Numerical Solution of Stiff Initial Value Problems Using Continuous Genetic Algorithms

Omar A. AbuArqob
Al-Balqa' Applied University, Jordan
E-mail: omar_abuarqob@bau.edu.jo

Zaer S. Abo-Hammour
University of Jordan, Jordan
E-mail: zaer_hmr@hotmail.com

ABSTRACT

The stiff initial value problems occur in many fields of engineering science, particularly in the studies of electrical circuits, vibrations, chemical reactions, and so on. In most cases, the model is too complex to allow one to find an exact solution: an efficient, reliable computer simulation is required. The techniques commonly used for stiff problems are implicit multistep methods. These methods are suited for linear problems. However, when solving the nonlinear problems, these method require some major modifications that include the use of some root finding technique. Furthermore, they require the use of other basic numerical techniques in order to obtain the solution. In this paper, a novel approach is proposed to solve stiff problems, which is based on the use of continuous genetic algorithms where smooth solution curves are used throughout the evolution of the algorithm to obtain the required nodal values. The proposed algorithm has the following distinct advantages over the conventional methods. First, it does not require any modification while switching from the linear to the nonlinear case. Second, its ability to solve stiff problems without the use of other numerical techniques. Third, the error in this method does not effect by the distance from the given initial value. Numerical example presented in this paper to illustrate the applicability, accuracy, and generality of the proposed method.

Key Words: Stiff Problems; Initial Value Problem; Continuous Genetic Algorithms; Finite Difference Approximation.

Mathematics Subject Classifications: 65Y20, 34K28, 46N40, 34A45.

1. Introduction

Continuous genetic algorithms (CGA's), depends on the evolution of curves in 2-dimensional space, and surfaces in 3-dimensional space. Generally, CGA's uses smooth operators and avoids sharp jumps in the parameter values. CGA's were developed, enhanced and applied for the solution of the Cartesian path generation problem of robotic manipulators, which is a highly nonlinear, coupled problem. Also it used to solve second order-two point boundary value problems. CGA's begin with a population of randomly generated candidates and evolve to-wards better solution by applying genetic operators which is reproduction; crossover and mutation, a procedure of genetic algorithms similar to the genetic processes which occur in nature. CGA's are a relatively new class of optimization technique, which are generating a growing interest in the engineering community. They are well suited for a broad range of problems encountered in science and engineering. The work presented in this paper is motivated by the needs for a new numerical method for the solution of the stiff problems with the following characteristics: First, it requires the minimal amount of informations about specific problems. Second, the method is not a mathematically guided scheme. Third, the algorithm is of global nature in terms of the solutions obtained as well as it is ability to solve other mathematical problems based on ordinary as well as partial differential equations. Fourth, it should not restore to more advanced mathematical tools; that is, the algorithm should be simple to understand and implement, and should be thus easily accepted in the engineering and mathematical application fields.

The concept of genetic algorithms was firstly proposed by Holland [8]. Smooth genetic algorithm introduced by Gutowski [7]. Abo-Hammour in his thesis [1] and in a research paper [3] developed the CGA's to solve a second-order, two-point boundary value problems. CGA's were developed, enhanced and applied for the solution of the Cartesian path generation problem of robotic manipulators [2]. AbuArqob in his thesis [4] developed the CGA's to solve a fuzzy initial value problems.

The reminder of the paper is organized as follows: the formulation of the CGA's is described in section 2. Description of the CGA's is covered in section 3. Numerical results are given in section 4. Finally, conclusion is presented in section 5.

2. Formulation of the Continuous Genetic Algorithms

In this section, a novel method for the solution of the stiff problems based on a CGA's is introduced. The proposed technique might be considered as a variation of the finite difference method in the sense that each derivatives in the system of ordinary differential equations is replaced by an appropriate difference-quotient approximation.

The general stiff problems discussed in this paper involves a system of first-order differential equation of the form: $H\left(t, Y[t], \dot{Y}[t]\right) = \dot{Y}[t] - F[t, Y[t]] = 0$, $Y[t_0] = Y_0$, $t \in T$, where $T = [t_0, t_f]$, $Y[t] = (Y_1[t], Y_2[t] ..., Y_m[t])$, and $F = (F_1, F_2, ..., F_m)$.

We first make the stipulation that the mesh points are equally distributed through the interval T. This condition is ensured by choosing positive integer N and selecting the mesh points $t_i = t_0 + ih$, for each $i = 0, 1, ..., N$, where the step size $h = (t_f - t_0)/N$ is the common distance between the points in the interval T. Thus, at the *interior mesh points*, t_i, $i = 1, 2, ..., N$, the system of ordinary differential equation to be approximated is given as:

$$H\left(t_i, \dot{Y}[t_i], Y[t_i]\right) = 0,\ Y[t_0] = Y_0,\ t_i \in T. \tag{I}$$

The continuous genetic algorithms approach for numerically approximating the solution to the stiff problem consists of replacing each derivative in Equation (I) by a appropriate finite-divided-difference formulas, which closely approximates that derivatives when h is small. The finite-divided-difference formulas with error up to $\mathrm{O}(h^n)$, where $n \in \mathbb{N}$, for approximating $\dot{Y}[t_i]$ for $i = 0, 1, ..., N$ can be obtained by using the following algorithm.

Algorithm 1 [8] *To approximate the derivative of the function $Y[t_i]$, $t_i \in T$ using an $(n+1)$-point formula, at $N+1$ equally spaced numbers in the intervals T, let $K = [n/2]$, then there are four steps:*

Step 1: For $i, j = 0, 1, ..., n$ set $d_{n+1,i,j} = (-1)^{i-j+1}\, i!\,(n-i)!/\left((j-i)\,j!\,(n-j)!\right)$ if $i \neq j$ and set $d_{n+1,i,i} = -\sum_{j=0, j\neq i}^{n} d_{n+1,i,j}$.

Step 2: For $i = 0, 1, .., K-1$ set $\dot{Y}[t_i] = \sum_{j=0}^{n} d_{n+1,i,j} Y[t_j]/h + \mathrm{O}(h^n)$.

Step 3: For $i = K, ..., N-K-1$ set $\dot{Y}[t_i] = \sum_{j=0}^{n} d_{n+1,K,j} Y[t_{i-K+j}]/h + O(h^n)$.

Step 4: For $i = N-K, ..., N$ set $\dot{Y}[t_i] = \sum_{j=0}^{n} d_{n+1,n+i-N,j} Y[t_{N-n+j}]/h + O(h^n)$.

Now, to complete the formulation of the stiff problem substituting the approximate value of $\dot{Y}[t_i]$ for $i = 1, 2, ..., N$ in Equation (I), discretized form of Equation (I) is obtained. The resulting equations will be a function of $Y[t_{i-n}]$, $Y[t_{i-(n-1)}]$,..., $Y[t_{i+n}]$, and t_i, where $n \in \mathbb{N}$.

After that, it is necessary to rewrite the discretized equation in the following form:

$$H(t_i, Y[t_{i-n}], ..., Y[t_{i+n}]) \approx 0, \quad i = 1, 2, ..., N.$$

The *residual* of the general interior node, i, denoted by Res(i), is defined for each $k = 1, 2, .., m$ as:

$$\text{Res}^k(i) = H_k(t_i, Y[t_{i-n}], ..., Y[t_{i+n}]), \quad i = 1, 2, ..., N.$$

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

$$\text{Oir} = \sqrt{\sum_{i=1}^{N} \left(\text{Res}^1(i)\right)^2 + ... + \sum_{i=1}^{N} \left(\text{Res}^m(i)\right)^2}.$$

A mapping of the overall individual residual, Oir, into a *fitness* function, Fit, is required in the CGA's in order to convert the minimization problem of Oir into a maximization problem of Fit. A suitable fitness function, used in this work is defined as:

$$\text{Fit} = \delta/(\delta + \text{Oir}), \ \delta \in \mathbb{R}^+. \qquad \text{(II)}$$

The individual fitness is improved if a decrease in the value of the overall individual residual is achieved. The optimal solution of the problem, nodal values, will be achieved when Oir approaches zero and Fit approach unity.

3. Described of the Continuous Genetic Algorithms

In this paper, we developed the CGA's to solve the stiff problems. Before going to the detailed description of the CGA's, the condition about the continuous functions that can be used in such algorithm should be clearly sated in [3]. In relation to the initialization function, any smooth function can be used and a mixture of functions will be beneficial in this case to result in a diverse initial population. The effect of the initial population usually dies after few tens of generations and the convergence speed (the average number of generations required for convergence) after that is governed by the selection mechanism, crossover and mutation operators. Regarding the crossover function, it should be within the range $[0, 1]$ such that the offspring solution curve will start with the solution curve of the first parent and gradually change their values till they reach the solution curve of the second parent at the other end. The mutation function may be any continuous function within the range $[0, 1]$ such that the mutated child solution curve will start with the solution curve of the child produced through the crossover process and gradually change its value till it reach the solution curve of the same child at the other end.

The CGA's proposed in this work consists of the following steps:

1. **Initialization:** In this phase, an initial population comprising of N_p smooth individuals is randomly generated. In this work, two smooth functions that satisfy the constraint condition used for initializing the population: the modified Gaussian function and the tangent hyperbolic function. The two function differ from each other by main criteria: the convex/concave nature. The modified Gaussian function is given by the equation:

$$p_j^k(i) = Y_k[t_0] + i\left(\beta^k - Y_k[t_0]\right)/N + A^k \exp\left(-0.5\left((i-\mu)/\sigma\right)^2\right),$$

(III)

while the tangent hyperbolic function is governed by the equation:

$$p_j^k(i) = Y_k[t_0] + 0.5\left(\beta^k - Y_k[t_0]\right) \left(1 + \tanh\left((i-\mu)/\sigma\right)\right), \tag{IV}$$

for each $i = 1, 2, ..., N$, $j = 1, 2, ..., N_p$, and $k = 1, 2, ..., m$, where β^k represents a random number within the range of $Y_k[t]$ if the range of $Y_k[t]$ known and any random number if it is unknown, $p_j^k(i)$ is the *i-th* variable value of the *k-th* curve for the *j-th* parent, A^k represents a random number within the range $\left[-2\left|\beta^k - Y^k[t_0]\right|, 2\left|\beta^k - Y^k[t_0]\right|\right]$, μ is a random number within the range $[N/4, 3N/4]$, and σ is a random number within the range $[1, d_{\min}/3]$, where $d_{\min}$ is a minimum of d_1 and d_2; the two values d_1 and d_2 representing the number of nodes to the left and right of μ. For both Gaussian and hyperbolic functions, μ specifies the center, while σ specifies its degree of dispersion.

2. **Evaluation:** The fitness, a nonnegative measure of quality used to reflect the degree of goodness of the individual, is calculated for each individual in the population as given in Equation (II).

3. **Selection:** In the selection process, individuals are chosen from the current population to enter a mating pool devoted to the creation of new individuals for the next generation such that the chance of selection of a given individual for mating is proportional to its relative fitness. This step ensures that the overall quality of the population increases from one generation to the next.

4. **Crossover:** Crossover provides the means by which valuable information is shared among the individuals in the population. The crossover process combines the features of two parent individuals, say j and h, to form two children individuals, say l and $l+1$, as given by the equations: $c_l^k(i) = w^k(i)\,p_j^k(i) + \left(1 - w^k(i)\right)p_h^k(i)$, $c_{l+1}^k(i) = \left(1 - w^k(i)\right)p_j^k(i) + w^k(i)\,p_h^k(i)$, and $w^k(i) = 0.5\left(1 + \tanh(i-\mu)/\sigma\right)$ for each $i = 1, 2, ..., N$ and $k = 1, 2, ..., m$, where p_l and p_h represent the two parents chosen from the mating pool, c_l and c_{l+1} are the two children obtained through crossover process, w^k represents the crossover weighting function within the range $[0, 1]$, and μ, σ are as given in the initialization process. In the proposed algorithm, pairs of individuals are crossed with probability P_{ci}. Within the pair of parents that should undergo crossover process, individual curves are crossed with probability P_{cc}.

5. **Mutation:** Mutation is often introduced to guard against premature convergence. Generally, over a period of several generations, the gene pool tends to become more and more homogeneous. The purpose of mutation is to introduce occasional perturbations to the parameters to maintain genetic diversity within the population. The mutation process is governed by the following formulas: $m_j^k(i) = c_j^k(i) + d^k g^k(i)$ and $g^k(i) = \exp\left(-0.5\left((i-\mu)/\sigma\right)^2\right)$ for each $i = 1, 2, ..., N$, $j = 1, 2, ..., N_p$, and $k = 1, 2, .., m$, where c_j represents the *j-th* child produced through the crossover process, m_j^k is the mutated *j-th* child, g is the Gaussian mutation function within the range $[0, 1]$, and d^k represents a random number within the range $[-Rang(k), Rang(k)]$, where $Rang(k)$ representing the difference between the minimum and maximum values of the *k-th* smooth curve of child c_j, and μ, σ are as given in

the initialization process. In mutation process, each individual child undergoes mutation with probability P_{mi}. However, for each child that should undergo mutation process, individual curves are mutated with probability P_{mc}.

6. **Replacement:** After generating the offspring's population through the application of the genetic operators to the parents population, the parents population is totally or partially replaced by the offspring's population depending on the replacement scheme used. This completes the "life cycle" of the population.

7. **Termination:** The CGA's is terminated when some convergence criterion is met. Possible convergence criteria are: the fitness of the best individual so far found exceeds a threshold value, the maximum nodal residual of the best individual of the population is less than or equal some predefined threshold value, the maximum number of generations is reached, or the progress limit; the improvement in the fitness value of the best member of the population over a specified number of generations is less than some predefined threshold, is reached. After terminating the algorithm, the optimal solution of the problem is the best individual so far found.

To summarize the evolution process in CGA's, an individual is a candidate solution that consists of m curves each of N 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 m 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.

The complete and unambiguous description of the CGA's is given by the following algorithm.

Algorithm 2 *To approximate the solution of the initial value problem:* $H\left(t, \dot{Y}[t], Y[t]\right) = 0$, $Y[t_0] = Y_0$, $t \in T$ *at* $N+1$ *equally spaced numbers in the interval* T.

Input: Endpoints of T; integer N; initial condition Y_0.

Output: Approximation Φ to Y at the $N+1$ values of t.

Step 1: Set $h = (t_f - t_0)/N$.

Step 2: For $i = 0, 1, ..., N$ Set $t_i = t_0 + ih$.

Step 3: Initialization process.

Step 4: Fitness evaluation process.

Step 5: Selectionprocess.

Step 6: Crossover process.

Step 7: Mutation process.

Step 8: Fitness evaluation process.

Step 9: Replacement process.

Step 10: If termination process doesn't hold then go to Step 5 else go to Step 11.

Step 11: Output $(t_i, \Phi[t_i])$.

Step 12: Stop.

Two additional operators were introduced to enhance the performance of the CGA's. These operators are summarized in the form of following:

1. **Elitism:** The preservation of the best solution or solutions and moving it or them to the next generation. Elitism is utilized to ensure that the fitness of the best candidate solution in the current population must be larger than or equal to that of the previous population. In other words, a good solution found should not be lost through some of the genetic operators.

2. **Extinction and Immigration:** This operator applied when all individuals in the population are identical or when the improvement in the fitness value of the best individual over a certain number of generations is less than some threshold value. The number of individuals in the population associated with better fitness grows exponentially. Therefore, after some generations, the mating pool will consist of almost identical members. This means that no new information will be obtained through crossover process. The CGA's thus tends to stagnate; "extinction and immigration" operator is used to bypass this difficulty. This operator, as indicated by its name, consists of two stages; the first stage is the extinction process where all of the individuals in the current generation are removed except the best-of-generation individual. The second stage is the mass-immigration process where the extinct population is filled out again by generating $N_p - 1$ individuals to keep the population size fixed. The generated population is divided into two equal segments each of $N_p/2$ size; the first segment, with $j = 2, 3, ..., Np/2$, is generated as in the initialization phase, while the other segment is generated by performing continuous mutation to the best-of-generation individual as given by formulas: $p_j^k(i) = p_1^k(i) + d^k g^k(i)$ and $g^k(i) = \exp\left(-0.5\left(\left(i-\mu\right)/\sigma\right)^2\right)$ for each $i = 1, 2, ..., N$, $j = Np/2 + 1, Np/2 + 2, ..., N_p$, and $k = 1, 2, .., m$, where p_j^k is the *j-th* parent generated using immigration operator, p_1^k represents the best-of-generation individual, g^k is the Gaussian mutation function, and d^k represents a random number within the range $[-Rang(k), Rang(k)]$, where $Rang(k)$ representing the difference between the minimum and maximum values of the *k-th* smooth curve of p_1^k, and μ, σ are as given in the initialization process.

4. Numerical Results and Discussion

The scenario of this section is to introduced a stiff problem with exact solution to compare the result obtained from CGA's with the corresponding exact solution to measure the efficiency of CGA's as a novel solver.

Example 3 *Consider the stiff initial value problem:* $\dot{Y}(t) = -15(Y_2(t), Y_1(t)) - 16(Y_1(t), Y_2(t)) + \frac{1}{5}(18, 21)$, $t \in [0, 1]$ *subject to initial condition:* $Y(0) = (0, 0)$. *The exact solution is given by:* $Y(t) = 3/10(1, -1)\exp(-t) - 39/310(1, 1)\exp(-31t) + 1/155(-27, 66)$.

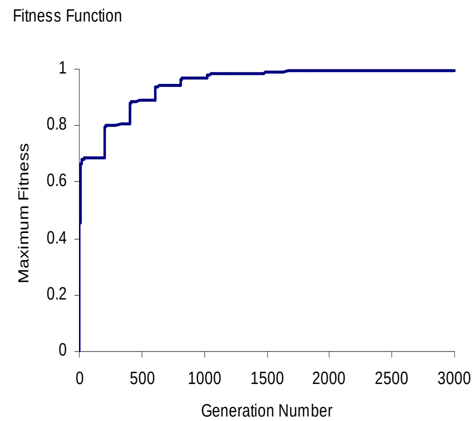
The CGA's proposed in this work is used to solve the given stiff problem. The input data to the algorithms is divided into two parts; the genetic algorithms related parameters and the stiff problem related parameters. The genetic algorithms related parameters include the population size, N_p, the crossover probability, P_{cc}, P_{ci}, the mutation probability, P_{mc}, P_{mi}, the value of δ in Equation (II), 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 stiff problems related parameters include the governing stiff differential equation, the interval $[t_0, t_f]$, the step size, h, the initial value, the number of nodes, N.

The initial settings of the CGA's related parameters are as follows: the population size is set to 500 individuals. The individual crossover probability, P_{ci}, is set to 0.5, the curve crossover probability, P_{ci}, is set to 0.5, the individual mutation probability, P_{mi}, is set to 0.5, the curve mutation probability, P_{mc}, is set to 0.5, and δ is set to 1. Mixed method for initialization schemes are used where half of the population is generated by the modified Gaussian given in Equation (III) while the other half generated using the tangent hyperbolic function given in Equation (IV). The rank-based selection strategy is used where the rank based ratio is set to 0.1. Generational replacement scheme is applied where the number of elite parents that are passed to the next generation equals one-tenth of the population size. Extinction and immigration operator is applied when the improvement in the fitness value of the best individual of the population over 100 generations is less than 0.01. For the problem, the step size, h, is set to 0.1 and the number of interior nodes is set to 10.

The CGA's is stopped when one of the following conditions is met. First, the fitness of the best individual of the population reaches a value of 0.9999; that is, the overall individual residual of the best individual of the population is less than or equal to 0.000100010001. Second, the maximum nodal residual of the best individual of the population is less than or equal to 0.0000000001. Third, a maximum number of 3000 generations is reached. Fourth, the improvement in the fitness value of the best individual in the population over 500 generations is less than 0.00001. 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). Due to the stochastic nature of CGA's, tenth 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.

The convergence data is given as follows: the problem take about 3000 generations, on average, to converge to a fitness value of about 0.98692556 with an average absolute nodal residual of the value 6.189294×10^{-4} and an average absolute difference between the exact values and the values obtained using CGA's of the value 7.232698×10^{-4}.

The evolutionary progress plots, of the best-fitness and minimum-residual individual are shown in Figure 1.



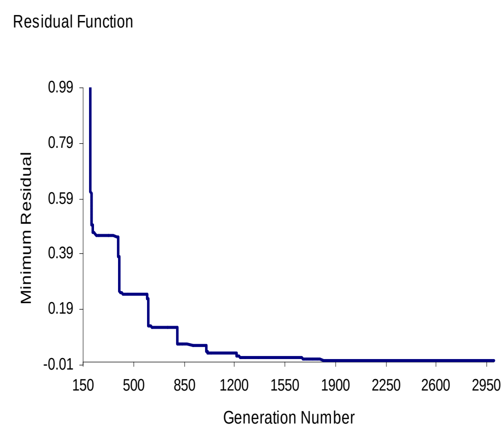
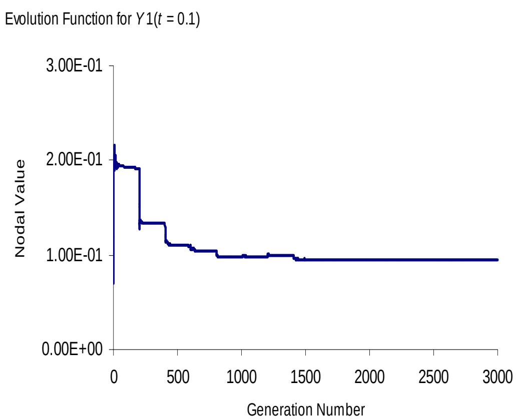


Figure 1: Evolutionary fitness and residual of the problem.

It is observed that from the evolutionary plots of the problems 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 reaches steady-state values and don't have oscillations by usual inspection.

The percentage of the fine-tuning stage till convergence from the total number of generations for the problem with respect to fitness evolution is 50%, where the number of generations in the fine-tuning stage is defined by: the average number of generations$-(j-50)$ such that Fit(k) $-$Fit$(k-50) \leq 0.001$ for each $k>j$ for some generation j. That means the approximate of CGA's converge to the actual solution very fast in the first 50% of the generations. In fact the individual fitness is improved if a decrease in the value of the overall individual residual is achieved. The optimal solution of the problem, nodal values, will be achieved when minimum overall individual residual approaches zero and best fitness approaches unity.

The way in which the nodal values evolve is studied next. Figure show the evolution of the first and ninth nodal value for the first variable, respectively, while the Figure 3 show the evolution of the first and ninth nodal value for the second variable, respectively.



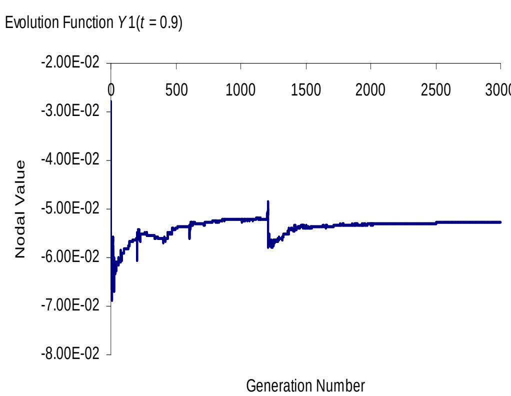
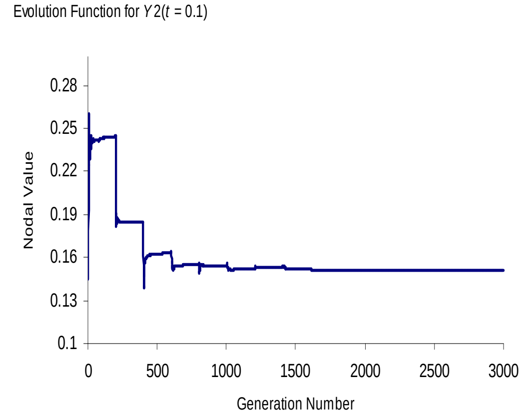


Figure 2: Evolution of the first and ninth nodal values for the first variable.



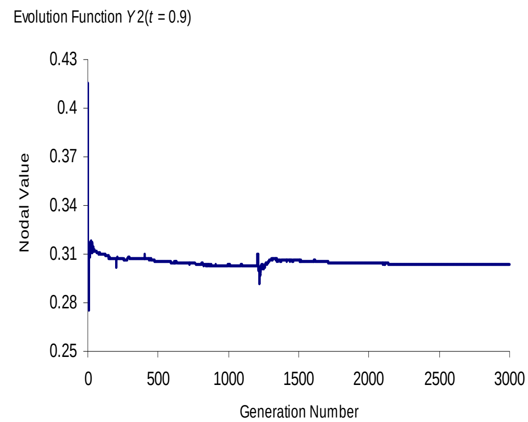


Figure 3: Evolution of the first and ninth nodal value for the second variable.

It is observed that all nodes, in the same problem, reaches the near optimal solution together. It is also concluded that the evolution has initial oscillatory nature for all nodes. As a result, the distance (number of nodes) from the initial point doesn't effect in the convergence speed.

The percentage of the fine-tuning stage till convergence from the total number of generations with at the first and ninth nodes evolution with respect to first and second variables are: 47%, 58% and 47%, 58%, respectively, where the number of generations in the fine-tuning stage is defined by: the average number of generations$-(j-50)$ such that $|\text{Evo}\,(t,k)-\text{Evo}\,(t,k-50)| \leq 0.001$ for each $k>j$ for some generation j, where Evo(t,k) denote to evolution of node t at generation k.

Table 1 through Table 4 show the results obtained for the problem using CGA's across all interior nodes for the first and the second variable, respectively.

Table 1: Numerical result for the 1st var.

t	Exact value	Approx. value
0.1	+0.09159020	+0.09421277
0.2	+0.07117036	+0.07066817
0.3	+0.04804042	+0.04851664
0.4	+0.02690195	+0.02645195
0.5	+0.00776563	+0.00797471
0.6	−0.00955006	−0.00940603
0.7	−0.02521796	−0.02579864
0.8	−0.03939486	−0.03897397
0.9	−0.05222265	−0.05270990
1.0	−0.06382972	−0.06699915

Table 2: Numerical result for the 1st var.

t	Absol. error	Absol. residue
0.1	2.622569×10^{-3}	8.660110×10^{-5}
0.2	5.021916×10^{-4}	2.496025×10^{-4}
0.3	4.762200×10^{-4}	9.042357×10^{-4}
0.4	4.500008×10^{-4}	1.377463×10^{-3}
0.5	2.090833×10^{-4}	1.035190×10^{-3}
0.6	1.440258×10^{-4}	2.664832×10^{-4}
0.7	5.806818×10^{-4}	5.658351×10^{-4}
0.8	4.208851×10^{-4}	8.325423×10^{-4}
0.9	4.872495×10^{-4}	8.136086×10^{-4}
1.0	3.169437×10^{-3}	5.738147×10^{-4}

Table 3: Numerical result for the 2nd var.

t	Exact value	Approx. value
0.1	+0.14868775	+0.15122578
0.2	+0.17993191	+0.17938585
0.3	+0.20354948	+0.20396309
0.4	+0.22470992	+0.22421968
0.5	+0.24384723	+0.24413581
0.6	+0.26116296	+0.26146166
0.7	+0.27683086	+0.27639496
0.8	+0.29100776	+0.29162541
0.9	+0.30383555	+0.30358507
1.0	+0.31544262	+0.31447228

Table 4: Numerical result for the 2nd var.

t	Absol. error	Absol. residue
0.1	2.538035×10^{-3}	7.895800×10^{-5}
0.2	5.460636×10^{-4}	2.113119×10^{-4}
0.3	4.136019×10^{-4}	8.635775×10^{-4}
0.4	4.902445×10^{-4}	1.312187×10^{-3}
0.5	2.885803×10^{-4}	1.117774×10^{-3}
0.6	2.986974×10^{-4}	4.566294×10^{-4}
0.7	4.358978×10^{-4}	5.611304×10^{-4}
0.8	6.176508×10^{-4}	8.751833×10^{-4}
0.9	2.504839×10^{-4}	8.370778×10^{-4}
1.0	9.703367×10^{-4}	5.972419×10^{-4}

The accuracy in CGA's doesn't effect by the distance from the initial node, while in conventional methods the accuracy effect by the distance from the initial node. The absolute average error which is calculated across all problem nodes for the first and the second variable are: 9.0623×10^{-4} and 6.8496×10^{-4}, respectively.

5. Conclusion

In this research, a new numerical method to tackle the stiff initial value problems is proposed. Central to the approach is the novel use of CGA's where smooth solution curves are used for representing the required nodal values. This new method promises to open new possibilities for applications in an important class of physical, engineering, and chemical problems.

REFERENCES

[1] Z. Abo-Hammour, Advanced Continuous Genetic Algorithms and their Applications in the Motion Planning of Robot Manipulators and in the Numerical Solution of Boundary Value Problems, Doctoral Dissertation, Quaid-Azam University, Islamabad, Pakistan, 2002.

[2] Z. Abo-Hammour, M. Yusuf, N. Mirza, S. Mirza, and M. Arif, "Cartesian Path Planning of Robot Manipulators using Continuous Genetic Algorithms," Robotics and Autonomous Systems, Vol. 41, No. 4, 2002, pp. 197 – 223.

[3] Z. 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," Int. J. Numer. Meth. Engng., Vo. 61, 2004 pp 1219 – 1242.

[4] O. AbuArqob, Numerical Solution of Fuzzy Initial Value Problems Using Contiunuous Genetic Algorithms, Doctoral Dissertation, University of Jordan, 2008.

[5] S. Chapra and R. Canale, Numerical Methods for Engineers, 5th ed.,

McGRAW-Hill International Edition, Singapore, 2006.

[6] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, 19^{th} ed., Addison Weslsy, USA, 1990.

[7] M. Gutowski, Smooth Genetic Algorithm. "Journal of Physical A Mathematical and General,", Vol 27, 1994, pp 7893 – 7904.

[8] J. Holland, Adaptation in Natural and Artificial Systems, 1^{th} ed., University of Michigan Press, USA.1975.

[9] J, Li, "General Explicit Difference Formulas for Numerical Differentiation," Journal of Computational and Applied Mathematics, Vol. 183, 2005, pp 29−52.