Nelder-Mead Simplex (NMS) approach for the optimal solutions of differential equations with an initial condition (IC)

[1]Shakoor Muhammad, [1]Nekmat Ullah, [3]Ubaid Ullah, [2]Abdul Rehman, [1]Waqas Ahmad

[1]Department of Mathematics, Abdul Wali Khan University Mardan, Khyber Pakhtunkhwa, Pakistan

[2]Department of Mathematics, University of Balochistan, Quetta 87300, Pakistan

[3]School of Mathematics and Statistic, Hainan Noraml University, 99 Longkun South Road, Haikou City, 571158, China

shakoor@awkum.edu.pk, nekmatmaths@gmail.com, uu627726@gmail.com, abdul_maths@yahoo.com, iwaqasahmad4498@gmail.com

Corresponding author email: shakoor@awkum.edu.pk

Corresponding author email: shakoor@awkum.edu.pk
Shakoor Muhammad[1] , Nekmat Ullah[1] , Fazal Hanan[2], Taimur Ali[1]

1 Department of Mathematics, Abdul Wali Khan University Mardan, Khyber Pakhtunkhwa, 23200, Pakistan

2 Department of Mathematics, AIOU, Islamabad, Pakistan

3 Deparment of Computer Science, COMSATS University Islamabad, Attock Campus, Pakistan

shakoormath@gmail.com, nekmatmaths@gmail.com, fazalhanan003@gmail.com, m tahir@cuiatk.edu.pk,sardaraz@cuiatk.edu.pk, marifmaths@awkum.edu.pk

Abstract: This paper proposes a new minimization technique for of the solutions of differential equation using non-linear Nelder-Mead Simplex algorithm and compare with genetic algorithm. We have solved differential problems both with Nelder-Mead simplex and genetic algorithm. The designed differential equation has been calculated as an error function for the minimization process. The proposed procedure has the capability to minimize the obtained solutions through any numerical techniques. The effectiveness of the proposed techniques guarantees the minimization of nonlinear differential equation with initial conditions. The resultant technique has a global validity for the minimization of differential equation. Genetic algorithm (GA) is suggested to minimize the differential problem, accessible to show the performance of the GA. Numerical outcomes obtained using Nelder-Mead Simplex guarantees its effectiveness.

——————————◆——————————

Introduction

Optimization is a useful technique for determining the best optimal solution to a problem. In other words, optimization is the problem of selecting suitable inputs under given circumstances in order to get the best possible output. For instance, optimization can be used in production models to adjust different inputs and make them more effective in order to get the best output for the production of a particular industry. In this scenario, once we have modeled a problem, it can be solved using the available optimization techniques to find the optimal solution [15].

Optimization methods are used for decision making, mainly in the selection of an optional subset of projects selected for investment. The degree of objective and dependence on data, as well as the structure of their results which differs widely across these methods [17].

Certificates of infeasibility can be useful within optimization algorithms in order to allow the fast determination of the inconsistency of the problem constraints, avoiding spending large computational times in infeasible problems, and also providing a guarantee that a problem is indeed not solvable. A series of results in interior-point based linear programming has been related to the construction of infeasibility certificates. The issue of detecting infeasibility in optimization problems has been particularly important in the context of mixed integer linear programming [16]. In [18] Shakoor et al tried to make use of conventional as well as local turbines, which lead to small hydropower plants that contribute to the proposed system as a case study in the northern areas of Pakistan in order to produce low-cost electricity. The purpose of the proposed study was to develop an optimal hybrid energy management system that generates more power at a minimum cost.

Differential equation is one of the most attractive topic in mathematics which is widely applicable in all fields of science, engineering, and applied economic [12]. Several numerical methods have been used for solving differential equations such as: Laplace Transform, RK-4, RK-6 and Predictor-Corrector method etc.

In this paper, we firstly solve various kind of differential equations by available analytical and numerical methods and then minimize it by using Nelder-Mead Simplex method. In the first phase, we convert the given differential equation into an energy function (cost function) and solve it by available techniques in the literature. In the second phase, we minimize the obtained solution using Nelder mead Simplex algorithm. In other words, we minimize the unconstrained function of n variables through NMS. Actually we check the functional values $P_0, P_1 ... P_n$ that are the (n+1) vertices of non-linear Simplex in an n-dimensional space. The proposed methodology is fast and quick in order to find out the local minimum in case of mono-objective optimization in n-dimensional spaces. According to the best of our knowledge, the methodology used here in this paper is a new approach in order to minimize the solution of differential equations with IC. The proposed methodology may be able to work for partial differential equations as well which is our future work in this regard. The remainder of the paper is arranged as follows: Section 2 is about non-linear Nelder Mead Simplex. Section 3 presents the genetic algorithm. Section 4 conversion of the constrained optimization problem to an unconstrained optimization problem. Section 5 describes the algorithmic procedure of the proposed methodology. Section 6 presents the numerical outcomes with examples and section 7 ends this paper with conclusion.

Non-Linear Nelder MeadS Simplex Method (NMS)

Nelder-Mead method is widely used algorithm for unconstrained optimization problems which does not rely the gradient of the functions. This algorithm is particularly applicable in the area of

chemistry, chemical engineering, and medicine, and engineering problems [1-3]. The function values are evaluated at each vertex during the iterative process. It is a direct search optimization method that works properly for stochastic problems as well. Each iteration in Rn space formed by n+1 vertices such as Y = {Y0, Y1…Yn}. The Simplex vertices are sorted in ascending order based on the function values. In other words, f (Y0) and f (Yn) are the best vertex and worst vertex respectively.

Basically NMS works on bases of three operations such as reection, expansion and contraction. In each iteration, the worst vertex Y^n is replaced by Y^c where $Y = Y^c + \delta(Y^c - Y^n)$ and $\delta \in R$ that is $Y^c = \sum_{i=0}^{n-1} Yi/n$ is the centroids best point in n vertices. The value of δ determines the type of repetition. For example δ = 1, δ = 2, δ = $\pm$1/2 represents reection, expansion and contraction respectively. Nelder-Mead simplex (NMS) may do shrink to form a new simplex. In shrinking process, all the vertices in Y may move towards the best head vertex.

Genetic Algorithm (GA)

The genetic algorithm is a random-based classical evolutionary algorithm. By random we mean that random changes will be applied to the available solutions in order to generate new solution using the GA. GA is also called Simple genetic algorithm (SGA) due to its simplicity compared to other evolutionary algorithm. GA is based on Darwins theory of evolution. It is a slow gradual process that works by making changes to the making slight and slow changes. Also, GA makes slight changes to its solutions slowly until getting the best optimal solution. Unlike traditional algorithms GA is not static but dynamic because it can evolve over time.

The genetic algorithm is a technique for solving and minimizing equally constrained and unconstrained optimization problems, which is base through the natural selection, the procedure which drives biological evolution. Genetic algorithm is a sort of optimization process; significantly it is used to search out the optimum solution(s) of an assume differential problem that maximizes or minimizes a particular function. Genetic algorithm has four operations. [13]

- Fitness Function
- Chromosome
- Crossover
- Mutation

Conversion of Differential Equation to an Optimization Problem

An equation involving the differential of one or more dependent variable with respect to one or more independent variable is called differential equation. Generally, we use the functions to represent physical quantities, the derivatives to represent their rates of change but the differential equation defines a relationship between these both physical quantities and their rates of change. Differential equations have an amazing capacity to forecast the world around us. These equations are used in a wide variety of disciplines as biology, economics, chemistry, physics, and engineering. Beside this they can be used to define exponential growth and decay, the population growth of species or the change in investment return over time.

The solution of a differential equation is a function which is free from derivatives. Differential equations have usually infinite number of solutions and hence its graph is actually a family of curves. Different approaches are used to solve a differential equation but our approach is quite opposite [10-11].

In this section, we convert the given differential equation with initial and boundary values to an optimization problem in order to solve it. This become a constrained optimization problem and later on it converted to an unconstrained optimization problem for the solution process within the interval interval $[\alpha, \beta]$. This unconstrained optimization problem can be solved by non-linear Nelder Mead Simplex method [4-9].

Proposed Methodology

Considered the general form of differential equation with initial and boundary conditions (IBC) as follows.

$$a_n y^n(t) + a_{n-1} y^{n-1}(t) \ldots, a_0 y(t) = g(t), \text{ where } t \in [\alpha, \beta] \quad (1)$$

with IC $y^n(t) = c_n, y^{n-1}(t) = c_{n-1} \ldots\ldots, y^0(t) = c_0,$

and $a_n, a_{n-1}, a_{n-2}, \ldots\ldots a_0$, which are any scaler functions.

It should be noted that the nonlinear differential equations have no products of $y(t)$ with its derivatives. Furthermore, if neither the function nor its derivatives exist other than the first power is called linear differential equations. If the product of $y(t)$ and their derivatives are exist other than first power is called non-linear differential equations.

In Equation (1), if $g(t)$ is zero then it is said be a homogeneous differential equation. Furthermore, $a_n \ldots a_0$ *determines as a scaler and* $y^n \ldots\ldots, y^0$ and $g(t)$ are any algebraic or transcendental functions. Here we proceed to solve (1) by different numerical methods which have been used for the solutions of linear and nonlinear differential equations.

Let, $y(t) = u(t)$ (2)

With $\alpha \leq t \leq \beta$ is the solution of (1). Equation (2) is an unconstrained optimization problem with $t \in [\alpha, \beta]$. In order to minimize (2), Nelder Mead Simplex may be used. Where t is random step size in iterative process. [14]

Algorithm for proposed Methodology

To solve this unconstrained optimization problem, we proceed as follows:

Step 1: Identify the interval $[\alpha, \beta]$ and convert the differential equation described in (1) to an unconstrained optimization problem (2).

Step 2: Choose points ti from $[\alpha, \beta]$ as an initial boundary value.

Step 3: Systemize the given differential equation described in (1). If it is an equation system in n unknown variables, then systemize the equation in order to solve it for n unknowns.

Step 4: Convert (1) to an unconstrained optimization problem (2) and solve it with available numerical and analytical techniques in the literature.

Step 5: Minimize the result obtained from step 4 through Nelder Mead Simplex in order to minimize the obtained solutions.

Numerical Experiments

In order to perform computational tests involving the proposed methodology, we will first convert the constrained optimization problem (1) to an unconstrained optimization problem (2). After that we will use non-linear Nelder-Mead Simplex algorithm in order to minimize the obtained solutions.

Example 1:

We have a second order initial value problem.

$$y'' + 16y = cos4t, \ \ with\ initial\ values\ \ y(0) = 0, y'(0) = 1 \quad (3)$$

Laplace transform for 2nd order differential equation is given below:

$$s^2Y(s) - sy(0) - y'(0) = F(s)\ldots \quad (4)$$

So the exact solution of the (3) is given below:

$$y(t) = \frac{1}{4} sin4t + \frac{1}{8} t\ sin4t \quad (5)$$

Now to minimize the function (5) by Nelder-Mead simplex algorithm at the point *t*=0, so the iterative process is given below in table 1:

Iteration	Function-Count	Min y(t)	NMS Process
0	1	0	
1	2	0	initial simplex
2	4	-0.000499875	Expand
3	6	-0.00149887	Expand
4	8	-0.00349376	Expand
5	10	-0.00747075	Expand
6	12	-0.01537	Expand
7	14	0.0309219	Expand
8	16	-0.0608249	Expand
9	18	-0.114264	Expand
10	20	-0.186041	Expand
11	22	-0.201926	Reflect
12	24	-0.201926	contract inside
13	26	-0.203271	contract inside
14	28	-0.203271	contract inside
15	30	-0.203271	contract inside
16	32	-0.203291	contract inside
17	34	-0.203291	contract inside
18	36	-0.203291	contract inside
19	38	-0.203292	contract inside
20	40	-0.203292	contract inside
21	42	-0.203292	contract inside
22	44	-0.203292	contract inside

Table 1

As of the above table, it is understandable that Nelder-Mead simplex algorithm minimize the given unconstrained optimization problem. The function value at a given point is 0, then applying the NMS then function value is minimized to -0.203292.

Now we have to minimize the above problem (3) through genetic algorithm: so iterative process is as under:

Generations	Func-Count	Best of f(x)	Mean of f(x)
1	100	-0.2838	0.03458
2	150	-0.2838	-0.04534
3	200	-0.2838	-0.04603
4	250	-0.2838	-0.08022
5	300	-0.2838	-0.1626
6	350	-0.2838	-0.2086
7	400	-0.2838	-0.2324
8	450	-0.2838	-0.2626
9	500	-0.2838	-0.2769
10	550	-0.2838	-0.2785

11	600	-0.2838	-0.2826
12	650	-0.2838	-0.2827
13	700	-0.2838	-0.2838
14	750	-0.2838	-0.2838
15	800	-0.2838	-0.2838
16	850	-0.2838	-0.2838
17	900	-0.2838	-0.2838
18	950	-0.2838	-0.2838
19	1000	-0.2838	-0.2838
20	1050	-0.2838	-0.2838
21	1100	-0.2838	-0.2838
22	1150	-0.2838	-0.2838
23	1200	-0.2838	-0.2838
24	1250	-0.2838	-0.2838
25	1300	-0.2838	-0.2838
26	1350	-0.2838	-0.2838
27	1400	-0.2838	-0.2838
28	1450	-0.2838	-0.2838
29	1500	-0.2838	-0.2838
30	1550	-0.2838	-0.2838
31	1600	-0.2838	-0.2838
32	1650	-0.2838	-0.2838
33	1700	-0.2838	-0.2838
34	1750	-0.2838	-0.2838
35	1800	-0.2838	-0.2838
36	1850	-0.2838	-0.2838
37	1900	-0.2838	-0.2838
38	2000	-0.2838	-0.2838
39	2050	-0.2838	-0.2838
40	2100	-0.2838	-0.2838
41	2150	-0.2838	-0.2838
42	2200	-0.2838	-0.2838
43	2250	-0.2838	-0.2838
44	2300	-0.2838	-0.2838
45	2350	-0.2838	-0.2838
46	2400	-0.2838	-0.2838
47	2450	-0.2838	-0.2838
48	2500	-0.2838	-0.2838
49	2550	-0.2838	-0.2838
50	2600	-0.2838	-0.2838
51	2650	-0.2838	-0.2838

Table 2

Now the graph of the above problem is

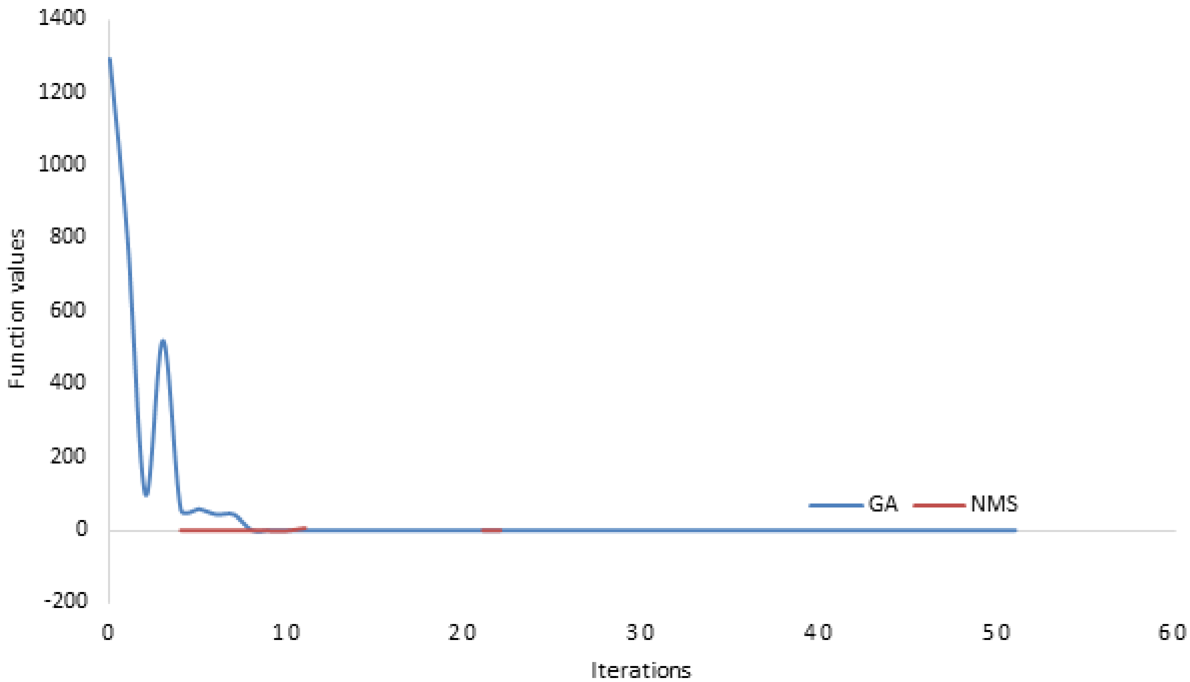


Fig 1: GA Vs NMS

As from the both techniques, it minimizes the differential equation. The NMS is more e_cient than GA because NMS find out the optimum value through a small number of iterations, while GA is not stable in iterative process. From the above Fig GA (1), it is clearly showed that GA is minimized the function, GA gives an optimal point -0.2838 after 51 iterations, the black doted shows the best fitness while the blue doted shows the mean tness and GA gave 51 iterations, on another hand NMS minimized the same function and give an optimal point -0.203292 at 22 iterations.

Example 2:
Consider the following example

$$\frac{dy}{dt} = -5y + 5t^2 + 5t \qquad where\ y(0) = \frac{1}{3} \tag{6}$$

The exact solution of (6) is given in the following equation:

$$y(t) = \frac{3e^{5t}t^2+1}{3e^{5t}} \tag{7}$$

Minimize the unconstrained optimization problem (7) by Nelder-Mead Simplex and iterative process given below.

Generations	Func-Count	Best of f(x)	Mean of f(x)
1	100	-0.2838	0.03458
2	150	-0.2838	-0.04534
3	200	-0.2838	-0.04603
4	250	-0.2838	-0.08022
5	300	-0.2838	-0.1626
6	350	-0.2838	-0.2086
7	400	-0.2838	-0.2324

8	450	-0.2838	-0.2626
9	500	-0.2838	-0.2769
10	550	-0.2838	-0.2785
11	600	-0.2838	-0.2826
12	650	-0.2838	-0.2827
13	700	-0.2838	-0.2838
14	750	-0.2838	-0.2838
15	800	-0.2838	-0.2838
16	850	-0.2838	-0.2838
17	900	-0.2838	-0.2838
18	950	-0.2838	-0.2838
19	1000	-0.2838	-0.2838
20	1050	-0.2838	-0.2838
21	1100	-0.2838	-0.2838
22	1150	-0.2838	-0.2838
23	1200	-0.2838	-0.2838
24	1250	-0.2838	-0.2838
25	1300	-0.2838	-0.2838
26	1350	-0.2838	-0.2838
27	1400	-0.2838	-0.2838
28	1450	-0.2838	-0.2838
29	1500	-0.2838	-0.2838
30	1550	-0.2838	-0.2838
31	1600	-0.2838	-0.2838
32	1650	-0.2838	-0.2838
33	1700	-0.2838	-0.2838
34	1750	-0.2838	-0.2838
35	1800	-0.2838	-0.2838
36	1850	-0.2838	-0.2838
37	1900	-0.2838	-0.2838
38	2000	-0.2838	-0.2838
39	2050	-0.2838	-0.2838
40	2100	-0.2838	-0.2838
41	2150	-0.2838	-0.2838
42	2200	-0.2838	-0.2838
43	2250	-0.2838	-0.2838
44	2300	-0.2838	-0.2838
45	2350	-0.2838	-0.2838
46	2400	-0.2838	-0.2838
47	2450	-0.2838	-0.2838
48	2500	-0.2838	-0.2838
49	2550	-0.2838	-0.2838
50	2600	-0.2838	-0.2838
51	2650	-0.2838	-0.2838

Table 3

Now we have to minimize the above problem (6) through genetic algorithm: so iterative process is as under

Generations	Func-Count	Best of f(x)	Mean of f(x)
1	100	0.0006796	1292
2	150	0.0006796	798.3
3	200	0.0006796	97.66
4	250	0.0006796	519.5

5	300	0.0006796	55.36
6	350	0.0006796	57.11
7	400	0.0006796	42.23
8	450	6.211e-05	42.23
9	500	6.211e-05	0.0516
10	550	6.211e-05	0.002106
11	600	6.211e-05	0.001977
12	650	1.406e-05	0.001838
13	700	1.406e-05	0.001734
14	750	1.406e-05	0.0002116
15	800	3.038e-06	0.0001208
16	850	3.038e-06	3.719e-05
17	900	3.038e-06	2.358e-05
18	950	5.69e-07	1.165e-05
19	1000	5.69e-07	1.035e-05
20	1050	5.69e-07	5.722e-06
21	1100	6.945e-08	4.249e-06
22	1150	6.945e-08	2.224e-06
23	1200	6.945e-08	1.836e-06
24	1250	6.945e-08	9.908e-07
25	1300	6.945e-08	3.439e-07
26	1350	1.994e-08	1.321e-07
27	1400	3.629e-10	1.142e-07
28	1450	3.629e-10	9.598e-08
29	1500	3.629e-10	5.727e-08
30	1550	3.629e-10	4.783e-08
31	1600	3.629e-10	3.549e-08
32	1650	3.629e-10	2.635e-08
33	1700	3.584e-10	1.382e-08
34	1750	3.533e-10	2.439e-08
35	1800	3.232e-10	6.556e-08
36	1850	3.244e-10	2.636e-07
37	1900	3.244e-10	1.54e-06
38	1950	3.244e-10	1.174e-06
39	2000	1.968e-10	5.013e-07
40	2050	1.968e-10	2.587e-07
41	2100	1.968e-10	1.389e-07
42	2150	1.968e-10	1.245e-07
43	2200	1.951e-10	1.157e-07
44	2250	1.901e-10	4.213e-08
45	2300	1.709e-10	8.126e-08
46	2350	9.948e-11	2.129e-07
47	2400	9.948e-11	7.632e-07
48	2450	8.572e-11	2.37e-07
49	2500	8.572e-11	2.316e-06
50	2550	3.793e-11	1.928e-06
51	2600	3.793e-11	4.005e-07
52	2650	2.963e-11	6.364e-07
53	2700	2.963e-11	2.071e-06
54	2750	2.963e-11	1.769e-06
55	2800	2.016e-11	1.132e-06
56	2850	2.653e-12	1.126e-06
57	2900	2.653e-12	2.737e-06
58	2950	2.653e-12	2.002e-06
59	3000	4.559e-13	1.508e-06
60	3050	4.559e-13	1.605e-06

61	3100	4.559e-13	1.41e-06
62	3150	2.97895e-16	1.58766e-08

Table 4

The graph of example 2 via obtained from genetic algorithm is given below.

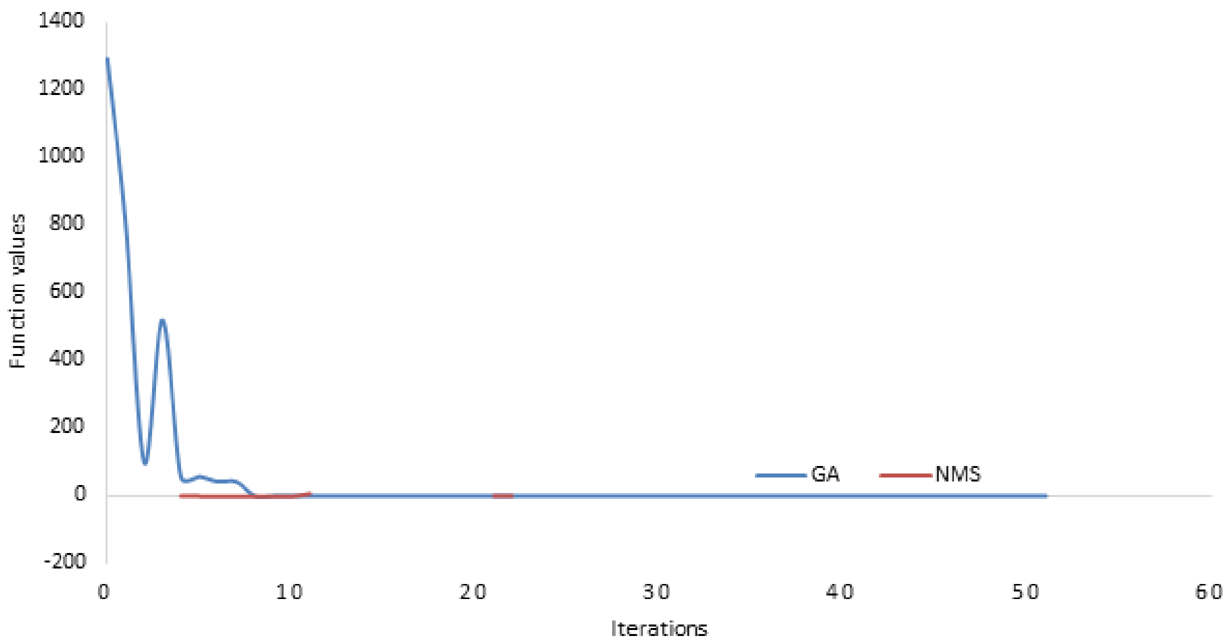


Fig 2: GA Vs NMS

Obviously from the both methods, it minimizes the differential equation. The NMS is more competent than GA because NMS find out the optimum value through a small number of iterations, while GA is not stable in iterative process. From the above Fig GA (2), it is clearly showed that GA is minimized the function, GA gives an optimal point 2.97895×1016, GA evaluate the process in 62 iterations, while NMS minimized the same function and give an optimal point 0, NMS take 34 iterations to give an optimal point.

Example 3:

$$\frac{dy}{dt} = \sin(t) - 2x\, tan(t) \quad where \quad y(0) = pi/3 \tag{8}$$

Transform equation (8) into an unconstrained optimization problem for exact solutions

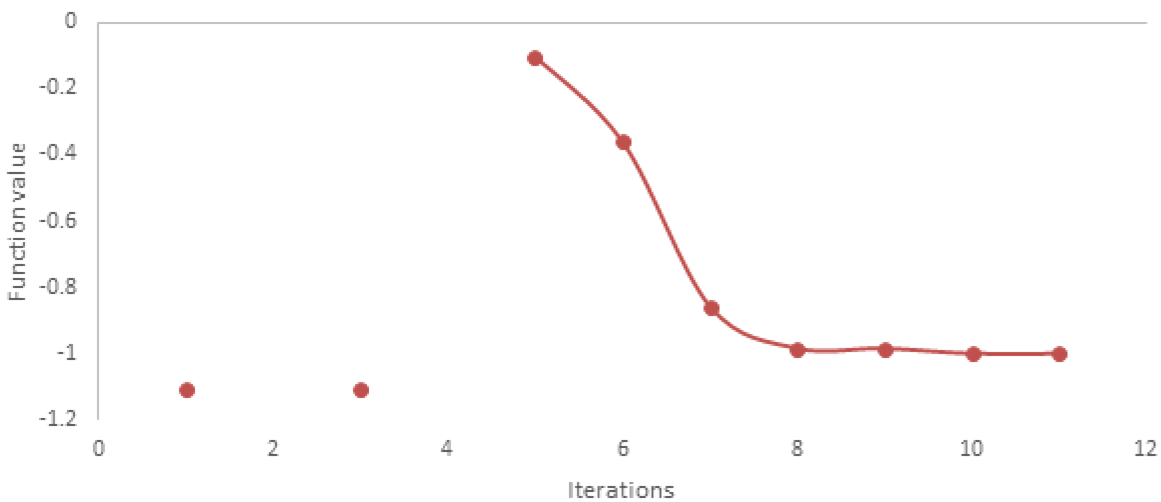
$$y(t) = \frac{\sec(t)-2}{sec^2(t)} \tag{9}$$

Minimize the unconstrained problem (9) by Nelder-Mead Simplex method with initial point *t=pi/3.* The iterative process is given in the following table:

No of Iterations	Func-Count	Min y(t)	NMS observations
0	1	-1.11022e^{-16}	
1	2	-1.11022e^{-16}	Initial Simplex
2	4	-0.103198	Expand
3	6	-0.361384	Expand
4	8	-0.85796	Expand
5	10	-0.983826	Reflect
6	12	-0.983826	Contract inside
7	14	-1	Contract inside
8	16	-1	Contract inside

Table 5

After five iterations the zeros of the objective functions is given in the following figure.



3: NMS

Number of iterations: 11
Number of Func-Count: 23
Fval: -1
Solve via GA: We have to solve the above problem (8) by genetic algorithm solver in matlab. Through a given process of matlab solver, graphically problem (8) is shown as:

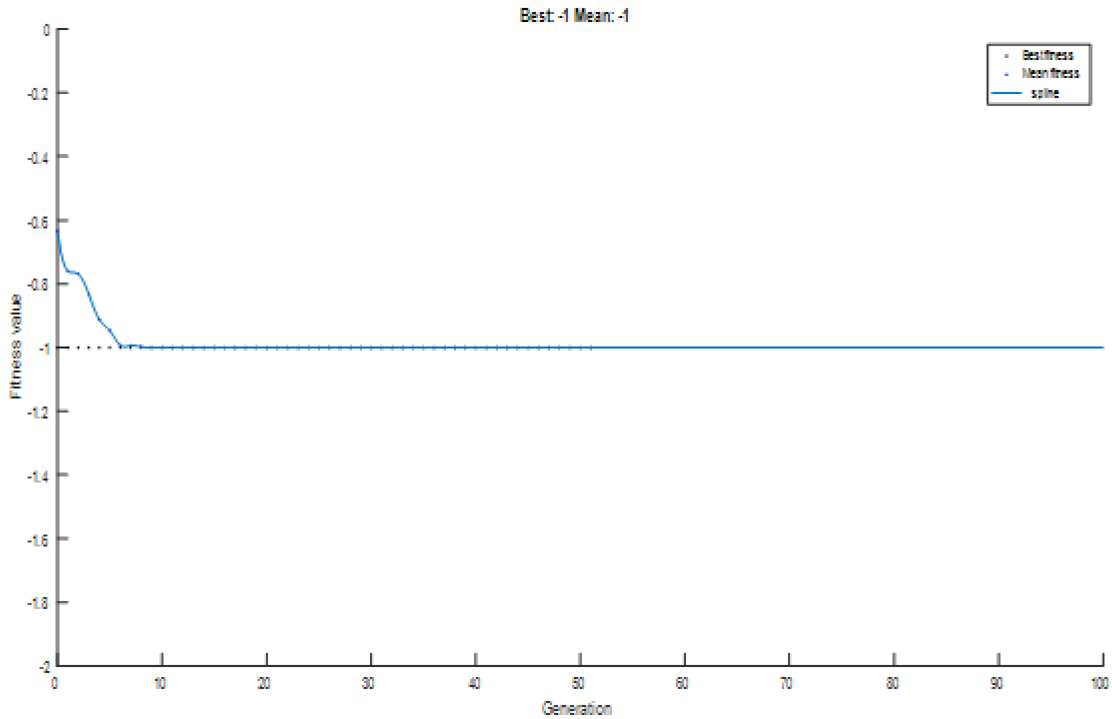


Fig 4: GA

From both techniques, the function is minimized to -1, while GA takes iterations than NMS.

Computational Analysis

The proposed technique was implemented in MATLAB [R2017a]. Computational experiments were carried out on a HP Core i3 with 2 GB of RAM, with 64-bit operating system windows 8.1 pro. The iterative process was repeated five times for each test problem that produces some promising results. All experiments produced the minimization of complicated differential equations with initial values.

Conclusion

In this paper, we have presented the proposed methodology in order to minimize the solution of differential equation obtained from different numerical and analytical techniques through Nelder Mead Simplex method to compare with genetic algorithm. We have changed the given differential equation to an unconstrained optimization problem in order to minimize it. The robustness of NMS method with initial values is highly appreciable and has good accuracy than GA. Furthermore, it can be used to minimize various kind of differential equation using the proposed methodology. We have validated the proposed methodology with different numerical examples. NMS works better as compare with GA because NMS is stable is iterative process and obtain unique optimal point while GA is not stable in iterative process and give a small change in optimum point. The proposed methodology may be able to work for partial differential equations and fractional-order differential equations which is our future work in this regard and compare with other optimization techniques.

References

1. Lagarias, J.C., Reeds, J.A., Wright, M.H. and Wright, P.E., 1998. Convergence properties of the Nelder-Mead simplex method in low dimensions. SIAM Journal on optimization, 9(1), pp.112-147.
2. McKinnon, K.I., 1998. Convergence of the Nelder{Mead Simplex Method to a Non stationary Point. SIAM Journal on Optimization, 9(1), pp.148-158.
3. Price, C.J., Coope, I.D. and Byatt, D., 2002. A convergent variant of the NelderMead algorithm. Journal of optimization theory and applications, 113(1), pp.5-19.
4. Nelder, J.A. and Mead, R., 1965. A Simplex method for function minimization. The computer journal, 7(4), pp.308-313.
5. Wright, M.H., 2010. Nelder, Mead, and the other simplex method. Documenta Mathematica, 7, pp.271-276.
6. Han, L. and Neumann, M., 2006. E_ect of dimensionality on the NelderMead simplex method. Optimization Methods and Software, 21(1), pp.1-16.
7. Gao, F. and Han, L., 2012. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. Computational Optimization and Applications, 51(1), pp.259-277.
8. Lagarias, J.C., Poonen, B. and Wright, M.H., 2012. Convergence of the Restricted Nelder{ Mead Algorithm in Two Dimensions. SIAM Journal on Optimization, 22(2), pp.501-532.
9. Chen, D.H., Saleem, Z. and Grace, D.W., 1986. A new simplex procedure for function mini- mization. International Journal of modeling and simulation, 6(3), pp.81-85.
10. Pashaie, M., Sadeghi, M. and Jafarian, A., Arti_cial Neural Networks with Nelder-Mead Optimization Method for solving nonlinear integral equations.
11. Jafarian, A., Nia, S.A.M., Golmankhaneh, A.K. and Balenu, D., 2013. Numerical solution of linear integral equations system using the Bernstein collection method. Advances in Difference Equations, 2013(1), p.123.
12. Pham, N., 2012. Improved Nelder Meads simplex method and applications (Doctoral dissertation).
13. Ali, R., Muhammad, S., & Takahashi, R. H. (2021). Decision making viva genetic algorithm for the utilization of leftovers. *International Journal of Intelligent Systems*, *36*(4), 1746-1769.
14. Bilal, M., Muhammad, S., Ullah, N., Hanan, F., & Ullah, S. (2022). Optimum solutions of partial differential equation with initial condition using optimization techniques.
15. Muhammad, S., Hanan, F., Shah, S. A., Yuan, A., Khan, W., & Sun, H. (2023). Industrial optimization using three-factor Cobb-Douglas production function of non-linear programming with application. *AIMS Mathematics*, *8*(12), 29956-29974.
16. Muhammad, S., Coelho, V. N., Guimarães, F. G., & Takahashi, R. H. (2016). An infeasibility certificate for nonlinear programming based on Pareto criticality condition. *Operations Research Letters*, *44*(3), 302-306.
17. Muhammad, S., Ali, T., Khan, Y., & Shafiullah, M. (2023). Application of Dynamic Programming for Optimal Hybrid Energy Management System: Hydro-Photovoltaic-Diesel-BESS. *IEEE Access*.