Optimization with Quantum Genetic Algorithm

Utpal Roy
Associate Professor
Department of Computer and System Sciences
Visva-Bharati, Santiniketan, India

Sudarshan Roy
Department of Computer and System Sciences
Visva-Bharati, Santiniketan, India

Susmita Nayek
Department of Computer and System Sciences
Visva-Bharati, Santiniketan, India

ABSTRACT

Recent development in quantum technology have shown that quantum computer can provide a dramatic advantage over classical computers for some algorithms. In particular, a polynomial-time algorithm for factoring, a problem which was previously thought to be hard for classical computers, has recently been developed [13]. Similarly, a quantum algorithm searching for unsorted database in square root of time it would take on a classical computer has also been described by Grover [4] - [3]. Both algorithms rely upon the inherent parallelism, superposition and entanglement property of quantum computing to achieve their improvements. Since most problems of real interest for genetic algorithms have a vast search space, it seems appropriate how quantum parallelism can be applied to Genetic Algorithms. In this paper we provide a brief background of quantum computers. We explain *why and how* quantum algorithms provides a fundamental improvements over classical ones for some problems. Further, we present here the Conventional Genetic Algorithm and the quantum approach of Genetical Algorithms(QGA) as well. The benefits and drawbacks of QGA are also analyzed.
Moreover, this paper provides an improved version over the conventional QGA. This improvement originates from the best partial immigration technique applied to the quantum chromosomes. The main objective of the best partial immigration is to consider the string of qubits from the quantum chromosomes having best fitness and transfer the same randomly to the chromosomes of next generation for better mixing. The process is reiterated. To observe the performance the best partial immigration technique we have considered some popular optimization problems and performed the experiment on it. These problems are namely Travelling Salesman Problem(TSP), Binpacking Problem and Vertex Cover Problem. It has been observed that the obtained results outperforms the conventional QGA.
■

1. INTRODUCTION

In Quantum Computers informations are stored in the form of qubits and processed through quantum mechanical principles. Qubits are two-level quantum system represented by superposition of two level basis sets with varying probability.

Quantum computation is contrary to classical computation. Quantum algorithm exploits the laws of Quantum Mechanics in order to perform computation. Further it exploits the properties of superposition, coherence, and the entanglement of different qubits of quantum state to realize quantum computation. Quantum computation is an integration of quantum mechanics applying in the field of algorithm. The ability of parallelism is the essential difference between quantum computation and classical computation. In the probability calculation, the system is not in an invariable state. Conversely, it has a certain probability, and the state probability vector is corresponding to different possible states. Quantum computation is similar to it, the probability amplitudes of quantum states is used in quantum computation.
QC brings a new philosophy to optimization due to underlying concepts. Recently, a practice has been started to realize the beauty of quantum algorithm through classical computers. A growing theoretical and practical interest is devoted to researches on culturing evolutionary computation with the help of quantum computing principle.
Unlike pure quantum computing, Quantum Genetical Algorithm(QGA) does not require the presence of quantum hardware to work with. Many quantum inspired algorithms have been used to solve successfully many interesting problem which are well known as intractable problems. Having many special merits, the study of quantum computing and quantum computers has become very hot in information science, especially after some good quantum computing algorithms, such as Shors factorizing algorithm [13], were explored. such as Grovers [4], [3], quantum search algorithm [4] and
Quantum genetic algorithm (QGA), a new and promising genetic algorithm developed in recent years, is the product of quantum computing theory and genetic algorithm. It is a new evolutionary algorithm of probability [13]. QGA is based on the concepts of quantum computing (qubit, quantum superposition, and quantum entanglement) and quantum theory, such as quantum logic gate. In QGA, qubit encoding is used to represent the chromosome, and evolutionary process is implemented by using quantum logic gate operation on the chromosomes. Now, much attention is paid to QGA because it has the characteristics of strong searching capability, rapid convergence, short computing time, and small population size, a great capability of global optimization and a good robustness. Narayanan and Moore [12] presented quantum-inspired ge-

netic algorithm (QIGA) to solve traveling salesman problem (TSP) successfully, which introduced the concepts and theory of quantum computing into genetic algorithm.
By introducing qubit representation and quantum logic gate operation, Han and Kim [7] and Han et al. [6] presented genetic quantum algorithm (GQA) and parallel quantum genetic algorithm (PQGA) to solve an NP-hard combination optimization problem (knapsack problem). Quantum crossover [12] and quantum mutation were used to improve the performances of GQA [6] in [10]. An improved QGA based on multi-qubit encoding and dynamically adjusting the rotation angle mechanism was presented to separate the blind sources [8]. Li and Jiao [11] proposed a hybrid parallel quantum evolutionary algorithm based on QGA [10] and parallel algorithm. Zhang et al. [16] presented an improved QGA by introducing population catastrophe operation and violent vibration. They also proposed a novel PQGA [15] by using a novel evolutionary strategy. The results [7] - [15] show that QGA and GQA are greatly superior to conventional genetic algorithm (CGA). The evolutionary strategy [7] - [16] is based on prior knowledge of the best solution of optimization problems. For example, in knapsack problem, the criterion of the optimal solution is that the number of 1 should be as big as possible within constraint conditions because more number 1 means bigger fitness of chromosome. However, the criterions of optimal solutions have not been gotten in continuous function optimization problems and in most practical cases. This paper an improved version of quantum genetic algorithm in which qubit phase comparison method is used to update the rotation angles of quantum logic gates, and the strategy of adjusting search grid self-adaptively is employed. Details of the formulation and description of quantum computing and quantum genetic algorithms are available in the references [2, 1]. To give the article a complete shape following few sections have been described briefly and details of which are available in the references [14, 5, 17, 9].

2. QUANTUM VERSUS CLASSICAL

The significance differences between Quantum computer and Classical computer. The most fundamental difference is that the classical computer stores information through classical bits where as quantum computer stores the information with quantum bits known as qubits. The second one is quantum mechanical feature entanglement, which allows a measurement on some qubits that effect the value of the other qubits. A classical bit is one of the two states, 0 or 1. The qubit can stay in the superposition of 0 and 1 states. The interesting fact is that until qubit is measured it is effectively in both states. For example, any computation using this qubits produces as an answer a superposition combining these results of the calculation having been applied to a 0 and to a 1. Thus the calculation for both the 0 and the 1 is performed simultaneously. Interestingly, when the result is measured only one value either 0 or 1 can be seen. This is the collapse of superposition. The probability of measuring the answer corresponding to original 0 bit is α^2 and the probability of measuring the answer corresponding to an original 1 bit is β^2.
Quantum register differs much from classical register. The superposition property of quantum mechanical states enables a quantum register stores exponentially more data than a classical register of the same size. Whereas a classical register with n bits can store one values out of 2^N, a quantum register can be in a superposition of all 2^N values. An operation applied to the quantum register produces one result. An operation applied to the quantum register produces a superposition of all possible results. This is what is meant quantum parallelism. Further, the difficulty is that a measurement of quantum results collapse the superposition so that only one result is measured. Depending upon the function being applied, the superposition of answers may have common features. If these features are ascertained by taking a measurements and then repeating the algorithm, it may be possible to divine the answer we are searching for, probabilistically. Essentially this is how Shor's algorithm works [13]. First you produce a superposition and then apply the desired functions.
The key feature of the quantum computation is to understand the quantum entanglement. Entanglement is the quantum connection among the superimposed states. In previous we have began with a qubit which is the superposition of the 0 and 1 states. we applied a calculation producing an answer which is the superposition of two possible answers. Measuring the superimposed answers collapse that answers into a single classical result. The quantum entanglement produces a quantum connection between the original superimposed qubit and the final superimposed answers, so that when the answers is measured, collapsing the superposition into one answer or the other, the original qubit also collapse into the value, 0 or 1, that produces the measured answers. Given this very brief introduction to superposition and entanglement, we now begin to address our Quantum Genetical Algorithm(QGA).

3. CONVENTIONAL GENETIC ALGORITHM (GA)

GA is an exploration algorithm based on genetic evolution and natural selection. It manipulates a population of individuals called chromosomes. In each time step a new generation is constructed by applying genetic operators between some selected chromosomes. The simplest way for coding chromosomes is to represents them by binary strings. The initial population has to start with random chromosomes uniformly distributed over the entire search space. The next step is the evaluation operation. Its role is to mark the individuals of population. After that individuals will be sorted according to their marks. The selection operation has a goal to elect some numbers of individuals to enable reproduction. The cross over operation can be performed by exchanging some parts of selected individuals in random positions which leads to create a new set of chromosomes replacing the old ones. Before repeating the process it is recommended to perform a mutation to correct stochastic error to avoid genetic drift and to ensure genetic diversity in the population. It consists of changing some random positions of the individuals according to a very small probability.

4. QUANTUM GENETIC ALGORITHM

4.0.1 Algorithm description. The smallest information unit in a two-state quantum computer is called a qubit. A qubit may be in the 0 state, in the 1 state, or in any superposition of the two. The state of a qubit can be represented as:

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

where α and β are the probability amplitudes of the corresponding states and satisfy the following normalization:

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

In eq. 2, $|\alpha|^2$ gives the probability that the qubit will be found in the "0" state and $|\beta|^2$ is in the "1" state. A system with m qubits contains information on 2^m states. The linear superposition of all possible states can be represented as

$$|\psi_i\rangle = \sum_{k=1}^{2^m} C_k |S_k\rangle \quad (3)$$

where C_k specifies the probability amplitude of the corresponding states S_k and subjects to the normalization condition $|C_1|^2 + |C_2|^2 + \cdots + |C_{2^m}|^2 = 1$.

Definition 1 The probability amplitude of one qubit is defined by a pair of complex numbers, (α, β), as

$$[\alpha\beta]^T \quad (4)$$

where α and β satisfy 1 and 2.

Definition 2 The phase of a qubit is defined with an angle ζ as

$$\zeta = \arctan(|\beta|/|\alpha|) \quad (5)$$

and the product $|\alpha| \cdot |\beta|$ is represented by the symbol d, *i.e.*

$$d = |\alpha| \cdot |\beta| \quad (6)$$

where d stands for the quadrant of qubit phase ζ. If d is positive, the phase ζ lies in the first or third quadrant; otherwise the phase ζ lies in the second or forth quadrant.

The probability amplitude of m qubits are represented as

$$p = \left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array} \right] \quad (7)$$

where $|\alpha|^2 + |\beta|^2 = 1$, $i = 1, 2, \cdots, m$. Hence, the phase of the ith qubit is

$$\zeta = \arctan(|\beta_i|/|\alpha_i|) \quad (8)$$

Let the population size be n. The chromosomes are represented with qubits as

$P = (p_1, p_2, ..., p_n)$, where $p_j (j = 1, 2, ..., n)$ is an individual of population shown in eq. 7. The quantum rotation gate $\mathbf{G}$ is chosen as quantum logic gate and it is

$$\mathbf{G} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (9)$$

where θ is the rotation angle of the quantum rotation gate and whose expression is as follows:

$$\theta = k.f(\alpha_i, \beta_i) \quad (10)$$

where k is the coefficient whose value influences the speed of convergence and $f(\alpha_i, \beta_i)$ is a searching direction function of convergence to a global optimum. The individuals are updated with the following expression using quantum rotation gate

$$\mathbf{p_i^{t+1}} = \mathbf{G(t).p_j^t}. \quad (11)$$

4.1 Structure of Quantum Chromosomes

Each biological cell contains a nucleus. In each nucleus chromosomes are present. Each chromosomes is a thread like structure packed with DNA molecule. In a chromosome double-helix DNA molecules are tightly coiled around a protein called histones. The histone protein mainly supports the structure of the chromosome. Chromosomes are visible under microscope during cell division only. In human cell there are 23 pairs of chromosomes. 22 pairs of these chromosomes look same for both males and females. 23rd pair of the chromosome is called the sex chromosome. It differs in male and female. Female has the two copies of X chromosomes where as male has the one copy of X and one copy of Y chromosomes.

A Quantum Chromosomes is simply a string of m qubits that can be stored in a quantum register, on the other way one can say that it forms a quantum register on m bits.

4.2 Initializing the Population

The easiest way to create the initial population is to initialize all the amplitudes of qubits by the value $\frac{1}{\sqrt{2}}$. This means that a chromosome represents all quantum superposition states with equal probability.

4.3 Evaluation of Individuals

The role of this phase is quantifying the quality of each quantum chromosome in the population to make a reproduction. The evaluation is based on an objective function that corresponds to each individual, after measuring, an adaptation value. It permits to mark individuals in the population.

4.4 Quantum Genetic Operations

(1) *Measuring Chromosomes:* In order to exploit effectively superposed states of qubits, we have to observe each qubit. This leads to convert each chromosomes to a binary string of reasonable length.

(2) *Interference:* This operation allows modifying the amplitudes of individuals in order to improve performance. It mainly consists of moving the state of each qubit in the sense of the value of the best solution. This is useful for intensifying the search around the best solution. It can be performed using a unit transformation that allows a rotation whose angle is a function of the amplitudes and the value of the corresponding bit in the reference solution. The value of the rotation angle $\delta\theta$ has to be chosen so that to avoid premature convergence. It is often empirically determined and its direction is determined as a function of the values of amplitudes and the value of the qubit located at the particular position position in the individual being modified.

(3) *Qubit Rotation Gates Strategy:* The rotation of individual's amplitudes is performed by quantum gates. Quantum gates can also be designed in accordance with the present problem. The population Q(t) is updated with a quantum gates rotation of qubits constituting individuals. The rotation strategy adopted is given by the following equation:

$$\begin{bmatrix} \alpha_i^{t+1} \\ \beta_i^{t+1} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \begin{bmatrix} \alpha_i^t \\ \beta_i^t \end{bmatrix} \quad (12)$$

Where $\Delta\theta i$ is the rotation angle of qubit quantum gate i of each quantum chromosome. It is often obtained from a lookup table that ensures convergence.

5. NEW FORMULATION OF QUBITS AND ITS APPLICATION TO QUANTUM GENETIC ALGORITHM(QGA)

The qubit is the smallest form of information in quantum computer and has been expressed in eq.(1) as follows:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{13}$$

where α and β are the probability amplitudes of the corresponding states and satisfy the following normalization:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{14}$$

We express here the qubit as

$$|\phi\rangle = \cos\theta|0\rangle + \sin\theta|1\rangle \tag{15}$$

Here $\cos^2\theta$ and $\sin^2\theta$ indicate the probabilities that qubit will be found in the state "0" and "1" respectively. So a qubit can be encoded as

$$\begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix} \tag{16}$$

A chromosome as a string of m qubit can be expressed as:

$$\begin{bmatrix}\cos(\theta_1) & \cos(\theta_2) & \cdots & \cos(\theta_m) \\ \sin(\theta_1) & \sin(\theta_2) & \cdots & \sin(\theta_m)\end{bmatrix} \tag{17}$$

where $\cos^2\theta$ and $\sin^2\theta$ are the probability amplitudes of the states "0" and "1" respectively. In this formulation $\cos^2\theta$ and $\sin^2\theta$ carries very significant role. The quantum chromosomes are defined in the space $\mathbf{S^m} = [\mathbf{-1}, \mathbf{1}]$. Each quantum chromosomes are defined as two gene chains. The $\cos^2\theta$ and $\sin^2\theta$ are no longer probability amplitudes. Each gene chain contains two gene chains. Obviously (from the expression) every gene chain represents an individuals and each gene consists of two individuals. In this manner we redefine here the quantum chromosome. The present technique of defining the qubit brilliantly avoid the randomicity and blindness in the process of measurement, and save massive translation among quantum coding, binary coding and real number coding. Interestingly, the chromosomes with two individuals could accelerate the optimizing speed.

5.1 Best Partial Immigration Technique

To incorporate some improvements over conventional Quantum Genetic Algorithm(QGA) one may consider a qubit string randomly from the best chromosome of a certain generation. This best chromosome means a quantum chromosome having maximum fitness value. This string of qubits from a best chromosomes is set in into a chromosome having lesser fitness value. Obviously same numbers of qubits are removed from the improved chromosomes. In this way getting some part of the best fit chromosome, lesser fit chromosome can improve its fitness value. We termed this procedure as **Best Partial Immigration Technique (BPI)**. Here, some qubits of the best chromosome are immegrated. This procedure actually helps to improve the entire population towards optimal solution.

Define **quantity** is the percentage of chromosomes upon which the best partial immigration will be applied and **partiality** represents the percentage of qubits in chromosomes which will be immigrated during best partial immigration technique. The value of quantity and partiality depends on the problem. Observe, when qunatity or partiality or both are zero, then it is becoming conventional QGA, becuase nothing are immegrated. In Section 6, we shall discuss convergence rate on changing the value of quantity and partiality.

On the other way some variants of this strategy can be as instead of taking the best chromosome one may choose some better chromosomes because it may happen that in case of the best chromosome, we actually choose *some part of the best chromose that may have insignificant contribution* toward the fitness value, so choosing better chromosome is a better option in that case.

Another variant of the improvement of the technique can be the frequency of applying this procedure in the quantum genetic evaluation. We may apply this procedure after every f generations. Choosing the value of f is problem dependent, because some problem may require frequent immigration or some problem do not. For example, in optimizing a function, keeping f high may yield better result.

It is to be noted that we can apply best immigration after or before quantum gate rotation because this operation includes immigrated qubits in its updation procedure.

6. EXPERIMENT AND DISCUSSION

We have considered here $0/1$ Knapsack Problem, Binpacking Problem, Vertex Cover Problem and Travelling Salesman Problem(TSP) as test cases for our experiment. Table I, Table II, Table III and Table IV represent respectively the results of $0/1$ Knapsack Problem, Binpacking Problem, Vertex Cover Problem and Travelling Salesman Problem(TSP).

6.1 0/1 Knapsack Problem

Given a set $\{a_1, a_2, \ldots, a_n\}$ of n items with specified size, $size(a_i) \in Z^+$ and profit, $profit(a_i) \in Z^+$ for each item $i = 1, 2, \ldots, n$ and a "Knapsack Capacity" B, the problem is to find a subset of given items whose total size is bounded by B and profit is maximized.

For this problem the quantum chromosome formation is very easy and direct. For each item there will be a qubit in the chromosome corresponding this item. If the qubit is measured to be 1 then the corresponding item is choosed, otherwise it would not be choosed. Therefore, each chromosome will represent a solution. In each generation, we find the chromosome (after measurement) that gives maximum profit as well as bounded by "Knapsack Capacity" and this chromosome must have maximum fitness.

Here, we have taken 100 chromosomes and 80 items and "Knapsack Capacity" to be 500.

The Table 1 is showing the convergence nature of Knapsack Problem. In first row quantity and partiality both are zero, hence it is only Quantum Genetic Algorithm. We can see that only 0.341375 fitness is achived in 4293 number of generations. But when we are applying BPI it is converging fast, i.e., taking lesser number of generations to achieve fitness more than the fitness achived in QGA. From the table, it is also clear that if quantity and partiality near to 50% then it is converging fast with slightly lesser fitness than maximum achivable fitness.

In the Fig. 1, we are showing the convergence nature of Knapsack Problem with respect to Genetic Algorithm (GA) and QGA and QGA with BPI.

Table 1. "Convergence nature of Knapsack Problem"

Quantity	Partiality	Maximum Fitness	Generation
0	0	0.341375	4293
0.2	0.2	0.367203	1713
0.2	0.45	0.381207	2226
0.2	0.85	0.383207	3212
0.45	0.2	0.353492	3521
0.45	0.5	0.365210	823
0.45	0.9	0.368918	2674
0.89	0.2	0.356818	1981
0.89	0.45	0.370010	1240
0.89	0.85	0.369299	4080

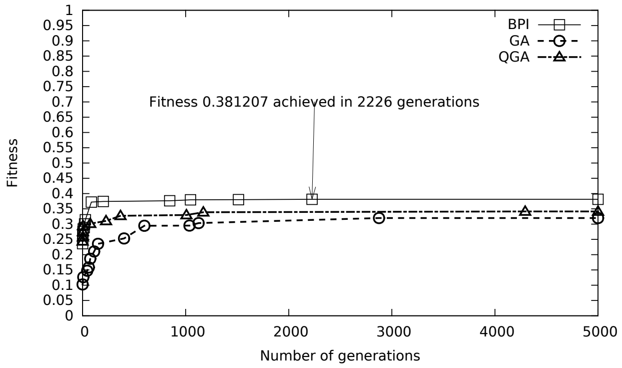

Fig. 1. Convergence nature of Knapsack Problem.

Table 2. "Convergence result of Vertex Cover Problem, Number of vertices = 50"

Quantity	Partiality	Maximum Fitness	Generation
0	0	0.670859	782
0.2	0.2	0.952373	143
0.2	0.5	0.946985	39
0.2	0.9	0.942860	20
0.5	0.2	0.942028	66
0.5	0.45	0.950663	30
0.5	0.9	0.943388	103
0.9	0.2	0.960959	16
0.9	0.45	0.941145	22
0.9	0.9	0.951641	22

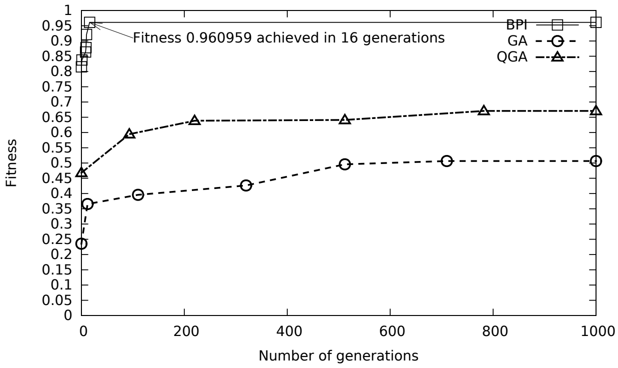

Fig. 2. Convergence nature of Vertex Cover Problem.

6.2 Vertex Cover Problem

Here, we have considered non-weighted version of vertex cover porblem. Given a non-weighted graph $G = (V, E)$, find a minimum size $V' \subseteq V$ such that every edge has at least one end-point in V'.
Chromosome formulation of vertex cover problem is also very easy and direct. For every vertex in the graph there will be a corresponding qubit in the chromosome. If the measure value of the qubit is 1 then the corresponding vertex will be included, otherwise not. Essentially, each chromosome will represent a subset of vertices. In each generation, for each chromosome we have to find that whether the subset represented by the chromosome is vertex cover or not and it size. Obviously, the chromosome representing a vertex cover with minimum size will be of maximum fitnes.
Here, we have considered a graph with 50 vertices and 800 edges.
In Table 2, the first row represent the result of application of QGA. QGA achieved only .670859 fitness in 782 generations. The subsequent rows showing the improved result obtained by applying QGA with BPI. We can see that if quantity is high and partiality is low then the fitness achives higher value and it also converges fast, i.e., in 16 generations.
In Fig. 2, we can see that QGA with BPI converges so fast with respect to GA and QGA.

6.3 Bin Packing Problem

Given a set of n items with sizes $a_1, a_2, \ldots, a_n \in (0, 1]$, find a packing in unit-sized bins that minimizes the number of bins.
In this problem chromosome formulation is easy but geting the feasible solution from the chromosome is not direct. First we generate a permutation of items from the chromosome. Then we pick each item from the permutation one-by-one and pack the item in a bin having enough empty space to pack the item, if there is no such bin with enough empty space then we will open a new bin. Therefore, the chromosome giving minimum number of bins will have maximum fitness.
Here, we have taken 500 items to pack. Unlike, previous two problems, here, number of qubits is not depend on the problem instance. Therefore, one can assume any number of qubits in the chromosomes. The result is obtained by taking 100 chromosomes having 64 qubits in each.
Table 3 shows the convergence result of bin-packing problem. First row shows that 782 generations are required to achive fitness 0.670859 in case of applying QGA. Applying QGA with BPI, we can see that maximum fitness 0.802366 achived when qunatity is low, i.e., 20% and partiality is medium, i.e., 40%. But if keep both parameter low (i.e., quantity and partiality both be equal to 20%) then we achieve fast convergence with very less deviation from the maximum fitness.
In Fig. 3 we can see that QGA with BPI is converging faster than QGA or GA.

6.4 Travelling Salesman Problem

Given a non-negative weighted graph $G = (V, E)$ the problem is to find a minimum weight cycle visting every vertex exactly once. Here, if between two vertices there is no edge then we put an edge with infinite cost. Therefore, the graph becomes complete graph.
The chromosome formation in this problem is easy and not dependent on the input instance but obtaining a solution from a chromosome is indirect. Initially, again like previous problem we generate a permutation of vertices from the chromosome. After that we check whether this permutation is really a TSP tour or not. If it is

Table 3. “Result obtained for Bin Packing problem”

Quantity	Partiality	Maximum Fitness	Generation
0	0	0.670859	782
0.2	0.2	0.797314	10
0.2	0.4	0.802366	118
0.2	0.78	0.736022	169
0.2	0.9	0.765306	50
0.4	0.2	0.736022	59
0.4	0.4	0.781692	51
0.4	0.78	0.776320	69
0.4	0.92	0.717058	66
0.8	0.2	0.770859	60
0.8	0.45	0.776320	62
0.8	0.8	0.759658	35
0.8	0.92	0.742097	73

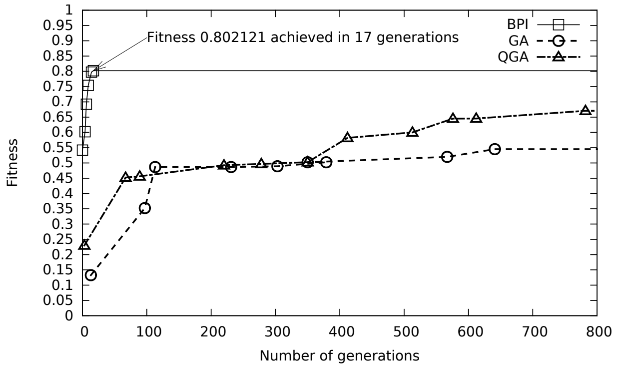

Fig. 3. Convergence nature of Binpacking Problem.

a TSP tour then we find its weight. The chromosome exhibiting minimum weight TSP tour on a generation is the chromosome of maximum fitness. Observe that a permutaion can not be a TSP tour when it contains edge with infinite weight. Now, in a generation if there is no chromosome exhibiting a TSP tour then we consider that chromosome with maximum fitness which has minimum infinite cost edges.

We have taken a graph with 50 vertices and 700 edges. Number of chromosomes is considered to be 100 with 64 qubits in each chromosome.

The first row of the Table 4 is showing maximum fitnes 0.512382 achieved in 6695 generations when we used only QGA. The subsequent rows are showing the effect of applying QGA with BPI. We see that when quantity is medium (i.e., 40%) and partiality is high (i.e., 90%) then we achieve maximum fitness in only 415 generations.

In Fig. 4, we can see the convergin rate of QGA with BPI is faster than conventional QGA or GA.

6.5 Overall Discussion of Results

For a particular problem the formulation of quantum chromosomes is problem dependent. But in general the input instances of each problem are converted into quantum chromosomes. The technique to obtain a (feasible) solution from quantum chromosomes is entirely problem dependent. In many cases obtaining the solution from the quantum chromosome are not so direct. For example, in our study obtaining solution from quantum chromosomes in case of TSP and Binpacking is not so easy. The ultimate result and the rate of convergence of a problem depends upon the formulation of quantum chromosomes and the technique to obtain a (feasible) solution from the quantum chromosomes.

The experimental results for the problems considered here are shown in the corresponding tables (Table 1, 2, 3 and 4) and the comparison of results and the nature of convergence are equally shown in the corresponding Figures(Figs. 1, 2, 3 and 4). We have done each experiment 3 times. Average value of each experiment is shown in the results. The present study represents shows that the nature of convergence in case of Best partial immigration is much better than the GA and the conventional QGA.

Table 4. “Convergence Nature of Travelling Salesman Problem”

Quantity	Partiality	Maximum Fitness	Generation
0.0	0.0	0.512382	6695
0.2	0.2	0.581782	5356
0.45	0.2	0.637627	4485
0.9	0.2	0.690291	315
0.4	0.2	0.669126	3047
0.4	0.5	0.715462	844
0.4	0.9	0.747567	415
0.85	0.2	0.692995	532
0.85	0.44	0.691977	453
0.85	0.85	0.689163	532

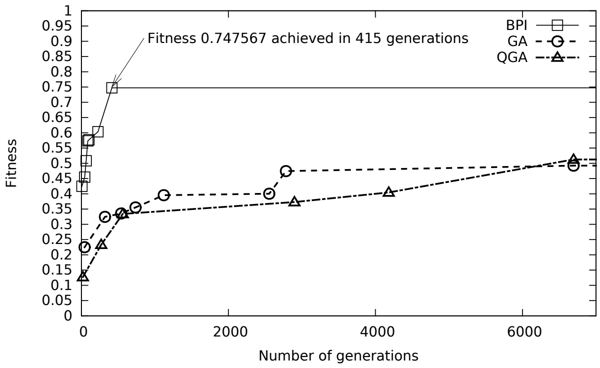

Fig. 4. Convergence nature of Travelling Salesman Problem.

7. DIFFICULTIES WITH QUANTUM GENETIC ALGORITHM(QGA)

There are some potential difficulties with the QGA presented here, even as a theoretical model. Some fitness functions may require ”observing” the superimposed individuals in a quantum mechanical sense. This would destroy the superposition of the individuals and ruin the quantum nature of the algorithm. Clearly it is not possible to consider all fitness functions in this context. However, since mathematical operations can be applied without destroying a superposition, many common fitness functions will be usable. As noted previously a one-to-one fitness function will also negate the advantages of the QGA. Another, more serious difficulty, is that it is not physically possible to exactly copy a superposition. This creates difficulties in both the crossover and reproduction stages of the algorithm. A possible solution for crossover is to use individuals consisting of a linked list rather than an array. Then crossover only requires moving the pointers between two list elements rather than copying array elements. However, without a physical model for our quantum computer it is unclear whether the notion of linked lists is compatible with maintaining a quantum superposition. The difficulty for reproduction is more fundamental. However, while it is not possible to make an exact copy of a superposition, it is possible to make an inexact copy. If the copying errors are small enough they can be considered as a "natural" form of mutation. Thus, those researchers who favor using only mutation may have an advantage in the actual implementation of a QGA. All these difficulties in implementing the QGA naturally flows to the implementing issues of the improved version of QGA, as presented in this paper.

8. CONCLUSION

Quantum computing has the characteristics of strong parallelism. How to combine the theory of quantum computing with the principles of genetic algorithm is still a promising research problem. By introducing qubit chromosome representation and the best partial immigration technique this paper proposes an improved version of QGA based on the concepts and principles of quantum computing. Because qubit can represent basic states and their superposition states simultaneously, the improved version of QGA only needs a small population size instead of degrading its performances. The evolutionary operation that quantum rotation gates are updated using the quantum phase comparison method is simple and valid. In essence it can be stated that the present improved version of QGA has the characteristics of good search capability, rapid convergence, short computing time, and ability to avoid premature convergence effectively, which are also indicated in the experiment results obtained during solving $0/1$ Knapsack Problem, Binpacking Problem, Vertex Cover Problem and TSP. Experimental results obtained so far indicates that the best partial immigration technique in quantum chromosomes make the QGA more powerful in all dimensions.

9. REFERENCES

[1] Comparison of genetic algorithm and quantum genetic algorithm. http://ccis2k.org/iajit/PDF/vol.9,no.3/2107-6.pdf.

[2] Quantum genetic algorithm. http://people.ibest.uidaho.edu/~foster/Papers/40147.pdf.

[3] L. K. Grover. A fast quantum mechanical algorithm for database search. *in Proc. of the 28th Annual ACM Symphosium on the Theory of Computing*, pages 212–219, 1996.

[4] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79(2):325–328, 1997.

[5] 1 Jun Zhi 2 Huaixiao Wang, 1 Jianyong Liu and Chengqun Fu. The improvement of quantum genetic algorithm and its application on function optimization. *Mathematical Problems in Engineering,*, 2013:10 pages, 2013.

[6] Kuk hyun Han, Kui hong Park, Chi ho Lee, and Jong hwan Kim. Parallel quantum-inspired genetic algorithm for combinatorial optimization problem. In *in Proc. 2001 Congress on Evolutionary Computation. Piscataway, NJ: IEEE*, pages 1422–1429. Press, 2001.

[7] Kuk hyun Han and Jong-Hwan Kim. Genetic quantum algorithm and its application to combinatorial optimization problem. In *in Proc. 2000 Congress on Evolutionary Computation. Piscataway, NJ: IEEE*, pages 1354–1360. Press, 2000.

[8] Zhenquan Zhuang Junan Yang, Bin Li. Research of quantum genetic algorith and its application in blind source separation. *Journal of Electronics*, 20(1):62–68, 2003.

[9] Zakaria Labouidi and Salim Chikhi. Comparison of genetic algorithm and quantum genetic algorithm. *The international Arab Journal of Information Technology*, 9:243, 2012.

[10] Bin Li and Zhenquan Zhuang. Genetic algorithm based-on the quantum probability representation. In Hujun Yin, Nigel M. Allinson, Richard T. Freeman, John A. Keane, and Simon J. Hubbard, editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2002, Third International Conference, Manchester, UK, August 12-14, Proceedings*, volume 2412 of *Lecture Notes in Computer Science*, pages 500–505. Springer, 2002.

[11] Li Ying Jiao Licheng. An effective method of image edge detection based on parallel quantum evolutionary algorithm [j]. *Signal Processing*, 1:016, 2003.

[12] Ajit Narayanan and Mark Moore. Quantum-inspired genetic algorithms. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 61–66. IEEE, 1996.

[13] P.W. Shor. Algorithm for quantum computation : discrete logarithms and factoring. *IEEE, Proceedings of the 35th Annual Symposium on Foundation of Computer Science, Piscataway*, IEEE Press, 1994.

[14] Donald A. Sofge. Proceedings of the second quantum interaction symposium (qi-2008), college publications. *Prospective of algorithms for Quantum*, UK, 2008.

[15] Gexiang Zhang, Weidong Jin, and Laizhao Hu. A novel parallel quantum genetic algorithm. In *Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on*, pages 693–697. IEEE, 2003.

[16] Gexiang Zhang, Weidong Jin, and Na Li. An improved quantum genetic algorithm and its application. In *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, pages 449–452. Springer, 2003.

[17] R. Zhou and J Cao. Quantum novel genetic algorithm based on parallel subpopulation computing and its applications;. *Artf. Intell. Rev.*