Journal of Advanced Research 29 (2021) 147–157

Contents lists available at ScienceDirect

# Journal of Advanced Research

journal homepage: www.elsevier.com/locate/jare

# Optimizing quantum cloning circuit parameters based on adaptive guided differential evolution algorithm

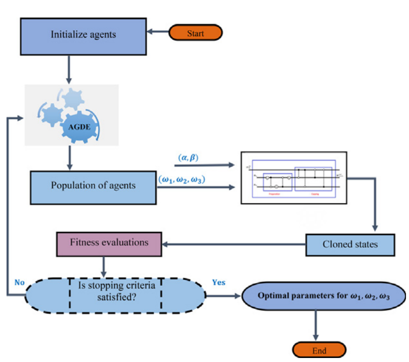
Essam H. Houssein [a,*], Mohamed A. Mahdy [b], Manal. G. Eldin [c], Doaa Shebl [b], Waleed M. Mohamed [a], Mahmoud Abdel-Aty [d]

[a] *Faculty of Computers and Information, Minia University, Minia, Egypt*
[b] *Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni-Suef, Egypt*
[c] *Department of Mathematics and Computer Science, Faculty of Science, Beni-Suef University, Beni-Suef, Egypt*
[d] *Department of Mathematics, Faculty of Science, Sohag University, Sohag, Egypt*

## HIGHLIGHTS

- AGDE is utilized to optimize quantum cloning circuit parameters.
- The experimental results reveal that AGDE is outperformed the other well-known metaheuristics algorithms.
- AGDE is minimized the parameter values of cloning difference error value down to $10^{-8}$.
- The qualitative and quantitative measurements proved the superiority of AGDE.

## GRAPHICAL ABSTRACT




## ARTICLE INFO

*Article history:*
Received 23 July 2020
Revised 8 October 2020
Accepted 11 October 2020
Available online 17 October 2020

*Keywords:*
Adaptive guided differential evolution
AGDE
Cloning fidelity
Cloned qubits
Meta-heuristics
Quantum cloning

## ABSTRACT

*Introduction:* Quantum cloning operation, started with no-go theorem which proved that there is no capability to perform a cloning operation on an unknown quantum state, however, a number of trials proved that we can make approximate quantum state cloning that is still with some errors.
*Objectives:* To the best of our knowledge, this paper is the first of its kind to attempt using meta-heuristic algorithm such as Adaptive Guided Differential Evolution (AGDE), to tackle the problem of quantum cloning circuit parameters to enhance the cloning fidelity.
*Methods:* To investigate the effectiveness of the AGDE, the extensive experiments have demonstrated that the AGDE can achieve outstanding performance compared to other well-known meta-heuristics including; Enhanced LSHADE-SPACMA Algorithm (ELSHADE-SPACMA), Enhanced Differential Evolution algorithm with novel control parameter adaptation (PaDE), Improved Multi-operator Differential Evolution Algorithm (IMODE), Parameters with adaptive learning mechanism (PALM), QUasi-Affine TRansformation Evolutionary algorithm (QUATRE), Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Cuckoo Search (CS), Bat-inspired Algorithm (BA), Grey Wolf Optimizer (GWO), and Whale Optimization Algorithm (WOA).

Peer review under responsibility of Cairo University.
* Corresponding author.
*E-mail addresses:* essam.halim@mu.edu.eg (E.H. Houssein), marafa.mahdy@fcis.bsu.edu.eg (M.A. Mahdy), Manal.Gamal@science.bsu.edu.eg (M.G. Eldin), dshebl@fcis.bsu.edu.eg (D. Shebl), waleedmakram@minia.edu.eg (W.M. Mohamed), amisaty@gmail.com (M. Abdel-Aty).

https://doi.org/10.1016/j.jare.2020.10.001

*Results:* In the present study, AGDE is applied to improve the fidelity of quantum cloning problem and the obtained parameter values minimize the cloning difference error value down to $10^{-8}$.
*Conclusion:* Accordingly, the qualitative and quantitative measurements including average, standard deviation, convergence curves of the competitive algorithms over 30 independent runs, proved the superiority of AGDE to enhance the cloning fidelity.


## Introduction

The quantum world is highly evolved and superior compared to the classical world [1], whereas many challenges we face in the quantum world rather than the classical world, classically, we are able to copy information between systems, but in quantum mechanics, it becomes a challenge; as all we are able to do is just imperfect copying of quantum qubits, which is defined according to 'no-go' theorem [2]. A number of trials are carried out to generalize no-cloning theorem.

Wootters and Zurek [3], firstly, proposed that it's impossible to copy unknown quantum states depending on unitary operations. After that, Bužek and Hillery [4] have introduced that it is possible to clone arbitrary states of single-spin quantum systems. In [5] et al., they used the quantum triplicator to provide appropriate cloning of a single-qubit with a cloning fidelity around 58% is used. Also, Wootters and Bužek [6] proposed a cloning machine that generates perfect copies as an output; if the input state is in the form of basis vector ($|0\rangle, |1\rangle$), then produced output copies are perfect. However the quality of produced copies is affected by the input state since the following input states from the output quality is poor.

Quantum physicists firstly introduced the quantum cloning machines (QCM) [7], that take as input, an arbitrary qubit state to output one or more copies of that input state but a given output produced with errors. Universal quantum cloning machine (UQCM) is one type of the QCMs besides other cloning machines such as probabilistic quantum cloning machine asymmetric quantum cloning machine, and phase co-variant cloning machine [8]. Unlike quantum cloning machines, UQCM generates cloned states so that their quality is not affected with the input state like QCM machines. Thapliyal, et al. [9] proposed two designs for integer division-based quantum circuits using Clifford + T gates, in order to optimize the quantum hardware in design, by reducing total qubits, the introduced two quantum circuits are based on restoring and non-restoring division algorithms integrated with quantum ADD operation, adder-subtractor, and subtraction circuits. Gyongyosi and Imre [10], proposed Quantum Triple Annealing Minimization (QTAM) algorithm, which based on the framework of simulated annealing (SA). Introduced QTAM target to optimizing the physical structure of the quantum circuit, including minimize of the quantum circuit area on hardware structure, and the number quantum gates required for input quantum systems, and measuring output, beside maximizing the objective function of computational problem. Since the world is fast evolving, it makes technology employed in various fields of life. Optimization occupy an important role in solving complex real-world challenges [11]. Many scientific fields such as engineering design problems, economics and system management employ optimization for achieving desired targets. Since optimization's purpose is to reach a nearly optimal value related to one or more objectives for a given problem; more specifically, in the optimization process, the promising solutions required for solving a specific problem is selected among the provided solutions, considering a given problem constraints.

Meta-heuristic algorithms [12] are widely spread in getting optimal solutions for real-world problems in the last decade. Meta-heuristic algorithms are being used in many fields including economics, engineering, information technology and, moreover, all life fields [13]. Meta-heuristic algorithms have remarkable existence in various fields due to their characteristics: Simplicity of algorithms mechanisms, algorithms are more flexible, derivative-free mechanism, and avoiding trapping into local optima sub-regions.

- Firstly, simplicity of meta-heuristic algorithms is mostly a resultant of algorithms inspiration ideas, since meta-heuristics mimic different natural concepts such as simulating swarm of birds, and animals behaviors, and physical phenomena. This simplicity motivate researchers and scientists to simulate various ideas from surrounding natures, introduce new inspired ideas, also hybridize more than one meta-heuristic algorithms together to improve their performance or solve an optimization problem.
- Secondly, meta-heuristics are fairly flexible to be applied for solving complex challenges without necessarily making root changes in used algorithm mechanism for adaptation on problem. Since meta-heuristics consider problems as black boxes, they allow designers to use algorithm with no specific problem adaptation.
- Thirdly, meta-heuristics mostly apply a problem derivative-free mechanisms, as they optimize problems in a stochastic way, starting with random initialized agent(s) and evolved through optimization process to get optimal parameters with no need to compute the derivative of problem search space, giving meta-heuristics an opportunity to be suitable for real challenges with complex search space, and non-predefined problem specific parameters.
- Finally, meta-heuristics have more ability to avoid and get out of local sub-regions in problem space due to, mainly, random nature of algorithms mechanism, besides collaborating agents in population-based meta-heuristics. This feature assist meta-heuristics converge through problem space extensively and outperform through real-time complex optimization problems [14].

Meta-heuristic algorithms are divided into two categories based on the number of employed solutions for searching optimization problem region: single solution algorithms, and multiple solutions (population) based algorithms, where in the single solution based meta-heuristics algorithms the search space is explored with only one individual solution to get improved through all optimization iterations, such as Simulated Annealing (SA) [15], and Hill Climbing [16]; on the other side, in multiple solutions algorithms a population of solutions are randomly initialized and get evolved through iterations of optimization process for searching problem space and achieving the optimal points through problem, population based algorithms. Using population-based meta-heuristic algorithms, a set of solutions are employed that increase the diversity of exploring agents in search space, therefore, increase the probability of search space convergence more than in single solution-based algorithms; in population-based meta-heuristics, population agents can share information about problem search space which saving time required to search and reach promising regions. Since agents can adjust their position according to weights of one or more best agents allow them, to avoid trapping into local optima in problem space [17].

One of the common features of popular meta-heuristics is that their ideas are inspired by the simulation of the best characteristics in nature. In General speaking, research area in meta-heuristics can be categorized into four areas. The first category includes introducing newly proposed meta-heuristic algorithms inspired by the social behavior of swarms like Particle Swarm Optimization (PSO) [18] and Ant Colony Optimization (ACO) [19], natural phenomena like Virus Colony Search (VCS) [20]; laws of natural/biology evolution like Genetic Algorithms (GAs) [21]; Evolution Strategies (ESs) [22], physical phenomena or chemical laws like Henry Gas Solubility Optimization (HGSO) [23] and Lévy flight distribution [24]. In the second researches category, joining two meta-heuristic algorithms together to exploit benefits of the new generated hybrid algorithm like performance in solving more optimization problems [25,26]. The third research area includes researches for evolving the performance of proposed algorithms by incorporating improvement mechanics, including random operators like Levy Flight [27], and mathematical operators [28]. The last research area includes applying meta-heuristic algorithms produced from previous categories for solving real-life challenges that appeared in various fields such as engineering [29], bioinformatics [30,31], information technology [32], feature selection [33], drug design [34,35], and wireless sensors networks [36,37].

The last two decades witnessed fast evolution in the optimization field, and many new meta-heuristic algorithms have been developed, this evolution is related to the No Free Lunch (NFL) theorem [38], which states that if an meta-heuristic algorithm performs well on a set of optimization problems, there are some other optimization problems this meta-heuristic algorithm, will not perform well, which conclude that a specific optimization problem can be solved well with some meta-heuristic algorithms than others. Therefore, we used a set of meta-heuristic algorithms which, provides a better opportunity to obtain overall best optimal parameters that maximize cloning fidelity. To address this issue, and with the rapid development of soft-computing techniques, many meta-heuristic algorithms have recently been designed and used as competitive alternative resolution methods to resolve many real-world issues, due to their simplicity and easy implementation, the Adaptive Guided Differential Evolution (AGDE) is used to solve the optimal parameters that maximize cloning fidelity. As a high performance optimizer and based on the experimental results illustrated in [39], revealed that AGDE is significantly better than, or at least comparable to state-of-the art approaches in terms of robustness, stability and quality of the solution achieved. To the best of our knowledge, this paper is the first of its kind to attempt using the AGDE as a meta-heuristic algorithms in order to obtain overall best optimal parameters that maximize cloning fidelity. In this paper, the experimental results proved that the AGDE was outperformed the eleven competitor algorithms including; Enhanced LSHADE-SPACMA Algorithm (ELSHADE-SPACMA) [40], Particle Swarm Optimization (PSO) [18], Gravitational Search Algorithm (GSA) [41], Whale Optimization Algorithm (WOA) [42], Grey Wolf Optimizer (GWO) [43], QUasi-Affine TRansformation Evolutionary algorithm (QUATRE) [44], Enhanced Differential Evolution algorithm with novel control parameter adaptation (PaDE) [45], Cuckoo Search (CS) [46], Bat-inspired Algorithm (BA) [47], Parameters with adaptive learning mechanism (PALM) [48], and Improved Multi-operator Differential Evolution Algorithm (IMODE) [49].

The paper is organized as follows: In Section "Quantum Cloning Circuit", there is a brief explanation about the used standard gates as well as their employed operations in the proposed Bužek network, also the cloning circuit is explained mathematically. The Adaptive Guided Differential Evolution (AGDE) is presented in Section "Adaptive Guided Differential Evolution (AGDE)". The methodology of the proposed method is explained in Section "Methodology of the proposed method". The experimental results and discussions are introduced in Section "Experimental Results". Finally, the paper is concluded in Section "Conclusion".

## Quantum cloning circuit

Bužek, et al. [6], proposed a network that contains a number of quantum gates to produce two copies of initial qubit states with cloning errors. This network takes three qubits as input, two qubits $(a_2, a_3)$ will be copies of the output, remaining qubit $(a_1)$ is the control target. Mainly, the network is decomposed into two parts, as shown in Fig. 1the first part is the preparation of quantum copier, where initial states of qubits $a_2$ and $a_3$ are prepared and reach state $|\phi\rangle^{(prep)}_{a_2a_3}$. In the network cloning part, original information of qubit $a_1$ is redistributed among the three qubits, the main two parts of the network are described in detail. The quantum gates that represent components in cloning circuit, have two types, the rotation and CNOT operators, one qubit rotation gate is explained as follows:

$$\widehat{R}(\omega) = \begin{pmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{pmatrix} \tag{1}$$

and when acting on basis vectors produces,

$$\widehat{R}_j(\omega)|0\rangle = \cos\omega|0\rangle_j + \sin\omega|1\rangle_j, \widehat{R}_j(\omega)|1\rangle = -\sin\omega|0\rangle_j + \cos\omega|1\rangle_j. \tag{2}$$

A controlled NOT gate (CNOT) is a two-qubit gate, in CNOT gate, input qubit controls the target qubit with a condition such that if input qubit is in state $|0\rangle$, there is no effect on target qubit, but if input state is $|1\rangle$, target qubit is switched to either $|1\rangle$ or $|0\rangle$, according to it's current state. Representing CNOT gate with $\hat{CN}_{mn}$, it affects on basis vectors as follows:

$$\begin{aligned}&\hat{CN}_{mn}|0\rangle_m|0\rangle_n = |0\rangle_m|0\rangle_n, \hat{CN}|0\rangle_m|1\rangle_n = |0\rangle_m|1\rangle_n, and\\ &\hat{CN}_{mn}|1\rangle_m|0\rangle_n = |1\rangle_m|1\rangle_n, \hat{CN}|1\rangle_m|1\rangle_n = |1\rangle_m|0\rangle_n.\end{aligned} \tag{3}$$

### *Preparation stage of network*

First part of network in which two quantum copier qubits $a_2$ and $a_3$ are prepared and represented with a specific state $|\psi\rangle^{(prep)}_{a_2a_3}$. We have initial input state $|\psi\rangle^{(in)}_{a_1}$ in basis set $|0\rangle_{a_1}$ and $|1\rangle_{a_1}$, this state is represented by

$$|\psi\rangle_{a_1} = \alpha|0\rangle + \beta|1\rangle, \tag{4}$$

where $\alpha = \cos(\frac{\omega}{2}), \beta = \exp(i\phi)\sin(\frac{\omega}{2})$, and values of $\omega$ and $\phi$ following the relations depict the operations that are carried out in the preparation part of the quantum cloning network as follows:

$$|\psi\rangle^{(prep)}_{a_2a_3} = \widehat{R}_3(\omega_3)\hat{CN}_{32}\widehat{R}_2(\omega_2)\hat{CN}_{23}\widehat{R}_1(\omega_1)|0\rangle_{a_2}|0\rangle_{a_3}. \tag{5}$$

Carrying out operations in Eq. (5), while using Eqs. (2) and (3), we get the preparation state in the following form:

$$|\psi\rangle^{(prep)}_{a_2a_3} = A_1|00\rangle + A_2|01\rangle + A_3|10\rangle + A_4|11\rangle, \tag{6}$$

where $A_1, A_2, A_3, A_4$ are real coefficients that are functions of $\omega_1$ , $\omega_2$ and $\omega_3$ and are defined by following equations:

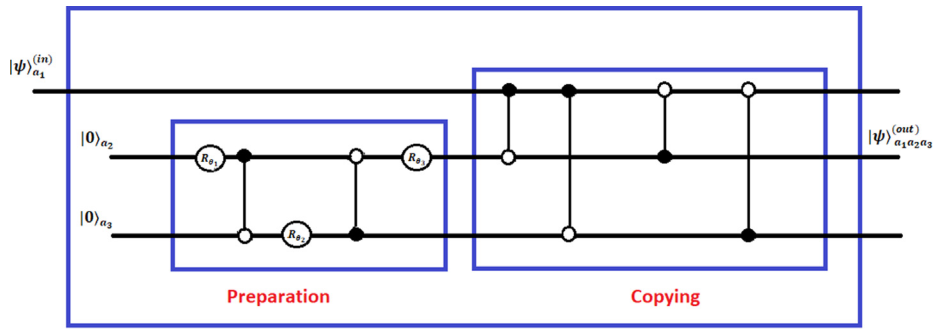



**Fig. 1.** Quantum Cloning Circuit. This circuit consists of two parts, the first part is the preparation, the second one is the copying part, the circuit includes rotation gates $R_j$ that takes square box on lines, and controlled-NOT gates $\hat{CN}_{mn}$ are represented with a dark dot (controlling qubit), and white dot is represented with a cross (target qubit).

$$\begin{aligned} A_1 &= \cos(\omega_1)\cos(\omega_2)\cos(\omega_3) + \sin(\omega_1)\sin(\omega_2)\sin(\omega_3), \\ A_2 &= -\cos(\omega_1)\sin(\omega_2)\sin(\omega_3) + \sin(\omega_1)\cos(\omega_2)\cos(\omega_3), \\ A_3 &= \cos(\omega_1)\cos(\omega_2)\sin(\omega_3) + \sin(\omega_1)\sin(\omega_2)\cos(\omega_3), \textit{and} \\ A_4 &= \cos(\omega_1)\sin(\omega_2)\cos(\omega_3) + \sin(\omega_1)\cos(\omega_2)\sin(\omega_3). \end{aligned} \tag{7}$$

Using proper values of $\omega_1$ , $\omega_2$, and $\omega_3$ enhances the preparation state output.

*Quantum copying stage*

In the second part of quantum copying network, where qubits of quantum copier are prepared in the preparation part, original information for initial state $|\psi\rangle^{in}_{a_1}$ is cloned, the process is performed with four controlled CNOTs operations as follows:

$$|\psi\rangle^{(out)}_{a_1a_2a_3} = \hat{CN}_{31}\hat{CN}_{21}\hat{CN}_{13}\hat{CN}_{12}|\psi\rangle^{(in)}_{a_1}|\psi\rangle^{(prep)}_{a_2a_3}, \tag{8}$$

carrying out previous operation produces a three-qubit state based on the following formula:

$$\begin{aligned} |\psi\rangle^{(out)}_{a_1a_2a_3} &= \alpha A_1|000\rangle + \alpha A_2|001\rangle + \beta A_2|010\rangle + \beta A_1|011\rangle \\ &\quad + \beta A_4|100\rangle + \beta A_3|101\rangle \alpha A_3|110\rangle + \alpha A_4|111\rangle. \end{aligned} \tag{9}$$

Final output state $|\psi\rangle^{(out)}_{a_1a_2a_3}$ is a mixture of three-qubits state, in order to convert into a single-qubit state, the principle of density operator algebra is being used, hence, previous state is expressed with the density operator with following equation:

$$\hat{p}^{(out)}_{a_1a_2a_3} \equiv |\psi\rangle^{(out)}_{a_1a_2a_3}\left\langle\psi\right|^{(out)}_{a_1a_2a_3}. \tag{10}$$

With the density operator, the resulted three-qubit state is represented with an $8\times8$ mixed density matrix, which can be converted into a reduced density matrix of single-qubit state and represented with reduced density operator $\hat{p}_{a_j}$, and j, ranging from 1 up to the number of input states which is 3. In order to find the reduced density matrix for one element, the other two elements are traced out of mixed density matrix, as shown in Eq. (11).

$$\begin{aligned} \hat{p}_{a_1} &= Tr_{a_2a_3}\left[\hat{p}^{(out)}_{a_1a_2a_3}\right], \\ \hat{p}_{a_2} &= Tr_{a_1a_3}\left[\hat{p}^{(out)}_{a_1a_2a_3}\right] \textit{and} \\ \hat{p}_{a_3} &= Tr_{a_1a_2}\left[\hat{p}^{(out)}_{a_1a_2a_3}\right]. \end{aligned} \tag{11}$$

Each reduced density operator gives $2\times2$ matrix that represents an individual output of each cloned state. In order to, accurately perform the cloning process, the resulted density matrices $\hat{p}_{a_1}, \hat{p}_{a_2}$ and $\hat{p}_{a_3}$ are compared with the input state density matrix, $\hat{p}^{(in)}_{a_1} \equiv |\psi\rangle^{(in)}_{a_1}\left\langle\psi\right|^{(in)}_{a_1}$, where

$$\hat{p}^{(in)}_{a_1} = \begin{pmatrix} |\alpha|^2 & \alpha * \beta \\ \alpha\beta* & |\beta|^2. \end{pmatrix} \tag{12}$$

A comparison between two density matrices is performed by measuring difference, which is represented with distance ($\widetilde{D}$) in the square of Hilbert–Schmidt norm.

$$\widetilde{D}\left(\hat{p}^{(in)}_{a_1}, \hat{p}_{a_j}\right) = ||\hat{p}^{(in)}_{a_1} - \hat{p}_{a_j}||^2. \tag{13}$$

Whereas, the fidelity to measure the similarity of two density matrices is

$$F = \mathrm{Tr}\left[\left(\hat{\rho}^{(in)}_{a_1}\right)^{1/2}\hat{\rho}_{a_j}\left(\hat{\rho}^{(in)}_{a_1}\right)^{1/2}\right]^{1/2}, \tag{14}$$

where fidelity value $F \in [0,1]$, the more two density matrices are nearly identical, the more fidelity value $F$ near to value 1 [50].

With optimal parameter values, gives us proper matrices for $\hat{p}_{a_1}, \hat{p}_{a_2}, \hat{p}_{a_3}$ in form:

$$\begin{aligned} \hat{p}_{a_1} &= \begin{pmatrix} |\alpha|^2\left(|A_1|^2+|A_4|^2\right)+|\beta|^2\left(|A_2|^2+|A_3|^2\right) & \alpha\beta^*\left(A_1A_4^*+A_4A_1^*\right)+\alpha^*\beta\left(A_2A_3^*+A_3A_2^*\right) \\ \alpha^*\beta\left(A_1A_4^*+A_4A_1^*\right)+\alpha\beta^*\left(A_2A_3^*+A_3A_2^*\right) & |\alpha|^2\left(|A_2|^2+|A_3|^2\right)+|\beta|^2\left(|A_1|^2+|A_4|^2\right) \end{pmatrix}, \\ \hat{p}_{a_2} &= \begin{pmatrix} |\alpha|^2\left(|A_1|^2+|A_2|^2\right)+|\beta|^2\left(|A_3|^2+|A_4|^2\right) & \alpha\beta^*\left(A_1A_2^*+A_2A_1^*\right)+\alpha^*\beta\left(A_3A_4^*+A_4A_3^*\right) \\ \alpha^*\beta\left(A_1A_2^*+A_2A_1^*\right)+\alpha\beta^*\left(A_3A_4^*+A_4A_3^*\right) & |\alpha|^2\left(|A_3|^2+|A_4|^2\right)+|\beta|^2\left(|A_1|^2+|A_2|^2\right) \end{pmatrix} \textit{and} \\ \hat{p}_{a_3} &= \begin{pmatrix} |\alpha|^2\left(|A_1|^2+|A_3|^2\right)+|\beta|^2\left(|A_2|^2+|A_4|^2\right) & \alpha\beta^*\left(A_1A_3^*+A_3A_1^*\right)+\alpha^*\beta\left(A_2A_4^*+A_4A_2^*\right) \\ \alpha^*\beta\left(A_1A_3^*+A_3A_1^*\right)+\alpha\beta^*\left(A_2A_4^*+A_4A_2^*\right) & |\alpha|^2\left(|A_2|^2+|A_4|^2\right)+|\beta|^2\left(|A_1|^2+|A_3|^2\right). \end{pmatrix} \end{aligned} \tag{15}$$

## Adaptive Guided Differential Evolution (AGDE)

Adaptive Guided Differential Evolution (AGDE) considers a novel alternative of DE algorithm, AGDE have been proposed by Wagdy and Khater [39] to solve performance problems of original DE algorithm such as slow exploitation rate, problem parameters dependency, low performance with dimensionality increasing. In order to cover DE shortcomings, AGDE algorithm employs a new mutation rule, and an adapted value of crossover parameter strategies.

*Initialization*

Population individuals are randomly initialized, such as each $j^{th}$ (i = 1, 2, . . ., NP) individual of population is randomly initialized as follows:

$$x_i = rand(0,1) \times (ub_i - lb_i) + lb_i, \tag{16}$$

where $ub_i$ and $lb_i$ represent upper and lower boundaries of $j^{th}$ individual, respectively, and rand(0,1) is a generated random number between [0,1].

*Mutation*

In order to balance exploration and exploitation processes, AGDE uses a new mutation rule, where population individuals are divided into three parts, first two vectors are located randomly from top and bottom 100p% agents of population, and third vector is selected randomly from the remaining middle [NP - 2(100p%)] individuals as follows:

$$v_i^{G+1} = x_r^G + F \cdot \left(x_{p\_best}^G - x_{p\_worst}^G\right). \tag{17}$$

where $x_{p\_best}^G$ is a random selected vector of the population top 100p % agents, $x_{p\_worst}^G$ is a random selected vector of the population bottom 100p% agents, and $x_r^G$ is a random selected vector of the population middle [NP - 2(100p%)] agents, where F represents mutation factors generated randomly within range [0.1,1].

*Crossover*

AGDE algorithm uses binomial crossovers, where values of target and mutated vectors are mixed to generate a trial vector with the following equation:

$$u_{j,i}^G = \begin{cases} v_{j,i}^G, & \text{if } \left(rand_{j,i} \leqslant \text{ CR or } j = j_{nand}\right) \\ x_{j,i}^G, & \text{Otherwise,} \end{cases} \tag{18}$$

where $i \in [1, NP], j \in [1, D]$, and $rand_{j,i}$ is a random number in [0, 1]. Where CR is the crossover probability, AGDE uses and adaptive CR values are as the following: At each generation G, CR parameter is selected adaptively from one of sets CR1 and CR2, where CR1 $\in [0.05, 0.15]$ and CR2 $\in [0.9, 1]$. Selecting one of these sets depends on each set of experiences of generating promising solutions over previous iterations of optimization process as follows:

$$\begin{aligned} &If\ G = 1 CR_i^1 = \begin{cases} CR_1, & \text{if } u(0,1) \leqslant 1/2. \\ CR_2, & \text{Otherwise.} \end{cases} \\ &\text{Else } CR_i^G = \begin{cases} CR_1, & \text{if } u(0,1) \leqslant p_1 \\ CR_2, & \text{if } p_1 < u(0,1) \leqslant p_1 + p_2 \end{cases} \text{ End If} \end{aligned} \tag{19}$$

where $p_j$ is the probability of selecting set $j$, and $j = 1, 2, \ldots, m$. $m$ denotes the number of sets, $\sum p_j = 1$, and it is initialized with value $1/j$, which is $1/2$. Based on the value of $p_j$, a roulette wheel selection method is used to select the appropriate set for each target vector. Through evolution process, value of $p_j$ is updated, accordingly, as the following:

$$p_j^{G+1} = \left((G-1) \times p_j^{G-1} + ps_j^G\right)/G, \tag{20}$$

$$ps_j^G = \frac{s_j^G}{\sum_{j=1}^m s_j^G} \tag{21}$$

and

$$s_j^G = \frac{ns_j^G}{\sum_{G=1}^G ns_j^G + \sum_{G=1}^G nf_j^G} + \varepsilon. \tag{22}$$

where $G$ represents the generation, $ns_j^G$ is the respective numbers of generated offspring vectors by the $j^{th}$ set that is included in the selection operation through the last G generations, and $nf_j^G$ is the number of generated offspring vectors by $j^{th}$ set and is excluded from the selection process through the last G generations. $s_j^G$ represents the success ratio of the vector generated by $j^{th}$ set and will be included within next generations, and $ps_j^G$ is the probability that $j^{th}$ set will be selected in the current generation, constant ε = 0.01 so that result value not assigned to zero.

**Algorithm 1.** The Pseudo code of AGDE algorithm.

---

$G = 0$
Create a random initial population $\vec{x}_i^G \forall i, i = 1, \ldots, NP$
Evaluate $f(\vec{x}_i^G) \forall i, i = 1, \ldots, NP$
$G = 1$
**while** (G ⩽Gmax) **do**
  **for** $i = 1$ to NP **do**
    Generate $F = \text{rand}(0.1, 1)$
    Compute the (crossover rate) $\text{Cr}_\text{i}$ according to Eq. (19).
    Randomly choose $x_{p\_best}^G$ as one of the $100p\%$ best vectors (top individuals).
    Randomly choose $x_{p\_worst}^G$ as one of the $100p\%$ worst vectors (bottom individuals).
    Randomly choose $x_r^G$ as one of the $(NP - 2(100p\%))$ vectors (middle individuals).
    $j_{rand} = \text{randint}(1, D)$,
    **for** $j = 1$ to $D$ **do**
      **if** $(\text{rand}_{\text{i,j}}[0, 1] < \text{CRorj} = \text{j}_\text{rand})$ **then**
        $v_i^{G+1} = x_r^G + F \cdot \left(x_{p\_best}^G - x_{p\_worst}^G\right)$
      **else**
        $u_{i,j}^G = x_{i,j}^G$
      **end if**
    **end for**
    **if** $(f(\vec{u}_i^G) \leqslant f(\vec{x}_i^G))$ **then**
      $\vec{x}_i^{G+1} = \vec{u}_i^G, (f(\vec{x}_i^{G+1}) = f(\vec{u}_i^G))$
      **if** $(f(\vec{u}_i^G) \leqslant f(\vec{x}_{\text{best}}^G))$ **then**
        $\vec{x}_{\text{best}}^{G+1} = \vec{u}_i^G, (f(\vec{x}_{\text{best}}^{G+1}) = f(\vec{u}_i^G))$
        $ns_j^G = ns_j^G + 1$
      **end if**
      $\vec{x}_i^{G+1} = \vec{x}_i^G$
      $n_j^G = nf_j^G + 1$
    **end if**
  **end for**
  Generate $p_j^{G+1}$, according to Eq. (21), for the next generation
  $G = G + 1$
**end while**

---

### Methodology of the proposed method

The quantum cloning problem consider one of the recent challenges in quantum computing evolved world, the main objective of current study is to perform quantum cloning process on the Bužek circuit (explained in Section *Quantum Cloning Circuit*) with the least error using optimization methods, so that get the original and circuit cloned quantum states nearly identical. In this study, the AGDE employed to solve the quantum cloning problem, and get the optimal parameters for the quantum cloning circuit. Fig. 2 illustrates the flowchart of the proposed methodology. Moreover object process diagram with applied strategy steps in details shown in Fig. 3, the flowchart shows the process follows, and operations carried out to reach optimal solutions, optimization strategy begin with initializing AGDE agents which represent solutions in order to be optimized, in current problem context these agents Xs represent the values of the parameters ($\omega_1, \omega_2,$ *and* $\omega_3$), AGDE optimizer build the population of agents ($X_1, X_2, \ldots, X_n$) according to Eqs. (16) and (18), where the objective function is the cloning fidelity between copied versions states, and original ones. The quantum cloning circuit shown in Fig. 3, according to Section

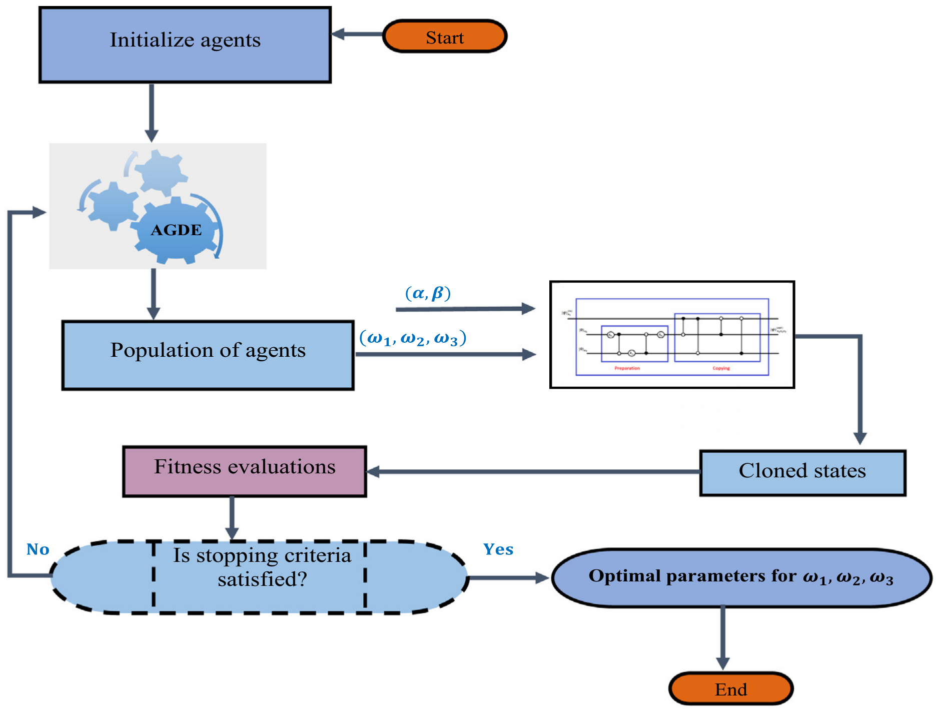



**Fig. 2.** Flowchart of the proposed method to obtain optimal parameters for the cloning circuit.

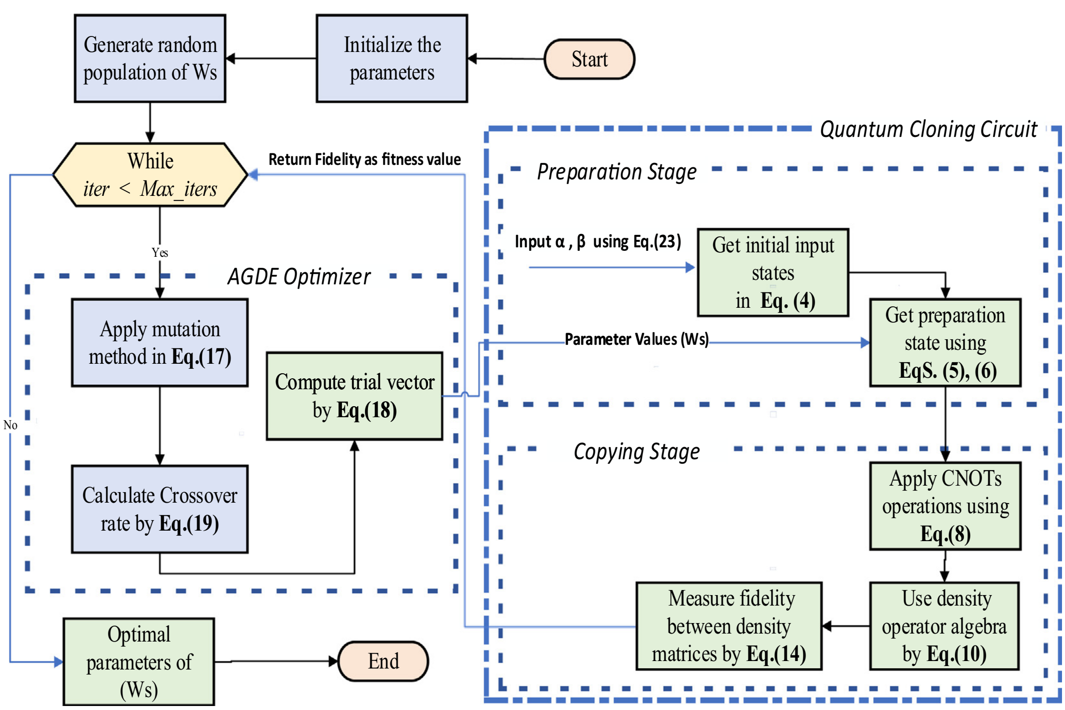



**Fig. 3.** Process diagram of the proposed methodology.

*Quantum Cloning Circuit*, has two main parts: i) Preparation part: receive a pure qupit inputs as in Eq. 4, with values of $(\alpha, \beta)$ parameters generated according to relation explained below in Eq. (23), to constitute a quantum initial state, followed by applying control gates in both Eqs. (5) and (6), depending mainly on parameters values of $(\omega_1, \omega_2, \text{ and } \omega_3)$, which represent solutions in AGDE algorithm. Second stage of cloning circuit perform copying process, where four controlled CNOTs operations applied on prepared state received from the previous preparing stage, through applying Eqs. (8) and (10), a density matrix is produced, then the density operator algebra employed to get density operators represent the copied qubit states. As illustrated in the Fig. 3, the fidelity between the input and output quantum states in Eq. (14), is computed as a fitness value. These steps are repeated till reach the pre-defined criteria, once the stopping criteria become satisfied, the parameter values of optimal solutions that have the ($\omega_1, \omega_2,$ *and* $\omega_3$) values with maximum fidelity are returned.

## Experimental results

This section illustrates the execution conditions, applied parameters, result statistics and explanation graphs. For a fair comparison, the meta-heuristic algorithms are executed in the same environment, experiments are implemented with software Matlab 2014a and carried out on a machine with resources Intel Core i7, 2.9 GHz Processor, and 8 GB of RAM.

The performance of AGDE algorithm, compared with other competitor algorithms. For a fair comparison the maximum number of objective function evaluations (FES) is set to 180,000 for all algorithms, besides number of employed agents in each meta-heuristic algorithm is 30 agents. To achieve a meaningful statistical results, all algorithms evaluated on 30 independent runs and the provided results, include best-so-far and worst fitness value, average, and standard deviation values of best solutions found in each run. Table 1, provides the names of the comparative algorithms and their parameter settings values. To have the best performance, the assigned parameter values are either recommended by algorithms corresponding developers, or within the range of recommendations [51]. The values of $\alpha$, and $\beta$ parameter values are generated according to specific conditions, where their values distributed over the range [0,1]. And the relation between these two parameters is considered during the generation.

From Eq. 4 of initial state where $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the unit vector form of parameters is $|\alpha|^2 + |\beta|^2 = 1$. The relation between $\alpha$ and $\beta$ is concluded as follows [52]:

$$\alpha = \sqrt{1-\beta^2}, \tag{23}$$

$\alpha, \beta \in R$.

The quantum cloning process have been optimized considering the following constraints:

- If $\beta$ = 0 or $\alpha$ = 0, which means $|\psi\rangle = |0\rangle$ or $|\psi\rangle = |1\rangle$, then resulted copy should be in the form $|\psi\rangle\prime = |0\rangle$ or $|\psi\rangle\prime = |1\rangle$.
- Both $\alpha$ and $\beta$ $\neq 0$ at the same time, in this state, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the cloned state is in the following form: $|\psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$

Where $|\psi\rangle$ and $|\psi\rangle\prime$ are the original and cloned states respectively.

### *Performance measures*

In order to evaluate the performance of compared meta-heuristic algorithms for maximizing the quantum cloning fidelity, a set of metrics are employed, including: average and standard deviations and worst and best of obtained fidelities.

- Average: Average of the fitness function value obtained after executing competitive algorithm $N$ times. The mean fitness function is calculated as follows:

$$\text{Average}_F = \frac{1}{N_{\text{run}}}\sum_{i=1}^{N_{\text{run}}} \text{Fit}W^i_{\text{best}} \tag{24}$$

- Standard deviation (STD): STD is computed to determine the deviation of the obtained function values over N times from their central value (average);

$$STD_F = \sqrt{\frac{1}{N_{run}-1}\sum_{i=1}^{N_{run}}\left(FitW^i_{best} - \text{Average}_F\right)^2}, \tag{25}$$

where $N_{run}$ is the total runs, and $FitW^i_{best}$ represents the best fitness obtained for each $j^{th}$ run.

- Best function value: It finds the maximum function value of cloning fidelity, and is obtained over N time as follows:

$$\text{Best}_F = \max_{1\leqslant i\leqslant N_{\text{run}}} \text{Fit}W^i_{\text{best}}. \tag{26}$$

- Worst function value: It finds the minimum function value of cloning fidelity and is obtained as follows:

$$\text{Best}_F = \min_{1\leqslant i\leqslant N_{\text{run}}} \text{Fit}W^i_{\text{best}}. \tag{27}$$

The competitive meta-heuristic algorithms have been executed to get the best values for $\omega_1, \omega_2$ *and* $\omega_3$ as parameters for the used quantum cloning circuit, in order to maximize cloning fidelity. The Convergence curves provided in Fig. 6, shows the convergence comparison between the AGDE algorithm, and competitor methods, to obtain optimal parameter values with maximum fidelity through optimization iterations. The implementation procedure executed based on parameter settings provided in Table 1, with a common implementation settings as explained previously.

The representation of convergence illustrated in Fig. 6, shows the performance of meta-heuristic algorithms through the optimizing process, showing the average of best obtained fitness values, against the function evaluation, since the objective fitness value represent cloning fidelity in Eq. (14), therefore fitness values within range $[0,1]$. It is observed that ADGE algorithm has a comparative convergence rate compared with other meta-heuristic algorithms, concluded with a highest performance in last iterations, while GSA and WOA with lowest convergence rates. Moreover the numerical statistics provided in Table 2, illustrates the average and standard deviations are computed in addition to worst (least) and best (highest) fidelities. The statistical results show the performance of compared algorithms with multiple metrics to assess the used comparative meta-heuristic algorithms on quantum cloning problem, and obtain the near optimal parameters for solving our cloning problem. It's observed from Table 2 that AGDE

**Table 1**
Parameter settings of the ten competitive and selected algorithms.

| Algorithms | Parameters setting |
|---|---|
| Common Settings | Population size: $N = 30$ |
| | Function evaluations (FES): $FE = 18E04$ |
| | Problem dimensions $Dim = 3$ |
| | Number of independent runs 30 |
| AGDE | $CR1 \in [0.05, 0.15], CR2 \in [0.9, 1.0]$ (Default) |
| ELSHADE-SPACMA | $max\_NP = NP, min\_NP = 4.0$ (Default) |
| | arc_rate = 1.4 |
| PaDE | $F = 0.8, CR = 0.6, numStra = 4,$ |
| | $Archfactor = 1.6, pmax = 0.11, pmin = 0.11$ |
| IMODE | $arch\_rate = 2.6$ |
| LPalmDE | $F = 0.8, Cr = 0.6$ |
| | $numStra = 19, ps\_min = 4$ |
| QUATRE | $F = 0.7$ |
| PSO | $Vmax = 6, wMax = 0.9, wMin = 0.2,$ |
| | $c1 = 2, c2 = 2$ |
| GSA | alpha $= 20, G0 = 100,$ |
| | $Rnorm = 2, Rpower = 1$ |
| CS | Nests number = 50 |
| | Discovery rate of alien eggs/solutions = 0.25 (Default) |
| BA | Frequency minimum Qmin = 0 |
| GWO | $a$ decreases linearly from 2 to 0 (Default) |
| WOA | $a$ decreases linearly from 2 to 0 |
| | $a2$ linearly decreases from $-1$ to $-2$ |

**Table 2**
The statistical results obtained from competitive algorithms for the quantum cloning problem.

| Algorithm | Best Fidelity | Mean | Worst Fidelity | STD | CPU Time |
|---|---|---|---|---|---|
| AGDE | **0.999999995** | 0.999999897 | 0.999997989 | 0.000002188 | 413.23 |
| ELSHADE-SPACMA | 0.999999961 | 0.999993703 | 0.999948999 | 0.000015937 | 412.31 |
| PaDE | 0.999999906 | 0.992306082 | 0.923102927 | 0.024315511 | 418.87 |
| IMODE | 0.999999788 | 0.999977232 | 0.999926410 | 0.000031902 | 420.03 |
| LPalmDE | 0.999999924 | 0.999996797 | 0.999993809 | 0.000002228 | 404.06 |
| QUATRE | 0.999999579 | 0.999994861 | 0.999980661 | 0.000006298 | 405.88 |
| PSO | 0.999996402 | 0.999954214 | 0.999766982 | 0.000069631 | 415.88 |
| GSA | 0.987901117 | 0.742289410 | 0.149622455 | 0.260589370 | 417.23 |
| CS | 0.999972212 | 0.998943697 | 0.991096027 | 0.002762260 | 411.64 |
| BA | 0.999999090 | 0.892310293 | 0.692310293 | 0.072762260 | 410.89 |
| GWO | 0.999999762 | 0.990158465 | 0.785189353 | 0.039822465 | 414.95 |
| WOA | 0.999993093 | 0.928632474 | 0.538466739 | 0.144053546 | 422.75 |

algorithm is located in the first rank with the best cloning fidelity of 0.999999995 and, accordingly, achieve the least qubit cloning error down to $10E-08$, in addition to best STD and average fitness values. ELSHADE-SPACMA algorithm come in the second rank following AGDE algorithm with a maximum cloning fidelity of 0.999999961, while GSA and CS algorithms consider the worst two meta-heuristic algorithms. Table 3, illustrates the best parameters, $W_1, W_2$ and $W_3$ obtained for each meta-heuristic algorithm, and related cloning fidelity provided in last column. The best values obtained are $2.35614453, 1.570810624, 2.356131604$ for parameters $W_1, W_2$ and $W_3$ respectively. Cloning may be useful to reduce the complexity of some quantum algorithms and quantum machine learning algorithms [53].

*Qualitative metrics*

This section illustrates the qualitative metrics in Figs. 4 and 5 to confirm the performance of AGDE algorithm through optimizing quantum cloning problem, these metrics include a 2D view of the cloning problem search space, search history, average fitness history, optimization history and diversity. The following points are observed from the resulted qualitative analysis. Regarding the quantum cloning problem domain's topology and search space: The first column of Fig. 4 illustrates the problem search space, and shape of problem topology. It's observed from the resulted shape that the search space of quantum cloning problem has many local sub-regions, which indicates the complexity of problem space. In terms of search history: The second column in Fig. 4 shows the search history of meta-heuristic algorithm agents through problem search space over optimization iterations, where lines in the background represent the contour lines; these lines show the gradation of fitness value from blue to red lines with increasing fitness value. The search history shows that AGDE algorithm is able to search through regions with low fitness values which, helps in achieving our target and maximizing the cloning fidelity.

Regarding the average fitness history: The first column of Fig. 5 illustrates the average fitness history, where the curve shows the average fitness history of meta-heuristic algorithm over optimization iterations; the resulted curve shape assess the performance of algorithm agents and how these agents collaborate to reach to optimal values, and that is reflected on the increase in fidelity value represented with the curve. In terms of optimization history: The optimization history curve in second column in Fig. 5 represents the objective function value obtained with best agent in each iteration from first to the last optimization iteration, the resulted curve illustrates that the objective function seems to increase over optimization iterations. In addition, the shapes of resulted curves in average fitness history and optimization history are mainly similar. This similarity reflects collaboration between search agents in meta-heuristic algorithm to reach an optimal state.

Regarding the population diversity: The last column in Fig. 4 represents the population diversity, this plot displays the average distances between population agents over optimization process. In the first iterations, the diversity value between population agents is high, as meta-heuristic algorithm explore the problem in the first iterations to find promising solutions over quantum cloning problem search space, whereas in the last iterations, diversity values between meta-heuristic algorithm agents decreases over iterations means algorithm in exploitation phase to find a global optimal or nearly an optimal solution between solutions found in previous exploration phase. Therefore, the resulted population diversity curve illustrates the balance between exploration and exploitation phases in order to get optimal parameters for maximizing cloning fidelity.

The purpose of this study is to optimize the parameters of quantum cloning circuits, so that maximize the cloning fidelity between produced copies of initial qubit states. AGDE algorithm and a set of eleven comparative algorithms are employed for solving quantum cloning problem. The experimental results and comparative study, performed demonstrate the reliability of the used methodology,

**Table 3**
The best solution obtained from competitive algorithms for the quantum cloning fidelity.

| Algorithm | $W_1$ | $W_2$ | $W_3$ | *Fidelity* |
|---|---|---|---|---|
| AGDE | 2.35614453 | 1.57081062 | 2.35613160 | **0.999999995** |
| ELSHADE-SPACMA | 0.78534239 | 0.00016784 | 0.78512326 | 0.999999961 |
| PaDE | 0.78524877 | 0.00025842 | 0.78495536 | 0.999999906 |
| IMODE | 0.78524568 | 0.00059293 | 0.78484702 | 0.999999788 |
| LPalmDE | 0.78521733 | 0.00006439 | 0.78495556 | 0.999999924 |
| QUATRE | 0.78551152 | 3.14142549 | 0.78669305 | 0.999999579 |
| PSO | 2.35588463 | 1.57387994 | 3.92759885 | 0.999996402 |
| GSA | 2.37604227 | 1.66599719 | 2.36851750 | 0.987901117 |
| CS | 0.77922253 | 3.14159265 | 0.78194392 | 0.999972212 |
| BA | 2.35730551 | 1.57032684 | 2.35514173 | 0.999999090 |
| GWO | 0.78489772 | 0.00000000 | 0.78486405 | 0.999999762 |
| WOA | 0.77321595 | 3.14159265 | 0.78880131 | 0.999993093 |

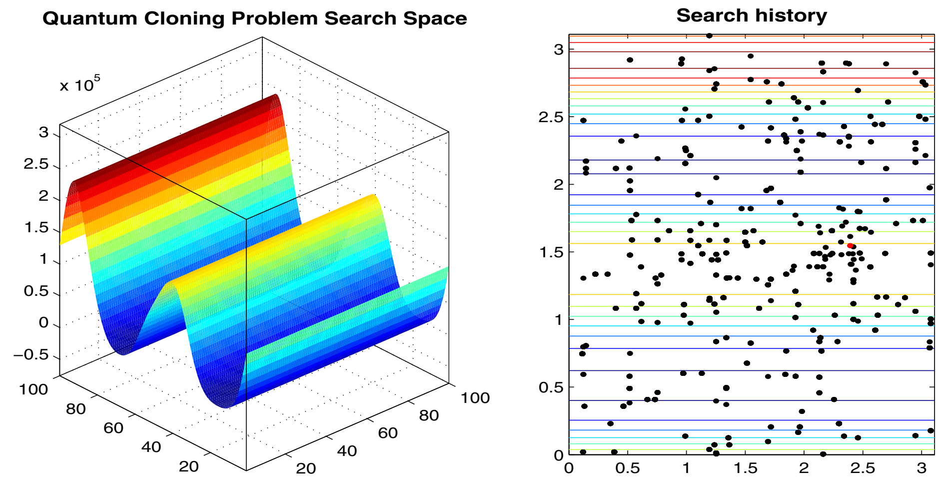



**Fig. 4.** Qualitative metrics on quantum cloning problem: 2D views of the problem search space and search history.

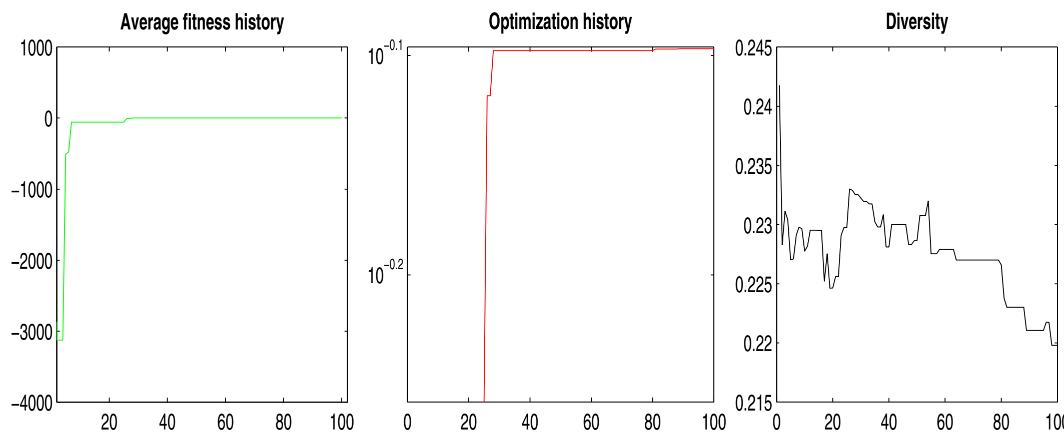



**Fig. 5.** Qualitative metrics on quantum cloning problem: Average fitness history, optimization history and diversity.

and achieving the objective of least cloning error. The employed methodology of applying meta-heuristics, presents certain advantages:

- AGDE used self-adaptation scheme for crossover rate, provides a smooth balance between exploration and exploitation process, besides integrated mutation method gives remarkable effect on convergence speed as illustrated in Fig. 6.
- AGDE algorithm performs well on low dimensional optimization problems according to experimental results of current study in Table 2.
- The existing literature also reports that, applying optimization techniques in quantum cloning domain is hot topic and needed further studies, due to the promising high quality parameter solutions than the traditional methods.

Besides benefits, the proposed methodology also poses some limitations as discussed below:

- Because AGDE is a self-adaptive crossover rate strategy, it is comparatively computationally low expensive than PALM, PaDE, and QUATRE algorithms.
- According to the No Free Lunch (NFL) theorem, that logically no superior meta-heuristic algorithm can solve all the optimization problems, so there is no guarantee that AGDE algorithm, may perform well on another optimization problem.

## Conclusion

Quantum cloning circuit reformulated in a new optimization problem context, along with cloning process constraints, described previously, where objective function, maximization of quantum cloning fidelity. This paper aims to obtain the optimal parameter values for angles to get cloned states with the least error down to $10^{-8}$ implemented on the Bužek quantum cloning circuit using the Adaptive Guided Differential Evolution (AGDE). The obtained best values for $\omega_1, \omega_2,$ *and* $\omega_3$ as parameters are used in quantum cloning circuit, in order to maximize cloning fidelity. To be specific, twelve competitive meta-heuristics including the AGDE were applied to get cloning circuit parameters with cloned qubits least error. The results demonstrate that AGDE can effectively obtain the optimal parameter values for angles, at the same time, AGDE is also better than other well-known meta-heuristic algorithms. Moreover, the experimental results proved that the superiority of AGDE on terms of convergence curves and obtaining the convenient parameter values required for minimizing quantum cloning error compared with competitor algorithms. Eventually, a real IBM simulator device is utilized to confirm the efficiency of the cloning operation based on proposed optimized parameters.

As future work, it would be interesting to extend the application of the AGDE to more practical optimization problems such as the classification and prediction problems as well as multi-objective problems with conflicting criteria. Also, due to the critical of get-

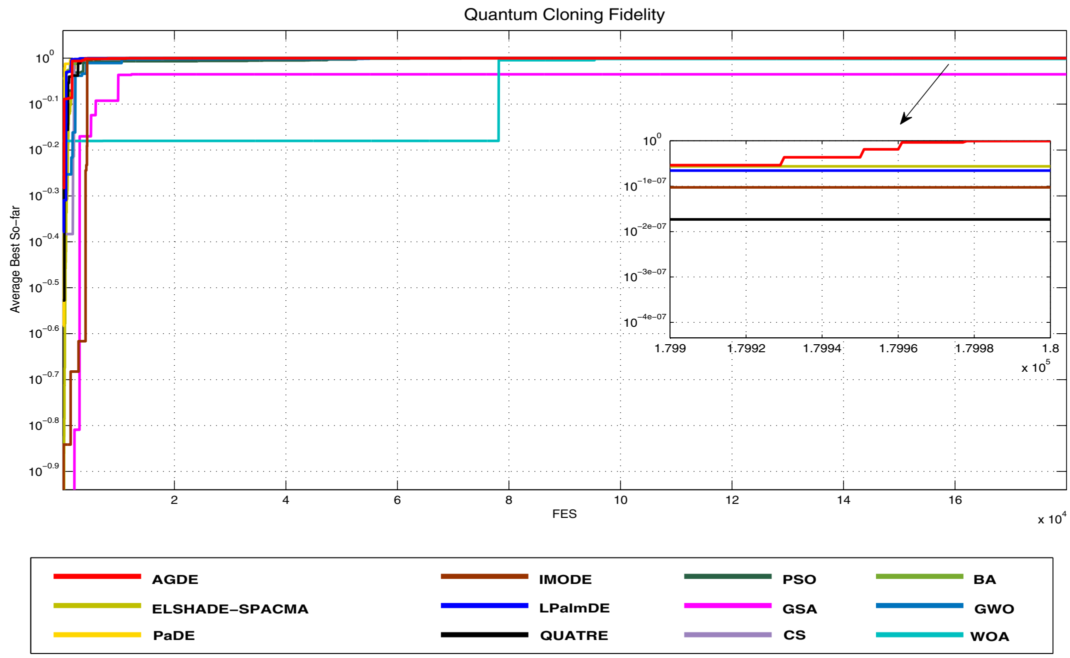



**Fig. 6.** Convergence curves of competitive algorithms.

ting the optimal parameter values for angles to get cloned states with the least error, it is urgent to propose a new optimization method to tackle with this great problem. Due to the promising findings, we suggest using the AGDE as an effective tool to solve complex optimization problems.

## Compliance with Ethics Requirements

*This article does not contain any studies with human participants or animals performed by any of the authors.*

## CRediT authorship contribution statement

**Essam H. Houssein:** Supervision, Project administration, Conceptualization, Methodology, Formal analysis, Writing - review & editing. **Mohamed A. Mahdy:** Software, Resources, Writing - original draft. **Manal. G. Eldin:** Methodology, Writing - review & editing. **Doaa Shebl:** Formal analysis, Visualization, Resources, Writing - review & editing. **Waleed M. Mohamed:** Conceptualization, Formal analysis, Visualization, Resources, Writing - review & editing. **Mahmoud Abdel-Aty:** Supervision, Writing - review & editing.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] Zidan M, Abdel-Aty A, Younes A, Zanaty E, El-khayat I, Abdel-Aty M. A novel algorithm based on entanglement measurement for improving speed of quantum algorithms. Appl Math Inf Sci 2018;12(1):265–9.
[2] Oldofredi A. No-go theorems and the foundations of quantum physics. J Gen Philos Sci 2018;49(3):355–70.
[3] Wootters WK, Zurek WH. A single quantum cannot be cloned. Nature 1982;299(5886):802–3.
[4] Bužek V, Hillery M. Quantum copying: beyond the no-cloning theorem. Phys Rev A 1996;54(3):1844.
[5] Cummins HK, Jones C, Furze A, Soffe NF, Mosca M, Peach JM, Jones JA. Approximate quantum cloning with nuclear magnetic resonance. Phys Rev Lett 2002;88(18):187901.
[6] Buzek V, Vedral V, Plenio M, Knight P, Hillery M. Broadcasting of entanglement via local copying. Phys Rev A 1997;55(5):3327.
[7] Fan H, Wang Y-N, Jing L, Yue J-D, Shi H-D, Zhang Y-L, Mu L-Z. Quantum cloning machines and the applications. Phys Rep 2014;544(3):241–322.
[8] Bužek V, Braunstein SL, Hillery M, Bruß D. Quantum copying: a network. Phys Rev A 1997;56(5):3446.
[9] Thapliyal H, Varun T, Munoz-Coreas E, Britt KA, Humble TS. Quantum circuit designs of integer division optimizing t-count and t-depth. In: 2017 IEEE International symposium on Nanoelectronic and Information Systems (iNIS). IEEE; 2017. p. 123–8.
[10] Zhang JS, Chen AX, Abdel-Aty M. Two atoms in dissipative cavities in dispersive limit: Entanglement sudden death and long-lived entanglement. J. Phys. B: Atom. Mol. Opt. Phys. 2010;43(2):025501.
[11] Houssein EH, Mina Y, Aboul E. Nature-inspired algorithms: a comprehensive review. In: Hybrid computational intelligence: research and applications. CRC Press; 2019. p. 1.
[12] Hussain K, Salleh MNM, Cheng S, Shi Y. Metaheuristic research: a comprehensive survey. Artif Intell Rev 2019;52(4):2191–233.
[13] Dey N, Ashour AS, Bhattacharyya S. Applied nature-inspired computing: algorithms and case studies. Springer; 2020.
[14] Dey N. Advancements in applied metaheuristic computing. IGI Global; 2017.
[15] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science 1983;220(4598):671–80.
[16] Hoffmann J, Nebel B. The ff planning system: fast plan generation through heuristic search. J Artif Intell Res 2001;14:253–302.
[17] Mavrovouniotis M, Li C, Yang S. A survey of swarm intelligence for dynamic optimization: algorithms and applications. Swarm Evol Comput 2017;33:1–17.
[18] Eberhart R, Kennedy J. A new optimizer using particle swarm theory, in: MHS'95. In: Proceedings of the sixth international symposium on micro machine and human science. IEEE; 1995. p. 39–43.
[19] Dorigo M, Di Caro G. Ant colony optimization: a new meta-heuristic. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), vol. 2, IEEE; 1999. p. 1470–7.
[20] Li MD, Zhao H, Weng XW, Han T. A novel nature-inspired algorithm for optimization: virus colony search. Adv Eng Softw 2016;92:65–88.
[21] Holland JH. Genetic algorithms. Sci Am 1992;267(1):66–73.
[22] Mezura-Montes E, Coello CAC. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 2008;37(4):443–73.
[23] Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S. Henry gas solubility optimization: a novel physics-based algorithm. Fut Gen Comput Syst 2019;101:646–67.
[24] Houssein EH, Saad MR, Hashim FA, Shaban H, Hassaballah M. Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 2020;94:103731.

[25] Tharwat A, Houssein EH, Ahmed MM, Hassanien AE, Gabel T. Mogoa algorithm for constrained and unconstrained multi-objective optimization problems. Appl Intell 2018;48(8):2268–83.
[26] Hussien AG, Hassanien AE, Houssein EH, Amin M, Azar AT. New binary whale optimization algorithm for discrete optimization problems. Eng Optim 2020;52(6):945–59.
[27] Heidari AA, Pahlavani P. An efficient modified grey wolf optimizer with lévy flight for optimization tasks. Appl Soft Comput 2017;60:115–34.
[28] Ewees AA, Abd Elaziz M, Houssein EH. Improved grasshopper optimization algorithm using opposition-based learning. Expert Syst Appl 2018;112:156–72.
[29] Del Ser J, Osaba E, Molina D, Yang X-S, Salcedo-Sanz S, Camacho D, Das S, Suganthan PN, Coello CAC, Herrera F. Bio-inspired computation: where we stand and what's next. Swarm Evol Comput 2019;48:220–50.
[30] Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W. A modified henry gas solubility optimization for solving motif discovery problem. Neural Comput Appl 2020;32(14):10759–71.
[31] Shen Q, Shi W-M, Kong W. Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data. Comput Biol Chem 2008;32(1):53–60.
[32] Molina G, Alba E. Location discovery in wireless sensor networks using metaheuristics. Appl Soft Comput 2011;11(1):1223–40.
[33] Neggaz N, Houssein EH, Hussain K. An efficient henry gas solubility optimization for feature selection. Expert Syst Appl 2020:113364.
[34] Houssein EH, Hosney ME, Elhoseny M, Oliva D, Mohamed WM, Hassaballah M. Hybrid harris hawks optimization with cuckoo search for drug design and discovery in chemoinformatics. Sci Rep 2020;10(1):1–22.
[35] Houssein EH, Hosney ME, Oliva D, Mohamed WM, Hassaballah M. A novel hybrid harris hawks optimization and support vector machines for drug design and discovery. Comput Chem Eng 2020;133:106656.
[36] Houssein EH, Saad MR, Hussain K, Zhu W, Shaban H, Hassaballah M. Optimal sink node placement in large scale wireless sensor networks based on harris' hawk optimization algorithm. IEEE Access 2020;8:19381–97.
[37] Ahmed MM, Houssein EH, Hassanien AE, Taha A, Hassanien E. Maximizing lifetime of large-scale wireless sensor networks using multi-objective whale optimization algorithm. Telecommun Syst 2019;72(2):243–59.
[38] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1(1):67–82.
[39] Mohamed AW, Mohamed AK. Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. Int J Mach Learn Cybernet 2019;10(2):253–77.
[40] Hadi AA, Wagdy A, Jambi K. Single-objective real-parameter optimization: Enhanced lshade-spacma algorithm, King Abdulaziz Univ., Jeddah, Saudi Arabia, Tech. Rep.; 2018.
[41] Rashedi E, Nezamabadi-Pour H, Saryazdi S. Gsa: a gravitational search algorithm. Inf Sci 2009;179(13):2232–48.
[42] Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw 2016;95:51–67.
[43] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw 2014;69:46–61.
[44] Meng Z, Pan J-S. Quasi-affine transformation evolutionary (quatre) algorithm: the framework analysis for global optimization and application in hand gesture segmentation. In: 2016 IEEE 13th International Conference on Signal Processing (ICSP). IEEE; 2016. p. 1832–7.
[45] Meng Z, Pan J-S, Tseng K-K. Pade: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization. Knowl-Based Syst 2019;168:80–99.
[46] Yang X-S, Deb S. Cuckoo search via lévy flights. In: 2009 World congress on nature & biologically inspired computing (NaBIC). IEEE; 2009. p. 210–4.
[47] Yang X-S. A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer; 2010. p. 65–74.
[48] Meng Z, Pan J-S, Kong L. Parameters with adaptive learning mechanism (palm) for the enhancement of differential evolution. Knowl-Based Syst 2018;141:92–112.
[49] Sallam KM, Elsayed SM, Chakrabortty RK, Ryan MJ. Improved multi-operator differential evolution algorithm for solving unconstrained problems. In: 2020 IEEE Congress on Evolutionary Computation (CEC). IEEE; 2020. p. 1–8.
[50] Schumacher B. Quantum coding. Phys Rev A 1995; 51 (4): 2738.
[51] Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH. Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst Appl 2020:113377.
[52] Zidan M, Abdel-Aty A-H, Nguyen DM, Mohamed AS, Al-Sbou Y, Eleuch H, Abdel-Aty M. A quantum algorithm based on entanglement measure for classifying boolean multivariate function into novel hidden classes. Res Phys 2019;15:102549.
[53] Abdel-Aty A-H, Kadry H, Zidan M, Al-Sbou Y, Zanaty E, Abdel-Aty M. A quantum classification algorithm for classification incomplete patterns based on entanglement measure. J Intell Fuzzy Syst 2020;38(3):2809–16.