Control Applications & Methods*, vol. 16, pp. 229–249, 1995.

[22] R. Fondacci, "Combinatorial issues in air traffic optimization," *Transportation Science*, vol. 32, pp. 256–267, 1998.

[23] E. de Klerk, C. Roos, T. Terlaky et al., "Optimization of nuclear reactor reloading patterns," *Annals of Operations Research*, vol. 69, pp. 65–84, 1997.

[24] Y. Cherruault, "Global optimization in biology and medicine," *Mathematical and Computer Modelling*, vol. 20, no. 6, pp. 119–132, 1994.

[25] J. G. Rowse, "On the solution of optimal tax models and other optimization models in economics," *Economics Letters*, vol. 18, pp. 217–222, 1985.

[26] G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell, *Engineering Optimization*, John Wiley & Sons, New York, NY, USA, 1983, Methods and applications.

[27] M. Sebag and A. Ducoulombier, "Extending population-based incremental learning to continuous search spaces," in *Parallel Problem Solving from Nature*, pp. 418–427, 1998.

[28] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Pena, "Optimization in continuous domains by learning and simulation of Gaussian networks," in *Proceedings of the Genetic and Evolutionary Computation Conference Workshop Program*, pp. 201–204, Las Vegas, Nev, USA, 2000.

[29] P. A. N. Bosnian and D. Thierens, "Expanding from discrete to continuous estimation of distribution algorithms: the IDEA," in *Parallel Problem Solving from Nature*, pp. 767–776, 2000.

[30] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[31] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, John Wiley & Sons, 2004.

[32] 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, Islamabad, Pakistan, 2002.

[33] Z. S. Abo-Hammour, "A novel continuous genetic algorithms for the solution of the cartesian path generation problem of robot manipulators," in *Robot Manipulators: New Research*, J. X. Lui, Ed., pp. 133–190, Nova Science Publishers, New York, NY, USA, 2005.

[34] Z. S. Abo-Hammour, N. Mirza, S. Mirza, and M. Arif, "Cartesian path planning of robot manipulators using continuous genetic algorithms," *Robotics and Autonomous Systems*, vol. 41, pp. 179–223, 2002.

[35] Z. S. Abo-Hammour, O. Alsmadi, S. I. Bataineh, M. A. Al-Omari, and N. Affach, "Continuous genetic algorithms for collision-free cartesian path planning of robot manipulators," *International Journal of Advanced Robotic Systems*, vol. 8, pp. 14–36, 2011.

[36] Z. S. Abo-Hammour, M. Yusuf, N. Mirza, S. Mirza, M. Arif, and J. Khurshid, "Numerical solution of second-order, two-point boundary value problems using continuous genetic algorithms," *International Journal for Numerical Methods in Engineering*, vol. 61, pp. 1219–1242, 2004.

[37] Z. S. Abo-Hammour, A. Al-Asasfeh, A. Al-Smadi, and O. Alsmadi, "A novel continuous genetic algorithm for the solution of optimal control problems," *Optimal Control Applications and Methods*, vol. 32, pp. 414–432, 2010.

[38] O. Abu Arqub, Z. Abo-Hammour, S. Momani, and N. Shawagfeh, "Solving singular two-point boundary value problems using continuous genetic algorithm," *Abstract and Applied Analysis*, vol. 2012, Article ID 205391, 25 pages, 2012.

[39] O. Abu Arqub, Z. S. Abo-Hammour, and S. Momani, "Application of continuous genetic algorithm for nonlinear system of second-order boundary value problems," *Applied Mathematics and Information Sciences*, vol. 8, pp. 235–248, 2014.

[40] O. Abu Arqub, *Numerical solution of fuzzy differential equation using continuous genetic algorithms [Ph.D. thesis]*, University of Jordan, Amman, Jordan, 2008.

[41] Z. Abo-Hammour, O. Alsmadi, S. Momani, and O. Abu Arqub, "A genetic algorithm approach for prediction of linear dynamical systems," *Mathematical Problems in Engineering*, vol. 2013, Article ID 831657, 12 pages, 2012.

[42] O. Abu Arqub, M. Al-Smadi, and S. Momani, "Application of reproducing kernel method for solving nonlinear Fredholm-Volterra integrodifferential equations," *Abstract and Applied Analysis*, vol. 2012, Article ID 839836, 16 pages, 2012.

[43] M. Al-Smadi, O. Abu Arqub, and S. Momani, "A computational method for two-point boundary value problems of fourth-order mixed integrodifferential equations," *Mathematical Problems in Engineering*, vol. 2013, Article ID 832074, 10 pages, 2013.

[44] O. Abu-Arqub, A. El-Ajou, S. Momani, and N. Shawagfeh, "Analytical solutions of fuzzy initial value problems by HAM," *Applied Mathematics & Information Sciences*, vol. 7, no. 5, pp. 1903–1919, 2013.

[45] O. A. Arqub, M. Al-Smadi, and N. Shawagfeh, "Solving Fredholm integro-differential equations using reproducing kernel Hilbert space method," *Applied Mathematics and Computation*, vol. 219, no. 17, pp. 8938–8948, 2013.

[46] O. Abu Arqub, A. El-Ajou, A. S. Bataineh, and I. Hashim, "A representation of the exact solution of generalized Lane-Emden equations using a new analytical method," *Abstract and Applied Analysis*, vol. 2013, Article ID 378593, 10 pages, 2013.

[47] O. Abu Arqub, Z. Abo-Hammour, R. Al-Badarneh, and S. Momani, "A reliable analytical method for solving higher-order initial value problems," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 673829, 12 pages, 2013.

[48] N. Shawagfeh, O. Abu Arqub, and S. Momani, "Analytical solution of nonlinear second-order periodic boundary value problem using reproducing kernel method," *Journal of Computational Analysis and Applications*, vol. 16, pp. 750–762, 2014.

[49] J. H. Jiang, J. H. Wang, X. Chu, and R. Q. Yu, "Clustering data using a modified integer genetic algorithm," *Analytica Chimica Acta*, vol. 354, pp. 263–274, 1997.

[50] E. M. Rudnick, J. H. Patel, G. S. Greenstein, and T. M. Niermann, "A genetic algorithm framework for test generation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, pp. 1034–1044, 1997.

[51] J. R. Vanzandt, "A Genetic Algorithm for Search Route Planning," Tech. Rep. ESD-TR-92-262, United States Air Force, 1992.