ISSN 2303-4521
Original Research
Periodicals of Engineering and Natural Sciences
Vol. 10, No. 6, December 2022, pp.49-56

Utilizing quantum genetic algorithm with TM to solve DEs

Ahmed Hamid Elias [1], Igor Klebanov [2], Salah A. Albermany[3]
[1] School of Electronic Engineering and Computer Science, South Ural State University, Russia.
[2] School of Electronic Engineering and Computer Science, South Ural State University, Russia.
[3] Computer Science department, Faculty of Computer Science and Mathematics, Kufa University, Iraq

ABSTRACT

In this paper a proposed approach to solve the DEs utilizing the TM with quantum Genetic algorithms. The aim of the proposed approach is to use the series functions of form quantum genetic to solve the DEs by decreasing the number of iterations and more increase by utilizing automata more with enhancement of TM.

Keywords: Quantum Genetic Algorithm, TM, DEs.

Corresponding Author:

Ahmed Hamid Elias
Computer Science
South Ural State University
Chelyabinsk, Russia
E-mail: ahmediteco@gmail.com

1. Introduction

A differential formula (DE) is an involving formula in the functions and their derivatives. DEs are named fractional DEs (pde) or DEs (ode) as ordinary based on if or not they comprise fractional derivatives. The DE order is the uppermost order derivative happening. A DE particular solution or solution of order n contains of a function well-defined and n times differentiable on a domain D possessing the character which is "the functional formula gotten through functions substituting along their n derivatives into the DE holds for each point in D. A classic DEs implementation continues along the following lines [1].

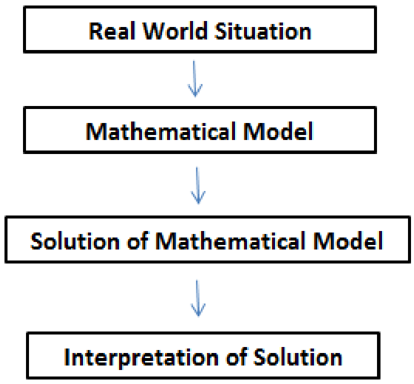

Figure 1. Lines of DEs implementation

A Turing Machine (TM) is a model as mathematical that contains an unlimited length tape splitted into cells where input is offers. It contains a head that reads the tape as input. A state register stores the TM state. Following input symbol reading, it is replaced with another symbol, its state as internal is altered, and it transfers from one cell to the left or right. When the TM touches the last state, the string as input is recognized, or else excluded. A TM able to be described formally as a 7-tuple (Q, X, Σ, δ, q0, B, F) in which [2]:

- Q is a finite state set • X is the alphabet tape
- Σ is the alphabet input
- δ is a function of transition; $\delta : Q \times X \rightarrow Q \times X \times \{$ Right _shift, Left _shift$\}$.
- q0 is the preliminary state • B is the symbol as blank
- F is the final states set

Quantum information processing (QIP) is a field which has observed wonderful growth over the several years since Richard Feynman's seminal mention the quantum computers (QC) utilize for simulating systems as physical [3]. When data bits are encoded on entangled or individual quantum states, a gain over outmoded systems able to be observed for few data processing functions. A well-known example is the familiar algorithm of Shor for factorization of prime number runned on a QC,in which a remarkable decline in resources is gained if compared to algorithms as classical [4]. Another chief QIP implementation is in security of communication, in which the fact that unidentified states of quantum cannot be cloned faithfully [5] is achieved for detecting an eavesdropper presence. Such conception was utilized as a central foundation after quantum key distribution (QKD), a protocol of communication intended for distributing random secretive means among remote parties4,5. As the 1st implementation to showcase in exercise the QIP benefits, QKD has experienced massive progresses ever since [6–9]. QKD is part of a further general protocol family named communications of quantum, that including other schemes i.e., entanglement swapping and teleportation of quantum [10][11], take aim to be the backbone of communication supporting QCs of future networks [12][13].
The algorithm genetic is a technique to solve both unconstrained and constrained optimization difficulties which is according to the selection as natural, the procedure which drives evolution biologically. The algorithm genetic modifies repeatedly an individual solutions population. At every step, the algorithm genetic picks individuals from the existing population as parents and utilizes them for producing the children for the subsequent generation. Over consecutive generations, the population evolves in the direction of an optimum solution. You are able to apply the algorithm genetic for solving a diversity of optimization difficulties which are not well suitable for optimization standard algorithms, include difficulties where the objective role is discontinuous, non-differentiable, extremely non-linear or stochastic. The algorithm genetic is able to address *mixed integer programming* difficulties in which few components are constrained as integer valued.
The flow chart summarizes the chief algorithmic steps [14] [15].

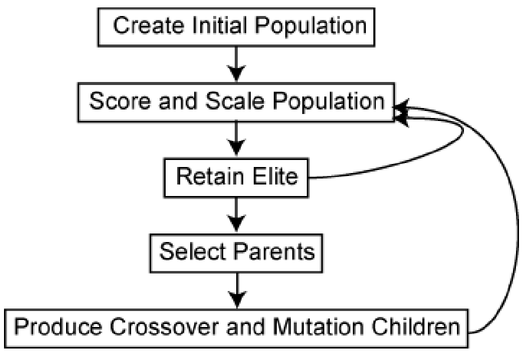

Figure 2. Genetic algorithmic steps

2. Related works

In [16] they display a quantum algorithm (QL) for systems of (probably inhomogeneous) linear DEs as ordinary with coefficients being constant. The algorithm yields a state as quantum which is comparative to the solution at a final desired time. The algorithm complexity is poly-nominal in the inverse error logarithm, an exponential enhancement over preceding QLs for such difficulty.

In [17] they describe QLs for simulation of Hamiltonian, DEs as ordinary (ODEs), and DEs as partial (PDEs). Formulas that are produced are utilized for simulating Hamiltonians that able to be expressed as terms sum that can ever be individually simulated. Through simulating every such term in sequence, the gross influence simulates almost the entire Hamiltonian. They catch that the product formulas error able to be enhanced through randomizing over the order where the simulation to Hamiltonian terms occur. They verify that such method is asymptotically better than formulas as ordinary product and show comparisons as numerical for small qubits numbers.

In [18] they display and through experimentation comprehend an implementable gate-based QL for powerfully solving the LDE difficulty: assumed an N × N matrix M, a vector as N-dimensional b, and a primary vector x (0), we attain a target vector x(t) as a time function t bases on the constraint dx(t)/dt = M x(t) + b. They display that the algorithm displays speedup as exponential over its traditional counterpart in definite circumstances, and a quantum gate-based circuit is formed that is friendly to the experimentalists and implementable in existing quantum methods. Furthermore, they experimentally solve a 4 × 4 linear DE utilizing their QL in a 4-qubit nuclear magnetic resonance quantum information processor.

In [19] they put into consideration 3 methods: (i) basis encoding and fixed-point arithmetic on a digital QC, and (ii) rep- re-sending and solving high-order Runge-Kutta approaches as optimization difficulties on quantum annealers. As realizations smeared to 2-dimensional linear ordinary differential formulas" they simulate and devise equivalent digital quantum circuits.

3. Proposed Approach

The solution of DEs in the interval [0, 1], by applied the suitable series by replace the truncated y(x) series into the linear or non-linear DEs as ordinary

Let $g(y^{(n)}, y^{(n-1)}, y^{(n-2)},. . ., y, x) = f(x) \qquad x \in [0,1]$

Ordinary DE (ODE) with $y^{(k)}(x_0) = y_k$, k=0 ,1 ,2 , . . .,n-1
Suppose the linear DEs as ordinary

$$|g\rangle = |f\rangle + \sum_{k=0}^{n} a_k \, |y^{(k)}\rangle \qquad \ldots(1)$$

$$|\, g\left(\sum_{k=0}^{n} B(x)^{(i)} |v_i >,\, x\right)\Bigg\rangle = |f\rangle + \sum_{k=0}^{n} \beta_k \, |B(x)^{(k)}\rangle$$

with $y^{(k)}(0) = y_k$, k=0 ,1 ,2 , . . ., n-1, and A set of vectors $|v_1\rangle$, ..., $|v_n\rangle$ is said to be **linearly dependent** if there exist corresponding coefficients for each vector ai∈ F, such that:
B(x) is a solution of (1) where $|B(x)\rangle = \sum_{k=0}^{L} a_k \, |\psi^{(k)}\rangle$ and $\mathscr{L}$ is the length of chromosome then B(x) satisfy of (1) with boundary condition

$$1\big|y^{(k)}(x_0) >= y_k \, , and$$

$$|g(B(x)^{(n)}, B(x)^{(n-1)}, B(x)^{(n-2)},. . ., B(x), x)\rangle = |f\rangle + \sum_{k=0}^{n} \beta_k \, |B(x)^{(k)}\rangle.$$

Let $x_i \in [a, b]$ i=0,1 ,2 ,… ,m-1 choose (m) equidistant point (x_0 ,x_1 , …, x_{m-1}) in the relative range

N size the population $1|g(x_i) >= 0$, i=0 ,1 ,2 , …,m-1, and $|x_i\rangle = i|\mathrm{h} + \mathrm{a}\rangle$

where,

$$|h\rangle = (b-a)|/\mathrm{m}\rangle$$

3.1. Fitness Function

the fitness function min *f(x)* where $|\boldsymbol{f} >= \sum_{\boldsymbol{i=0}}^{\boldsymbol{m-1}} |\boldsymbol{g}(\mathrm{xi})| >= 0$ or

$$|\boldsymbol{f}(\boldsymbol{x})>=\sum_{\boldsymbol{i}=\boldsymbol{0}}^{\boldsymbol{m}-\boldsymbol{1}}(\boldsymbol{g}(\boldsymbol{x}\boldsymbol{i}))^2>=0$$

3.2. Length of chromosome

Since $x \in [0,1]$ if the length of decimal is (d) and the number of ($\mathscr{L}$) then the length of chromosome is $\mathscr{L} \times \mathfrak{R}$ where $\mathfrak{R} = \text{upper}(\log_2 10^d)$

Definition (3.1)
The transformation map TQ from a set of sequence of production rules *R= {r1, r2…}* with applied TM TM set as an one tape of 7 tuple, and into a set of quantum functions *QF = {qf1, qf2 …}* which is define by utilzing **TQ: R ×TM → QF.**

If *f is* a function in $\mathbb{R}^2$, then denoted for the number of terms in *f* by Tn, it's calculated by the number of plus operations, and minus operations in the function *f* , which denoted by N_{PM} , and the number of operations in *f* denoted by N_{op} in general , with denoted the number of pluses, minus, multiplication, division, power and reminder operations in *f* by N_0, N_1, N_2, N_3, N_4 and N_5 with respectively, such that

$N_{PM}=<V|W_{1,2}>$

$N_{op}=<V|W>$,

$\Sigma=\{0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F\}$

The base tensor vector of dimension n is $e_i = [e_i^j] = (0 \quad 0 \ \cdots \ 1 \quad \cdots 0 \quad 0)$,

where,

$$e_i^j = \begin{cases} 1 & i = j \\ 0 & otherwise \end{cases}$$

$$W^t = \sum_{i=1}^{5} e_i$$

and

$$W_{1,2}{}^t = \sum_{i=1,2} e_i$$

$$< \mathrm{V}| = (\mathrm{N}_0 \quad \mathrm{N}_1 \quad \mathrm{N}_2 \ \mathrm{N}_3 \quad \mathrm{N}_4 \quad \mathrm{N}_5\ ) \text{ by bra notation}$$

then with design TM of grammatical evolution, and TM of digital number with production rules and symbols corresponding expression of v variables see Appendix Figure (1), Figure (2), and Table (1), for example the sequence of production rules and symbols corresponding data in tape (for example 002502002500030115 corresponding of expressing x^2+1) .
From the sequence of production rules corresponding to function of one or more variables, then can be calculated independent terms of expression. In mathematics algebra the term is either a one of variables, single number, or numbers and variables multiplied together (6x, 1.5y, 0.83 xy, …). The terms are separated by plus, or minus operation (+ sign, or − sign), then the number of terms Tn in expression calculated by summation the number plus and minus out of brackets.

In Genetic Algorithm each expression or production rules of t-terms correspond the chromosome of Tn terms. Crossover is a genetic operator that processes further than 1 parent solutions and yields a child solution from them. Following the process of reproduction (selection), supplementation for the population is with better **individuals**. Through re-combining operator that continues in 3 steps.

Definition (3.2): The two the sequences corresponding to two expressing functions are *congruent* if and only if the functions corresponding to the sequences first sequence and second sequence are equal, denoted by $\cong$.

Definition (3.3): If there are two sequences of production rules, then the definition of *blade function* Ω is a sequence of production rules such that

$$\Omega \text{ (first production rules)} \equiv \text{second production rules.}$$

The blade function Ω removed identity element from sequence of production rules for each operations: identity of addition and subtraction element: x = x ± 0 = 0 ± x, identity of multiplication and division element: x = 1*x = x*1, x = x/1, identity of power operation element: a = a^1, and equivalent expressions for example a + a = 2a, sin(θ+π) = -sin(θ),…

The number of terms in function by utilizing production rules corresponding to specific function, the terms are separated by plus operation or minus operation to calculate the number of terms by summation the number pluses and minuses out of brackets.

Definition (3.4): The *pure sequence of production rules* T_p define by

$$\Omega\,(T_p) \equiv T_p.$$

Since each term have combination of *basic function,* where basic function divided into four types:
1- *Standard Function* (SF): g(x) = α f(βx), where f(x)∈{sin(x), cos(x), tan(x), …, e^x, log(x), ln(x), $\sqrt{x}$, ….}, and α,β ∈ℝ, I.e., g(x) = 2.1 sin(4x), or g(x) = -4.1 $e^{(5x)}$,…
2- *Constant Function* (CF): c, where c is a constant c ∈ℝ, I.e., 45.67676, or -4532.87, ….
3- *Power Function* (PF): αx^n, where a ∈ℝ, and n ∈ N, I.e., 6.27 x^4, 3.958 x^7, …
4- *Line Function* (LF): αx, where a ∈ℝ, I.e., 9.23x, -4x, 5x,…

3.3. Random selection basic functions

The generation of the Random selection basic functions $B_i(x) \in \{SF, CF, PF, LF\}$ is in order to build the new production rule corresponding function *f(x),* where:

$$\psi(x) = \langle \alpha | B(\langle \beta | x \rangle) \rangle$$

Suppose the DE below it is N_{th} order then

$$\sum_i a_i \frac{d^i}{dx^i} \left| \psi(x) \right\rangle = G(x) |\psi(x)\rangle$$

Where the vector solution $|\psi(x)\rangle$ written in a somewhat special notation, and by apply GA in TM design, by continue selection, mutation, and crossover with find the suitable coefficients of $\psi(x)$.

4. Analysis and evaluation of the proposed approach

4.1 ODE Example

By utilizing the above, where n=25 number of terms, and DE is

$$\sum_{i=0}^{2} a_i \frac{d^i}{dx^i} |\psi(x)\Big\rangle = G(x)|\psi(x)\rangle$$

$$a_0 = -100, a_1 = 0, a_2 = 1, \text{and } G(x) = 0$$
$$\frac{d}{dx}|\psi(0) >= 10, \qquad and \;\; 1|\psi(0) >= 0,$$

Suppose the solution of above DEs is linear of two coefficients *a*, and *b*, with type functions. The comparing of the absolute Error by Quantum TM and GA with the absolute Error by pure GA, see Table (1), and Figure (3) of above.

Table 1. Absolute error comparing by quantum TM and GA with the absolute Error by pure GA

xi	The Absolute Error by Quantum TM with GA	The Absolute Error by GA
0	0	0
0.1	3.26e-13	4.43e-08
0.2	4.12e-13	3.86e-09
0.3	2.09e-12	4.99e-08
0.4	3.13e-12	5.17e-08
0.5	4.71e-12	5.81e-08
0.6	5.45e-12	5.26e-08
0.7	5.05e-11	5.81e-08
0.8	5.56e-11	5.17e-08
0.9	4.68e-11	5.29e-08
1	4.91e-11	5.62e-08

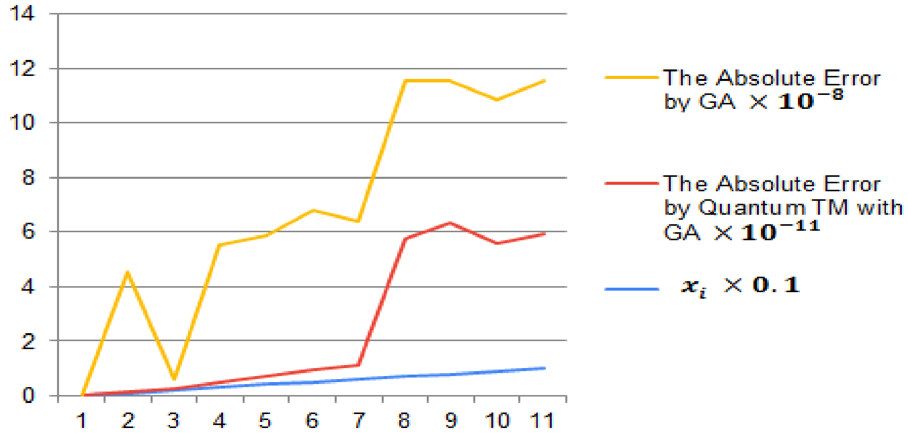

Figure 3. Error comparing between pure GA, with QTM_GA

4.2 PDE Example
To solve the following PDE

$$\sum_{i=0}^{2} a_i \frac{\partial^2}{\partial x_i^2} |u(x,y)\Big\rangle = G(x,y)\rangle$$

$$R = \{(x,y)|\, a < x < b, and\ c < y < d\}$$

With the boundary condition of PDE

$$|u(a,y) >= f_1(y) >, \qquad and \;\; u(b,y) >= f_2(y) >$$

$$|u(x,c)>= g_1(x)>, \quad and \quad |u(x,d)>= g_2(x)>$$

For example, a=c=0, b=2, d=1, f1(y)=0, f2(y)=2e^y, g1(x)=x, and g2(x)=e^x, with h=6, and k=5.

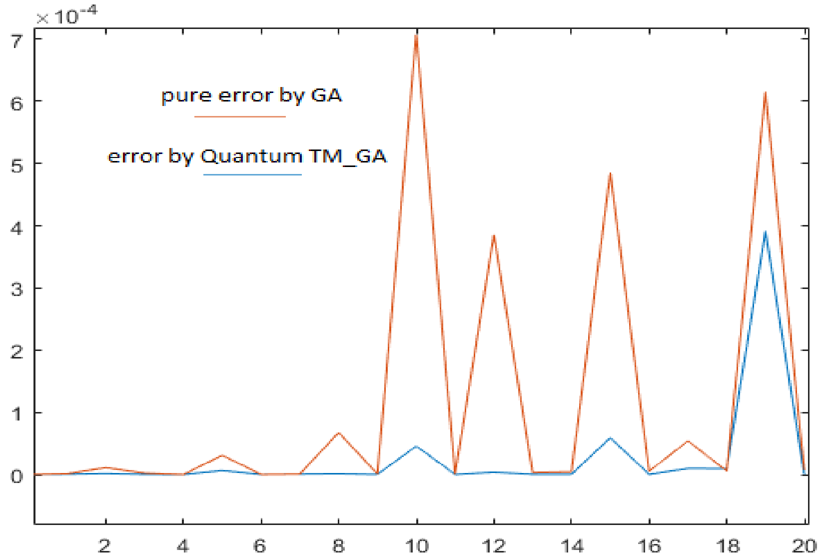

Figure 4. Error comparing between pure GA, with QTM_GA

5. Conclusions

In the proposed work the genetic algorithm with the TM is utilized I order to solve the DE. The proposed Quantum-Genetic-Turning Machine algorithm increased the speed of solving the DEs and the accuracy. In the proposed work the difficulty of the genetic algorithm processing time is processed by utilizing both quantum and turning machine.

References

[1] DEs I MATB44H3F, Version September 15, 2011-1949.

[2] B. Jack Copeland ed. 2004, The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life plus The Secrets of Enigma, Clarendon Press (Oxford University Press), Oxford UK.

[3] Feynman, R. P. Simulating physics with computers. Int. J. Theor. Phys. vol. 21, 467, 1982.

[4] Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a QC. SIAM J. Sci. Stat. Comput. vol. 26, 1484, 1997.

[5] Wootters, W. K. & Zurek, W. H. A single quantum cannot be cloned. Nature vol. 299, pp. 802–803, 1982.

[6] Bennett, C. H. & Brassard., G. Quantum cryptography: public key distribution and coin tossing. Proc. IEEE Int. Conf. Comput. Syst. Signal Process. vol. 175, no. 8, 1984.

[7] Gisin, N., Ribordy, G., Tittel, W. & Zbinden, H. Quantum cryptography. Rev. Mod. Phys. vol. 74, pp. 145–195, 2002.

[8] Lo, H.-K., Curty, M. & Tamaki, K. Secure quantum key distribution Nat. Photon vol. 8, 595, 2014.

[9] Diamanti, E., Lo, H.-K., Qi, B. & Yuan, Z. Practical challenges in quantum key distribution. npj Quantum Inf. vol. 2, 16025, 2016.

[10] Xu, F., Ma, X., Zhang, Q., Lo, H.-K. & Pan, J.-W. Quantum cryptography with realistic devices. Preprint at http://arxiv.org/abs/1903.09051, 2019.

[11] Pirandola, S. et al. Advances in Quantum Cryptography. Preprint at http://arxiv.org/abs/1906.01645, 2019.

[12] Gisin, N. & Thew, R. Quantum communication. Nat. Photon. vol. 1, pp. 165–171, 2007.

[13] Pirandola, S., Eisert, J., Weedbrook, C., Furusawa, A. & Braunstein, S. L. Advances in quantum teleportation. Nat. Photon. vol. 9, pp. 641–652, 2015.

[14] Gerges, Firas; Zouein, Germain; Azar, Danielle (12 March 2018). "Genetic Algorithms with Local Optima Handling to Solve Sudoku Puzzles". *Proceedings of the 2018 International Conference on Computing and Artificial Intelligence*. ICCAI 2018. New York, NY, USA

[15] Eiben, A. E. et al. "Genetic algorithms with multi-parent recombination". PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Diffuclity Solving from Nature: pp. 78–87, 1994.

[16] Dominic W. Berry, et al, "QL for linear DEs with exponentially improved dependence on precision", Communications in Mathematical Physics vol. 356, pp. 1057-1081, 2017.

[17] Aaron Jacob Ostrander, "QLs For DEs „, Doctor of Philosophy, 2019, University of Maryland Institute for Advanced Computer Studies.

[18] Tao Xin, et al, "QL for solving linear DEs: Theory and experiment“, Phys. Rev. A 101, 032307 – Published 6 March 2020.

[19] Benjamin Zanger, et al, " QLs for Solving DEs as ordinary via Classical Integration Methods", Quantum vol. 5, 502, 2021.