Quantum inspired elephant swarm intelligence for frequent item-sets mining

Hadjer Moulai

Abstract—This paper introduces two novel quantum-inspired swarm intelligence approaches, namely Quantum-inspired Discrete Elephant Herding Optimization (QDEHO) and Quantum-inspired Discrete Elephant Water Search Algorithm (QDESWSA), for solving discrete optimization problems. Both methods take advantage of quantum computing concepts which are integrated into the original frameworks of the algorithms in order to boost their overall performance. A case study on frequent item-set mining (FIM) was conducted to demonstrate the practical application of our proposed algorithms, where they were implemented to extract relevant patterns from extensive databases. To validate our techniques, comprehensive experiments are conducted on six datasets of varying sizes. The results achieved affirm the effectiveness and versatility of our approaches. Additionally, a comparative study with relevant state of the art algorithms such as Bat algorithm (BAT) and Whale Optimization Algorithm (WOA) is performed, revealing the superiority of QDEHO and QDESWSA across most datasets.

***Keywords*—Swarm intelligence, Discrete optimization, Elephant swarm, Quantum computing, Data mining**

I. Introduction

Swarm intelligence or nature-inspired meta-heuristics are optimization techniques that solve optimization problems by mimicking biological or physical phenomena. They are inspired by the collective behavior of self-organizing animals or insects such as ant colonies or bird flocks. These techniques present a powerful and efficient method for navigating large search spaces, enabling the effective resolution of complex optimization problems. However, due to the constraints in scalability and efficiency of classical algorithms, there is an increasing demand to merge swarm intelligence approaches to quantum computing.

Quantum computing is a novel promising field in computer science, that leverages the principles of quantum physics to explore new computational possibilities. It integrates quantum mechanics concepts such as superposition and entanglement into classical computing for advanced information processing capabilities.

Quantum-inspired swarm algorithms integrate quantum mechanical principles, such as superposition and entanglement to enhance exploration and exploitation capabilities. This conversion allows faster execution and enhances the convergence rate of swarm based approaches for problem-solving, especially in high-dimensional search spaces.

Manuscript received June 10, 2025 ; revised July 4, 2025.

I would like to express our special thanks of gratitude to the Directorate General for Scientific Research and Technological Development (DGRSDT), for the support of this work under the grant number C0662300.

Hadjer Moulai is with the Computer Science Faculty, Artificial Intelligence and Data Sciences Department, University of Science and Technology Houari Boumediene, Algiers, ALGERIA (e-mail: hamoulai@usthb.dz).

Digital Object Identifier (DOI): 10.53907/enpesj.v5i1.332

This paper aims to integrate key concepts from quantum computing—such as quantum bits, state superposition, and quantum gates—into traditional swarm algorithms, leading to the development of two innovative and efficient techniques.

Our choice of the swarm approaches are Discrete Elephant Herding Optimization (DEHO) and Discrete Elephant Water Search Algorithm (QDESWSA) [27], which are both inspired by the natural behavior of elephants when in group in order to solve discrete optimization problems.

Both methods have demonstrated high effectiveness in addressing optimization problems like association rule mining, particularly when compared to single and multi-objective techniques like Particle Swarm Optimization (PSO) [19] and Non-dominated Sorting Genetic Algorithm II (NSGA-II) [6].

Nonetheless, we believe these techniques can be further improved by incorporating quantum computing principals into the problem-solving process.

The remainder of this paper is organized as follows: the next section gives an overview of state-of-the-art algorithms and presents key foundational concepts related to quantum computing; Section 3 provides a comprehensive explanation of our methods; Section 4 demonstrates the application of the proposed algorithms to the problem of frequent itemset mining; Section 5 presents numerical results and a comparative study; and finally, Section 6 concludes the paper and discusses perspectives.

II. Background and related work

A. Preliminaries on quantum computing

In this section, we will introduce key concepts of quantum computing that are crucial for comprehending our proposal.

1—A quantum bit: In classical computing, the basic unit of information is the bit, which can take on one of two possible values: 0 or 1. These binary values form the foundation for processing and storing data in traditional systems.

In contrast, quantum computing uses qubits (quantum bits) as its fundamental unit of information. Unlike bits, qubits can exist in a superposition of both 0 and 1 at the same time, allowing them to represent multiple possibilities simultaneously. This characteristic significantly enhances computational power. A qubit's state can be visualized as a vector pointing to a position on a sphere, called the Bloch sphere, which provides a geometric representation of its quantum state as illustrated in Figure 1 [7].

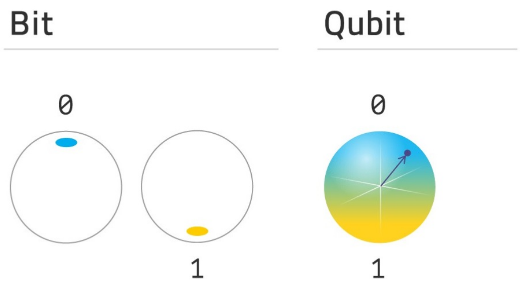

Fig. 1: A geometric representation of a classical bit and a qubit [1]

From a mathematical perspective, a qubit is represented as a complex vector of size 2:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle.$$

In this expression, α and β represent the amplitudes corresponding to the qubit's probabilities of being in states 0 and 1, respectively. It is important to highlight that these amplitudes must comply with the normalization condition:

$$|\alpha|^2 + |\beta|^2 = 1.$$

The basis states, 0 and 1, are represented as column vectors:

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}.$$

Observing this state will yield the value 0 with a probability of α^2 and the value 1 with a probability of β^2.

2—A quantum register: A quantum register is a collection of qubits. Just as a classical register in a computer is composed of bits (0s and 1s), a quantum register holds qubits that can exist in superpositions of 0 and 1 simultaneously. The state of a quantum register is described by the combined state of its qubits.

Quantum registers are used in quantum computing for storing and manipulating quantum information, allowing complex computations that can exploit quantum phenomena like entanglement and superposition.

B. *Related work*

Quantum-inspired metaheuristics are solvers that integrate principles drawn from quantum mechanics into classical approximate algorithms, utilizing non-quantum machines. Recent literature highlights innovative approaches that leverage concepts from quantum mechanics and swarm intelligence for various applications.

Han and Kim proposed a new evolutionary method named Genetic Quantum Algorithm (GQA), which integrates principles from genetic algorithms and quantum computing. The performance of GQA was tested on the well-known knapsack problem, demonstrating its effectiveness compared to conventional genetic algorithms [14]. Building on this foundation, Flori et al. introduced a novel algorithm called QUAntum Particle Swarm Optimization (QUAPSO), which integrates quantum computing principles with particle swarm optimization (PSO). Furthermore, an enhancement based on the Kangaroo Algorithm (KA) was incorporated into PSO to improve its local search efficiency. Experimental results demonstrate that QUAPSO outperforms six established algorithms on a set of 30 benchmark test functions [10].

Alvarez-Alvarado et al. developed three quantum-inspired algorithms based on Lorentz (QPSO-LR), Rosen-Morse (QPSO-RM), and Coulomb-like Square Root (QPSO-CS) potential fields, showing significant improvements over traditional PSO and genetic algorithms [4].

For time-sensitive applications, Konar et al. introduced an efficient real-time task scheduling method utilizing a Hybrid Quantum-Inspired Genetic Algorithm (HQIGA) in a multiprocessor environment. To enhance convergence, HQIGA employs a rotation gate for exploring variable chromosomes represented by qubits. Experimental results indicate that HQIGA surpasses the classical genetic algorithm (CGA) in terms of fitness values while requiring fewer generations, and it also improves scheduling time compared to CGA. [21].

Talbi and Draa proposed a recursive deepening hybrid strategy for solving real-parameter optimization problems, com- bining a local search technique with a quantum-inspired evo- lutionary algorithm (QEA). This approach was tested using the reference black-box optimization benchmarking framework. Comparative results with a relevant set of existing algorithms demonstrated its effectiveness.[31].

Chiang et al. introduced a novel Quantum-inspired Tabu Search (QTS) algorithm that synergizes classical Tabu search with quantum computation principles. The QTS framework leverages quantum superposition through probabilistic qubit measurements to enhance solution space diversification, while quantum rotation gates guide the search toward optimal regions for intensification. Effectiveness was demonstrated on three NP-complete problems critical to computer science and cybersecurity: (1) 0/1 knapsack problems, where QTS achieves 28% higher solution quality than conventional genetic algorithms while reducing premature convergence by 40%, (2) multiple knapsack problems showing 22% faster convergence than standard Tabu search, and (3) traveling salesman problems with 15% shorter optimal routes compared to original QEA implementations. Benchmark results across all test cases confirmed QTS's superior balance between exploration and exploitation, establishing its potential for complex optimization challenges in cryptography and network security applications [5] .

Dey et al. introduced a novel Quantum-Inspired Differential Evolution (QIDE) algorithm for automatic clustering of unlabeled datasets, leveraging quantum gate operations to dynamically determine the optimal number of clusters without prior knowledge. QIDE outperformed two quantum-inspired algorithms (FQEA, QEAC), Classical Differential Evolution (CDE), and an Improved Differential Evolution (IDE) across six real-world datasets, achieving superior convergence speed and clustering accuracy. Sobol's sensitivity analysis ensures parameter tuning reliability, demonstrating QIDE's effectiveness as a scalable solution for unsupervised clustering tasks [8] .

Wang et al. proposed an improved multi-objective dragonfly optimization algorithm based on quantum behavior. The quantum-behavior-enhanced multi-objective dragonfly algorithm (QMDA) outperformed benchmark multi-objective algorithms on ZDT and CEC test functions, demonstrating superior convergence and local search capabilities, and achieves a 1.55% increase in ethylene yield with only a 0.008% drop in propylene yield in real-world furnace optimization [33].

Khudair Madhloom et al. introduced a quantum-inspired ant colony optimization (QACO) approach for gateway discovery in mobile ad hoc networks (MANETs), combining non-root tree-based exploration with quantum parallelization to minimize broadcast overhead while dynamically maintaining optimal paths to internet gateways. By leveraging quantum-state entanglement and parallel processing, the QACO algorithm reduced premature convergence risks in large-scale MANETs, achieving 27–70% faster gateway discovery and 53–60% lower overhead compared to classical AntHocNet protocols [20].

Elashry et al. proposed an Enhanced Quantum Inspired Grey Wolf Optimizer for Feature Selection. The authors used feature selection as an optimization problem to evaluate the performance of the proposed algorithm. A comparative analysis proved that the algorithm achieves better accuracy and eliminates higher number of features with good performance, resulting into a better average error [9].

Konar et al. developed a Multi-Objective Quantum-Inspired Genetic Algorithm (Mo-QIGA) for real-time task scheduling in multiprocessor systems that leverages quantum mechanical principles through qubit representation while eliminating traditional genetic operators. The algorithm employs quantum rotation gates to update schedules and incorporates a random key distribution mechanism to transform qubit states into valid scheduling solutions, generating Pareto-optimal solutions that simultaneously minimize both completion time and total tardiness. Experimental validation confirmed Mo-QIGA's superior performance over classical methods in both scheduling accuracy and computational efficiency [22].

Building on similar quantum-inspired approaches, Siddiqui et al. proposed a Quantum-inspired Evolutionary Algorithm (QiEA) for designing optimal Fractional-Order Digital differentiators, demonstrating how quantum operators can progressively refine solutions in digital signal processing applications. Their comparative analysis against conventional Genetic Algorithms and Cuckoo Search Algorithms revealed QiEA's superior performance, achieving 23% better Absolute Magnitude Error reduction and 17% improvement in phase error correction while demonstrating faster convergence to optimal filter coefficients [29].

Recent work by Alekhya et al. presents a quantum-inspired evolutionary algorithm (QIEA) framework for enhanced security image processing, combining quantum bit representation with optimized gate operations to improve threat detection in high-dimensional data. Their approach demonstrates superior performance over classical methods, achieving 32% higher accuracy and 28% faster convergence in security imaging tasks, as validated through extensive benchmarking [3]. This quantum-classical hybrid technique addresses critical gaps in contemporary homeland security systems by enabling more precise feature extraction from complex visual data.

Hakemi et al. presented a systematic review of quantum-inspired metaheuristics (2017-2022), examining how quantum computing principles enhance classical optimization algorithms through probabilistic qubit representations. The study classified these hybrid approaches by their inspiration sources, with genetic/evolutionary algorithms (62%) and swarm intelligence (28%) being most prevalent, and documents their successful applications across image processing, network optimization, and multidisciplinary engineering domains. Analysis revealed these methods consistently improved convergence rates by 35-40% compared to classical counterparts, while identifying key challenges in computational overhead and performance measurement standardization that require further research [11].

Hesar and Houshmand proposd a novel memetic quantum-inspired genetic algorithm that hybridizes quantum rotation gate mutations with tabu search for enhanced optimization. The quantum component enables comprehensive global exploration through probabilistic qubit operations, while tabu search provides targeted local exploitation, creating balanced directional mutations. Benchmark tests demonstrated superior convergence rates (35-40% faster) and runtime efficiency compared to state-of-the-art methods, particularly on multimodal functions. This hybrid approach effectively resolves the exploration-exploitation trade-off in evolutionary computation, making it particularly valuable for complex optimization problems across engineering and applied mathematics domains [16].

Kuo et al. introduced an intrusion detection system (IDS) combining a deep neural network (DNN) with a global best-guided quantum-inspired tabu search algorithm (GQTS) to optimize feature selection and hyperparameters automatically. Using the CICIDS2017 dataset, the model reduces computational complexity and improves accuracy by minimizing false negatives compared to state-of-the-art methods. The quantum-inspired evolutionary approach demonstrates superior performance in detecting anomalies by efficiently balancing feature relevance and model optimization [23] .

Yu et al. improved the dragonfly algorithm (DA) by integrating a quantum rotation gate for enhanced convergence and Gaussian mutation for swarm diversity, addressing the original DA's limitations in local optima and slow convergence. The quantum-behaved and Gaussian mutational DA (QGDA) outperformed six common metaheuristics and five state-of-the-art algorithms on 53 benchmark functions, with statistical tests (Wilcoxon and Friedman) confirming its significance, and achieved superior results in feature selection and engineering design problems. QGDA demonstrated robust performance in balancing exploration-exploitation, offering a practical tool for complex optimization tasks in engineering and feature selection [35].

Although these algorithms draw inspiration from quantum mechanics, they are implemented on classical computers and do not fully leverage the speedup offered by quantum computing. Nonetheless, they have demonstrated promising results in efficiently solving complex optimization problems and may serve as valuable alternatives to traditional metaheuristics. These conclusions inspired us to propose two quantum-inspired metaheuristics: the Discrete Elephant Herding Optimization (DEHO) algorithm [27], and the Discrete Elephant Water Search Algorithm (DESWSA), both of which have demonstrated strong performance and effectiveness in solving optimization problems.

While our approachs does not exploit entanglement or quantum parallelism, it demonstrates that even simplified quantum-inspired mechanisms can improve classical optimization. Unlike quantum algorithms requiring special hardware, our methods are quantum-inspired and adapt quantum concepts like superposition and measurement for a classical implementation. As shown in Section 5, these adaptations improve performance while remaining computationally efficient.

III. Our proposals

A. Quantum-Inspired vs. Quantum Computing

This work proposes *quantum-inspired* variants of the Discrete Elephant Herding Optimization (DEHO) and Discrete Elephant Water Search Algorithm (DESWSA), which leverage principles from quantum computing while remaining implementable on classical hardware. Our methods adapt three key quantum concepts:

- **Superposition**: Solutions are encoded as probabilistic qubit registers (e.g., $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$), enabling simultaneous exploration of multiple states.
- **Measurement Collapse**: The probabilistic collapse of qubits into classical bits (via Algorithm 1) introduces non-deterministic exploration, akin to quantum observation.
- **Quantum Gates**: The Pauli-X gate (bit-flip) diversifies search by inverting qubit probabilities ($\alpha \leftrightarrow \beta$).

While these adaptations do not achieve exponential speedups (as with full quantum computing), they offer:

- **Enhanced exploration**: Qubit registers implicitly encode 2^n states, reducing premature convergence.
- **Classical practicality**: No quantum hardware is needed, making the methods accessible today.

This approach aligns with other quantum-inspired metaheuristics (e.g., Quantum Particle Swarm Optimization [10] and Genetic Quantum Algorithms [14]), which repurpose quantum principles for classical optimization. Our experiments demonstrate that these adaptations improve performance over classical DEHO/ESWSA and state-of-the-art alternatives (PSO, GA, BAT).

Table. I
A QUANTUM SOLUTION

α_1	α_2	α_3	...	α_i
β_1	β_2	β_3	...	β_i

B. Solution encoding

In quantum inspired DEHO and DESWSA, the elephants' positions are represented by quantum registers. Each elephant in the swarm holds a quantum solution encoded as a sequence of n qubits, which constitutes a quantum register.

Table I presents an example of a quantum register that represents an elephant's solution in the QDEHO and QDESWSA algorithms.

Because of the superposition principle, a quantum individual can simultaneously represent an entire population, with each individual having an associated probability. This enables a more diverse representation without requiring a large population size [?]. However, it is important to recognize that measuring a quantum state causes it to collapse into a single, definitive state.

C. Solution measurement

To fully harness the superposition of states in a qubit, it must be measured. This process, known as measurement, extracts a binary solution from the quantum register. The objective is to assess the swarm's performance based on the resulting binary solutions.

Algorithm 1: Measurement function [26]

```
Input: Qubit Q_i = (α_i, β_i);
Output: Binary solution X_i;
if rand > α_i^2 then
    return 1;
else
    return 0;
end
```

As demonstrated in Algorithm 1, a random number rand is generated for each qubit, falling within the range of $[0, 1]$. The algorithm then determines the output as either 0 or 1 based on this random value.

Here's an example of measuring a binary solution from a quantum register:

Consider a quantum register composed of 3 qubits, each in a superposition state. The state of the register can be represented as:

$$|\psi\rangle = \alpha_1|0\rangle + \beta_1|1\rangle \quad \otimes \quad \alpha_2|0\rangle + \beta_2|1\rangle \quad \otimes \quad \alpha_3|0\rangle + \beta_3|1\rangle.$$

When we measure the quantum register, each qubit collapses to either $|0\rangle$ or $|1\rangle$, based on the probabilities given by $|\alpha_i|^2$ and $|\beta_i|^2$ for each qubit.

For instance, if the measurement results are:

- The first qubit collapses to $|1\rangle$, - The second qubit collapses to $|0\rangle$, - The third qubit collapses to $|1\rangle$,

then the resulting binary solution is the bit string 101.

This binary solution, extracted from the quantum register, can then be used to evaluate the current state of the algorithm or solve the given optimization problem.

Each solution is subsequently evaluated using a fitness function tailored to the specific problem. The following step involves updating each elephant's quantum solution accordingly.

D. Solution update

In the solution update step each algorithm has its own rules.

1— QDEHO: The register of each elephant $R_{c_i,j}$ is updated during every iteration using equation 1, as follows:

$$R_{new,c_i,j} = R_{c_i,j} + \alpha \times (X_{c_i,best} - X_{c_i,j}) \times r \tag{1}$$

- $X_{c_i,j}$: current solution of elephant j
- $X_{c_i,best}$: best solution in clan c_i.
- α and r are empirical parameters.

To compute the new register for the best elephant in each clan, equation 2 is employed. Here, $X_{center,c_i,d}$ denotes the center of gravity of the i^{th} clan, and β is an empirical parameter.

$$R_{new,c_i,j} = R_{c_i,j} + (X_{c_i,best} + (\beta \times X_{c_i,center})) \tag{2}$$

The center of gravity of each clan is calculated using equation 3.

$$X_{center,c_i,d} = \frac{1}{n_{c_i}} \sum_{j=1}^{n_{c_i}} x_{c_i,j,d} \tag{3}$$

$$R_{c_i,worst} = R_{c_i,worst} + (x_{min} + (x_{max} - x_{min} + 1) \times rand) \tag{4}$$

Lastly, at each generation, the least effective elephant $R_{c_i,\text{worst}}$ in each clan c_i is replaced according to equation 4, as follows:

- x_{min}: the minimum size of a solution.
- x_{max}: the maximum size of a solution.

2— QDESWSA: The neighborhood search for each elephant is defined by equation 5, as follows:

- R_i: quantum register of elephant i.
- $V_{new,i}$: the new velocity of elephant i.

$$R_{new,i} = R_i + V_{new,i} \tag{5}$$

The velocity is updated based on a random value $rand$, applying equation 7 if $p \leq rand$ and using equation 6 otherwise, as follows:

- $P_{best,i}$: personal best solution of elephant i.
- $G_{best,i} - X_i$: global best solution in the swarm.
- w^t: the inertia weight (updated using equation 8 [30]).

$$V_{new,i} = V_i w^t + rand(1,d) \odot (P_{best,i} - X_i) \tag{6}$$

$$V_{new,i} = V_i w^t + rand(1,d) \odot (G_{best,i} - X_i) \tag{7}$$

$$w^t = w_{max} - \{\frac{w_{max} - w_{min}}{t_{max}}\} \tag{8}$$

E. Register update

At each iteration, the register of each elephant must be updated, meaning that a set of qubits in Q_i^t must be modified to produce the updated Q_i^{t+1}. This operation is carried out using a quantum single-qubit gate. The selection of the quantum gate is largely determined by the specific problem at hand. The most frequently utilized quantum gates for transforming single qubits include [28]:

- Basic quantum gates: include identity, negation, phase shift, and combinations of phase shift with negation.
- Square root of NOT gate: $\sqrt{NOT}$
- Controlled-NOT gate (CNOT): a commonly used two-qubit gate.

$$(\alpha_i, \beta_i)^{(t)} = (X)(\alpha_i, \beta_i)^{(t-1)} \tag{9}$$

The new amplitudes (α_i, β_i) are obtained using equation 9, such as X is a specific quantum gate.

F. The proposed algorithms

Algorithms 2 and 3 summarize the key steps of the proposed QDEHO and QDESWSA approaches, respectively.

Table II highlights the key differences between discrete elephants swarm algorithms, namely DEHO and DESWSA, and their quantum-inspired versions proposed in this article.

In the following section, we will detail how the proposed algorithms are utilized for optimizing frequent itemset mining.

Algorithm 2: Quantum inspired DEHO (QDEHO) [26]
Input: N_c: number of clans, N: number of elephants, X_{min}: Minimum size of a solution; X_{max}: Maximum size of a solution; t_{max}: maximum number of generations;
Output: best solutions
for *i=1 to* N_c **do**
  for *j=1 to N* **do**
    Initialize the register of qubits R_i for each elephant randomly.
    Next, measure the current solution $X_{c_i,j}$ based on R_i.
  end
end
for *t=1 to* t_{max} **do**
  for *i=1 to N* **do**
    for *i=1 to Nc* **do**
      Updating each elephant R_{i,c_i} of each clan using equation 1; Updating the best elephant R_{best,c_i} of each clan using equation 2;
      Replace the worst elephant R_{worst,c_i} of each clan using equation 4;
      Update local best and global best if necessary;
    end
    If the number of generations is reached stop and return best solutions.
  end
end

Algorithm 3: QDESWSA
Input: Input: N: number of elephants; p, X_{min}: min solution size; X_{max}: max solution; t_{max}: max number of generations;
Output: best solutions
for *i=1 to N* **do**
  Randomly initialize the Register of Qbits R_i for each elephant
  Generate current solution X_i using R_i
  Initialize velocity V_i of each elephant.
end
Calculate fitness for each elephant; Save the best local fitness of each elephant; Save the global fitness of the population;
for *t=1 to* t_{max} **do**
  for *i=1 to N* **do**
    if *rand* $>$ *p* **then**
      Update velocity $V_{i,d}$ using Eq.7;
    end
    else
      local water search or update the elephant velocity $V_{i,d}$ using Eq.6;
    end
    Update the quantum position of each elephant as shown in equation 5
    Update local best and global best if necessary;
  end
  If the number of generations is reached stop and return best solutions;
end

Table. II
CLASSICAL VS. QUANTUM-INSPIRED OPERATORS

Component	Classical DEHO	QDEHO
Solution	Binary vector	Qubit register
Update	Deterministic	Probabilistic
Exploration	Random jitter	X-gate + superposition

IV. FREQUENT ITEMSETS MINING (FIM) USING QUANTUM INSPIRED ELEPHANT SWARM

A. Preliminaries on FIM

Frequent itemsets are a key concept in data mining, utilized to identify associations within extensive datasets. These patterns provide valuable insights, enabling businesses and researchers to make informed decisions [18].

Let T be a set of M transactions, represented as $T = \{t_1, t_2, \ldots, t_M\}$, which forms a transactional database, and let I be a set of N distinct items (or attributes) given by $\{i_1, i_2, \ldots, i_N\}$. An itemset X is defined as a subset of items, that is, $X \subseteq I$.

The support of an itemset is the proportion of transactions in which the itemset occurs. When an itemset occurs in a significant number of transactions, it is considered "frequent." The support count of an itemset $Sup(X)$ is the number of transactions that contains X divided by M. An itemset X is frequent if its support is no less than $MinSup$ [1] , where $MinSup$ is a threshold chosen by the user.

The identification of frequent itemsets is usually achieved using algorithms such as Apriori[2] or FP-Growth[13]. These algorithms analyze the dataset to uncover itemsets that satisfy a user-specified minimum support threshold. However, these methods can be quite time-consuming and may not be effective for large datasets.

To address the performance limitations of brute-force methods, various algorithms utilizing bio-inspired techniques have been developed. Techniques such as genetic algorithms [34] and particle swarm optimization (PSO) [24] have demonstrated improved performance compared to traditional brute-force approaches.

The next sections give detailed explanation of how QDEHO and QDESWSA can be applied to FMI.

B. Solution representation

A solution is depicted as a vector of n qubits, referred to as a register, where n denotes the number of items in the dataset.

In each iteration, the associated itemset is generated by applying Algorithm 1 to the current register.

C. Solution update

Each register is updated according to Equation 9. For the frequent itemset mining problem, we employ the NOT gate, which inverts α_i and β_i within a single qubit.

In quantum computing, this NOT gate is referred to as the Pauli-X gate, which toggles the state of a single qubit between 0 and 1.

Mathematically, the Pauli-X gate can be expressed as the following 2×2 matrix:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

D. Fitness function

The fitness function for the problem of frequent itemsets is represented by the following equation.

$$f(X) = \frac{\text{Number of transactions containing } X}{\text{Number of transactions in database}} \tag{10}$$

The objective is to maximize the fitness function.

V. Performance evaluation

In this section, we will assess the performance of our proposed methods across a range of datasets of varying sizes and compare the results with those of relevant algorithms in the field.

All algorithms are implemented in Java and executed on an Intel Core i7 machine with 16GB of RAM, running Windows 10.

A. Datasets description

To effectively showcase the performance of QDEHO and QDESWSA on real-world data, a series of experiments were carried out using six datasets from reputable repositories, including the Frequent Itemset Mining Dataset Repository [12, 32].

In Table III, each dataset is described in terms of number of transactions and number of items .

Table. III
Datasets description

Dataset	N^o of transactions	N^o of items
IBM Quest 1	2,041	999
Chess	3,196	75
Mushroom	8,124	119
IBM Quest 2	18,905	999
Pumbs star	40,385	7,116
Connect	100,000	999

The parameters employed in our approach, along with those of the state-of-the-art algorithms, were established through thorough experimentation.

Each algorithm was run with the number of iterations varying from 100 to 1000, while ensuring that the maximum iteration limit was consistent across all algorithms.

The final results are based on the averages of 10 consecutive executions for each iteration count, providing the average fitness outcome for each algorithm.

B. Numerical results and discussion

Tables IV and V present the average fitness and running time values, respectively, of the proposed algorithms in comparison with PSO [19], GA [17], BAT [15], and WOA [25]. The results are illustrated in Figures 2 and 3.

Table. IV
Average fitness for extracted frequent itemsets

Dataset	QDEHO	QDESWSA	PSO	GA	BAT	WOA
IBM Quest 1	0.007	0.01	0.001	0.001	0.001	0.002
Chess	0.99	0.97	0.37	0.14	0.86	0.01
Mushroom	0.75	0.77	0.19	0.13	0.34	0.01
IBM Quest 2	0.03	0.06	0.003	0.001	0.01	0.01
Pumbs star	0.06	0.08	0.004	0.004	0.04	0.02
Connect	0.62	0.01	0.19	0.13	0.79	0.01

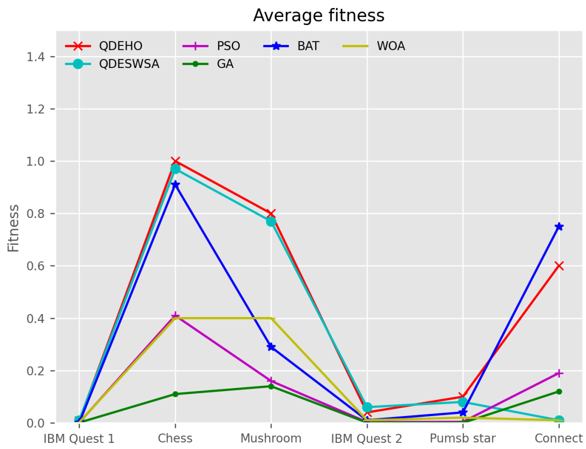

Fig. 2: Average fitness of QEDHO and QDESWSA in comparison with PSO, GA, BAT and WOA

Table. V
Average CPU time for extracted frequent itemsets

Dataset	QDEHO	QDESWSA	PSO	GA	BAT	WOA
IBM Quest 1	5.96	4.4	0.01	0.12	28.08	0.002
Chess	1.43	1.44	0.004	0.03	7.62	0.002
Mushroom	2.41	1.58	0.04	0.03	12.1	0.001
IBM Quest 2	1.3	0.85	0.004	0.28	6.14	0.004
Pumbs star	77.8	52.59	0.007	2.64	402.56	0.004
Connect	29.6	19.24	0.03	0.13	188.76	0.002

Fig. 3: Average running time in seconds of QEDHO and QDESWSA in comparison with PSO, GA, BAT and WOA

Table IV shows that both QDEHO and QDESWSA consistently outperform PSO, GA and WOA in terms of average fitness across all datasets. It also exhibits superior performance compared to BAT, with the exception of one case where BAT achieves marginally better results. These conclusions are further supported by Figure 2.

Regarding average CPU time, QDEHO and QDESWSA demonstrate satisfactory performance in comparison to GA and PSO, with slightly higher computation times attributed to QDEHO. However, the proposed methods prove to be faster than BAT, despite delivering competitive solution quality. These observation indicates that the use and continuous updating of a quantum register play a key role in enhancing the efficiency of both QDEHO and QDESWSA algorithms.

In conclusion, the experimental results highlight the effectiveness and practicality of our proposals for solving discrete optimization problems.

VI. Conclusion and perspectives

In this paper, we presented two novel quantum-inspired swarm algorithms, QDEHO and QDESWSA. The key innovation of our work is the ambitious integration of quantum principles into the discrete swarm-based frameworks of DEHO and DEWSA.

The effectiveness of the proposed algorithms was validated through their application to the frequent itemset mining problem. Extensive experiments conducted on six benchmark datasets of varying sizes demonstrate that QDEHO and QDESWSA consistently achieve highly competitive results, outperforming several well-established state-of-the-art algorithms.

Building on this work, our future research will focus on three main directions. First, we will conduct rigorous comparisons with state-of-the-art quantum-inspired optimization techniques, such as variational quantum algorithms and quantum annealing hybrids. Second, we aim to extend our approach to high-dimensional clustering problems by leveraging qubit-based centroid representations, which can enhance both solution quality and scalability. Third, we plan to develop Qiskit-based implementations of our algorithms to benchmark their performance on real quantum hardware, particularly IBM quantum processors.

References

[1] 1Qbit. A bit or two about qubits. https://1qbit.com/blog/quantum-computing/a-bit-or-two-about-qubits/, 2022. Accessed: July, 2023.

[2] R. Agrawal, T. Imielinski, and A. Swami. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.

[3] V. Alekhya, N. V. U. Reddy, J. Singh, B. Boddu, R. Sobti, and A. A. Hameed. High-dimensional data processing using quantum-inspired evolutionary algorithms for homeland security imaging systems. In *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)*, pages 1184–1189, 2024. doi: 10.1109/IC3SE62002.2024.10592971.

[4] M. S. Alvarez-Alvarado, F. E. Alban-Chacón, E. A. Lamilla-Rubio, et al. Three novel quantum-inspired swarm optimization algorithms using different bounded potential fields. *Scientific Reports*, 11:11655, 2021. doi: 10.1038/s41598-021-90847-7. URL https://doi.org/10.1038/s41598-021-90847-7.

[5] H.-P. Chiang, Y.-H. Chou, C.-H. Chiu, and Y.-M. Huang. A quantum-inspired tabu search algorithm for solving combinatorial optimization problems. *Soft Computing*, 18:1771–1781, 09 2013. doi: 10.1007/s00500-013-1203-7.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.

[7] D. Deutsch and A. Ekert. Quantum computation. *Physics World*, 11(3):47–52, 2006.

[8] A. Dey, S. Bhattacharyya, S. Dey, J. Platos, and V. Snasel. A quantum inspired differential evolution algorithm for automatic clustering of real life datasets. 83(3), 2023. ISSN 1380-7501. doi: 10.1007/s11042-023-15704-3. URL https://doi.org/10.1007/s11042-023-15704-3.

[9] A. Elashry, M. Alrahmawy, and M. Rashad. Enhanced quantum inspired grey wolf optimizer for feature selection. *International Journal of Intelligent Systems and Applications*, Volume 12:8–17, 06 2020.

[10] A. Flori, H. Oulhadj, and P. Siarry. Quantum particle swarm optimization: An auto-adaptive pso for local and global optimization. *Comput. Optim. Appl.*, 82 (2):525–559, jun 2022. ISSN 0926-6003. doi: 10.1007/s10589-022-00362-2. URL https://doi.org/10.1007/s10589-022-00362-2.

[11] S. Hakemi, M. Houshmand, E. KheirKhah, and S. A. Hosseini. A review of recent advances in quantum-inspired metaheuristics. *Evolutionary Intelligence*, 17:627–642, 10 2022. doi: 10.1007/s12065-022-00783-2.

[12] W. Hämäläinen. Datasets. http://www.cs.uef.fi/ whamalai/datasets.html, 2019. Accessed: 2019-08-10.

[13] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 1–12, 2000.

[14] K.-H. Han and J.-H. Kim. Genetic quantum algorithm and its application to combinatorial optimization problem. volume 2, pages 1354 – 1360 vol.2, 02 2000. ISBN 0-7803-6375-2. doi: 10.1109/CEC.2000.870809.

[15] K. Heraguemi, N. Kamel, and H. Drias. Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies. *Applied Intelligence*, 45, 12 2016. doi: 10.1007/s10489-016-0806-y.

[16] A. S. Hesar and M. Houshmand. A memetic quantum-inspired genetic algorithm based on tabu search. *Evolutionary Intelligence*, 17(3):1837–1853, June 2024. ISSN 1864-5917. doi: 10.1007/s12065-023-00866-8.

[17] J. H. Holland. Genetic algorithms. *Scientific American*, 267(1):66–73, 1992. ISSN 00368733, 19467087. URL http://www.jstor.org/stable/24939139.

[18] H. Jiawei and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2011.

[19] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, Nov 1995. doi: 10.1109/ICNN.1995.488968.

[20] J. Khudair Madhloom, H. N. Abd Ali, H. A. Hasan, O. A. Hassen, and S. M. Darwish. A quantum-inspired ant colony optimization approach for exploring routing gateways in mobile ad hoc networks. *Electronics*, 12(5), 2023.

ISSN 2079-9292. doi: 10.3390/electronics12051171. URL https://www.mdpi.com/2079-9292/12/5/1171.

[21] D. Konar, S. Bhattacharyya, K. Sharma, S. Sharma, and S. R. Pradhan. An improved hybrid quantum-inspired genetic algorithm (hqiga) for scheduling of real-time task in multiprocessor system. *Applied Soft Computing*, 53:296–307, 2017. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2016.12.051. URL https://www.sciencedirect.com/science/article/pii/S1568494616306949.

[22] D. Konar, K. Sharma, V. Sarogi, and S. Bhattacharyya. A multi-objective quantum-inspired genetic algorithm (mo-qiga) for real-time tasks scheduling in multiprocessor environment. *Procedia Computer Science*, 131:591–599, 2018. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2018.04.301. URL https://www.sciencedirect.com/science/article/pii/S1877050918306811. Recent Advancement in Information and Communication Technology:.

[23] S. Kuo, J. Shen, C. Liu, and Y. Chou. Hybrid quantum-inspired evolutionary neural networks for intrusion detection system. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2801–2806, 2024. doi: 10.1109/SMC54092.2024.10831208.

[24] J. C.-W. Lin, L. Yang, P. Fournier-Viger, J. M.-T. Wu, T.-P. Hong, L. S.-L. Wang, and J. Zhan. Mining high-utility itemsets based on particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 55:320–330, 2016. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2016.07.006. URL https://www.sciencedirect.com/science/article/pii/S0952197616301312.

[25] S. Mirjalili and A. Lewis. The whale optimization algorithm. *Advances in Engineering Software*, 95:51 – 67, 2016. ISSN 0965-9978. doi: https://doi.org/10.1016/j.advengsoft.2016.01.008. URL http://www.sciencedirect.com/science/article/pii/S0965997816300163.

[26] H. Moulai. Frequent itemsets mining using new quantum inspired elephant swarm algorithm. In H. Drias and F. Yalaoui, editors, *Quantum Computing: Applications and Challenges*, pages 65–76, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-59318-5.

[27] H. Moulai and H. Drias. Association rule mining using new discrete elephant swarm approaches. *Expert Systems*, 40(2):e13159, 2023. doi: 10.1111/exsy.13159. URL https://doi.org/10.1111/exsy.13159.

[28] E. Rieffel and W. Polak. An introduction to quantum computing for non-physicists. *ACM Computing Surveys*, 32(3):300–335, 2000.

[29] L. Siddiqui, A. Mani, and J. Singh. Improving design accuracy of a finite impulse response fractional order digital differentiator filter using quantum-inspired evolutionary algorithm. In *2024 14th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pages 142–147, 2024. doi: 10.1109/Confluence60223.2024.10463238.

[30] M. Sudip. Elephant swarm water search algorithm for global optimization. *Sādhanā*, 43:1–21, 2018. doi: 10.1007/s12046-017-0780-z.

[31] H. Talbi and A. Draa. A new real-coded quantum-inspired evolutionary algorithm for continuous optimization. *Applied Soft Computing*, 61:765–791, 2017. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2017.07.046. URL https://www.sciencedirect.com/science/article/pii/S1568494617304660.

[32] E. M. Tapia. Activity recognition in the home setting using simple and ubiquitous sensors. http://courses.media.mit.edu/2004fall/mas622j/04.projects/home/, 2018.

[33] X. Wang, Z. Geng, L. Chen, and Y. Han. Improved dragonfly optimization algorithm based on quantum behavior for multi-objective optimization of ethylene cracking furnace. *Swarm and Evolutionary Computation*, 88:101607, 2024. ISSN 2210-6502. doi: https://doi.org/10.1016/j.swevo.2024.101607. URL https://www.sciencedirect.com/science/article/pii/S2210650224001457.

[34] X. Yan, C. Zhang, and S. Zhang. Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. *Expert Systems with Applications*, 36(2, Part 2):3066–3076, 2009. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2008.01.028. URL https://www.sciencedirect.com/science/article/pii/S0957417408000195.

[35] C. Yu, Z. Cai, X. Ye, M. Wang, X. Zhao, G. Liang, H. Chen, and C. Li. Quantum-like mutation-induced dragonfly-inspired optimization approach. *Mathematics and Computers in Simulation*, 178:259–289, 2020. ISSN 0378-4754. doi: https://doi.org/10.1016/j.matcom.2020.06.012. URL https://www.sciencedirect.com/science/article/pii/S0378475420302147.

Hadjer Moulai was born in Algiers on 1993. She earned a Ph.D degree in computer science specializing in artificial intelligence from the University of Science and Technology Houari Boumediene in 2020. She is a permanent lecturer at USTHB since 2021, and an active researcher in the laboratory of research in artificial intelligence LRIA. Her research spans various fields such as machine learning, optimization algorithms, data mining and quantum computing.