Quantum Modeled Clustering Algorithms for Image Segmentation

[1]Ellis Casper and [2]Chih-Cheng Hung
[1]*Center for Biometrics Research, Southern Polytechnic State University, USA,*
ecasper@spsu.edu
[2]*Anyang Normal University, Anyang, China,*
Center for Biometrics Research, Southern Polytechnic State University, USA,
chung@spsu.edu

Abstract

The ability to cluster data accurately is essential to applications such as image segmentation. Therefore, techniques that enhance accuracy are of keen interest. One such technique involves applying a quantum mechanical system model, such as that of the quantum bit, to generate probabilistic numerical output to be used as variable input for clustering algorithms. This work demonstrates that applying a quantum bit model to data clustering algorithms can increase clustering accuracy, as a result of simulating superposition as well as possessing both guaranteed and controllable convergence properties. For accuracy assessment purposes, four quantum-modeled clustering algorithms for multi-band image segmentation are explored and evaluated. The clustering algorithms of choice consist of quantum variants of K-Means, Fuzzy C-Means, New Weighted Fuzzy C-Means, and the Artificial Bee Colony. Data sets of interest include multi-band imagery, which subsequent to classification are analyzed and assessed for accuracy. Results demonstrate that these algorithms exhibit improved accuracy, when compared to classical counterparts. Moreover, solutions are enhanced via introduction of the quantum state machine, which provides random initial centroid and variable input values to the various clustering algorithms, and quantum operators, which bring about convergence and maximize local search space exploration. Typically, the algorithms have shown to produce better solutions.

Keywords: *clustering algorithms, quantum mechanics, quantum computing, image segmentation*

1. Introduction

Data clustering involves grouping data set objects that share similar application dependent characteristics or patterns into clusters. Clustering differs from classification, where these objects are assigned predefined class labels, in that object classes are not assigned predefined labels. Cluster analysis can be classified as either supervised or unsupervised, with the previous requiring some degree of human interaction and prior knowledge of the data set, while the later enjoys increased popularity due to the inherent lack of prior dataset knowledge dependency. Some typical data clustering algorithms that are commonly utilized are partitioning algorithms such as K-Means [23] and Fuzzy C-Means [26], and metaheuristic swarm intelligence based algorithms such as the Artificial Bee Colony [27] and Particle Swarm Optimization [4].

In recent years, many bodies of work have applied quantum modeled algorithms to data clustering [3-6,8,9,11,15,16]. Some, but not all, works have assessed clustering accuracy. In order to be considered viable, quantum modeled algorithms need to provide tangible benefits, particularly in the form of a contribution to overall accuracy. Hence, the purpose of this body of work, which is to determine the viability of quantum modeled algorithms by addressing the following two research questions: Can quantum modeled algorithms effectively contribute to the overall accuracy of data clustering? If so, can these algorithms be applied practically, to a common application such as image segmentation?

Since the advent of the digital age, high performance computing has been on the forefront of many technical endeavors. This computational ability has provided academia, as well as private industry, with the ability to solve complex and/or time consuming problems with a high level of precision. Historically, the pinnacle of high performance computing has manifested itself technically in cutting edge hardware (i.e. supercomputers) as well as software (i.e. high performance algorithms such as

doi: 10.4156/pica.vol2.issue1.1

Quick Sort). However, these traditional pinnacles have the potential to be exceeded by the emerging field of quantum computing and all it has to offer. Of particular interest, is exploiting the various quantum properties such as superposition, entanglement, coherence and interference.

Quantum properties can be simulated via quantum system modeling algorithms, and used in applications that can benefit from these inherent properties. One such application is image segmentation, or thresh-holding, which has proven to benefit from the diversity derived from simulating quantum superposition via quantum bit (qubit) modeling. Moreover, quantum bits can be manipulated via quantum operators, such as a rotation gate, which consists of a unitary matrix and can be used to bring algorithmic convergence to fruition. Typical fields of application that utilize image thresh-holding involve remote sensing, biomedical, and bioinformatics. Image segments can be created via data clustering algorithms, and four data clustering algorithms were chosen. The clustering algorithms of choice consist of quantum variants of K-Means, Fuzzy C-Means, New Weighted Fuzzy C-Means, and the Artificial Bee Colony. These algorithms were chosen for several reasons, the first of which is that they represented a diverse sample of clustering algorithms, hard and soft partitioning as well as metaheuristic swarm intelligence based algorithms. Second, these algorithms were chosen for their simplicity and adeptness at accepting output from the quantum probabilistic Turing machine as initial centroid input. The third and final reason that four clustering algorithms were deemed sufficient to draw a reasonable empirical conclusion as to the quantum models effectiveness as it relates to accuracy.

The following outlines the remainder of this body of work: In Section 2, cluster analysis and quantum computing are reviewed. The quantum modeled clustering algorithms are described in Section 3. Experimental results and analysis are discussed in Section 4. Conclusions are drawn in Section 5.

2. Related works

In recent years, quantum mechanical computing models have been applied in various scientific fields to a plethora of different applications, ranging from image thresh-holding, to classic NP hard type problems, such as the knapsack combinatorial optimization problem [2], and Flow shop scheduling [3]. With regard to previous methodologies associated with applying quantum mechanical computing models, most involve either a quantum bit representation, or choosing an appropriate potential well case, and subsequently solving the Schrödinger equation for particle motion [4].

2.1. Clustering Algorithms

As previously stated, data clustering involves grouping data set objects that share similar application dependent characteristics or patterns into clusters [17]. Clustering differs from classification, where these objects are assigned predefined class labels, in that object classes are not assigned predefined labels. Cluster analysis can be classified as either supervised or unsupervised, with the previous requiring some degree of human interaction and prior knowledge of the data set, while the later enjoys increased popularity due the inherent lack of prior dataset knowledge dependency. Clustering classification also includes hierarchal, spectral, and partitional clustering. Partitional clustering can be classified as either hard or soft. In hard partitional clustering, each data set object is assigned to exactly one cluster, where as in soft partitional clustering each data object may belong to multiple clusters, with a certain probability of membership to each cluster. Moreover, Common fields of application include image processing, pattern recognition, machine learning, bioinformatics, and data mining.

The K-Means clustering algorithm is a hard partitioning algorithm, which assigns data members to the nearest calculated mean, represented by K clusters [23]. With regard to K-Means, each cluster centroid is initialized, and each data set member is assigned to a cluster, via minimum Euclidean distance or some other chosen distance metric, between the member and the centroid. Afterwards, each centroid is recalculated by averaging all members assigned to a particular cluster, and using this new centroid value to reassign members. The goal of K-Means cluster analysis is to minimize the following objective function:

$$J_m(U,\ V) = \sum_{j=1}^{n}\sum_{i=1}^{c} d^2(X_j, V_i) \tag{1}$$

where $d^2(X_j, V_i) \ = \ \|X_j - V_i\|^{\ 2}$ and represents a distance measure of choice, X_j is j^{th} data member of the i^{th}cluster of cluster center V_i, and m is any real number greater than one. Moreover, low time complexity and simplicity are the most prominent benefits of K-Means, while drawbacks consist of difficulty determining the exact number of real clusters, susceptibility to poor centroid initialization, sensitive to noise and outliers, and non-globular shape limitations.

The Fuzzy C-Means clustering algorithm is a soft partitioning algorithm [26]. With regard to Fuzzy C-Means, or FCM, the following objective function is minimized:

$$J_m(U,\ V) = \sum_{j=1}^{n}\sum_{i=1}^{c}(u_{ij})^m d^2(X_j, V_i) \tag{2}$$

where $d^2(X_j, V_i) \ = \ \|X_j - V_i\|^{\ 2}$ and represents a distance measure of choice, u_{ij} is the degree of fuzzy membership for j^{th} data member X_j in the i^{th}cluster of cluster center V_i, and m is any real number greater than one. The algorithm begins by initializing fuzzy membership for each pixel randomly, and each data set member is assigned to a cluster center, via minimum Euclidean distance or some other chosen distance metric, between the member and the centroid. Afterwards, membership values are normalized and each clusters mean is calculated from these initial fuzzy membership values. Once the mean for each cluster has been calculated, these mean values are used to recalculate fuzzy membership. Unlike K-Means, FCM allows data member membership to more than one cluster, with a certain probability of membership to a particular cluster. However, like K-Means, fuzzy C-Means is sensitive to poor centroid initialization, and requires the number of clusters to be determined prior to clustering. Moreover, fuzzy C-Means is more computationally expensive than K-Means.

In contrast to the standard Fuzzy C-Means algorithm described in the previous section, the New Weighted Fuzzy C-Means algorithm [12] minimizes the following objective function:

$$J_m(U,\ M) = \sum_{j=1}^{n}\sum_{i=1}^{c}(u_{ij})^m d^2(X_j, M_{ij}) \tag{3}$$

where $d^2(X_j, M_{ij}) \ = \ \|X_j - M_{ij}\|^{\ 2}$ and represents a distance measure of choice, u_{ij} is the degree of fuzzy membership for j^{th} data member X_j in the i^{th}cluster, M_{ij} is the unsupervised weighted mean, and m is any real number greater than one. Moreover, the weighted mean M_{ij} is represented by equation (4):

$$M_{ij} \ = \ \sum_{\substack{k=1 \\ k\neq j}}^{n} \frac{\|v_i - x_k\|^{\ -1} u_{ik}}{\|v_i - x_t\|^{\ -1} u_{it}} x_k \tag{4}$$

In addition, with the introduction of Lagrange Multiplier methods, a reformulated objective function can be realized as equation (5):

$$(U,\ \xi) = \sum_{j=1}^{n}\sum_{i=1}^{c}(u_{ij})^m d^2(X_j, M_{ij}) + \sum_{j=1}^{n}\xi_j(\sum_{i=1}^{c}u_{ij} - 1) \tag{5}$$

where ξ_j with j = 1, …, n represents the Lagrange multipliers for n constraints. As a result of differentiating this new objective function, equations (6) and (7) are obtained:

$$\xi_j = (\sum_{i=1}^{c}(\|v_i - \ x_k\|^{\ 2} \cdot m\sum_{i=1}^{c}u_{ij}^{m-1})^{1/(1-m)})^{1-m} \tag{6}$$

$$u_{ij} \ = \ \xi_j^{1/(m-1)}(\|x_j - \ M_{ij}\|^{\ 2} \cdot \ m\sum_{i=1}^{c}u_{ij}^{m-1})^{1/(1-m)} \tag{7}$$

Just as in the standard Fuzzy C-Means algorithm previously described, the New Weighted Fuzzy C-Means (NW-FCM) algorithm begins by initializing fuzzy membership for each pixel randomly, and each data set member is assigned to a cluster center, via minimum Euclidean distance or some other chosen distance metric, between the member and the centroid. Afterwards, membership values are normalized and the iterative process begins. First, each clusters mean is calculated from these initial fuzzy membership values. This is followed by calculating the weighted meansM_{ij}, and once the mean value is obtained it is used to update the Lagrange multiplier ξ_j , which in turn is used to update fuzzy membership. Then, the process is repeated until algorithmic convergence criteria are met. While NW-FCM is much more computationally expensive than FCM, it exhibits greater stability and accuracy in general.

The Artificial Bee Colony (ABC) Algorithm as defined by Karaboga in 2005 [27], is a metaheuristic swarm intelligence based algorithm applied to optimization problems. The algorithm simulates the foraging behavior of honey bees, in which the population utilizes a multidimensional search space to select optimal food (nectar) sources. In the process, information is exchanged between some population individuals (employed and onlooker bees) and combined with information already possessed by each individual, and positions are adjusted accordingly. Moreover, other individuals (scouts) choose food sources randomly. Upon discovering a new more plentiful food source, the position of the new source is memorized and the previous source is forgotten. Through the activities of these three categories of artificial honey bees, a good balance of exploration and exploitation is achieved.

With regard to ABC, every food source is investigated and employed by exactly one employed bee. The following Equation (8) is utilized to generate initial food sources

$$x_{ij} = x_j^{min} + rand(0,1)\left(x_j^{max} - x_j^{min}\right) \tag{8}$$

where i = 1, …, SN with SN representing the number of food sources, j = 1, …, D with D representing the number of parameters to be optimized, and x_j^{min} and x_j^{max} representing the minimum and maximum initial boundary values of parameter j.

Each employed bee investigates and evaluates the amount or quality of a new food source via Equation (9):

$$v_{ij} = x_{ij} + \varphi_{ij}\left(x_{ij} - x_{kj}\right) \tag{9}$$

where v_{ij} is the generated new food source, φ_{ij} represents a random number between -1 and 1, and x_{kj} is a neighboring food source. Once an employed bee's food source has been depleted, determined via a trial limit condition, the employed bee becomes a scout bee. The trial limit condition involves a specified number of attempts allowed to improve a food source, and if exceeded the source is abandoned. Food sources are selected by onlooker bees probabilistically according to Equation (10):

$$p_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_i} \tag{10}$$

where $fitness_i$ is the fitness value associated with the i^{th} food source. In order to apply the ABC algorithm to data clustering [14], the optimization parameters of each food must serve as cluster centroids, and evaluated by some cluster validity index based objective function for fitness value. Typically, the fitness value is the reciprocal of the objective function value. Moreover, optimizing parameters is crucial to the performance of ABC [21,22].

In order to evaluate the quality of a particular solution, a cluster validity index must be utilized. From this index value, the associated "fitness" or cluster quality can be estimated. Within this body of work, for the clustering algorithms in the previous sections, two different cluster validity indices will be utilized.

The first cluster validity index is the Davies-Bouldin (DB) Index method [7], or DB Index. The DB index takes into consideration both the inter-cluster scatter as well as the intra-cluster distance as follows:

$$DB = \overline{R} \equiv \frac{1}{N} \sum_{i=1}^{N} R_i \tag{11}$$

where

$$R_{ij} \equiv \frac{S_i + S_j}{M_{ij}} \tag{12}$$

and

$$S_i = \left\{ \frac{1}{T_i} \sum_{j=1}^{T_i} \left| X_j - A_i \right|^q \right\}^{1/q} \tag{13}$$

where A_i is the centroid of the i^{th} cluster, X_j is a pixel in the i^{th} cluster, and T_i is the number of pixels in the i^{th} cluster, and q is any positive real number.

$$M_{ij} = \left\{ \sum_{k=1}^{N} \left| a_{ki} - a_{kj} \right|^p \right\}^{1/p} \tag{14}$$

where a_{ki} is the k^{th} component of the n-dimensional vector a_i, which is the centroid of the i^{th} cluster, and p is any positive real number. Using the DB Index, a fitness function can be formulated:

$$f = \frac{1}{(1 + DB)} \tag{15}$$

The second cluster validity index of interest is specifically for the fuzzy clustering algorithms, FCM and NW-FCM. It is known as the Xie and Beni's (XB) Index method [17], or XB Index. The XB index takes into consideration both the inter-cluster scatter as well as the intra-cluster distance as follows:

$$XB = \frac{\sum_{j=1}^{n} \sum_{i=1}^{c} (u_{ij})^2 \left\| V_i - X_j \right\|^2}{n \min ij \left\| V_i - V_j \right\|^2} \tag{16}$$

where u_{ij} is the fuzzy weight or membership of pixel X_j in the i^{th} cluster, V_i is the fuzzy cluster centroid of the i^{th} cluster, V_j is the fuzzy cluster centroid of the j^{th} cluster , n is the number of pixels in the image, and c is the total number of clusters. Using the XB Index, a fitness function can be formulated:

$$f = \frac{1}{(1 + XB)} \tag{17}$$

2.2. Quantum computing

The model of choice for this work is that of the qubit, in the form of a string of qubits. This model was chosen specifically for the diversity obtained from representing the quantum mechanical superposition of states. Previous works have applied this model to various algorithms, such as Genetic Algorithms [5,9], evolutionary algorithms [6], and Particle Swarm Optimization [11]. Other bodies of work have also applied the qubit model to partitioning algorithms, such as K-Means [5,15], as well as Fuzzy C-Means [16,29]. This body of work will apply the qubit model not only to K-Means and Fuzzy C-Means, but also to New Weighted Fuzzy C-Means [12] and the Artificial Bee Colony [14,30]. These particular clustering algorithms were chosen both for their simplicity, and their adeptness for accepting randomly generated quantum centroid values as input.

To understand how a qubit works, one needs to acquire a fundamental understanding of quantum state systems. The basis of quantum physics is derived from the Schrodinger equation for matter waves [28], which in time-dependent general form can be expressed as

$$i\hbar \frac{\partial}{\partial t} \Psi = \widehat{H} \Psi \tag{18}$$

where Ψ represents the wave function of a particle, and $\widehat{H}$ is the Hamiltonian, which represents the sum of the kinetic and potential energies, i is the imaginary component, and $\hbar$ is the Planck constant. This equation completely describes the evolution of a matter wave associated with a given particle over time. Moreover, the Hamiltonian is equivalent to

$$\hat{H} = \hat{T} + \hat{V} \tag{19}$$

where $\hat{T}$ and $\hat{V}$ are the kinetic and potential energies respectively. Also, for a single particle in an electric field the kinetic energy can be represented as

$$\hat{T} = -\frac{\hbar^2}{2m}\,\nabla^2 \tag{20}$$

where ∇^2 represents the three dimensional gradient such that

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \tag{21}$$

Therefore, the total energy represented by the Hamiltonian is equivalent to

$$\hat{\mathrm{H}} = -\frac{\hbar^2}{2m}\,\nabla^2 + V \tag{22}$$

and substituting this back into the time-dependent Schrodinger equation gives

$$i\hbar\frac{\partial}{\partial t}\Psi(\boldsymbol{r},t) = \left[\frac{-\hbar^2}{2m}\,\nabla^2 + V(\boldsymbol{r},t)\right]\Psi(\boldsymbol{r},t) \tag{23}$$

Also, particles have quantum spin states associated with them, which represent intrinsic angular momentum, the value of which varies according to particle classification. For a fermion, such as an electron, the spin state is $\frac{1}{2}$ and corresponds to a vector quantity that represents the magnitude of the intrinsic angular momentum of the particle. Two spin states are possible for a fermion, spin up and spin down. In a two state qubit system, spin up and spin down are analogous to classical binary bit states 0 and 1 respectively. For visualization purposes, the spin states of a fermion can be represented by a Bloch sphere.

In quantum mechanics, a superposition or linear combination of states can be can be represented in Dirac notation [1] such that

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{24}$$

where ψ represents an arbitrary state vector in Hilbert space, α and β represent probability amplitude coefficients, and $|0\rangle$ and $|1\rangle$ represent basis states. These basis states correspond to spin up and spin down respectively. The state vector in normalized form can be represented as

$$\langle\,\psi\,|\,\psi\,\rangle = 1 \tag{25}$$

or equivalently

$$|\alpha|^2 + |\beta|^2 = 1 \tag{26}$$

where $|\alpha|^2$ and $|\beta|^2$ are the probabilities of quantum particle measurement, yielding a particular state. Moreover, due to the superposition of quantum states, the particle may be in either a single state or multiple states *simultaneously.*

These probability amplitudes are complex numbers, and in matrix form can represent a qubit

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \tag{27}$$

Moreover, a series of qubits can form a string such that

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array}\right] \tag{28}$$

In addition, states $|0\rangle$ and $|1\rangle$ can be represented in bit form as

$$|0\rangle = \begin{array}{c} \mathbf{0} \\ \mathbf{1} \end{array} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{29}$$

$$|1\rangle = \begin{array}{c} \mathbf{0} \\ \mathbf{1} \end{array} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{30}$$

Likewise, the qubit string in equation (11) can be expressed in the same notation [10]. For instance, the following eight bit string

11010010

can be represented as

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{31}$$

or in tensor product form

$$|1\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \tag{32}$$

In addition to the qubit model, another quantum model that has been utilized in other bodies of work and applied to particle swarm optimization [4] is the quantum potential well model. The concept involves a particle being trapped in a quantum potential well, bounded by the sides of the well, while being attracted to the center. The Schrödinger equation can be solved for this special case, with known solutions. For particle swarm optimization, for each particle the Schrödinger equation can be solved iteratively for position, with algorithmic convergence being achieved once the particle reaches the center of the well. There are many potential well distributions to choose from [28], a few of which will be mentioned here, the simplest of which is the delta potential well:

$$V(r) = -\lambda\delta(r) \tag{33}$$

where λ is negative number and $\delta(r)$ represents the Dirac delta function. Another potential well distribution is the square potential well:

$$V(r) = \begin{cases} 0 & , |r| \leq W/2 \\ V_0 & , \text{otherwise} \end{cases} \tag{34}$$

where W is the width of the well, and V_0 represents the energy threshold required to avoid being bound with the walls of the well, between the points $r = \pm W/2$. Finally, the last potential well distribution that will be noted is the harmonic oscillator potential:

$$V(r) = \frac{1}{2} kr^2 \tag{35}$$

where k is a constant that determines the strength of the well.

Given the choice to evaluate either the qubit or the potential well model, the model of choice for this work is the qubit model. With this in mind, the goal of this body of work is to ascertain the effectiveness of the quantum bit model, with as few additional enhancements as possible. Thereby, revealing the true essence of the models potential. With that being stated, the quantum bit model requires only a quantum rotation gate, which is used to rotate each qubit to the desired state, in order to bring convergence to fruition. In addition to the gate, for purposes of enhancing diversity and escaping the local optimum, two quantum variants of classical genetic operators, crossover and mutation are applied.

3. Quantum Modeled Clustering Algorithms

In order to simulate the quantum mechanical property of superposition of states, a probabilistic Turing machine must be utilized. Each of the quantum modeled algorithms that follows will use the output of this state machine as input to the clustering algorithms described previously. Moreover, the probability of obtaining a particular outcome can be controlled by operating on the associated qubits directly with a quantum rotation gate. In order to implement the quantum state machine, for each algorithmic iteration a series of black box quantum "*oracle"* objects are utilized, each of which in this context poses a solution. The qubit string representation previously described represents a superposition of states, and each oracle possesses *n* qubit strings with *m* qubits per string. Since *m* represents the string length in qubits, the value of *m* for each string is chosen according to the estimated optimal centroid initial value. The total length of each qubit string is the product of *m*, the number of bands associated with the image being segmented, and the number of clusters chosen beforehand. Once the desired centroid value is determined, the number of qubits is specified accordingly, along with the appropriate number of clusters for the dataset of interest. Moreover, qubit strings may be applied to any random variable, not just cluster centers. Simply apply qubit string values to all desired random variables, obtain some measure of fitness via the inverse of some appropriate cluster validity index, determine the fittest solution thus far, and use the rotation gate and the state information of the best solution to converge the output of the state machine to that of the best solution. The quantum rotation gate is a unitary matrix operator, and can be formulated as follows:

$$\mathrm{G} = \begin{bmatrix} cos\,\theta & -sin\,\theta \\ sin\,\theta & cos\,\theta \end{bmatrix} \tag{36}$$

The angle of rotation θ is determined via a lookup table based upon the variable angle distance method [6]. The rotation gate is only one of a variety of gates that can be applied to the probability amplitude coefficients of each qubit [9].

By applying the rotation gate, the probability that the output values will reflect the state of the best current solution will increase as the qubit state values approach that of the best solution, the rate of which is determined according to the angle of the rotation gate operating on the qubits following each iteration. More specifically, to control the algorithmic rate of convergence, for each iteration, as stated before a series of black box quantum "*oracles"* are utilized. Each oracle poses a solution, and consists of an instantiated object that generates its own input. This input is generated by observing the probability amplitude values associated with each of the various self-contained qubit strings, thereby collapsing the superposition of states of each qubit to a single state, and encoding each collapsed state into a binary string.

During the *Initialize Q(t)* stage, the oracle population is initialized, and upon instantiation of each oracle object, each probability amplitude in each qubit string is initialized to $\frac{1}{\sqrt{2}}$. With this initial value, the probability of being observed in either binary state is equivalent. This binary encoding is followed by decoding these binary values into decimal values, and then providing these decoded values as input to the various clustering algorithms, as either centroids or random parameters. The processes of state observation, binary encoding and decimal decoding occur during the *Make P(t)* stage. Once the clustering algorithm of interest has received these values as input from the oracle, data partitioning will occur. Afterwards, the cluster fitness is evaluated during the *Evaluate P(t)* stage, via the inverse of some appropriate cluster validity index, and the fitness value is stored in the oracle. Afterwards, *Store B(t)* is called, the fitness of the solution is compared to the stored best solution, and the better one is retained. Following evaluation, two additional quantum variants of classical genetic operators are employed. The first is a quantum two-point crossover operator [9], which after each iteration, according to a specified probability selects two individuals randomly and applies two-point crossover to subsequent generations at random crossover points. The second operator is quantum mutation inversion [9], and this operator randomly selects two individuals from the population and inverts the probability coefficients *α* and *β* of a randomly chosen qubit. The idea behind applying these quantum genetic operators is to attempt to further increase diversity and discover a better solution.

This body of work will focus upon applying the qubit model to the K-Means, Fuzzy C-Means, New Weighted Fuzzy C-Means, and Artificial Bee Colony clustering algorithms. These particular clustering algorithms were chosen both for their simplicity, and their adeptness for accepting randomly generated quantum centroid values as input. As an alternative, rather than assigning quantum random 8 bit values as centroids, pixel values may be assigned as centroids, while assigning the quantum values to the pixel array as random index values. Furthermore, by assessing the quantum model for all four of these clustering algorithms individually, and then comparing the results, a more comprehensive overall assessment is possible.

3.1. Quantum K-Means Algorithm

The Quantum K-Means (QKM) algorithm begins by initializing a population of quantum oracles, and then a call is made to *Make P(t)*. Following *Make P(t)*, the decoded decimal values are provided as initial centroid input to K-Means, and hard partitioning will then ensue until a stopping criterion has been determined. Afterwards, the cluster fitness is evaluated, via the Davies-Bouldin (DB) Index method [7], and the fitness value is stored in the oracle. The fitness function utilized in QKM is in noted in equation (15).

Afterwards, the fitness of the oracle is compared to the stored best solution as of yet, and the oracle that posses a superior fitness value is stored as the new best solution. In order to guide the current solution toward the best stored solution, and hence bring convergence to fruition, subsequent to oracle population fitness evaluation, the quantum rotation gate in equation (36) is applied to the probability amplitudes *α* and *β* respectively for each qubit in each string. Following quantum gate rotation, quantum crossover and mutation operators are applied to individuals chosen randomly. The previously described process continues for the specified number of iterations.

Step 1: *Initialize Q(t)* – (population initialization)
Step 2: *Make P(t)* – (Observe, encode, decode)
Step 3: *K-Means Clustering*
Step 4: *Evaluate P(t)* – (XB Index)
Step 5: *Store B(t)* – (Store best solution)
Step 6: Repeat Steps 3-5 for the entire population
Step 7: *Quantum Rotation Gate*
Step 8: *Quantum Crossover*
Step 9: *Quantum Mutation Inversion*
Step 10: *Make P(t)* – (Observe, encode, decode)
Step 11: Repeat Steps 3-10 for every iteration

3.2. Quantum Fuzzy C-Means Algorithm

The Quantum Fuzzy C-Means (QFCM) algorithm begins by initializing a population of quantum oracles, and then a call is made to *Make P(t)*. Following *Make P(t)*, the decoded decimal values are utilized as initial membership weights and normalized. Following membership normalization, soft partitioning will then ensue until a stopping criterion has been determined. Once soft partitioning has completed, solution fitness is evaluated, via the Xie and Beni's (XB) Index method [17], and the fitness value is stored. The fitness function utilized in QFCM is noted in equation (17).

Afterwards, the fitness of the oracle is compared to the stored best solution as of yet, and the oracle that possesses a superior fitness value is stored as the new best solution. In order to guide the current solution toward the best stored solution, and hence bring convergence to fruition, subsequent to oracle population fitness evaluation, the quantum rotation gate in equation (36) is applied to the probability amplitudes *α* and *β* respectively for each qubit in each string. Following quantum gate rotation, quantum crossover and mutation operators are applied to individuals chosen randomly. The previously described process continues for the specified number of iterations.

The following describes the steps of the QFCM algorithm:

Step 1: *Initialize Q(t)* – (population initialization)
Step 2: *Make P(t)* – (Observe, encode, decode)
Step 3: *Fuzzy C-Means Clustering*
Step 4: *Evaluate P(t)* – (XB Index)
Step 5: *Store B(t)* – (Store best solution)
Step 6: Repeat Steps 3-5 for the entire population
Step 7: *Quantum Rotation Gate*
Step 8: *Quantum Crossover*
Step 9: *Quantum Mutation Inversion*
Step 10: *Make P(t)* – (Observe, encode, decode)
Step 11: Repeat Steps 3-10 for every iteration

3.3. Quantum New Weighted Fuzzy C-Means Algorithm

The Quantum New Weighted Fuzzy C-Means (QNW-FCM) algorithm begins by initializing a population of quantum oracles, and then a call is made to *Make P(t)*. Following *Make P(t)*, the decoded decimal values are utilized as initial membership weights and normalized. Following membership normalization, soft partitioning will then ensue until a stopping criterion has been determined. Once soft partitioning has completed, solution fitness is evaluated, via the Xie and Beni's (XB) Index method [17], and the fitness value is stored. The fitness function utilized in QNW-FCM is noted in equation (17).

Afterwards, the fitness of the oracle is compared to the stored best solution as of yet, and the oracle that possesses a superior fitness value is stored as the new best solution. In order to guide the current solution toward the best stored solution, and hence bring convergence to fruition, subsequent to oracle population fitness evaluation, the quantum rotation gate in equation (36) is applied to the probability amplitudes α and β respectively for each qubit in each string. Following quantum gate rotation, quantum crossover and mutation operators are applied to individuals chosen randomly. The previously described process continues for the specified number of iterations.

The following describes the steps of the QNW-FCM algorithm:

Step 1: *Initialize Q(t)* – (population initialization)
Step 2: *Make P(t)* – (Observe, encode, decode)
Step 3: *New Weighted Fuzzy C-Means Clustering*
Step 4: *Evaluate P(t)* – (XB Index)
Step 5: *Store B(t)* – (Store best solution)
Step 6: Repeat Steps 3-5 for the entire population
Step 7: *Quantum Rotation Gate*
Step 8: *Quantum Crossover*
Step 9: *Quantum Mutation Inversion*
Step 10: *Make P(t)* – (Observe, encode, decode)
Step 11: Repeat Steps 3-10 for every iteration

3.4. Quantum Artificial Bee Colony Algorithm

The Quantum Artificial Bee Colony (QABC) algorithm begins by initializing a population of quantum oracles, and then a call is made to *Make P(t)*. Following *Make P(t)*, the decoded decimal values are utilized as initial food sources and random parameter values. The ABC algorithm is then executed, with food source fitness being evaluated during employed and onlooker bee phases once each cycle , via the Davies-Bouldin (DB) Index method [7], and the fitness value is stored in the oracle. The fitness function utilized in QABC is in noted in equation (15).

Afterwards, the fitness of the oracle is compared to the stored best solution as of yet, and the oracle that posses a superior fitness value is stored as the new best solution. In order to guide the current solution toward the best stored solution, and hence bring convergence to fruition, subsequent to oracle population fitness evaluation, the quantum rotation gate in equation (36) is applied to the probability amplitudes α and β respectively for each qubit in each string. Following quantum gate rotation, quantum crossover and mutation operators are applied to individuals chosen randomly. The previously described process continues for the specified number of iterations.

The following describes the steps of the QABC algorithm:

Step 1: *Initialize Q(t)* – (population initialization)
Step 2: *Make P(t)* – (Observe, encode, decode)
Step 3: *Artificial Bee Colony Clustering*
Step 4: *Evaluate P(t)* – (DB Index)
Step 5: *Store B(t)* – (Store best solution)
Step 6: *Quantum Rotation Gate* – (unitary matrix)
Step 7: *Quantum Crossover* - (2 point, swap qubits)
Step 8: *Quantum Mutation Inversion* – (flip α and β)
Step 9: *Make P(t)*– (Observe, encode, decode)
Step 10: Repeat Steps 3-9 for every iteration

4. Experimental results and analysis

The four proposed algorithms were implemented in java and conducted 30 times each on a satellite image, a hazy landscape image, a raptor jet fighter image, an image of the earth from the surface of the moon, and an image of a desert arch and a tree. In addition, QKM and KM only were conducted on the Iris data set from the UCI Machine Learning Repository. Moreover, due to the fact that the fitness value obtained via the Davies Bouldin index is higher for Iris solutions that possess less than the optimal number of clusters, parameters were controlled according to data set properties for ease of assessment purposes. For the satellite and hazy landscape images, overall accuracy assessments were performed via ground truth, with the best segmentation results displayed in Figures 1 and 2, and the best, worst and mean segmentation accuracy listed in Tables 1 – 4 below. The raptor, the earth from the moon, and the desert arch with a tree images were accessed via visual inspection, with the best segmentation results displayed in Figures 3 – 5 below. For the Iris data set, a cross-validated single fold accuracy assessment was performed over a varying number of iterations (50, 100, and 200 iterations), with the best, worst and mean segmentation accuracy listed in Table 5 below. The following initialization methods were utilized by each algorithm:

Method 1 - Random number in 8 bit range
Method 2 - ((Random number in 8 bit range / Total Cluster No) * (Current Cluster No)) + offset

With regard to initialization Method 2, "Current Cluster No." represents the index value of the current cluster in a loop that iterates from 0 to the total number of clusters (K) – 1. Also, for Hazy Landscape image, the offset = + 20, and for all other images the offset = -1.

Table 1: Comparison of 30 experiments for satellite image using method 1 and 3 clusters

Clustering Algorithm	Overall Accuracy Best	Worst	Mean	Variance
KM	96.55	82.29	86.80	5.72×10^1
QKM	97.63	83.91	91.63	2.74×10^1
FCM	68.19	68.19	68.19	0.00×10^0
QFCM	68.19	68.19	68.19	0.00×10^0
NW-FCM	61.17	61.17	61.17	0.00×10^0
QNW-FCM	61.17	61.17	61.17	0.00×10^0
ABC	85.93	50.64	77.96	3.70×10^1
QABC	87.84	73.94	81.07	1.89×10^1

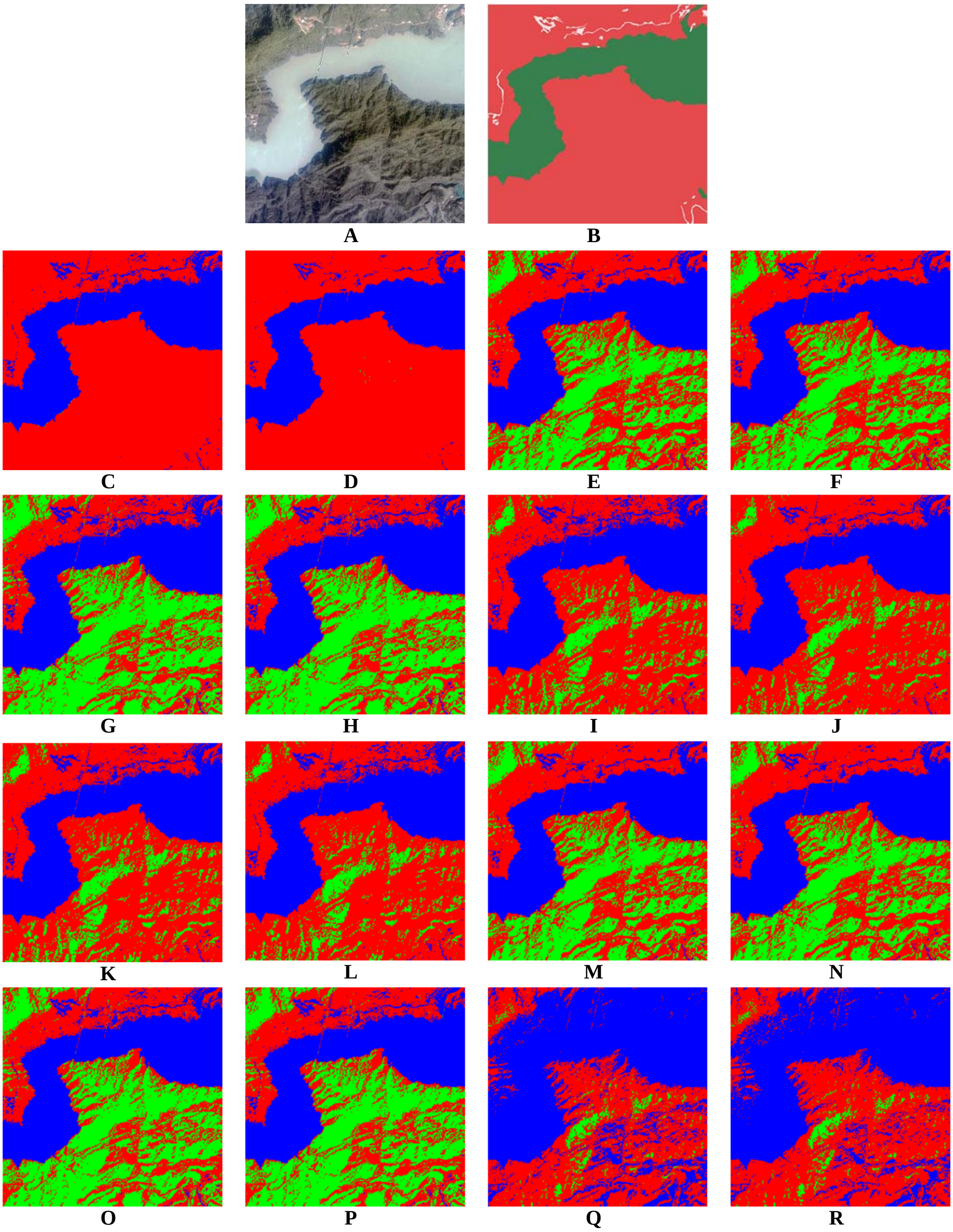

Figure 1. Satellite image: A. Original, B. Ground truth. Best segmentation results utilizing initialization method 1 and 3 clusters: C. KM, D. QKM, E. FCM, F. QFCM, G. NW-FCM, H. QNW-FCM, I. ABC, J. QABC. Best segmentation results using method 2 and 3 clusters: K. KM, L. QKM, M. FCM, N. QFCM, O. NW-FCM, P. QNW-FCM, Q. ABC, R. QABC.

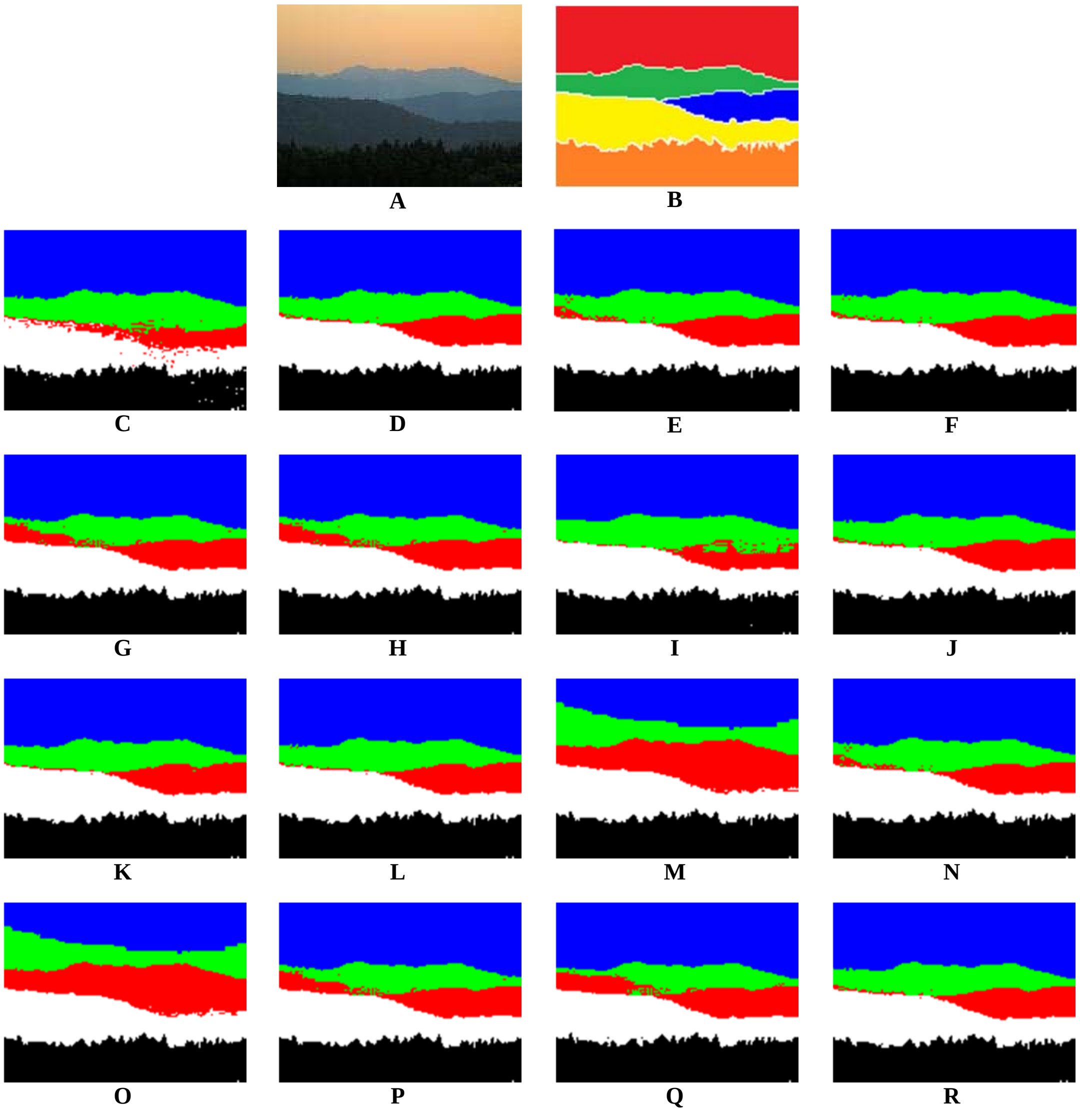

Figure 2. Hazy landscape image: A. Original, B. Ground truth. Best segmentation results utilizing initialization method 1 and 5 clusters: C. KM, D. QKM, E. FCM, F. QFCM, G. NW-FCM, H. QNW-FCM, I. ABC, J. QABC. Best segmentation results using method 2 and 5 clusters: K. KM, L. QKM, M. FCM, N. QFCM, O. NW-FCM, P. QNW-FCM, Q. ABC, R. QABC.

Table 2: Comparison of 30 experiments for satellite image using method 2 & 3 clusters

Clustering Algorithm	Overall Accuracy			
	Best	Worst	Mean	Variance
KM	84.84	81.46	83.23	1.54×10^{0}
QKM	85.71	84.09	84.92	2.97×10^{-1}
FCM	68.19	68.19	68.19	0.00×10^{0}
QFCM	68.19	68.19	68.19	0.00×10^{0}
NW-FCM	61.17	61.17	61.17	0.00×10^{0}
QNW-FCM	61.17	61.17	61.17	0.00×10^{0}
ABC	68.30	56.92	62.16	2.19×10^{1}
QABC	73.25	60.67	66.82	1.99×10^{1}

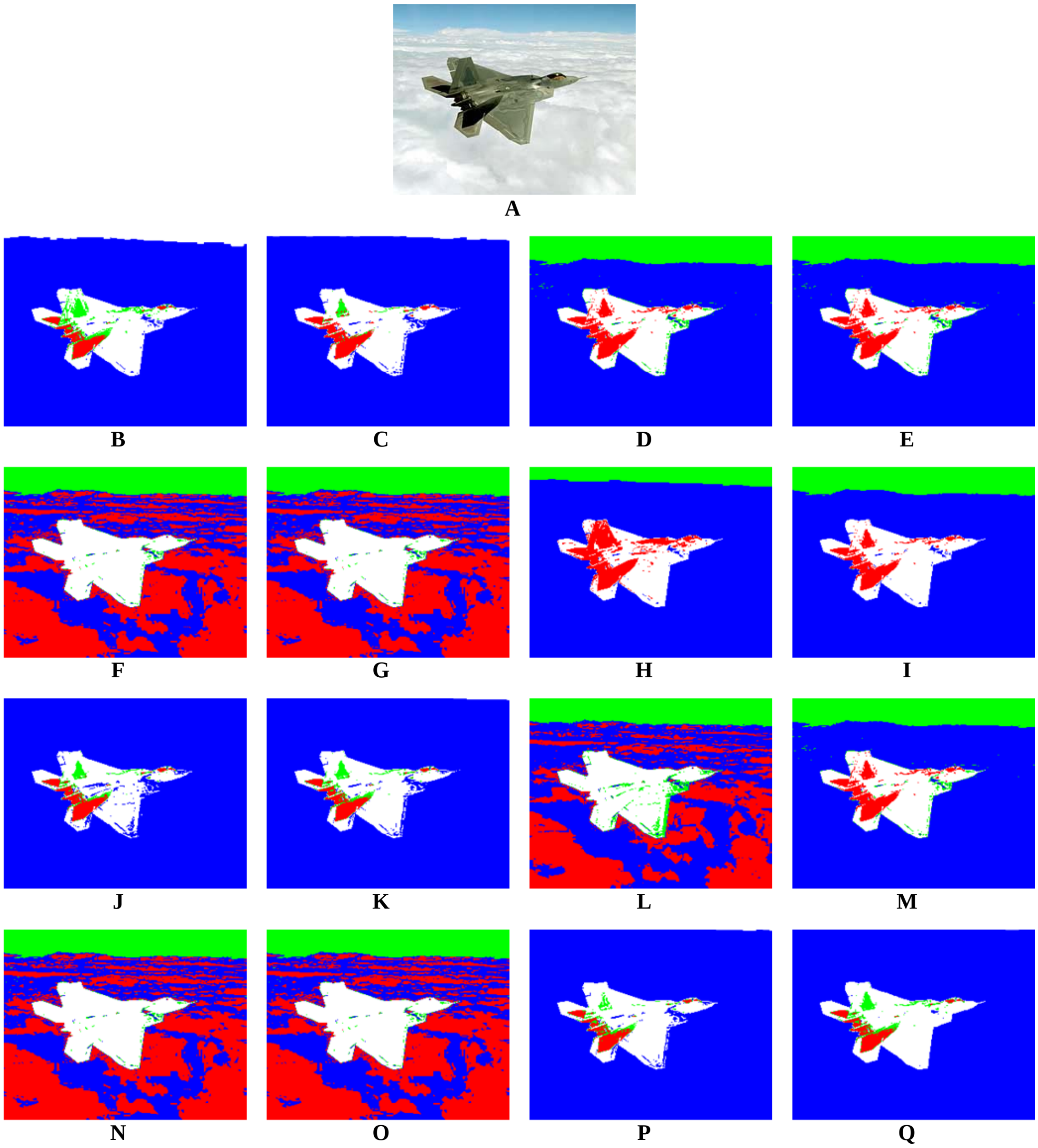

Figure 3. Raptor image: A. Original. Best segmentation results using method 1 and 4 clusters: B. KM, C. QKM, D. FCM, E. QFCM, F. NW-FCM, G. QNW-FCM, H. ABC, I. QABC. Best segmentation results using method 2 and 4 clusters: J. KM, K. QKM, L. FCM, M. QFCM, N. NW-FCM, O. QNW-FCM, P. ABC, Q. QABC.

As can be ascertained from analyzing the data set segmentation results in Figures 1 - 5 visually or Tables 1-5 statistically, it is apparent that the various quantum modeled algorithms exhibit improved accuracy, when compared to their classical counterparts. With regard to accuracy, the greatest differences were observed while utilizing initialization method 2. This initialization method often trapped the classical algorithms into a subpar local solution, and restricted solution diversity substantially. To the contrary, the quantum variants were often able to escape this local trap and find a global solution, thereby exhibiting greater diversity. This is more the case with QFCM, and less so with QNW-FCM, with QNW-FCM escaping in approximately one in three data sets that QFCM could.

This is primarily due to the increased degree of stability NW-FCM possesses relative to FCM, and QNW-FCM exhibited less susceptibility to the effects of the quantum state machine than QFCM in general. Moreover, due to the computational cost associated with QNW-FCM and NW-FCM, due to the size of some of the data sets, for the sake of expediency, pixels were skipped when measuring the Euclidean distance between the pixels and the means, which adversely affected the segmentation quality.

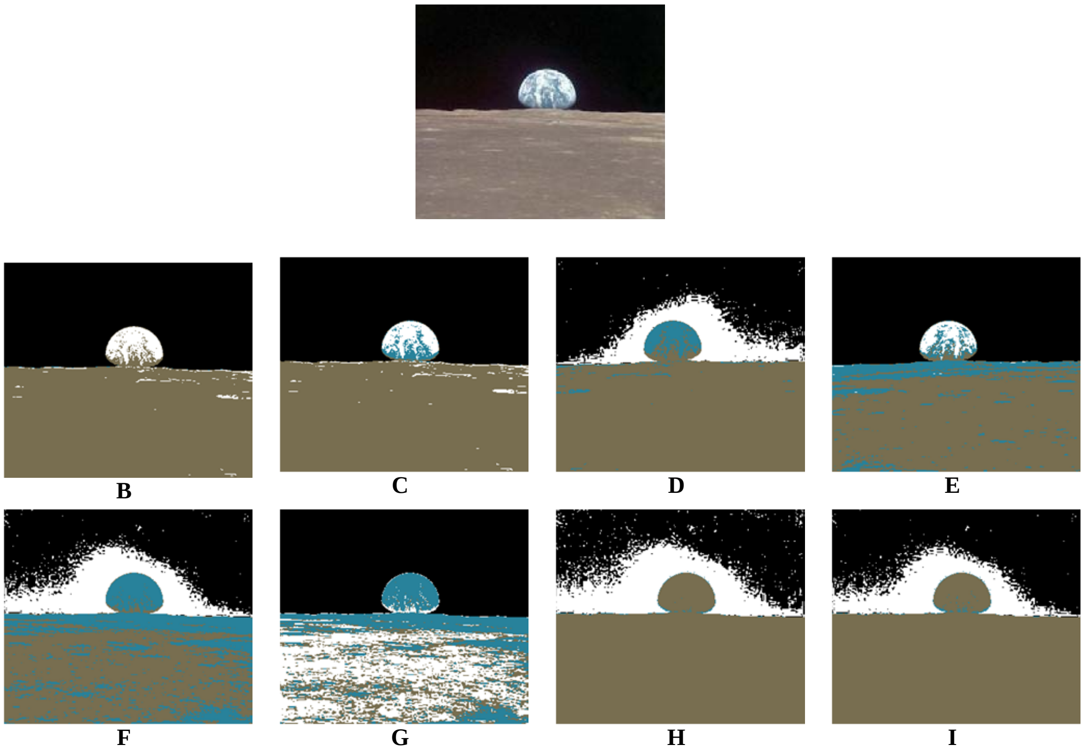

Figure 4. Earth from moon image: A. Original. Best segmentation results using method 2 and 4 clusters: B. KM, C. QKM, D. FCM, E. QFCM, F. NW-FCM, G. QNW-FCM, H. ABC, I. QABC.

]=

Table 3: Comparison of 30 experiments for hazy landscape image using method 1 and 5 clusters

Clustering Algorithm	Overall Accuracy			
	Best	Worst	Mean	Variance
KM	91.72	70.36	83.14	5.36 x 10^{1}
QKM	97.52	70.91	90.17	1.07 x 10^{2}
FCM	96.89	96.47	96.83	4.64 x 10^{-4}
QFCM	97.32	96.81	96.87	2.48 x 10^{-2}
NW-FCM	95.15	95.11	95.13	3.47 x 10^{-4}
QNW-FCM	95.16	95.11	95.13	4.33 x 10^{-4}
ABC	91.48	68.86	79.13	5.65 x 10^{1}
QABC	97.29	68.53	84.59	5.69 x 10^{1}

Table 4: Comparison of 30 experiments for hazy landscape image using method 2 and 5 clusters

Clustering Algorithm	Overall Accuracy			
	Best	Worst	Mean	Variance
KM	97.39	86.39	92.76	1.47 x10^{1}
QKM	97.61	86.99	93.30	1.66 x 10^{1}
FCM	78.95	78.75	78.84	7.94 x 10^{-3}
QFCM	96.89	96.75	96.82	1.42 x 10^{-3}
NW-FCM	78.82	78.71	78.77	3.03 x 10^{-3}
QNW-FCM	95.13	95.07	95.10	8.47 x 10^{-4}
ABC	94.01	75.26	84.40	4.22 x 10^{1}
QABC	97.36	79.91	90.02	4.04 x 10^{1}

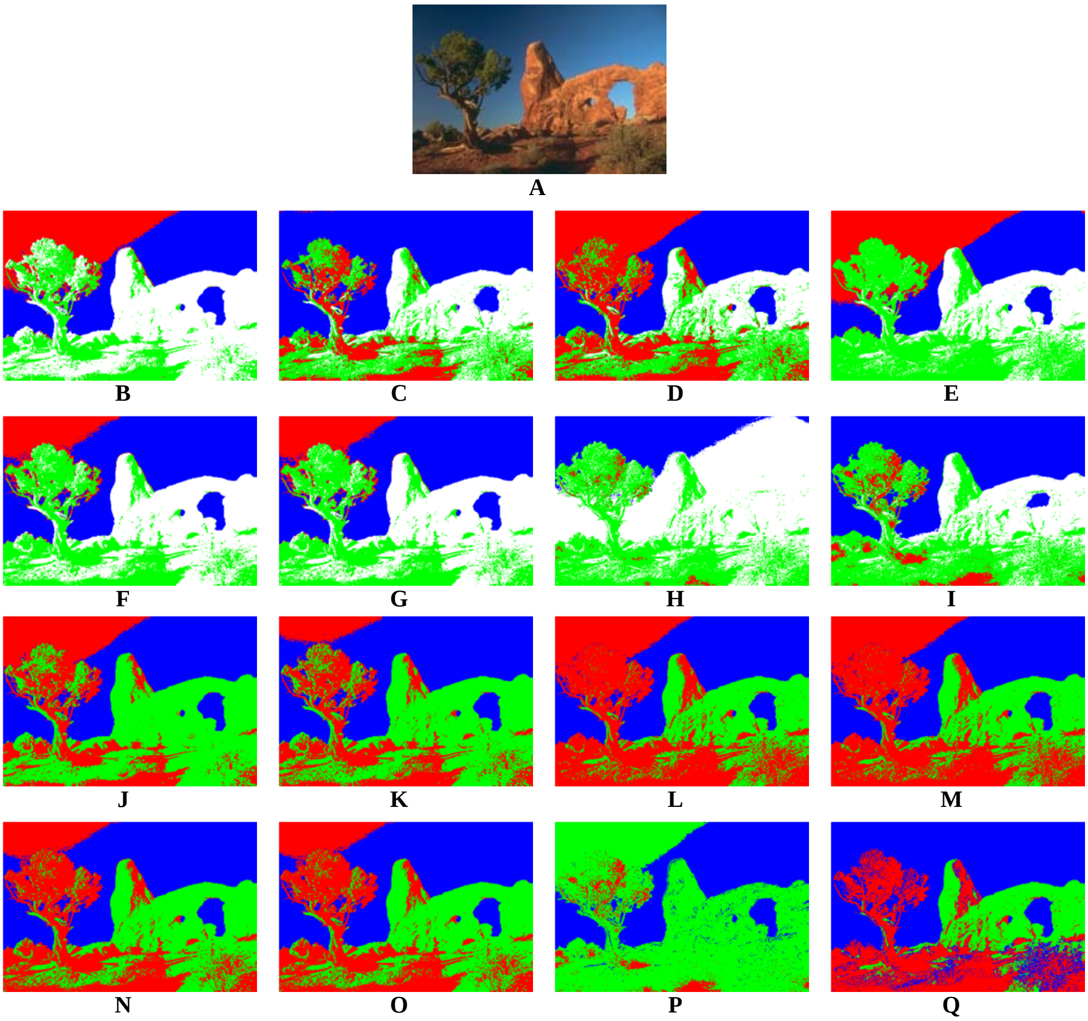

Figure 5. Desert arch with tree image: A. Original. Best segmentation results using method 2 and 4 clusters: B. KM, C. QKM, D. FCM, E. QFCM, F. NW-FCM, G. QNW-FCM, H. ABC, I. QABC. Best segmentation results using method 2 and 3 clusters: J. KM, K. QKM, L. FCM, M. QFCM, N. NW-FCM, O. QNW-FCM, P. ABC, Q. QABC.

Table 5: Comparison of 30 experiments for Iris data using method 1, 3 clusters, at various iterations

Clustering Algorithm	Overall Accuracy Best	 Worst	 Mean	 Variance
50 iterations				
KM	98.66	93.33	96.71	$5.19 \text{ x } 10^{0}$
QKM	100.00	95.33	97.73	$4.58 \text{ x } 10^{0}$
100 iterations				
KM	100.00	94.66	97.80	$3.02 \text{ x } 10^{0}$
QKM	100.00	97.33	99.02	$5.00 \text{ x } 10^{-1}$
200 iterations				
KM	100.00	96.66	98.46	$1.76 \text{ x } 10^{0}$
QKM	100.00	98.00	99.68	$1.30 \text{ x } 10^{-1}$

Moreover, while utilizing initialization method 1, the greatest difference in accuracy was between QABC and ABC. This was primarily due to the simplicity of the ABC algorithm as adapted to image segmentation, which did not utilize a histogram as originally proposed [14], but simply assigned pixels to random centroids as food sources. Also, of note is the fact that the initialization method 2 offset utilized in all algorithms for each data sets was (-1) except for QABC and ABC on the hazy landscape image only, which was assigned an offset of (+20) to account for the shortcomings of the image segmentation portion of the algorithm. Furthermore, with these algorithmic shortcomings in mind, the quantum variant of this algorithm is highlighted by being able to achieve substantially more accurate segmentations, due primarily to the inherent quantum properties of simulated superposition and the qubits being operated upon by the unitary quantum rotation gate. These effects are more apparent when comparing QABC and ABC utilizing initialization method 1, than the other clustering algorithms.

The convergence speed and precision the algorithms possess are primarily due to the effects of the gate operating on the qubits, and algorithmic convergence speed varies, and is determined by the rotation gate angle value applied to the qubits. Smaller values of θ allow for finer adjustments in the probability amplitude coefficients of each qubit, while larger values increase the convergence speed. The overall effect is analogous to the convergence properties exhibited by classical genetic and other optimization algorithms. Therefore, due these inherent properties, the quantum model could be applied to applications that would normally utilize these types of algorithms. The difference being, that in the case of the quantum variant, the probability of algorithmic convergence is absolute. This would be the case with either before-mentioned model, be it the qubit model, or the various potential well models. Regardless of model selection, convergence would be guaranteed.

Even with this guaranteed algorithmic convergence, the most significant limitation factor is the quality of the fitness function. This was the case with the Iris data set, where during the evaluation of k-means clustering, the Davies Bouldin index alone was not sufficient to determine the most accurate solution. Solutions with poor accuracy possessed either very high or very low fitness, while very accurate solutions possessed a median fitness value. Therefore, the algorithm would converge to the poor solution with very high fitness. Hence, in order to avoid this, the choice was made to apply parameter controls based upon the symmetrical properties of known class labels. This way, an acceptable convergence target was achievable, and a proper assessment could ensue. Moreover, the ability of the qubit modeled clustering algorithms to discover fitter solutions compared to their classical counterparts seems to be one of the most noted characteristics in the literature [5]. Also, of note is the fact that by analyzing Table 4.5, the effects of algorithmic convergence on accuracy are quite apparent. The difference between the mean accuracy of QKM and KM was 1.02 % for 50 iterations, and 1.22% for both 100 and 200 iterations. Thus, from this we can determine that both the algorithm had converged by 100 iterations, and going from partial convergence at 50 iterations to full convergence at 100 iterations, increased accuracy by 0.22%. Therefore, algorithmic convergence brought to fruition by the rotation gate contributed to solution enhancement. In addition, the quantum crossover and mutation inversion operators helped vary some of the converged centroid values for selected members of the population, thereby effectively enhancing search space exploration. Furthermore, the costs associated with the quantum operators and Turing machine collectively result in an increase in algorithmic execution time of approximately 1.25 times that of the classical variants on average.

In order to better understand the convergence properties associated with quantum modeled algorithms, let us examine the evolution of stored solutions written to the QKM directory during *Store B(t)* for the hazy landscape image:

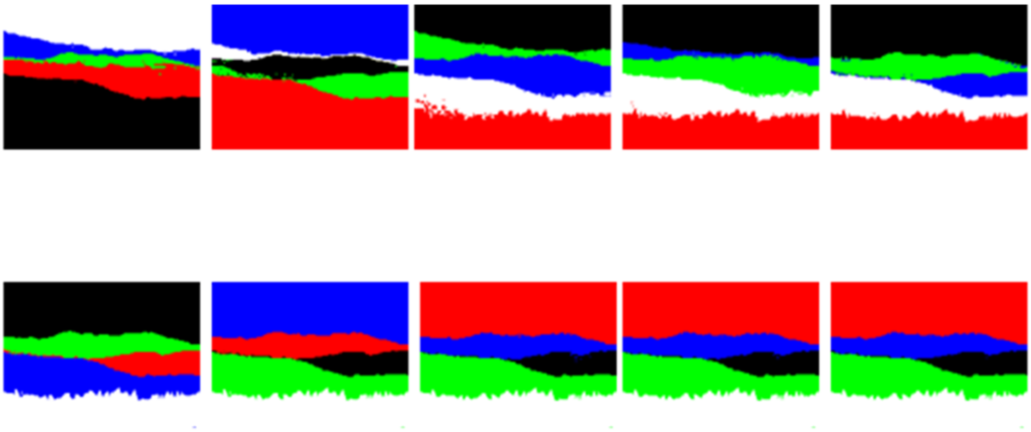

Figure 6. Evolution of stored solutions for hazy landscape image

Moreover, the following graph emphasizes the relationship between solution accuracy and solution fitness for the same series of stored solutions during a single run of the hazy landscape image:

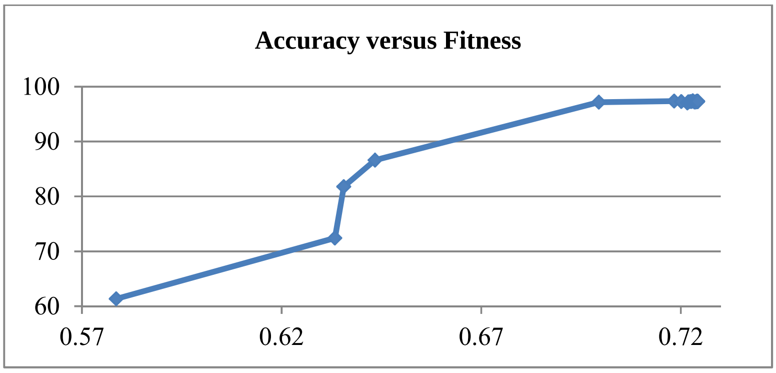

Figure 7. Accuracy versus fitness graph for hazy landscape image

While referring to the stored solution segmentation results in Figure 6 and the graph in Figure 7, notice that for the third stored solution there is a large jump in accuracy from ~72.5% to ~82%, and a slight jump in fitness. This corresponds with the convergence of a single centroid value (cluster 5, band 1) between the second and third stored solutions. Note that the gate operates on the qubits for the first time *after* the first iteration, so the third solution is the first solution that has been operated upon. The first two solutions are random, with an equal probability of all possible solutions occurring, since the gate has not had a chance to operate upon the qubits, so no convergence has began to occur. Following this, the fitness and accuracy for the fourth solution both continue to climb. Similarly, this corresponds with a single converged centroid value. Now note the jump in both accuracy and fitness between the fourth and fifth solutions. The difference in both fitness and accuracy between these two solutions represents the largest in the graph, and corresponds to five converged centroid values. Moreover, between the fifth and sixth stored solutions both fitness and accuracy increase again, though the difference in accuracy is much more modest than between the previous two solutions. From this point forward, the fitness continues to rise but the accuracy fluctuates up and down slightly, with the twelfth stored solution measuring as the most accurate at ~97.43%.

In order to understand how quantum achieves more diverse solutions, analyzing a contiguous sequence of solution fitness values is useful. The following graph highlights the effect of the quantum Turing machine on measured fitness values on the desert arch with tree image, while utilizing initialization method 2 and 3 clusters:

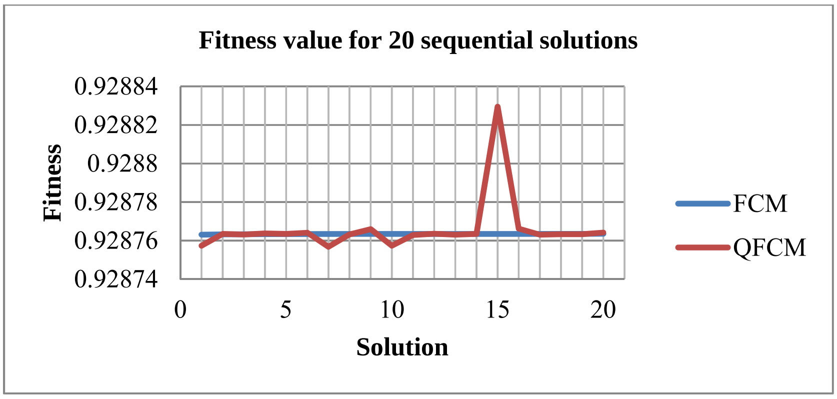

Figure 8. Fitness versus solution graph for desert arch with tree image and 3 clusters

Likewise, the following graph highlights the effect of the quantum Turing machine on measured fitness values on the desert arch with tree image while utilizing initialization method 2 and 4 clusters:

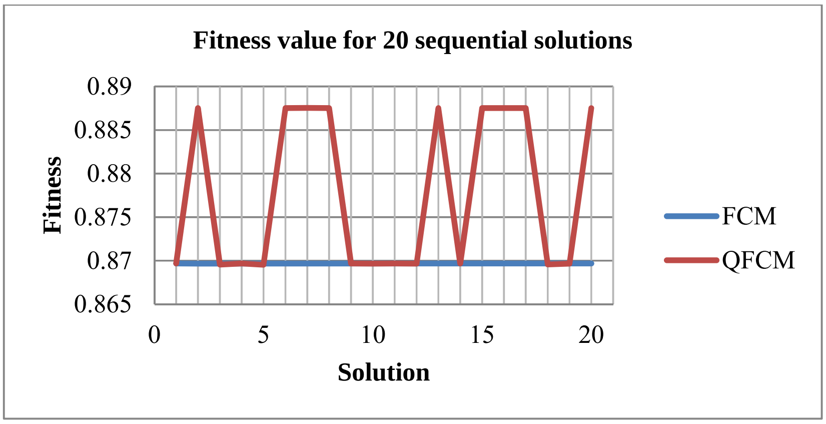

Figure 9. Fitness versus solution graph for desert arch with tree image and 4 clusters

Of note, is the fact that the same phenomena is occurring in both graphs. Furthermore, the desert arch with tree image segmentation results with 3 clusters reflects no detectable difference in overall accuracy, while results with 4 clusters reflect a very substantial increase in overall accuracy.

5. Conclusions

In order to assess the effectiveness of the quantum bit model, four quantum-based clustering algorithms where proposed. Experimental results yielded improved accuracy when compared to each respective classical variant. In addition, the algorithms exhibited a controllable rate of convergence, with eventual algorithmic convergence guaranteed. Moreover, promise was exhibited with regard to future endeavors involving applying quantum computing models to clustering algorithms, as well as any application that would normally utilize a classical genetic or other optimization algorithm.

Applicability far exceeds image segmentation, and could be extended to a multitude of other applications that require clustering data sets and/or can benefit from the properties inherent to quantum mechanical models. In order to maximize the effectiveness of the quantum model as applied to data clustering, the capabilities of the fitness function for solution evaluation must be optimized, so as to allow the qubits to converge to a more accurate solution. Since algorithmic convergence is guaranteed, properly evaluating what constitutes a "fit" solution is key to maximizing the potential of the model. As a result of these findings, future work will focus upon applying quantum modeling to other clustering or optimization algorithms in order to reap the benefits of simulating quantum mechanical properties, and enhancing the fitness function capabilities which directly guide algorithmic convergence.

In particular, these quantum modeled algorithms have proven that enhanced solutions are obtainable via simulated superposition, and that the quantum Turing machine serving as a standalone random number generator, absent of any operators, can exceed the random capability of the Java Random class, with a little additional computational cost. This fact is evident not only from the accuracy assessment of solutions, but also from the increased variance in fitness values, such as the graphs in Figures 8 and 9 illustrate. With regard to image segmentation, it can be said that simulating the properties of superposition can produce more diverse solutions. Thereby, increasing the probability that the search space will be maximized, and also that the global optima will be discovered. It can further be stated that the quantum model can be applied in a practical manner, especially if implemented absent of any operators, since the complications inherent to employing proper fitness functions would be alleviated. Therefore, it can be concluded that quantum modeled algorithms are viable as evidenced in this body of work, via empirical experimentation and subsequent data analysis.

REFERENCES

[1] Nielsen, M. and Chuang, I. *Quantum Computation and Quantum Information.* New York. Cambridge University Press, 2010.

[2] Chou, Y., Yang, Y. and Chiu, C. Classical and quantum-inspired Tabu search for solving 0/1 knapsack problem. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp.1364-1369, 2011.

[3] Chang, J., An, F. and Su, P. A quantum-PSO algorithm for no-wait flow shop scheduling problem. In *Control and Decision Conference (CCDC)*, pp.179-184, 2010.

[4] Mikki, S. M. and Kishk, A. A., Quantum Particle Swarm Optimization for Electromagnetics. In *IEEE Transactions on Antennas and Propagation*, vol.54, no.10, pp.2764-2775, Oct. 2006.

[5] Xiao, J., Yan, Y., Zhang, J. and Tang, Y. *A quantum-inspired genetic algorithm for k-means clustering*. School of Computer Science, South China Normal University, Guangzhou 510631, PR China. Key Lab of Machine Intelligence and Sensor Network (Sun Yat-sen University), Ministry of Education, Guangzhou 510006, PR China, 2009.

[6] Liu, W., Chen, H., Yan, Q., Liu, Z., Xu, J. and Yu, Zheng, A novel quantum-inspired evolutionary algorithm based on variable angle-distance rotation. In *IEEE Congress on Evolutionary Computation (CEC)*, pp.1-7, 2010.

[7] Davies, D. L. and Bouldin, D. W. A Cluster Separation Measure, *IEEE Transactions Pattern Analysis and Machine Intelligence*. Vol. PAMI-1, No. 2, 224-227, Apr. 1976.

[8] Spector, L. Evolution of quantum algorithms, In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation (GECCO '10)*. pp 2739-2768. 2010

[9] Mohammed, A.M., Elhefhawy, N.A, El-Sherbiny, M.M. and Hadoud, M.M. Quantum Crossover Based Quantum Genetic Algorithm for Solving Non-linear Programming, In *8th International Conference on Informatics and Systems (INFOS)*, pp.BIO-145-BIO-153, 14-16 May 2012

[10] Yanofsky, N.S., and Mannucci, M.A. *Quantum Computing for Computer Scientists.* New York. Cambridge University Press, 2008.

[11] Agarwal, K. and Srivastava, G. Towards software test data generation using discrete quantum particle swarm optimization. In *Proceedings of the 3rd India software engineering conference (ISEC '10),* p.p. 65-68, 2010.

[12] Hung, C.-C., Kulkarni, S., Kuo, B.-C., A New Weighted Fuzzy C-Means Clustering Algorithm for Remotely Sensed Image Classification, In *IEEE Journal of Selected Topics in Signal Processing,,* vol.5, no.3, pp.543-553, June 2011.

[13] Li, C.-H., Huang, W.-C., Kuo, B.-C and Hung, C.-C., A Novel Fuzzy Weighted CMeans Method for Image Classification, In *International Journal of Fuzzy Systems*, Vol. 10, No. 3, pp. 168 – 173, September 2008.

[14] Hancer, E., Ozturk, C., and Karaboga, D., Artificial Bee Colony based image clustering method, In *IEEE Congress on Evolutionary Computation (CEC),* pp. 1-5, 10-15 June 2012.

[15] Casper, E., Hung, C.-C., Jung, E., Yang, M., A Quantum-Modeled K-Means Clustering Algorithm for Multi-band Image Segmentation. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium (RACS '12)*. pp. 158-163, October 2012.

[16] Li, Y., Shi, H., Gong, M., and Shang, R., Quantum-inspired evolutionary clustering algorithm based on manifold distance. In *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation (GEC '09)*, pp. 871-874, 2009.

[17] Xie , X. L. and Beni, G., “A Validity Measure for Fuzzy Clustering”, *IEEE Transactions Pattern Analysis and Machine Intelligence*. Vol. 13, No. 8, 841-847, Aug. 1991.

[18] Nakahara, M. and Tetsou, O. *Quantum Computing From Linear Algebra to Physical Realizations.* Boca Raton. CRC Press, 2008.

[19] Yanofsky, N. and Mannucci, M., *Quantum Computing for Computer Scientists.* New York. Cambridge University Press, 2008.

[20] Karaboga, D. and Gorkemli, B., A quick artificial bee colony -qABC- algorithm for optimization problems, In *International Symposium on Innovations in Intelligent Systems and Applications (INISTA),,* pp. 1-5, 2-4 July 2012.

[21] Karaboga, D. and Basturk, B., On the performance of artificial bee colony (ABC) algorithm, In *Journal of Applied Soft Computing (ASC)* Vol. 8, Issue 1, pp. 687-697, January 2008.

[22] Hancer, E., Ozturk, C., and Karaboga, D., Parameter Tuning for the Artificial Bee Colony Algorithm, In *Proceedings of the 1st International Conference on Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems (ICCCI '09),* pp. 608-619, 2009.

[23] Duda, R. O. and Hart, P. E,. *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[24] Bezdek , J. C., *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York. Plenum Press, 1981.

[25] Rouseeuw, P. J., Kaufman, L., and Trauwaert, E., Fuzzy clustering using scatter matrices, *Comput. Statist. Data Anal*., vol. 23, pp. 135–151, 1996.

[26] Bezdek , J. C. and . Pal, S. K, Fuzzy *Models for Pattern Recognition*. New York: IEEE Press, 1992.

[27] Karaboga, D., An Idea Based On Honey Bee Swarm for Numerical Optimization, Technical Report-TR06,Erciyes University, Engineering Faculty, Computer Engineering Department 2005.

[28] Griffiths, D., *Introduction to Quantum Mechanics.* Upper Saddle River. Pearson, 2005.

[29] Hung, C.-C., Casper, E., Kuo, B.-C., Liu, W., Jung, E., Yang, M., A Quantum-Modeled Fuzzy C-Means Clustering Algorithm for Remotely Sensed Multi-band Image Segmentation. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS ‘13).* In press, 2013.

[30] Hung, C.-C., Casper, E., Kuo, B.-C., Liu, W., Jung, E., Yang, M., A Quantum-Modeled Artificial Bee Colony Clustering Algorithm for Remotely Sensed Multi-band Image Segmentation. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS ‘13).* In press, 2013.

[31] Retrieved on <Oct 2012> from http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.names