Soft Computing
https://doi.org/10.1007/s00500-022-07460-7

FOCUS

Hybrid quantum genetic algorithm with adaptive rotation angle for the 0-1 Knapsack problem in the IBM Qiskit simulator

Enrique Ballinas[1] · **Oscar Montiel**[1]

Accepted: 15 August 2022

Abstract
A Hybrid Quantum Genetic Algorithm with an Adaptive Rotation Angle (HQGAAA) for the 0-1 knapsack problem is presented. This novel proposal uses the Deutsch-Jozsa quantum circuit to generate quantum populations, which synergistically works as haploid recombination and mutation operators taking advantage of quantum entanglement providing exploitative and explorative features to produce new individuals. Furthermore, the created individuals are updated using an adaptive rotation angle operator that helps refine new individuals to converge to the optimal solution. We performed comparative tests with other quantum evolutionary algorithms and the classical genetic algorithm to demonstrate that this proposal performed better with the tested problem. Results showed that quantum algorithms performed similar but better than the classic genetic algorithm regarding accuracy. Moreover, statistic tests demonstrated that our proposal is faster than the other quantum algorithms tested.

Keywords Quantum-inspired algorithms · Adaptive rotation angle · Knapsack problem

1 Introduction

The 0-1 knapsack problem is a widely studied combinatorial optimization problem with NP-hard computational complexity (García et al. 2018; Ozsoydan and Baykasoglu 2019; Lai et al. 2019). In the literature, there are two classes of approaches that have solved this problem (Jourdan et al. 2009): The first class includes exact methods based on mathematical programming. If the knapsack problem is on a small scale, it is possible to obtain exact solutions with this class. However, as the problem grows, it becomes impractical to use these methods to solve the problem. The second class consists of the use of approximation methods based on metaheuristics.

Metaheuristics have shown to be more effective in solving NP-hard problems than exact methods (Montiel Ross 2020). Some successful examples applied to solve the 0-1 knapsack problem are the Ant Colony Algorithm (ACO) (Calabrò et al. 2020), Tabu Search Algorithm (TS) (Wei and Hao 2021), Sparse Search (SS) (Sapra et al. 2017), Local Search (Wei and Hao 2019), Simulated annealing algorithm (SA) (Delahaye et al. 2019), Iterative local search (Lourenço et al. 2019), Genetic algorithm (GA) (Rezoug et al. 2018), Particle swarm optimization algorithm (PSO) (Adeyemo and Ahmed 2017), Simulated annealing algorithm with noising methods (Zhan et al. 2020), Cuckoo search algorithm with repair operator (Bhattacharjee and Sarmah 2017), and Galactic Swarm Optimization algorithm (Vásquez et al. 2020) among others. Usually, these metaheuristics require long computational times to find the optimal solutions (Wang and Wang 2021) but can solve problems that would be impossible to solve using exact methods.

Quantum computing is an emerging area, which in recent years has gained significant relevance, thanks to the research and developments carried out by companies such as Google (Arute et al. 2019), IBM (Pednault et al. 2019), and D-wave (King et al. 2017). Quantum phenomena properties like superposition, entanglement, and interference have provided quantum computers with exceptional computational capacities such as quantum parallelism for data processing, teleportation of information, and the capability to process millions of times faster than the world's faster supercom-

Communicated by Oscar Castillo.

Enrique Ballinas and Oscar Montiel contributed equally to this work.

✉ Oscar Montiel
oross@citedi.mx

Enrique Ballinas
lballinas@citedi.mx

[1] Instituto Politécnico Nacional - CITEDI, 1310 Ave. Instituto Politécnico Nacional, 22435 Tijuana, Baja California, México

Published online: 05 September 2022

puter. These quantum advantages have raised the interest in designing quantum metaheuristics since it is well known that classical metaheuristics present various problems that negatively impact convergence time and population diversity.

At present, quantum metaheuristic proposals can be divided into two branches:

1. Quantum-inspired metaheuristics. Most of these algorithms were intended to run in classical computers since they mainly date when quantum computers were unavailable. However, quantum phenomena like superposition and entanglement inspire these algorithms; they do not truly exploit quantum phenomena, although research results demonstrate that they can provide better solutions than their classical counterparts. Some quantum-inspired meta-heuristics used to solve the knapsack problem are: Quantum inspired wolf pack algorithm (Gao et al. 2018), Differential evolution algorithm with Grey wolf optimizer (Wang and Wang 2021), Diversification-based quantum particle swarm optimization algorithm (Lai et al. 2018), Quantum-inspired evolutionary algorithms (Han and Kim 2002; Xiang et al. 2017), and Quantum genetic algorithms (Kuk-Hyun and Jong-Hwan 2000; Kuk-Hyun et al. 2001; Valerii and Tkachuk 2018).
2. Quantum metaheuristics (Zhang 2011). These algorithms are designed to be executed into a quantum computer exploiting quantum phenomena indeed. At present, quantum computers are very noisy and limited; hence, the existing proposals are hybrid; i.e., proposals in which one part of the algorithm is executed into a quantum computer and the other part is executed into a classical computer, which imposes many limitations to the software design; even though scientific reports show that they are viable and efficient and can provide solutions to complex functions (Rubio et al. 2021).

As we mentioned, today's most advanced quantum computers are very noisy; hence, it is said that they belong to the "NISQ" (Noisy Intermediate-Scale Quantum) era (Preskill 2018). This era is imposing some implementation restrictions on algorithm design for real applications on quantum computers; some of these are limited number of physical qubits available for developing programs and testing, the quantum circuit depth and coherency of information linked to noise and loss of qubit state phases primarily due to propagation time.

Hybrid quantum-classical computing is a popular way of programming modern quantum processors. The main idea of this kind of computation is to use the classical computer to parametrize the quantum algorithm and divide the program into two parts, the classical part and the quantum part. The classical part will submit a job to the quantum computer to perform some algorithmic tasks; the results are sent back to the classical part. The classical computer will create the new quantum circuits, which will be sent again as a job to the quantum computer. The process will be repeated until the algorithm fulfills the termination criteria.

All quantum-inspired meta-heuristics presented above have solved the knapsack problem better than their classical counterparts. Some of them use the lookup table implemented by Kuk-Hyun and Jong-Hwan (2000) to update the rotation angle. The selection of the correct rotation angle by using the lookup table needs to consider many actors that affect the efficiency of the algorithm (Wang et al. 2013). On the other hand, selecting a fixed value for rotation angle harms the search and convergence process of the algorithm (Valerii and Tkachuk 2018). Because quantum-inspired metaheuristics were not designed to be implemented on a quantum computer, most of them will not work on a quantum computer or quantum simulator for a quantum computer for several reasons. The numeric representation of a quantum bit and a register is one of the main reasons. For example, some quantum-inspired algorithms are implemented using floating-point codification instead of binary codification, which allows a more natural quantum conversion. Another one is handling of the circuit's depth, which is related directly to the loss of information coherency; high-circuit depths produce a high loss of information coherency; quantum-inspired metaheuristics do not have this problem since classical computers only simulate the quantum states.

In the literature review, we did not find any solution proposal for the 0-1 knapsack problem for a quantum computer or its simulator; therefore, there are no previous published results that we can refer to compare to this original work.

The main contributions of this work are:

- We present the first algorithm able to run in a quantum computer for solving the 0-1 knapsack problem.
- Implementing a variable rotation angle operator in our quantum proposal is also one of the contributions. This operator has advantages over selecting a fixed value or a value from the lookup table. In the literature, there are some quantum-inspired-oriented implementations such as Wang et al. (2013), Valerii and Tkachuk (2018), and Valerii (2018) that implement operators similar to the adaptive rotation angle, but they do not have to grapple with the number of physical qubits and coherence problems that a quantum computer has.
- The use of the Deutsch-Jozsa quantum circuit (see subsec. 2.2) to generate quantum populations is the third contribution. Adding this circuit to the metaheuristic provides the algorithm with real quantum advantages such as quantum parallelism, entanglement, and quantum haploid recombination and mutation that works synergistically. This feature provides the algorithm with attributes

and capacities impossible to obtain with a classical computer.

The proposed algorithm is fully scalable to optimize bigger problems —problems with more variables—today's state-of-the-art technology only limits it. Unfortunately, one of the limitations for testing our quantum proposal showing all its potential is the number of qubits that the simulator can manipulate and the availability and disponibility of current quantum computers with an adequate number of qubits. It was tested on the IBMQ Quantum Simulator that is 100% compatible with the IBM quantum computers. Currently, this simulator can only handle a maximum of 32 qubits. The size of a quantum chromosome is the number of qubits that they make up. To solve the knapsack problem, the length of the chromosome should be equal to the number of objects. Therefore, we can only solve knapsack problems with a maximum of 32 objects. The IBM platform has simulators with more qubits, but the quantum gates used in the quantum circuit of the HQGAAA are not compatible with these simulators.

We maintained the integrity of the produced results of these new versions of quantum-inspired algorithms concerning the original versions. These new adapted versions of quantum-inspired algorithms that can be executed on a quantum computer are also contributions since they were not tested before in the 0-1 knapsack problem. The simulator is 100% compatible with their quantum computers, allowing it to test software even if one of their quantum devices is unavailable, which is very common due to several factors, such as the high demand and limited access. Furthermore, the software developed for their quantum simulators can be targeted by modifying some configuration parameters for different quantum computers. Therefore, all the quantum software presented in this paper was tested in the quantum simulator accessed through the cloud.

The organization of this paper is as follows. Section 1 corresponds to this introduction. Section 2 shows the main theoretical concepts involved in this work. Section 3 describes the structure of the proposed HQGAAA. Section 4 presents the experiments and results obtained when implementing the HQGAAA in the IBM Qiskit simulator. Section 5 contains the conclusions. Finally, in Sect. 6 we comment on future work.

2 Theoretical framework

Quantum computing began in 1959 with the research of Richard Feynman (Feynman 1959), where he mentioned that technology would get smaller as the years passed. He also anticipated that quantum mechanics, rather than classical mechanics, would govern those tiny physical devices, and would not necessarily still behave the same as their classical counterparts.

Feynman also revealed that quantum computers could perform operations much faster than conventional computers due to their properties, such as entanglement, superposition, and the inherent quantum parallelism associated with this property. The evolutionary algorithm can use these quantum properties to outperform classical counterparts for specific problems, in particular, for combinatorial problems such as the knapsack problem with a large number of instances, which is a widely studied problem with NP-hard computational complexity (García et al. 2018; Lai et al. 2019; Ozsoydan and Baykasoglu 2019).

Different works in the literature have demonstrated the effectiveness of quantum-inspired evolutionary algorithms with quantum computing in solving the knapsack problem. One of the pioneering works in this area is the Genetic quantum algorithm (GQA) proposed by Kuk-Hyun and Jong-Hwan (2000), and the Quantum-inspired evolutionary algorithm (QEA) introduced by Han and Kim (2002). Based on the experimental results, both algorithms demonstrated their efficacy and applicability for the knapsack problem compared to classical evolutionary algorithms.

Quantum-inspired genetic algorithms have shown significant potential in solving combinatorial optimization problems of the NP-hard class, even providing better results than classical algorithms present in state-of-the-art. In Kuk-Hyun et al. (2001), the authors showed a comparison of a parallel quantum-inspired genetic algorithm, a quantum-inspired genetic algorithm, and a classical genetic algorithm; the results demonstrated the parallel quantum-inspired genetic algorithm's superiority over the other two algorithms that solved the knapsack problem. Parallel computing improves exploitation and exploration capabilities. .

In Valerii and Tkachuk (2018), a Higher-order Quantum-inspired Genetic algorithm to solve the 0-1 knapsack problem with 200, 500, and 1000 objects is presented. The algorithm uses an adaptive quantum gate based on a mathematical formula that considers the amplitude values of the qubits, the possible states if the quantum register were classical, and a rotation angle that varies between 0 and 1, avoiding using the lookup table implemented by Kuk-Hyun and Jong-Hwan (2000) and reducing the execution time compared with the traditional quantum genetic algorithm.

In Valerii (2018), a Quantum-inspired Genetic Algorithm (QGA) to solve the knapsack problem is implemented. QGA uses an adaptive quantum gate to update its quantum population. This gate reduces the amplitudes of the qubit states of the previous results. The experimental results showed that the adaptive quantum gate generates a fast local convergence when solving the knapsack problem.

Another approach using adaptive angles in quantum genetic algorithms is presented in Wang et al. (2013); they

implement a method that automatically updates the value of the rotation angle used by the quantum rotation gate without the need to use the lookup table. The experiments showed that the improved quantum genetic algorithm effectively solves the knapsack problem better than the conventional one.

In Hao (2019), a quantum-inspired evolutionary algorithm for the quadratic knapsack problem is presented, in which the qubits are initialized according to the density value. Unlike the previous works, they used the lookup table to update the rotation angle. In contrast to a classical genetic algorithm and other quantum-inspired evolutionary algorithms, the experimental results show improved effectiveness and convergence when solving the quadratic knapsack problem with multiple instances.

In Jindal and Bansal (2019), a Quantum-inspired evolutionary algorithm for the multidimensional knapsack problem is presented. In this work, experiments were carried out with different repair functions to verify which of them generates the best results. The algorithm uses the lookup table to update the rotation angle.

An improved quantum-inspired evolutionary algorithm for the knapsack problem is proposed in Xiang et al. (2017). First, the algorithm creates an elite group of solutions selected according to their fitness value. Then, elite solutions are used to calculate the rotation angle, comparing it with current solutions.

In Lai et al. (2018), a diversification-based quantum particle swarm optimization algorithm (DQPSO) for the multidimensional knapsack problem is implemented; it uses the quantum particle swarm algorithm (QPSO) and a diversification criterion to generate a better diversity than QPSO. The diversification criterion was also used in Lai et al. (2020). In addition to generating better diversity than the PSO's most popular population update strategy, it also prevents premature convergence.

In Huang et al. (2019), a binary multi-scale quantum harmonic oscillator algorithm (BMQHOA) to solve the knapsack problem is designed; it is based on the probabilistic interpretation of the wave function, using properties such as the quantum tunnel effect, avoiding local optimum. The results showed the superiority of the BMQHOA in precision, convergence, and stability compared to state-of-the-art algorithms.

A quantum-inspired wolf pack algorithm to solve the knapsack problem is presented in Gao et al. (2018). The algorithm is based on the behavior of wolves when hunting. The algorithm uses quantum rotation gates to update the solutions' position. Compared with algorithms present in the literature, the results demonstrated the proposed algorithm's effectiveness, especially for cases where the dataset is large. Another similar approach is presented in Wang and Wang (2021), where they combine a Grey wolf optimizer and a differential evolution algorithm.

A Quantum Annealing Algorithm (QA) that uses parallel computing properties to solve the Multidimensional Knapsack Problem was introduced in Forno et al. (2018). QA is a technique derived from Simulated Annealing Algorithm (SA). Thanks to the quantum tunnel effect, the QA avoids local minima. The experiments with 500 objects showed that QA outperforms its non-parallel version.

We end this section with an algorithm based on Adiabatic quantum computation (AQC) implemented in the IBM Qiskit simulator, presented by López-Sandoval and Cobos (2020). AQC is a quantum annealing (QA) class. Comparing it to its classical counterpart, the results are more effective.

2.1 NISQ era

Currently, one of the main limitations in quantum computers is the noise generated by their qubits and quantum coherence, which can cause unreliable results when measuring (Ballance et al. 2016; Barends et al. 2014). For example, in a quantum computer based on superconductors circuits, the error (noise) generated by a 2-qubit quantum gate is approximately 0.01% Tannu and Qureshi (2019). Therefore, if we want to execute a quantum circuit with more than 1000 quantum gates in series with a depth of 1000, the error would be about 10%. This limitation belong to an era known as the NISQ (Noisy Intermediate-Scale Quantum) era. "Intermediate-Scale" refers to the size of quantum computers that will be available in the coming years, with a number of qubits ranging from 50 to about 100 qubits approximately. "Noisy" emphasizes that we will have imperfect control over those qubits; noise will impose severe limitations on the performance of quantum devices (Preskill 2018).

To achieve sustainable quantum supremacy (the ability of a quantum computer to solve problems that a classical computer cannot), the researchers estimate that we will need machines running between 208 and 420 qubits (Hirzel 2020), depending on the type of circuit used. The most powerful quantum computers presented by IBM (Pednault et al. 2019) and Google (Arute et al. 2019) have 53 qubits. This means that quantum supremacy is not really fully achieved, as Google mentioned in 2019 (Arute et al. 2019). Despite those mentioned above, quantum computers have solved mathematical problems faster than classical computers. However, at this time, quantum computing has to be hybrid and consists mainly of classical computing components (Hirzel 2020; Preskill 2018). These classic elements handle various tasks, from data preparation and parameter selection to data post-processing and analysis. Quantum elements are limited to very specific, though powerful, acceleration or co-processing roles for particular problems.

For the foreseeable future, quantum devices themselves will tend to be quite specialized, with different types of devices (superconductors, trapped ions) especially suitable

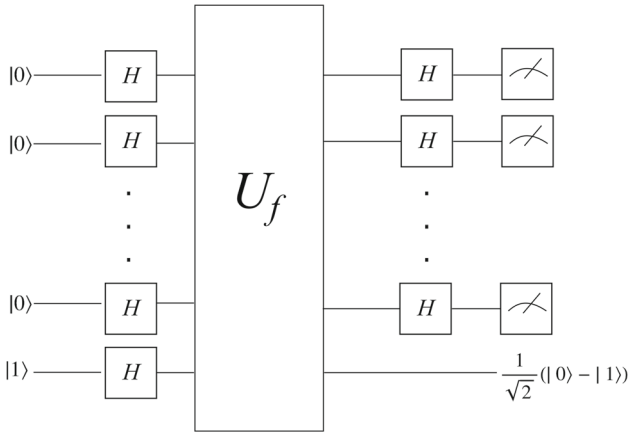

Fig. 1 Block diagram of the Deutsch-Jozsa circuit. U_f represents the Oracle or "black-box"

for different types of problems. For the various components to be orchestrated effectively, the hybrid nature necessitates the implementation and management of workflows.

2.2 Deutsch-Jozsa circuit

The problem is as follows: Let x be a binary number of n bits and $f(x)$ is a function that generates a single output value (0 or 1) for each value of x. In advance, it is known that the function can only have two states, constant, if $f(x)$ returns the same value for all n inputs, or balanced if $f(x)$ returns any other pattern of values (Williams 2011).

The mathematical function $f(x)$ is a "black box" function, so when given an input x the black box (sometimes also called Oracle) will respond with a correct output value for $f(x)$. The goal is to determine when $f(x)$ is constant or balanced by calling the Oracle as few times as possible. The number of times you would have to call the Oracle with n bits using a classical algorithm to identify when $f(x)$ is constant or balanced is $(\frac{1}{2} \times 2^n) + 1 = 2^{n-1} + 1$ (Williams 2011; Cao et al. 2018; Johansson and Larsson 2017; Paredes López et al. 2018); therefore, as the input bit string n increases, the number of calls to the Oracle grows exponentially. If, instead of using a classical algorithm to solve this problem, we use a quantum algorithm, the number of calls to the black box would be only once; this would represent an exponential acceleration at the time of making the decision. In contrast, the problem mentioned above can be solved using the Deutsch–Jozsa quantum circuit presented in Fig. 1. The Deutsch–Jozsa circuit is made up of Hadamard gates and an Oracle based on C-NOT quantum gates. The Oracle is represented by U_f. In the Deutsch-Jozsa circuit, the information presented to the control inputs (data register) of the C-NOT gates are set in the superposition state through the Hadamard gates, in this way the quantum parallelism is being achieved. The last qubit is connected to all the target inputs (target register) of the C-NOT gates (Oracle) to provide a phase kickback to the information presented to this Oracle; so, it must be in the quantum state $|-\rangle$, i,e., $H|1\rangle \rightarrow |-\rangle$. In the IBMQ, all the initial states are set to $|0\rangle$ therefore we need to add a Pauli-X gate to go from $X|0\rangle \rightarrow |1\rangle$.

Figure 1 shows the block diagram of the Deutsch–Jozsa circuit. Mathematically, it can be explained as follows. We need to prepare n qubits of data in accordance with $|\psi_0\rangle = |x\rangle^{\otimes n}|1\rangle$. Then it is necessary to apply a Hadamard gate to each qubit to obtain $|\psi_1\rangle = \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(|0\rangle - |1\rangle)$. The Oracle's operation can be written as $U_f(|x\rangle^{\otimes n}|y\rangle) = |x\rangle^{\otimes n}|y \oplus f(x)\rangle$, since for each x the value of $f(x)$ can be either 0 or 1. Note that $|y\rangle = |-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. Therefore, applying the quantum Oracle we have

$$\begin{aligned} |\psi_3\rangle &= \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(|f(x)\rangle - |1 \oplus f(x)\rangle) \\ &= \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(||0\rangle\rangle - |1\rangle) \end{aligned} \tag{1}$$

The qubit of the Oracle's target output can be ignored. Therefore, apply a Hadamard gate to each qubit of the control register (first register) to obtain 2, where $x \cdot y$ is given by $x_0y_0 \otimes x_1y_1 \otimes \cdots x_{n-1}y_{n-1}$:

$$\begin{aligned} |\psi_3\rangle &= \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\Big[\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}|y\Big] \\ &= \frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}\Big[\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot y}|y\Big] \end{aligned} \tag{2}$$

The last step is to measure the output of the first register. The outcome probability is given by $|0\rangle^n = |\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|^2$.

In our proposed algorithm HQGAAA, we included the Deutsch-Jozsa circuit as a quantum operator that mimics haploid recombination and performs mutations. We found that this circuit accelerates convergence to the optimal value without losing exploratory capabilities.

2.3 Knapsack problem

Combinatorial optimization is the mathematical process of finding the minimal or maximal value of an objective function whose domain belongs to discrete variables. One of the best-known combinatorial optimization problems is the 0-1 knapsack problem, which is computationally challenging due to the NP-hard complexity that it presents (García et al. 2018; Ozsoydan and Baykasoglu 2019; Lai et al. 2019). The problem consists in given a set of n objects, each with weight w_i and value p_i, to determine which objects should be part

of a collection with the condition that the weight WX is less than or equal to a specific limit and the total value PX is what largest possible (Ozsoydan and Baykasoglu 2019).

The mathematical representation of the model is presented in equation 3.

$$\begin{aligned} &\text{maximize } f(x_1, x_2, \ldots, x_n) = PX = \sum_{i=1}^{n} p_i x_i \\ &\text{subject to } WX = \sum_{i=1}^{n} w_i x_i \leq V \end{aligned} \tag{3}$$

where $x_i \in \{0, 1\}$, $i = 1, 2, \ldots, n$. Where $P = (p_1, p_2, \ldots, p_n)$, represent the vector of values and $W = (w_1, w_2, \ldots, w_n)$ the vector of weights of all objects. V is the maximum capacity of the knapsack, $x_i = 1$ indicates that the object is inside of the knapsack and $x_i = 0$ that it is not.

2.4 Quantum computing

Richard Feynman's research laid the foundation for the development of quantum computing. However, the concept of quantum computing had not yet been defined. In 1985, physicist David Deutsch described the concept of a quantum computer as a computational device that uses the properties of quantum mechanics capable of efficiently simulating an arbitrary physical system (Deutsch 1985).

Quantum computing is characterized by using qubits instead of bits, as a classical computer does. Qubits are a unit of information that describes a two-dimensional quantum system (Yanofsky and Manucci 2008). One of the most prominent characteristics of quantum computing is the superposition state. This can be defined as the state in which the qubit is simultaneously at $|0\rangle$ and $|1\rangle$.

Mathematically this is represented as a matrix of complex numbers

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle \equiv \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} \tag{4}$$

where $|c_0|^2 + |c_1|^2 = 1$. Thus, $|c_0|^2$ and $|c_1|^2$ represent the probability of finding the qubit after being measured in the state $|0\rangle$ and $|1\rangle$, respectively (Yanofsky and Manucci 2008; Williams 2011).

Dirac's notation is used to describe a quantum system's state formally. For each "ket" $|\psi\rangle$ there is a corresponding "bra" $\langle\psi|$. The ket and the bra contain equivalent information about the quantum state. Mathematically, they are dual with each other, that is

$$\langle\psi| = c_0^*\langle 0| + c_1^*\langle 1| = (c_0^* c_1^*). \tag{5}$$

In general, a qubit can be found in superposition of the $|0\rangle$ and $|1\rangle$ states. For a quantum register of n qubits, it can generate 2^n possible superposition states. With 2 qubits, the quantum register can be represented as follows

$$|\psi\rangle = c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle \tag{6}$$

where $|c_0|^2 + |c_1|^2 + |c_2|^2 + |c_3|^2 = 1$. This would mean that in a single quantum register we could contain many strings of qubits, each with its corresponding amplitude value.

The general form of a n-qubit quantum memory register is

$$\begin{aligned} |\psi\rangle &= c_0|00\ldots0\rangle + c_1|00\ldots1\rangle + \ldots + c_{2^n-1}|11\ldots1\rangle \\ &= \sum_{i=0}^{2^n-1} c_i|i\rangle \end{aligned} \tag{7}$$

where $\sum_{i=0}^{2^n-1} |c_i|^2 = 1$, and $|i\rangle$ represents the eigen state of the computational base whose bit values match those of the decimal number expressed in base 2 notation.

2.5 Quantum-inspired algorithms

Quantum-inspired evolutionary algorithms are computational methods inspired by quantum mechanics and natural evolution concepts. Its beginnings date back to 1996 with the investigations of Narayanan and Moore (1996). The main objective was to compare the performance of a classical algorithm and a quantum-inspired algorithm in the Traveling Salesman Problem (TSP). The results showed that the quantum-inspired genetic algorithm outperformed the classical version. After that, a basic methodological principle to design a quantum algorithm was presented in Narayanan (1999). The main objective was to identify the novelty and potential of the quantum algorithm in tackling NP-hard problems.

One of the main metaheuristics derived from quantum-inspired algorithms are the Genetic quantum algorithms (GQA) proposed by Kuk-Hyun and Jong-Hwan (2000), and the Quantum-inspired evolutionary algorithms (QEA) designed by Han and Kim (2002). GQA is based on quantum concepts and principles such as qubits and superposition instead of binary, numeric, or symbolic representation. The experimental results in Kuk-Hyun and Jong-Hwan (2000) demonstrated its effectiveness and applicability by experimental results on the 0-1 Knapsack Problem.

The GQA Kuk-Hyun and Jong-Hwan (2000) is a metaheuristic characterized by using quantum registers representing quantum chromosomes. It provides the algorithm with excellent global searchability since quantum chromosomes are intrinsically in the superposition state, allowing it to handle multiple classical chromosome values that are modified

according to an objective value. With this characteristic, the algorithm can find better solutions in a shorter time than the classical algorithms (Kuk-Hyun et al. 2001). For instance, a quantum register of four qubits makes it possible to represent $2^4 = 16$ states. Comparatively, a classical computer requires sixteen registers to represent the same sixteen possible states. For large instances of N qubits (large problems), a quantum algorithm would only need a 32-qubit register to process all required possible states, whereas a classical computer would need 4,294,967,296 registers to process the same information, which is significantly large.

The QEA (Han and Kim 2002) can be considered an improved version of the GQA (Kuk-Hyun and Jong-Hwan 2000), the QEA similar to its predecessor, was designed to solve the knapsack problem. The search process to find the optimal solution is similar in both algorithms. The main difference lies in the migration concept used by QEA. The migration is a process that can induce a variation of the probabilities of a quantum chromosome (Montiel Ross 2020).

When quantum computing and evolutionary computing are combined, three approaches can be identified (Zhang 2011): The first is the Evolutionary-Designed Quantum Algorithm (EDQAs); here, genetic programming is used to generate new quantum algorithms. The second is known as Quantum Evolutionary Algorithms (QEAs), which focus on developing evolutionary algorithms for quantum computers. The last one is Quantum-Inspired Evolutionary Algorithms (QIEAs), which use quantum mechanics concepts such as qubits, superposition, quantum gates, and quantum measurements to develop evolutionary methods for classic computers.

So far, the characteristics offered by quantum computing look promising. The development of quantum computers and simulators by companies such as Google (Arute et al. 2019), IBM (Pednault et al. 2019), and D-wave (King et al. 2017) have impulsed the development of a diversity of novel algorithms. Quantum-inspired evolutionary algorithms were not developed for quantum computers; however, as it was mentioned in (Montiel Ross 2020) it is worth investigating which ideas and algorithms of quantum-inspired metaheuristics can be successfully used in a quantum computer in the NISQ era (Preskill 2018).

2.6 IBM Qiskit simulator

IBM Qiskit is open-source software for working with quantum computers at the level of circuits, pulses, and algorithms. Currently, the platform has 22 quantum computers and five quantum simulators, which can be used through an internet connection. Qiskit is made up of four fundamental elements (IBM 2020). They are: (1) **Terra** that provides a base for composing quantum programs at the circuit level and pulses; with this module, we can perform optimizations for the constraints of a specific device. (2) **Aer** that allows accelerating the development of applications via simulators and noise models. (3) **Ignis** dedicated to fighting noise and errors; it is meant for those who want to work designing quantum error correction codes. (4) **Aqua** is where algorithms for quantum computing are built; this module focuses on constructing solutions for real-world application problems.

3 Hybrid quantum genetic algorithm with adaptive rotation angle for solving the knapsack problem in the IBM Qiskit simulator

This section presents the Hybrid Quantum Genetic Algorithm with Adaptive Rotation Angle (HQGAAA) for solving the knapsack problem in the IBM Qiskit simulator. The HQGAAA and the other quantum algorithms that we developed used the Deutsch-Joza quantum circuit to generate the new quantum population. Figure 2 shows the quantum circuit of the HQGAAA and the other quantum circuits in the IBM Qiskit simulator to solve the Knapsack Problem with 10, 15, and 20 objects.

The circuit shown in Fig. 2 is based on the circuit used to solve the Deutsch-Jozsa problem (see subsect. 2.2). This circuit uses two main properties of quantum algorithms: quantum parallelism and entanglement. In the Deutsch-Jozsa problem, a single evaluation using the Oracle allows us to determine the state of 2^n possible configurations (Cao et al. 2018; Johansson and Larsson 2017; Paredes López et al. 2018). This characteristic was successfully used as a recombination operator working together with the rotation gates. The complete circuit includes a Pauli-X gate, C-NOT gates, rotation gates, and measurement operators. The Pauli-X gate serves as an ancillary gate that will enable the C-NOT (Controlled NOT) gates; the Pauli-X gate can be removed if the q_m qubit is in state one ($|1\rangle$). We decided to use it in this case because the Qiskit simulator initializes all the qubits in state zero. The C-NOT gates generate entanglement between each of the qubits; these gates are part of the Oracle. The Hadamard gates are used at the beginning of the circuit to put in an equal superposition state of the information provided. In the final, the Hadamard gates help produce constructive or destructive interference. The rotation gates ($R_y(\theta)$) are used to update the quantum population using the angle θ. Finally, the measurement operators are included to indicate that after this circuit level, we take measurements. All the experiments performed with the HQGAAA used the circuit shown in Fig. 2.

The length of a quantum circuit is very important since errors are accumulated in serial quantum circuits, and maintaining a low length is crucial to obtaining reliable results in the NISQ era. The hybrid way of programming the quantum

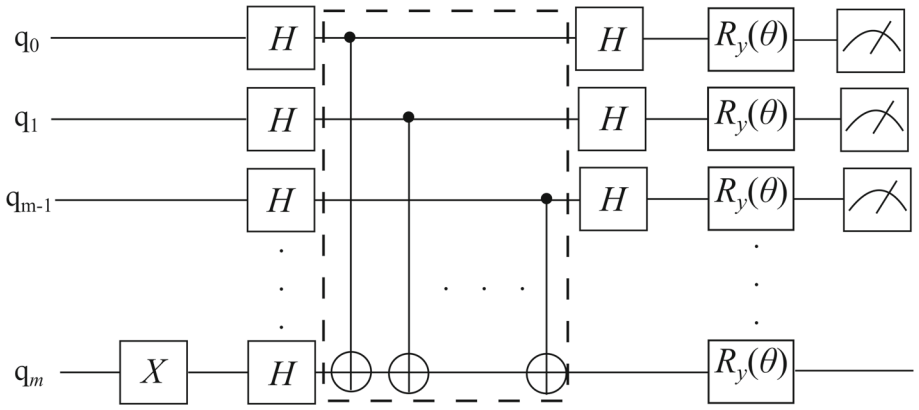

Fig. 2 The HQGAAA uses this quantum circuit as a recombination and nutation operator. The Pauli X-gate was added because the IBM Qiskit simulator initializes all the qubits in the state $|0\rangle$ and the quantum circuit requires an $|1\rangle$ in this position. In the figure, the Oracle is represented by a square dotted line

metaheuristics makes viable maintaining quantum circuit length small. In our proposal, the quantum circuit's length is five, which is considered small. The quantum circuit's width is another important characteristic since it is related to the quantum word length, in our case, the quantum chromosome. In our case, we codified in this quantum chromosome the number of objects to be included in the knapsack and the ancilla qubits Kuk-Hyun and Jong-Hwan (2000); Han and Kim (2002); Kuk-Hyun et al. (2001). Therefore, our experiments were limited to a maximum of 20 objects.

Figure 3 shows a block diagram of the HQGAAA. On the left side are the computational steps executed by the classical computer (CC). On the right side are the steps performed by the quantum computer (QC). This computation model will help us maintain quantum circuits with a small length, avoiding losing coherency. Algorithm 1 and Algorithm 2 will help us to understand better than Fig. 3. The quantum computer performs steps 3, 5, 8, and 9 of Algorithm 1; the CC executes the remaining. The CC initializes the HQGAAA's parameters. The first step is to initialize the generation counter variable t; then, we define the size of quantum chromosomes, i.e., the number of qubits that each quantum chromosome has. That information is sent to the QC. Next, we applied the Deutsch-Jozsa circuit and $R_y(\theta)$ quantum adaptive rotation gates to quantum chromosomes $|\Psi_i^t\rangle, |\Psi_{i+1}^t\rangle, \ldots, |\Psi_n^t\rangle$ to create a quantum population $Q(t)$. Then, we performed n measurements to create the classical population P. In CC, the classical population P is sorted in descending order, repaired, and evaluated. Finally, the best classical solutions are stored in a set of best solutions $B(t)$. The process is repeated until the algorithm reaches the maximal number of generations MAX_GEN.

Algorithm 1 describes our algorithmic implementation proposal of the Hybrid Quantum Genetic Algorithm with adaptive rotation angle (HQGAAA) to solve the knapsack problem. The input of the HQGAAA is a set of quantum chromosomes, i.e, the initial quantum population is defined as $Q(t) = [|\Psi_1^t\rangle, |\Psi_2^t\rangle, \ldots, |\Psi_n^t\rangle]$, where n is the size of the population and t is the generation number. A quantum chromosome is defined as follows:

$$|\Psi_i^t\rangle = \begin{bmatrix} \alpha_{i,1}^t & \alpha_{i,2}^t & \cdots & \alpha_{i,m}^t \\ \beta_{i,1}^t & \beta_{i,2}^t & \cdots & \beta_{i,m}^t \end{bmatrix} \tag{8}$$

where m is the number of qubits and $i = 1, 2, \ldots, n$. The length of a qubit string is the same as the number of items. The algorithm's output will be the best classical solution through generation saved in $B(t)$. As in most of the evolutionary algorithms present in the state of the art, it begins by using a generation counter t.

Algorithm 1 Hybrid Quantum Genetic Algorithm with adaptive rotation angle

Require: The number n of quantum chromosomes, and the maximal number of generations MAX_GEN
Ensure: The best solution b
1: **Begin**
2: $t \longleftarrow 0$;
3: Initialize $Q(t)$ using a Deutsch-Jozsa circuit and R_y quantum rotation gates
4: **while** $t < MAX\_GEN$ **do**
5: Measure $Q(t)$ using equation 9 to generate $P(t)$
6: Repair $P(t)$ using Algorithm 2
7: Evaluate $RP(t)$ using the fitness function $\sum_{i=1}^{n} p_i x_i$
8: Generate the new population $Q(t)$ using the Deutsch-Jozsa circuit
9: Update $Q(t)$ using R_y quantum rotation gates with an adaptive rotation angle
10: Store the best solution b of $RP(t)$ in $B(t)$
11: $t = t + 1$
12: **end while**
13: **End**

In step 2, the generation counter is initialized. In step 3, the quantum chromosomes $Q(t)$ are set in zero states, then, using the Deutsch-Jozsa circuit (see Fig. 2), the quantum chromosomes are put in the superposition state, and quantum entanglement is generated. For the experiments, a population of one quantum chromosome was used. Step 4 is a while loop; the condition criteria is the maximum number of generations MAX_GEN. In step 5, the quantum population is measured (observed) to generate classical population (0 and

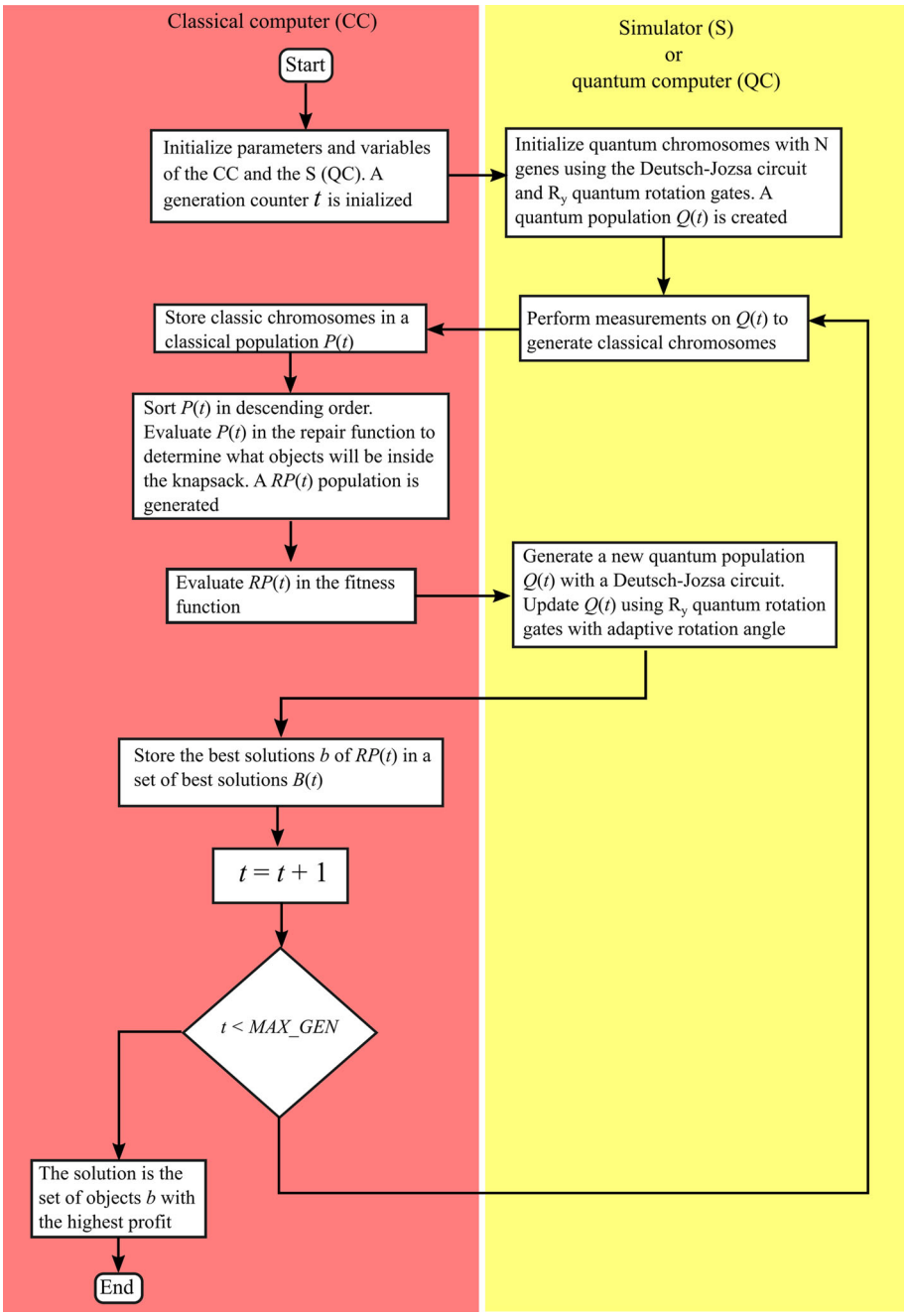

Fig. 3 Hybrid quantum genetic algorithm. On the left side are the classical computer's steps. The steps executed by the quantum computer or quantum simulator are on the right side. In the figure, RP is the classic population after having been evaluated in the repair function, and MAX_GEN is the maximum number of generations

1) according to $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α represents the probability that the qubit collapses to state zero ($|0\rangle$) and β the probability that the qubit collapses to state one ($|1\rangle$) after being measured through a measurement operator M which is a Hermitian operator that has a spectral decomposition $M = \sum \lambda_m P_m$; where P_m is a projector onto the eigenspace of M with eigenvalue λ_m Rubio et al. (2021). Applying Born's rule, the probability of measuring λ_m is calculated, then we can calculate the post-measured state $|\psi'\rangle$ using:

$$|\psi'\rangle = \frac{P_m|\psi\rangle}{\sqrt{\langle\psi|P_m|\psi\rangle}} \tag{9}$$

A repair algorithm was used in step 6, the main objective of this algorithm (Algorithm 2) is to add or remove items from the knapsack respecting the restriction presented in equation 3. The inputs of Algorithm 2 are classic chromosomes $P(t)$, and the output will be classic repair chromosomes $RP(t)$. First, algorithm 2 initialize a variable knapsack-full; this indicates if the knapsack is filled (true) or not (false). If the sum of the weight w_j of each object x_j is greater than the total capacity V of the knapsack, knapsack-full = true,

Algorithm 2 Repair Algorithm

Require: Classic chromosomes $P(t) = 0\ldots000+\ldots+1\ldots111_{2^n-1}$, n is the number of qubits
Ensure: $RP(t)$
1: **Begin**
2: knapsack-full ← false
3: **if** $\sum_{j=1}^{m} w_j x_j > V$ **then** knapsack-full ← true
4: **end if**
5: **while** knapsack-full = true **do**
6: Select a j-th item from the knapsack
7: $x_j \leftarrow 0$
8: **if** $\sum_{j=1}^{m} w_j x_j < V$ **then** knapsack-full ← false
9: **end if**
10: **end while**
11: **while** knapsack-full = false **do**
12: Select a j-th item from the knapsack
13: $x_j \leftarrow 1$
14: **if** $\sum_{j=1}^{m} w_j x_j > V$ **then** knapsack-full ← true
15: **end if**
16: **end while**
17: **End**

and the algorithm continues in step 5; otherwise, it continues in step 11. In step 7 of Algorithm 1, the profit of a solution x is evaluated by $\sum_{i=1}^{n} p_i x_i$, and it is used to find the best solution b to store in $B(t)$ (step 10) after the update of $|\Psi_i\rangle$, $i = 1, 2, \ldots, n$. A new quantum population $Q(t)$ is created (step 8) by using a Deutsch-Jozsa circuit (see Fig. 2) and updated (step 9) with $R_y(\theta)$ rotation gates. The Deutsch-Jozsa circuit works analogously to a classical recombination operator. The $R_y(\theta)$ rotation gate improves the outputs of the Deutsch-Jozsa circuit. The j-th qubit value (α_j, β_j) is updated as

$$\begin{bmatrix} \alpha'_j \\ \beta'_j \end{bmatrix} = \begin{bmatrix} \cos(\theta_j) & -\sin(\theta_j) \\ \sin(\theta_j) & \cos(\theta_j) \end{bmatrix} \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix}. \tag{10}$$

To determine the rotation angle for the quantum gate $R_y(\theta)$ the method proposed by Wang et al. (2013) was used. The formula used to find the proper rotation angle is shown below

$$\theta_i = \theta_{max} - \left(\frac{\theta_{max} - \theta_{min}}{itmax} \right) * iter. \tag{11}$$

Where θ_i is the value of the rotation angle of the i-th generation, θ_{max} is the maximal rotation angle, θ_{min} is the minimal rotation angle, $itmax$ is the maximal generation, and $iter$ is the current generation. According to formula (11), we can conclude that the value of θ_i will gradually increase with increasing $iter$ generations, only if the value θ_{max} is negative. Otherwise, θ_i will be reduced. Finally, in step 11, the generations counter t increases until a maximum number of generations MAX_GEN is reached.

4 Experiments and results

We designed four sets of comparative experiments to determine different aspects of our proposed algorithm and then to demonstrate the contributions of the proposal. In general terms, we determined the rotation angle range, the overall performance, the convergence time, and the accuracy and precision.

All the experiments were achieved on the IBMQ software development platform. This environment has five simulators compatible with their quantum computers. We used the ibm_qasm_simulator to perform all the experiments. The maximum number of qubits in this simulator is 32; this is a technological limitation for testing our proposal with more objects; however, the developed algorithms are scalable and can be tested with more objects in the near future.

For the experiments, we generated a data set using (12)

$$\begin{aligned} w_i &\sim \mathrm{U}[1, 10) \\ p_i &= w_i + 5 \end{aligned} \tag{12}$$

where U represents a discrete continuous distribution; w_i and p_i represent the weight and the profit of each object, respectively. The average knapsack capacity was calculated with $V = 1/2 \sum_{i=1}^{m} w_i$. The complexity of an instance can be defined as the correlation between the weights of each object and its profit (López-Sandoval and Cobos 2020; Pisinger 2005). This complexity is classified as: Uncorrelated, Weakly correlated, Strongly correlated, Inverse strongly correlated, Almost strongly correlated, and Subset sum.

Strongly correlated instances are challenging to solve (Pisinger 2005), and they were used in this work to test the algorithms' performance. We compared the HQGAAA for 10, 15 an 20 objects. The algorithms selected to achieve the comparison are the Quantum-inspired Evolutionary Algorithm with Adaptive Angle using local migration (QIEAAA-local), the Quantum-inspired Evolutionary Algorithm with Adaptive Angle using global migration (QIEAAA-global), and the Classical Genetic Algorithm (GA).

In current literature, the method to determine the rotation angle and its direction is based on the lookup table proposed by Kuk-Hyun and Jong-Hwan (2000). However, using this table involves considering multiple conditions, affecting the efficiency of the algorithm (Wang et al. 2013). In this paper, we use an adaptive rotation angle (see equation 11) instead of the lookup table to calculate the rotation angle for the quantum rotation gate.

The first set of experiments consisted in determining the valid range of the rotation angle. The results presented in Table 1 were performed with 15 objects. In the left column are the three tested algorithms. The column θ_{max} and θ_{min} shows four different cases for each algorithm. The next three columns show the statistical results of 10 experiments for

Table 1 Experimental results of testing different θ angles to find the most appropriate rotation angles for the HQGAAA, QIEAAA-global, and QIEAAA-local

Algorithm	$\theta_{max}, \theta_{min}$	Best	Average	Worst	Std
HQGAAA	$0.005\pi, 0.05\pi$	88.31	88.08	88.06	0.08
	$0.01\pi, 0.1\pi$	95.19	91.89	88.06	2.32
	π, -π	**114.60**	**114.60**	**114.60**	**0**
	$-0.005\pi, 2\pi$	**114.60**	**114.60**	**114.60**	**0**
QIEAAA-global	$0.005\pi, 0.05\pi$	94.14	88.56	85.62	2.60
	$0.01\pi, 0.1\pi$	94.14	91.84	88.06	2.01
	π, -π	**114.60**	113.47	**113.18**	**0.59**
	$-0.005\pi, 2\pi$	**114.60**	**113.61**	**113.18**	0.68
QIEAAA-local	$0.005\pi, 0.05\pi$	88.41	87.60	85.58	1.06
	$0.01\pi, 0.1\pi$	94.14	92.54	88.43	1.96
	π, -π	114.60	**114.45**	113.18	**0.45**
	$-0.005\pi, 2\pi$	**114.60**	114.31	**113.18**	0.59

Best results in bold

each case. For example, for the second case of the HQGAAA, the θ_{max} value is 0.01π and the θ_{min} value is 0.1π, the Best value is 95.19, the Average value is 91.89, the Worst value is 88.06, and the Standard deviation (Std) is 2.32. The best results are shown in bold. For the three tested algorithms, we conclude that formula 11 can calculate the adequate rotation angle in the range $[0, 2\pi]$, and it is not necessary to impose any restriction.

The second set of experiments consisted in determining the overall performance of the proposed HQGAAA with no migration, and the other quantum versions named QIEAAA-local (local migration) and QIEAAA-global (global migration); we performed comparative experiments. To achieve this task, we used the results obtained with a classical GA as ground truth, which is fair and valid. It is fair because the HQGAAA, as well as the QIEAAA-local and the QIEAAA-global, are quantum versions of a GA, i.e., the solution coding using chromosomes might be considered equivalent since each bit and qubit codifies one object, and the evolutionary idea is similar, although the quantum version takes advantage mainly from quantum parallelism. The GA is one of the EA that can satisfy these criteria better. Furthermore, it is valid since it is known that the GA can produce optimal results when optimizing the 0-1 knapsack problem for 10, 15, and 20 objects. In Table 2, we summarized the results of this second set of experiments devoted to determining the performance of our proposal, the HQGAAA. The left column shows the number of objects, in the next column are the tested algorithms, and the last four columns contain statistical results of the four tested algorithms. All the experiments were executed 30 times during 200 generations. For the case of 10 objects, all the algorithms reach the same optimal value obtained by the GA; however, in general, the three quantum proposals performed better than the GA since the standard deviation (Std) of the results was thousands of times better. For the case of 15 objects, all the algorithms reached the optimal values; in general, the three quantum proposals performed better than the GA, although the HQGAAA and the QIEAAA-local were the best since the standard deviation of these two algorithms was 50% smaller than the obtained by the GA. Similarly, for the case of 20 objects, the three quantum algorithms performed better than the GA; however, the HQGAAA performed better than the others. In this last case, the standard deviation of the HQGAA was 11.6% smaller than the QIEAAA-local, 27.6% smaller than the QIEAAA-global, and 74.8% smaller than the GA.

The third set of experiments was focused on analyzing the convergence time of the proposed algorithms. The objective of the experiment is to determine statistically if the quantum proposals have advantages and which ones are the best. Therefore, we needed to perform hypothesis tests using the data of the four algorithms for 10, 15, and 20 objects. For every case, we ran the algorithms 30 times during 200 generations. Similar to the experiments mentioned above, the GA is the ground truth. Using the obtained results, we made Tables 3 to 8 for 10, 15, and 20 objects.

We followed the next methodology to determine which hypothesis test should be applied, parametric or non-parametric.

1. Perform the Shapiro–Wilk hypothesis test to determine if the obtained results of each algorithm are normally distributed (null hypothesis H0 equals True) or not and then determine the type of hypothesis test (parametric or non-parametric) that we should apply. We used a significance level $\alpha = 0.01$, i.e., the probability of rejecting H0 when it is True is 0.01. The three tables (Tables 3, 5, and 7) show these results.

Table 2 Best, worst, average solutions, and standard deviation (Std) of the quantum metaheuristics and classical metaheuristic for the knapsack problem with 10, 15, and 20 objects

Objects	Algorithm	Best	Average	Worst	Std
10	HQGAAA	**82.35**	**82.35**	**82.35**	**4.33E-14**
	QIEAAA-local	82.35	82.35	82.35	4.33E-14
	QIEAAA-global	82.35	82.35	82.35	4.33E-14
	GA	82.35	80.75	81.87	0.74
15	HQGAAA	**114.59**	**113.18**	**114.50**	**0.35**
	QIEAAA-local	114.59	113.18	114.50	0.35
	QIEAAA-global	114.59	113.18	113.37	0.48
	GA	114.59	113.18	113.98	0.71
20	HQGAAA	**155.47**	**153.18**	154.74	**0.76**
	QIEAAA-local	155.47	152.37	**155.009**	0.86
	QIEAAA-global	155.47	152.08	153.86	1.05
	GA	155.47	146.58	150.51	3.02

Best results in bold

2. At this point, there are three situations to consider.

   (a) The results of the Shapiro–Wilk test are True for the four tested algorithms, i.e., the results indicate that we are dealing with data whose statistical distribution is normal. In this case, we performed the Welch parametric hypothesis test using a significance level $\alpha = 0.001$.

   (b) The results of the Shapiro–Wilk test are False for the four tested algorithms, i.e., the data do not have a normal distribution. In this case, we performed a non-parametric hypothesis test, such as the Wilcoxon signed ranks test (Derrac et al. 2011) using a significance level $\alpha = 0.001$.

   (c) The results of the Shapiro–Wilk test are True for some cases and False for the other cases. In this case, it is necessary to use the statistical procedure named bootstrapping to reshape the False datasets and then be able to apply the Welch hypothesis test. These results are shown in Tables 4, 6, and 8 with an appended label "boots" next to algorithm's name.

For the case of 10 objects, Table 3 shows that the distribution probability of the GA and the QIEAAA-local is non-normal, and the distribution of the HQGAAA and the QIEAAA-global is normal. Therefore, we are in the case 2(c), and we have to achieve the bootstrapping method and the results are shown in Table 4 where now all the algorithms have a normal distribution. In the Shapiro–Wilk test section are the mean and standard deviation of all the tests, and it can be seen that the HQGAAA needed fewer generations to reach the optimal value in comparison with the other quantum algorithms. In the last section of this table, the results of the hypotheses of Welch's test show significant differences between the quantum algorithms and the classical GA. Apparently, in favor of the GA, however, this value is not important since the execution time of quantum algorithms is some millions of times faster than the most sophisticated classical computer system (Wu et al. 2021; Zhong et al. 2021); hence, the quantum algorithms are faster.

For the case of 15 objects, Table 5 shows that the distribution probability of the GA and the HQGAAA are normal, and the distribution of QIEAAA-local and QIEAAA-global is non-normal, and we also are in the case 2(c) of the method. Therefore, we also have to apply the bootstrapping method obtaining Table 6. This table shows that also the HQGAAA is the faster of the quantum methods since it needs fewer generations to reach the optimal values. The Welch's hypothesis test also shows a significant difference between the quantum algorithms and the classical; the observation about this issue is similar to the above mentioned.

For the case of 20 objects, Table 7 shows that the distribution probability given by the Shapiro–Wilk test is normal only for the QIEAAA-global; the rest of the algorithms have non-normal distribution. Hence, we also are in the case 2(c). Therefore, we have to apply the bootstrapping method; the results are present in Table 8. The statistical results show that the HQGAAA is faster than the other quantum algorithms. Similarly, to the comments of the above cases, it is not worth performing a comparison of the quantum algorithms with the classical algorithms since, as we mentioned, quantum computers can execute the algorithms millions of times faster (Wu et al. 2021; Zhong et al. 2021) and some few more generations that quantum algorithms required to reach the optimal values are insignificant.

The fourth set of experiments aims to demonstrate if there is a significant difference in accuracy among the algorithms. The methodology to perform this test is the same as the previ-

Table 3 Shapiro–Wilk test regarding convergence time with 10 objects

Run	GA	HQGAAA	QIEAAA-local	QIEAAA-global
1	1	23	46	119
2	1	28	46	110
3	1	27	46	122
4	1	28	46	122
5	1	27	46	120
6	1	27	46	120
7	1	25	46	123
8	1	27	46	121
9	1	22	46	118
10	1	26	46	115
11	1	23	46	121
12	1	29	46	115
13	1	23	1	119
14	1	30	46	114
15	5	25	46	123
16	2	20	1	120
17	5	22	1	118
18	1	27	52	120
19	2	27	52	119
20	1	21	46	114
21	1	22	46	110
22	1	27	46	115
23	1	32	46	115
24	1	25	46	117
25	1	24	46	114
26	1	27	46	112
27	1	28	46	116
28	2	24	46	112
29	1	30	46	114
30	1	26	46	117
Mean	1.366	25.733	41.9	117.166
Variance	30.966	241.866	5,642.70	404.166
Stand. Dev.	1.015	2.839	13.714	3.67
n	30	30	30	30
W	0.4009	0.969	0.405	0.956
Critical value	0.900	0.900	0.900	0.900
p-value	5.54E-10	0.526	6.14E-10	0.246
Hypothesis H0	False	True	False	True

The table presents the generations where the best solutions were obtained. In all the cases, $\alpha = 0.01$ represents the significance level value

ous set of experiments. The results of the experiments for 10, 15, and 20 objects are shown in Tables 9 to 13. For the case of 10 objects, we need to apply the method shown in 2(c) since the distribution probability of the GA is non-normal (see Table 9). The results of the bootstrapping method for this case are shown in Table 10, which shows that quantum algorithms are the best methods since they obtained the most accurate values since they have the higher mean values compared with the classical GA; in addition, all the quantum algorithms exhibit the best precision since they have the lowest standard deviation.

For the case of 15 objects, Table 11 shows that all the algorithms distributions are non-normal; hence, we applied the method explained in 2(b). The Wilcoxon signed ranks test

Table 4 Statistical test with 10 objects regarding convergence time

Run	GA-boots	HQGAAA	QIEAAA-local_boots	QIEAAA-global
1	1.4	23	40	119
2	1.3	28	38.5	110
3	1.3	27	44.7	122
4	1.3	28	41.7	122
5	1.5	27	44.9	120
6	1.6	27	41.5	120
7	1.6	25	41.9	123
8	1.5	27	43.8	121
9	1.2	22	42.1	118
10	1.5	26	38.9	115
11	1.2	23	42.3	121
12	1.4	29	38.9	115
13	1.4	23	46.6	119
14	1.2	30	43.4	114
15	1.4	25	43	123
16	1.4	20	40.2	120
17	1.1	22	43.2	118
18	1.1	27	41.9	120
19	1.2	27	43.4	119
20	1.5	21	39.3	114
21	1.6	22	41.7	110
22	1.6	27	38.9	115
23	1.4	32	45.1	115
24	1.5	25	42.3	117
25	1.1	24	37	114
26	1.3	27	43.6	112
27	1.5	28	41.7	116
28	1.3	24	43.4	112
29	1.4	30	40.8	114
30	1.2	26	42.1	117
Shapiro–Wilk test ($\alpha = 0.01$)				
Mean	1.357	25.733	41.893	117.166
Variance	0.64	241.866	143.038	404.166
Stand. Dev.	0.146	2.839	2.183	3.67
n	30	30	30	30
W	0.976	0.969	0.977	0.956
Critical value	0.900	0.900	0.900	0.900
p-value	0.731	0.526	0.753	0.246
Hypothesis H0	True	True	True	True
Welch's hypothesis t-test ($\alpha = 0.001$)				
d.o.f		29	29	29
t-value		−46.171	−99.749	−169.778
Critical value		3.659	3.659	3.659
Hypothesis H0		False	False	False

The table presents the generations where the best solutions were obtained. In the table, α represents the significance value and d.o.f means degree of freedom

Table 5 Shapiro–Wilk test with 15 objects regarding convergence time

Run	GA	HQGAAA	QIEAAA-local	QIEAAA-global
1	5	36	46	134
2	7	42	46	138
3	3	44	46	141
4	3	41	46	137
5	6	47	46	135
6	3	36	46	164
7	6	41	46	132
8	5	41	46	139
9	7	41	46	134
10	1	31	46	134
11	12	39	46	135
12	6	46	46	133
13	10	43	46	134
14	17	37	46	130
15	4	37	46	135
16	5	33	47	129
17	6	39	46	139
18	6	47	51	135
19	7	35	46	130
20	9	57	47	168
21	7	49	46	139
22	3	34	48	141
23	9	38	46	139
24	7	56	46	132
25	3	37	46	132
26	5	54	46	131
27	11	36	46	144
28	4	46	46	133
29	2	28	46	166
30	8	41	46	124
Mean	6.233	41.066	46.30	137.90
Variance	321.366	1,407.866	28.30	3,130.699
Stand. Dev.	3.272	6.85	0.971	10.215
n	30	30	30	30
W	0.916	0.957	0.35	0.732
Critical value	0.900	0.900	0.900	0.900
p-value	0.022	0.27	1.94E-10	4.74E-06
Hypothesis H0	True	True	False	False

The table presents the generations where the best solutions were obtained. In all the cases, $\alpha = 0.01$ represents the significance level value

results show a significant difference in favor of the HQGAAA and the QIEAAA-local. These two algorithms have the best accuracy and precision.

For the case of 20 objects, Table 12 shows that the only algorithm that produced results with normal distribution is the QIEAAA-global. Therefore, we need to apply the method explained in 2(c). The bootstrapping results are shown in Table 13, which shows a significant difference in favor of the quantum algorithms. Moreover, the mean and standard deviation statistical values indicate that the HQGAAA and the QIEAAA-local are the most accurate and precise algorithms. We also note that the QIEAAA-local has the better accuracy, whereas the HQGAAA has the best precision, although the difference is minimal in both cases.

Table 6 Statistical test with 15 objects regarding convergence time

Run	GA	HQGAAA	QIEAAA-local_boots	QIEAAA-global_boots
1	5	36	46.4	139.4
2	7	42	46.8	139.4
3	3	44	46.3	137.1
4	3	41	46.1	139.1
5	6	47	46.1	138.7
6	3	36	46.2	137.4
7	6	41	46.2	135.8
8	5	41	46.2	138.4
9	7	41	46.3	140.7
10	1	31	46.1	135.5
11	12	39	46.7	138.0
12	6	46	46.2	138.5
13	10	43	46.1	139.2
14	17	37	46.1	141.6
15	4	37	46.4	139.5
16	5	33	46.4	139.3
17	6	39	46.4	141.0
18	6	47	46.4	138.9
19	7	35	46.2	142.5
20	9	57	46.3	139.1
21	7	49	46.2	137.8
22	3	34	46.4	137.2
23	9	38	46.6	138.5
24	7	56	46.0	135.6
25	3	37	46.5	138.9
26	5	54	46.3	137.2
27	11	36	46.3	138.7
28	4	46	46.7	135.8
29	2	28	46.1	138.9
30	8	41	46.5	135.3
Shapiro–Wilk test ($\alpha = 0.01$)				
Mean	6.233	41.066	46.3	138.4
Variance	321.366	1,407.866	1.121	90.566
Stand. Dev.	3.272	6.85	0.193	1.737
n	30	30	30	30
W	0.916	0.957	0.933	0.953
Critical value	0.900	0.900	0.900	0.900
p-value	0.022	0.27	0.062	0.212
Hypothesis H0	True	True	True	True
Welch's hypothesis t-test ($\alpha = 0.001$)				
d.o.f		41	29	44
t-value		−25.129	−66.967	−195.418
Critical value		3.551	3.659	3.526
Hypothesis H0		False	False	False

The table presents the generations where the best solutions were obtained. In the table, α represents the significance value and d.o.f means degree of freedom

Table 7 Shapiro–Wilk test regarding convergence time with 20 objects

Run	GA	HQGAAA	QIEAAA-local	QIEAAA-global
1	13	38	48	138
2	9	44	150	162
3	3	157	174	155
4	107	52	51	158
5	1	150	147	168
6	4	54	155	155
7	6	40	60	153
8	17	156	46	142
9	4	44	52	164
10	9	149	166	160
11	24	45	158	141
12	4	144	151	150
13	11	46	152	148
14	17	42	46	161
15	4	145	153	151
16	138	153	59	148
17	4	50	62	158
18	4	55	46	163
19	20	45	157	147
20	5	46	46	150
21	2	146	159	155
22	6	147	53	152
23	34	160	50	153
24	6	50	46	155
25	3	155	46	165
26	8	49	156	160
27	1	40	46	159
28	4	42	63	152
29	12	51	157	144
30	61	45	46	141
Mean	18.033	84.666	96.7	153.6
Variance	28,096.966	77,350.666	83,372.299	1,749.2
Stand. Dev.	30.603	50.777	52.716	7.635
n	30	30	30	30
W	0.538	0.697	0.726	0.978
Critical value	0.900	0.900	0.900	0.900
p-value	1.41E-08	1.43E-06	3.908E-06	0.78
Hypothesis H0	False	False	False	True

The table presents the generations where the best solutions were obtained. In all the cases, $\alpha = 0.01$ represents the significance level value

5 Conclusions

This paper proposed the Hybrid Quantum Genetic Algorithm with an Adaptive Rotation Angle (HQGAAA) for solving the combinatorial problem known as the 0-1 knapsack problem, which is of computer science interest since it is classified as an NP-hard problem.

This work has presented at least three statistically sustained contributions to the state-of-the-art in computer science.

The first one corresponds to the HQGAAA, which is the first quantum algorithm developed to solve the 0-1 knapsack in a quantum computer. In this topic, secondary contributions are the reformulation of two quantum-inspired

Table 8 Statistical test with 20 objects regarding convergence time

Run	GA-boots	HQGAAA-boots	QIEAAA-local_boots	QIEAAA-global
1	11.6	88.1	76.4	138
2	15.4	73.7	77.1	162
3	19.2	79.8	110.1	155
4	10.9	76.4	98.6	158
5	18.6	94.2	112.2	168
6	30.6	93.5	87.3	155
7	27.6	80.0	90.5	153
8	12.3	80.9	92.1	142
9	17.4	87.2	92.0	164
10	23.6	103.8	95.9	160
11	12.3	85.6	78.7	141
12	9.7	77.6	82.4	150
13	19.3	91.1	100.5	148
14	19.8	80.7	82.1	161
15	17.2	88.2	95.9	151
16	28.2	81.8	102.5	148
17	17.0	97.5	75.3	158
18	31.0	99.6	102.4	163
19	13.1	83.9	92.5	147
20	14.8	81.6	99.4	150
21	19.8	82.2	93.1	155
22	19.1	91.6	109.4	152
23	15.9	91.6	100.9	153
24	18.3	92.9	93.7	155
25	29.0	80.7	90.0	165
26	15.8	78.3	96.7	160
27	15.8	86.7	83.2	159
28	11.8	89.9	96.2	152
29	34.8	82.9	105.3	144
30	19.6	84.1	97.8	141
Shapiro–Wilk test ($\alpha = 0.01$)				
Mean	18.980	86.200	93.7	153.6
Variance	1,262.33	1,591.24	2,923.75	1,749.2
Stand. Dev.	6.48	7.11	9.87	7.635
n	30	30	30	30
W	0.911	0.968	0.967	0.978
Critical value	0.900	0.900	0.900	0.900
p-value	0.016	0.494	0.478	0.78
Hypothesis H0	True	True	True	True
Welch's hypothesis t-test ($\alpha = 0.001$)				
d.o.f		57	50	56
t-value		−37.596	−34.052	−72.355
Critical value		3.46	3.496	3.46
Hypothesis H0		False	False	False

The table presents the generations where the best solutions were obtained. In the table, α represents the significance value and d.o.f means degree of freedom

Table 9 Shapiro–Wilk test with 10 objects regarding precision and accuracy

Run	GA	HQGAAA	QIEAAA-local	QIEAAA-global
1	82.35	82.35	82.35	82.35
2	80.75	82.35	82.35	82.35
3	82.35	82.35	82.35	82.35
4	82.35	82.35	82.35	82.35
5	82.35	82.35	82.35	82.35
6	82.35	82.35	82.35	82.35
7	80.75	82.35	82.35	82.35
8	82.35	82.35	82.35	82.35
9	80.75	82.35	82.35	82.35
10	82.35	82.35	82.35	82.35
11	82.35	82.35	82.35	82.35
12	82.35	82.35	82.35	82.35
13	82.35	82.35	82.35	82.35
14	80.75	82.35	82.35	82.35
15	82.35	82.35	82.35	82.35
16	82.35	82.35	82.35	82.35
17	82.35	82.35	82.35	82.35
18	82.35	82.35	82.35	82.35
19	82.35	82.35	82.35	82.35
20	80.75	82.35	82.35	82.35
21	82.35	82.35	82.35	82.35
22	80.75	82.35	82.35	82.35
23	80.75	82.35	82.35	82.35
24	82.35	82.35	82.35	82.35
25	80.75	82.35	82.35	82.35
26	82.35	82.35	82.35	82.35
27	82.35	82.35	82.35	82.35
28	82.35	82.35	82.35	82.35
29	82.35	82.35	82.35	82.35
30	80.75	82.35	82.35	82.35
Mean	81.87	82.35	82.35	82.35
Variance	16.05	−2.91E-10	−2.91E-10	−2.91E-10
Stand. Dev.	0.73	4.26E-14	4.26E-14	4.26E-14
n	30	30	30	30
W	0.577	1	1	1
Critical value	0.900	0.900	0.900	0.900
p-value	3.91E-08	1	1	1
Hypothesis H0	False	True	True	True

The table presents the best solutions in each run. In all the cases, the significance level value is $\alpha = 0.01$

algorithms, the QIEAAA-local and the QIEAAA-global, for executing in a quantum computer. To evaluate these proposals, we designed three sets of experiments containing statistical hypothesis tests to determine the solutions' overall performance, convergence time, and accuracy and precision. The designed tests were made in fair and valid conditions since the tested algorithms were selected according to impartial criteria, such as chromosome codification and evolution methodology. The GA was chosen to be the ground truth since it is known to be fast and can provide accurate and precise results for the tested examples. The quantum tested algorithms were the QIEAAA-local, and the QIEAAA-global. The statistical results show that the quantum algorithms generally have better performance,

Table 10 Statistical test regarding precision and accuracy with 10 objects

Run	GA-boots	HQGAAA	QIEAAA-local	QIEAAA-global
1	81.98	82.35	82.35	82.35
2	81.87	82.35	82.35	82.35
3	81.66	82.35	82.35	82.35
4	81.87	82.35	82.35	82.35
5	81.87	82.35	82.35	82.35
6	81.77	82.35	82.35	82.35
7	81.98	82.35	82.35	82.35
8	81.77	82.35	82.35	82.35
9	82.08	82.35	82.35	82.35
10	81.50	82.35	82.35	82.35
11	81.71	82.35	82.35	82.35
12	82.08	82.35	82.35	82.35
13	81.77	82.35	82.35	82.35
14	81.87	82.35	82.35	82.35
15	81.82	82.35	82.35	82.35
16	81.87	82.35	82.35	82.35
17	82.03	82.35	82.35	82.35
18	81.93	82.35	82.35	82.35
19	81.71	82.35	82.35	82.35
20	82.14	82.35	82.35	82.35
21	81.66	82.35	82.35	82.35
22	82.03	82.35	82.35	82.35
23	81.87	82.35	82.35	82.35
24	82.03	82.35	82.35	82.35
25	82.08	82.35	82.35	82.35
26	81.66	82.35	82.35	82.35
27	81.61	82.35	82.35	82.35
28	81.98	82.35	82.35	82.35
29	81.77	82.35	82.35	82.35
30	81.93	82.35	82.35	82.35
Shapiro–Wilk test ($\alpha = 0.01$)				
Mean	81.86	82.35	82.35	82.35
Variance	0.76	−2.91E-10	−2.91E-10	−2.91E-10
Stand. Dev.	0.15	4.26E-14	4.26E-14	4.26E-14
n	30	30	30	30
W	0.969	1	1	1
Critical value	0.900	0.900	0.900	0.900
p-value	0.512	1	1	1
Hypothesis H0	True	True	True	True
Welch's hypothesis t-test ($\alpha = 0.001$)				
d.o.f		29	29	29
t-value		−16.44	−16.44	−16.44
Critical value		3.659	3.659	3.659
Hypothesis H0		False	False	False

The table presents the best solutions obtained in each run. In all the cases, α represents the significance level value; d.o.f. means degree of freedom

Table 11 Statistical test regarding precision and accuracy with 15 objects

Run	GA	HQGAAA	QIEAAA-local	QIEAAA-global
1	114.60	113.18	114.60	113.18
2	114.60	114.60	114.60	113.18
3	113.18	114.60	114.60	113.18
4	114.60	114.60	114.60	113.18
5	114.60	114.60	114.60	113.18
6	114.60	114.60	114.60	114.60
7	113.18	114.60	114.60	113.18
8	113.18	114.60	114.60	113.18
9	113.18	114.60	114.60	113.18
10	113.18	114.60	114.60	113.18
11	113.18	114.60	113.18	113.18
12	113.18	114.60	114.60	113.18
13	113.18	114.60	114.60	113.18
14	114.60	114.60	113.18	113.18
15	114.60	114.60	114.60	113.18
16	113.18	114.60	114.60	113.18
17	114.60	113.18	114.60	113.18
18	114.60	114.60	114.60	113.18
19	114.60	114.60	114.60	113.18
20	114.60	114.60	114.60	114.60
21	113.18	114.60	114.60	113.18
22	113.18	114.60	114.60	114.60
23	114.60	114.60	114.60	113.18
24	114.60	114.60	114.60	113.18
25	113.18	114.60	114.60	113.18
26	114.60	114.60	114.60	113.18
27	114.60	114.60	114.60	113.18
28	114.60	114.60	114.60	113.18
29	113.18	114.60	114.60	114.60
30	114.60	114.60	114.60	113.18
Shapiro–Wilk test ($\alpha = 0.01$)				
Mean	113.984	114.501	114.501	113.372
Variance	14.651	3.712	3.712	6.894
Stand. Dev.	0.698	0.351	0.351	0.479
n	30	30	30	30
W	0.631	0.275	0.275	0.404
Critical value	0.900	0.900	0.900	0.900
p-value	1.823E-07	4.401E-11	4.401E-11	5.981E-10
Hypothesis H0	False	False	False	False
Wilcoxon signed ranks test ($\alpha = 0.001$)				
N		15	13	17
W-value		16	7	18
Critical value		6	2	11
Sum of pos. Ranks		16	7	135
Sum of neg. Ranks		104	84	18
p-value		0.004	0.002	0.0016
Hypothesis H0		False	False	False

The table presents the best solutions obtained in each run. In all the cases, α represents the significance level value

Table 12 Shapiro–Wilk test with 20 objects regarding precision and accuracy

Run	GA	HQGAAA	QIEAAA-local	QIEAAA-global
1	146.59	154.11	155.47	154.42
2	147.51	155.47	154.42	153.46
3	152.40	153.46	155.47	152.40
4	154.11	155.47	154.11	153.19
5	151.48	154.42	155.47	155.47
6	155.47	155.47	155.47	153.46
7	152.09	155.47	155.47	155.47
8	152.09	153.46	153.19	153.46
9	148.76	153.46	153.46	153.46
10	152.09	155.47	155.47	153.46
11	154.11	154.42	155.47	152.09
12	149.41	155.47	155.47	153.19
13	154.11	155.47	155.47	152.40
14	147.51	153.19	155.47	155.47
15	155.47	154.42	155.47	154.11
16	155.47	155.47	155.47	153.46
17	147.58	155.47	155.47	154.42
18	148.99	155.47	155.47	154.11
19	147.51	154.42	152.38	155.47
20	148.34	155.47	155.47	155.47
21	154.11	154.42	153.46	154.42
22	148.99	154.42	154.42	152.09
23	147.58	155.47	155.47	153.19
24	147.96	154.42	155.47	153.46
25	147.51	155.47	155.47	154.11
26	147.51	154.42	155.47	153.46
27	153.46	154.11	155.47	152.54
28	147.51	155.47	155.47	155.47
29	152.09	154.42	155.47	154.42
30	147.51	154.11	154.42	154.42
Mean	150.51	154.74	155.009	153.86
Variance	264.53	16.98	21.64	32.25
Stand. Dev.	2.96	0.75	0.84	1.03
n	30	30	30	30
W	0.861	0.806	0.608	0.918
Critical value	0.900	0.900	0.900	0.900
p-value	1.10E-03	8.79E-05	9.24E-08	0.024
Hypothesis H0	False	False	False	True

The table presents the best solutions in each run. In all the cases, the significance level value is $\alpha = 0.01$

convergence time, accuracy, and precision than the GA. Regarding the comparison among the quantum algorithms, the statistics demonstrate that the HQGAAA achieved the best overall performance for 10, 15, and 20 object. Concerning the comparison of convergence time, the HQGAAA for the case of 10 objects was $\approx$ 39% better than the QIEAAA-local, and $\approx$ 78% better than the QIEAAA-global; for 15 objects, the HQGAAA was $\approx$ 11% better than the QIEAAA-local, and $\approx$ 70% than the QIEAAA-global; for 20 objects, the HQGAAA was $\approx$ 8% better than the QIEAAA-local, and $\approx$ 44% better than the QIEAAA-global. With respect to the accuracy and precision, for 10 objects, the three quantum algorithms behave similarly; for the case of 15 objects, the three quantum algorithms presented the same accuracy, but

Table 13 Statistical test regarding precision and accuracy with 20 objects

Run	GA-boots	HQGAAA-boots	QIEAAA-local_boots	QIEAAA-global
1	150.94	154.80	155.15	154.42
2	149.62	154.71	154.95	153.46
3	151.26	154.85	155.19	152.40
4	150.94	154.48	155.13	153.19
5	149.89	154.71	154.84	155.47
6	150.94	154.83	155.13	153.46
7	151.59	154.85	155.32	155.47
8	150.61	154.56	155.11	153.46
9	150.87	154.99	155.14	153.46
10	149.47	154.98	155.04	153.46
11	150.60	154.83	155.09	152.09
12	149.84	154.75	154.94	153.19
13	150.80	155.06	154.79	152.40
14	150.24	154.92	155.00	155.47
15	150.31	154.66	154.75	154.11
16	150.36	154.81	154.99	153.46
17	150.51	154.94	154.55	154.42
18	150.33	154.82	155.06	154.11
19	150.10	154.95	155.11	155.47
20	150.04	154.61	154.99	155.47
21	150.81	155.00	155.11	154.42
22	150.38	154.80	155.01	152.09
23	150.65	154.71	155.25	153.19
24	150.16	154.90	155.02	153.46
25	150.46	154.48	155.14	154.11
26	151.13	154.86	155.18	153.46
27	150.60	154.74	155.08	152.54
28	150.52	154.86	154.83	155.47
29	150.77	154.77	155.12	154.42
30	151.00	154.81	155.01	154.42
Shapiro–Wilk test ($\alpha = 0.01$)				
Mean	150.52	154.80	155.030	153.86
Variance	6.72	0.6	0.74	32.25
Stand. Dev.	0.47	0.14	0.15	1.03
n	30	30	30	30
W	0.990	0.963	0.925	0.918
Critical value	0.900	0.900	0.900	0.900
p-value	0.994	0.385	0.037	0.024
Hypothesis H0	True	True	True	True
Welch's hypothesis t-test ($\alpha = 0.001$)				
d.o.f		34	35	40
t-value		−46.57	−48.66	−15.79
Critical value		3.601	3.591	3.551
Hypothesis H0		False	False	False

The table presents the best solutions obtained in each run. In all the cases, α represents the significance level value

the HQGAAA and the QIEAAA-local obtained the best precision. For the case of 20 objects, the differences in accuracy of the three quantum algorithms are negligible; however, the precision of the HQGAAA was $\approx$ 7% better that obtained by the QIEAAA-local, and $\approx$ 86% better than the QIEAAA-global.

The second contribution was the successful implementation of the adaptive rotation algorithm strategy to the quantum algorithms tested in this work. We consider this implementation as a contribution since it was only tested in quantum-inspired algorithms, and now we demonstrate that it can be used successfully in implementations for quantum computers.

The third contribution was the original idea of using the Deutsch-Jozsa circuit as a quantum operator of the HQGAAA to perform the task of recombination and mutation to take advantage of quantum parallelism. Including this operator was the key to obtaining the reported results that outperformed the other algorithms.

We had some technical limitations for testing the HQGAAA with larger problems since, at present, quantum technology is limited because of the NISQ era; hence, qubits are noisy and perform implementations with more quantum stages increments the depth of the quantum circuits, which introduces problems. A second limitation is the width of quantum registers in simulators and in quantum computers. Increasing the width of the quantum register in our problem for testing it in simulations represents a considerable increment in computational time. The first versions of quantum algorithms were tested on the IBM Quantum computer, and they work similar to the reported results in this work; unfortunately, to complete the experiments reported here, the IBM Quantum computer access was restricted.

All the presented quantum programs are scalable, and they are limited only by the above-mentioned problems.

As a general conclusion, the HQGAAA outperformed all the other tested algorithms significantly.

6 Future work

The future of hybrid quantum algorithms is promising. However, the NISQ era is a stage of quantum computing that is just beginning; this is because the noise in the qubits is a characteristic that will take some time to decrease but never disappear. As the technology advances, more stages of the hybrid quantum metaheuristic, those that are currently executed into the classical computer, could be tested on the quantum computer to improve efficiency.

Funding This work has been founded by Instituto Politécnico Nacional under Grant Numbers SIP20210320 and SIP20220079. We also thanks to the Comisión de Fomento y Apoyo Académico del IPN (COFAA), and the Mexican National Council of Science and Technology (CONACYT) for supporting our research activities.

Data availability We can provided the application software and detailed instructions to obtain the reported data.

Code Availability Software application can be shared under requirement.

Declarations

Conflict of interest The authors declare that they do not have any conflict of interest.

References

Adeyemo H, Ahmed M (2017) Solving 0/1 knapsack problem using metaheuristic techniques. In: 2017 9th IEEE-GCC Conference and Exhibition (GCCCE), pp. 1–6. https://doi.org/10.1109/IEEEGCC.2017.8448239

Arute F, Arya K, Babbush R et al (2019) Quantum supremacy using a programmable superconducting processor. Nature 574:505–510

Ballance CJ, Harty TP, Linke NM, Sepiol MA, Lucas DM (2016) High-fidelity quantum logic gates using trapped-ion hyperfine qubits. Physical Review Letters **117**(6). https://doi.org/10.1103/physrevlett.117.060504

Barends R, Kelly J, Megrant A, Veitia A, Sank D, Jeffrey E, White TC, Mutus J, Fowler AG, Campbell B et al (2014) Superconducting quantum circuits at the surface code threshold for fault tolerance. Nature 508(7497):500–503. https://doi.org/10.1038/nature13171

Bhattacharjee KK, Sarmah SP (2017) Modified swarm intelligence based techniques for the knapsack problem. Appl Intell 46(1):158–179. https://doi.org/10.1007/s10489-016-0822-y

Calabrò G, Torrisi V, Inturri G, Ignaccolo M (2020) Improving inbound logistic planning for large-scale real-world routing problems: a novel ant-colony simulation-based optimization. Eur Transp Res Rev 12(1):21. https://doi.org/10.1186/s12544-020-00409-7

Cao Z, Uhlmann J, Liu L (2018) Analysis of deutsch-jozsa quantum algorithm. IACR Cryptol ePrint Arch 2018:249

Delahaye D, Chaimatanan S, Mongeau M (2019) In: Gendreau, M., Potvin, J.-Y. (eds.) Simulated Annealing: From Basics to Applications, pp. 1–35. Springer, Cham. https://doi.org/10.1007/978-3-319-91086-4_1

Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18. https://doi.org/10.1016/j.swevo.2011.02.002

Deutsch D (1985) Quantum theory, the church-turing principle and the universal quantum computer. R Soc 400:97–117

Feynman RP (1959) There's plenty of room at the bottom. IEEE 1:60–66

Forno E, Acquaviva A, Kobayashi Y, Macii E, Urgese G (2018) A parallel hardware architecture for quantum annealing algorithm acceleration. In: 2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), pp. 31–36. https://doi.org/10.1109/VLSI-SoC.2018.8644777

Gao Y, Zhang F, Zhao Y, Li C (2018) Quantum-inspired wolf pack algorithm to solve the 0–1 knapsack problem. Math Probl Eng 2018:1–10. https://doi.org/10.1155/2018/5327056

García J, Crawford B, Soto R, Castro C, Paredes F (2018) A k-means binarization framework applied to multidimensional knapsack problem. Appl Intell 48(2):357–380. https://doi.org/10.1007/s10489-017-0972-6

Han K-H, Jong-Hwan Kim (2002) Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Trans Evol Comput 6(6):580–593. https://doi.org/10.1109/TEVC.2002.804320

Hao L (2019) An angle-expressed quantum evolutionary algorithm for quadratic knapsack problem. IOP Conf Ser: Mater Sci Eng 631:052054. https://doi.org/10.1088/1757-899x/631/5/052054

Hirzel T (Agosto 2020) Building the Quantum Stack for the NISQ Era. https://www.hpcwire.com/2020/08/24/building-the-quantum-stack-for-the-nisq-era/

Huang Y, Wang P, Li J, Chen X, Li T (2019) A binary multi-scale quantum harmonic oscillator algorithm for 0–1 knapsack problem with genetic operator. IEEE Access 7:137251–137265. https://doi.org/10.1109/ACCESS.2019.2942340

IBM (2020) The Qiskit Elements. https://quantum-computing.ibm.com/docs/qiskit/the_elements

Jindal A, Bansal S (2019) Effective methods for constraint handling in quantum inspired evolutionary algorithm for multi-dimensional 0-1 knapsack problem. In: 2019 4th International Conference on Information Systems and Computer Networks (ISCON), pp. 534–537. https://doi.org/10.1109/ISCON47742.2019.9036166

Johansson N, Larsson J-A (2017) Efficient classical simulation of the deutsch–jozsa and simon's algorithms. Quantum Information Processing **16**(9). https://doi.org/10.1007/s11128-017-1679-7

Jourdan L, Basseur M, Talbi E-G (2009) Hybridizing exact methods and metaheuristics: A taxonomy. Eur J Oper Res 199:620–629. https://doi.org/10.1016/j.ejor.2007.07.035

King J, Yarkoni S, Raymond J, Ozfidan I, King AD, Nevisi MM, Hilton JP, McGeoch CC (2017) Quantum Annealing amid Local Ruggedness and Global Frustration

Kuk-Hyun Han, Jong-Hwan Kim (2000) Genetic quantum algorithm and its application to combinatorial optimization problem. In: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), vol. 2, pp. 1354–13602

Kuk-Hyun Han, Kui-Hong Park, Ci-Ho Lee, Jong-Hwan Kim (2001) Parallel quantum-inspired genetic algorithm for combinatorial optimization problem. In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), vol. 2, pp. 1422–14292

Lai X, Hao J-K, Yue D (2019) Two-stage solution-based tabu search for the multidemand multidimensional knapsack problem. Eur J Op Res 274(1):35–48. https://doi.org/10.1016/j.ejor.2018.10.001

Lai X, Hao J-K, Fu Z-H, Yue D (2020) Diversity-preserving quantum particle swarm optimization for the multidimensional knapsack problem. Expert Syst Appl 149:113310. https://doi.org/10.1016/j.eswa.2020.113310

Lai X, Hao J, Yue D, Gao H (2018) A diversification-based quantum particle swarm optimization algorithm for the multidimensional knapsack problem. In: 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), pp. 315–319. https://doi.org/10.1109/CCIS.2018.8691247

López-Sandoval D, Cobos C (2020) Adiabatic quantum computing applied to the solution of the binary knapsack problem. RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao 38:214–227

Lourenço HR, Martin OC, Stützle T (2019) In: Gendreau, M., Potvin, J.-Y. (eds.) Iterated Local Search: Framework and Applications, pp. 129–168. Springer, Cham. https://doi.org/10.1007/978-3-319-91086-4_5

Montiel Ross OH (2020) A review of quantum-inspired metaheuristics: Going from classical computers to real quantum computers. IEEE Access 8:814–838. https://doi.org/10.1109/ACCESS.2019.2962155

Narayanan A (1999) Quantum computing for beginners. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406) 3, 2231–22383

Narayanan A, Moore M (1996) Quantum-inspired genetic algorithms, pp. 61–66. https://doi.org/10.1109/ICEC.1996.542334

Ozsoydan FB, Baykasoglu A (2019) A swarm intelligence-based algorithm for the set-union knapsack problem. Future Gener Comput Syst 93:560–569. https://doi.org/10.1016/j.future.2018.08.002

Paredes López M, Meneses Viveros A, Morales-Luna G (2018) Algoritmo cuántico de Deutsch y Jozsa en GAMA. Revista mexicana de física 64:181–189

Pednault E, Gunnels JA, Nannicini G, Horesh L, Wisnieff R (2019) Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits

Pisinger D (2005) Where are the hard knapsack problems? Comput Op Res 32(9):2271–2284. https://doi.org/10.1016/j.cor.2004.03.002

Preskill J (2018) Quantum computing in the nisq era and beyond. Quantum **2**, 79. https://doi.org/10.22331/q-2018-08-06-79

Preskill J (2018) Quantum computing in the nisq era and beyond. Quantum 2, 79. https://doi.org/10.22331/q-2018-08-06-79

Rezoug A, Bader-El-Den M, Boughaci D (2018) Guided genetic algorithm for the multidimensional knapsack problem. Memet Comput 10(1):29–42. https://doi.org/10.1007/s12293-017-0232-7

Rubio Y, Olvera C, Montiel, O.: Quantum-inspired evolutionary algorithms on ibm quantum experience. Engineering Letters 29(4), 1573–1584, (2021) Publisher Copyright: © 2021. International Association of Engineers, All rights reserved

Sapra D, Sharma R, Agarwal AP (2017) Comparative study of metaheuristic algorithms using knapsack problem. In: 2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence, pp. 134–137. https://doi.org/10.1109/CONFLUENCE.2017.7943137

Tannu SS, Qureshi MK (2019) Not all qubits are created equal: A case for variability-aware policies for nisq-era quantum computers. In: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. ASPLOS '19, pp. 987–999. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3297858.3304007

Valerii T (2018) An adaptive quantum evolution algorithm for 0-1 knapsack problem. System research and information technologies 77–88. https://doi.org/10.20535/SRIT.2308-8893.2018.2.08

Valerii T, Tkachuk O (2018) Higher-order quantum genetic algorithm for 0–1 knapsack problem. System research and information technologies 52–67. https://doi.org/10.20535/SRIT.2308-8893.2018.3.05https://doi.org/10.20535/SRIT.2308-8893.2018.3.05

Vásquez C, Lemus-Romani J, Crawford B, Soto R, Astorga G, Palma W, Misra S, Paredes F (2020) Solving the 0/1 knapsack problem using a galactic swarm optimization with data-driven binarization approaches. In: Gervasi O, Murgante B, Misra S, Garau C, Blečić I, Taniar D, Apduhan BO, Rocha AMAC, Tarantino E, Torre CM, Karaca Y (eds) Computational Science and Its Applications - ICCSA 2020. Springer, Cham, pp 511–526

Wang Y, Wang W (2021) Quantum-inspired differential evolution with grey wolf optimizer for 0–1 knapsack problem. Mathematics. https://doi.org/10.3390/math9111233

Wang H, Liu J, Zhi J, Fu C (2013) The improvement of quantum genetic algorithm and its application on function optimization. Mathematical Problems in Engineering 2013. https://doi.org/10.1155/2013/730749

Wei Z, Hao J-K (2019) Iterated two-phase local search for the set-union knapsack problem. Futur Gener Comput Syst 101:1005–1017. https://doi.org/10.1016/j.future.2019.07.062

Wei Z, Hao J-K (2021) Kernel based tabu search for the set-union knapsack problem. Expert Syst Appl 165:113802. https://doi.org/10.1016/j.eswa.2020.113802

Williams CP (2011) Explorations in Quantum Computing,

Wu Y, Bao W-S, Cao S, Chen F, Chen M-C, Chen X, Chung T-H, Deng H, Du Y, Fan D, Gong M, Guo C, Guo C, Guo S, Han L, Hong L, Huang H-L, Huo Y-H, Li L, Li N, Li S, Li Y, Liang F, Lin C, Lin J, Qian H, Qiao D, Rong H, Su H, Sun L, Wang L, Wang S, Wu D, Xu Y, Yan K, Yang W, Yang Y, Ye Y, Yin J, Ying C, Yu J, Zha C, Zhang C, Zhang H, Zhang K, Zhang Y, Zhao H, Zhao Y, Zhou L, Zhu Q, Lu C-Y, Peng C-Z, Zhu X, Pan J-W (2021) Strong quantum computational advantage using a superconducting quantum processor. Phys. Rev. Lett. 127:180501. https://doi.org/10.1103/PhysRevLett.127.180501

Xiang S, He Y, Chang L, Wu K, Zhang C (2017) An improved quantum-inspired evolutionary algorithm for knapsack problems. In: Sun X, Chao H-C, You X, Bertino E (eds) Cloud computing and security. Springer, Cham, pp 694–708

Yanofsky N, Manucci M (2008) Quantum Computing for Computer Scientists,

Zhan S, Wang L, Zhang Z, Zhong Y (2020) Noising methods with hybrid greedy repair operator for 0–1 knapsack problem. Memet Comput 12(1):37–50. https://doi.org/10.1007/s12293-019-00288-z

Zhang G (2011) Quantum-inspired evolutionary algorithms: a survey and empirical study. J Heuristics 17:303–351. https://doi.org/10.1007/s10732-010-9136-0

Zhong H-S, Deng Y-H, Qin J, Wang H, Chen M-C, Peng L-C, Luo Y-H, Wu D, Gong S-Q, Su H, Hu Y, Hu P, Yang X-Y, Zhang W-J, Li H, Li Y, Jiang X, Gan L, Yang G, You L, Wang Z, Li L, Liu N-L, Renema JJ, Lu C-Y, Pan J-W (2021) Phase-programmable gaussian boson sampling using stimulated squeezed light. Phys Rev Lett 127:180502. https://doi.org/10.1103/PhysRevLett.127.180502