International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 3 Issue 8 ǁ August 2014 ǁ PP.01-07

Quantum Evolutionary Algorithm for Solving Bin Packing Problem

Bhagaban Swain[1], Rajdeep Ghosh[2], Pranjal Borah[3]
[1,2,3,](Department Of Information Technology, Assam University Silchar ,Assam, India)

ABSTRACT: *Quantum computing is a recent and new frontier of computer science. It has opened up new prospects for various areas in this field. It has an alternative aspect for every problem in this domain and can be extended to various areas like image processing, optimization problems etc. It theoretically defines a new type of computer called as Quantum computer. The performance of which is going to be significantly better than the current classical computer. But construction of such computer is still a challenge to engineers and scientists. Quantum Algorithms are supposed to run in these computers. In this paper, we try to develop an analogue of Quantum algorithm and simulate the algorithm in classical computer .We go a step further and combine evolutionary algorithm with this quantum algorithm to form a Quantum Evolutionary Algorithm to solve a very well known problem called the bin packing problem .Later on we simulate the algorithm and test the algorithm with the benchmark datasets available for this class of problem and find that the algorithm provides us with optimal results.*

KEYWORDS: *BPP, Bin packing problem, Crossover, Mutation, Qbits, QEA, Quantum algorithm, Quantum Evolutionary algorithm.*

I. INTRODUCTION

The origins of Quantum Mechanics can be rooted to the 1900 where Planck first proposed 'Quantum Hypothesis' and Einstein explaining the photoelectric effect in 1905.Quantum computing had its beginning in the year 1984 when Feynman first conceptualized the idea of simulating Quantum systems in classical computers. After that Quantum computing has diversified and expanded immensely. The Quantum factorization algorithm by Peter Shor in 1994 further added interest to researchers in the field of Quantum computing. Quantum computing from then has been applied to various areas in computer science and has succeeded in almost every areas like image processing, database, cryptography, optimization. Thus the power of Quantum computing cannot be neglected.

In this paper we are converging the evolutionary algorithm and quantum computing concepts to form a quantum evolutionary algorithm. The basic concepts of quantum computing and evolutionary algorithms are described in the section 3 and 4 respectively.

II. BIN PACKING PROBLEM

Bin Packing Problem is a NP-hard combinatorial optimization problem first introduced by Garey and Johnson in 1979, it is a basic task that takes in almost everywhere in industries in packing items in a definite space. Some areas where bin packing can occur is the cutting stock, packing, cargo loading etc. Being NP hard no procedure can solve the algorithm in polynomial time. Recently some bio-inspired techniques have been proposed for solving the problem using Quantum inspired cuckoo search algorithm, firefly algorithm as in [5][8][9].

2.1. Formal Statement

Formally Bin Packing Problem can be defined as like if we are given a bin with capacity **V** and **n** items with sizes $\mathbf{w_1,w_2,\ldots\ldots,w_n}$ the aim is to find an integer **Z** and whether there exists a partition $\mathbf{S_1 U S_2 U \ldots\ldots U\ S_n}$. The objective is to minimize the value of **B**[5] **:**

$$\mathbf{Z}= \sum_{i=0}^{n} y_i \qquad (1)$$

such that :

$$\sum_{i \in s_k} w_i \leq V \qquad (2)$$

and subject to :

$$\sum_{j=1}^{n} w_j x_{ij} \leq V y_i, where\ i = \{1, .., n\} \quad (3)$$

$$\sum_{i=0}^{n} x_{ij} = 1, where\ j = \{1, .., n\} \quad (4)$$

$$y_i = \{0,1\},\ where\ i = \{1, .., n\}$$
$$x_{ij} = \{0,1\}, where\ i = \{1, .., n\}, and\ \ j = \{1, .., n\} \quad (5)$$

III. QUANTUM COMPUTING

Quantum computing is a relatively new theory which has emerged from merging computer science and the basic principles of quantum mechanics. Its major objective is to investigate the various quantum effects which take place in a computer. The origin of quantum computing dates back to 1981 with Richard Feynman proposing a model for quantum computer which could simulate the evolution of a quantum computer. Quantum computing has arouse much interest to scientists, engineers and researches since it appears more robust to almost all classes of problems in computer science. Indeed, the flexibility that it provides is helpful in molding it to solve almost any kind of problem and even reduce the algorithmic complexity. Such is its ability that optimization problems which are NP hard can be solved such that the algorithmic complexity is reduced. The basic laws and principles of quantum information theory are too vivid and varied to be described in this paper. Although the readers can refer to[10] for definitions and laws of quantum mechanics.

3.1 Qubit

In classical computers we have **bits** representing data. But a quantum computer on the other hand has **Q-bits** for representing data. The Q-bit is an analogue of the classical bit in classical computer, the qubit is the fundamental unit of information. Qubit and Qbit are used interchangeably. Classical bits can have values of 0 or 1. But Q-bits can be in one of the two states denoted by $|0\rangle$ or $|1\rangle$ or in a linear superposition of both the states. The notation $|\ \rangle$ is used to represent a state in quantum mechanics, to denote a vector and is called a **ket**. The notation was formalized by Dirac. Thus a Q-bit can be both zero or one simultaneously. A Q-byte is composed of eight Q-bits and has all values from zero to 255 in a state of superposition. In general the state of a Q-bit is the linear combination of two states and is given by:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (6)$$

Where, $|\psi\rangle$ is the state of the Qubit. α, β **are** complex numbers. $|\alpha|^2$ represents the probability amplitude of finding the superposition state $|\psi\rangle$ in state 0. $|\beta|^2$ represents the probability amplitude of finding the superposition state $|\psi\rangle$ in state 1. This formalism for a Qubit is a representation of a bit in quantum computer with the difference that it does not have discrete values but can also be in a combination of both.[1][2]

IV. GENETIC ALGORITHM

Genetic algorithms were first formalized by *John Holland in 1975* as a candidate solution search algorithm for spanning the solution space. In genetic algorithms, this principle is to find the best individuals with optimal values which are represented as chromosomes in the algorithm. So, each chromosome represents an arrangement of the items in the given problem, the algorithm starts from an initial population of chromosomes, with each iteration the chromosomes evolve into new generation of chromosomes performing a parallel search through the entire solution space. The fitness value for each chromosome is measured by a fitness function which is dependent on the problem being solved. Basically, a genetic algorithm depends on some major basic operations like crossover, mutation, selection etc. The selection operation has the fitness function which evaluates each individual and retains only the fittest chromosome from the population. In addition to those fittest chromosome, some less fitter ones could also be selected according to a fixed probability. The others are removed from the population generated. The crossover recombines two individuals to form new one which might be better, i.e. it introduces a better chance in the process to reach an optimum value. The mutation operator induces a random change in a relatively few number of individuals or chromosomes. Its purpose is diversify the population during the optimization process.

V. QUANTUM EVOLUTIONARY ALGORITHM (QEA)

Quantum Evolutionary Algorithm utilizes the concepts of quantum interference, superposition etc. in a framework of evolutionary algorithm to solve problems in classical computer. We can explain the algorithm as like evolutionary algorithm with a modification where the individual chromosome are represented as a sequence of Q-bits in concatenation i.e. in place of the representation of chromosome in binary values we represent them probabilistically with non-discrete value. At first we treat them as Quantum individual then perform the basic operation of evolutionary algorithm like crossover, mutation etc. and then map it to the corresponding numerical values where we evaluate and validate the values in accordance with the fitness function, then select the individuals for next generation and finally convert them back to the Q-bit formalism and proceed to the next iteration. [3][4]

Quantum Individual Representation

A Q-bit is the fundamental unit of information which represents a bit in quantum computation and can be defined as $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$. A combination of such Q-bits may form an individual gene, a combination of such gene form a chromosome, a combination of such chromosome form a population. A population of m individual chromosomes can be represented as:

$$S(t) = \{s^t_1 , s^t_2 ,\ldots\ldots, s^t_m\} \quad (7)$$

at generation *t* ,where, s^t_j, j=1,2,….,n, is each chromosome .
Each chromosome can be represented as:

$$s^t_j=\{c^t_{1j}, c^t_{2j,\ldots\ldots\ldots\ldots}, c^t_{nj}\} \quad (8)$$

Here c^t_{ij} represents each gene of the chromosome and n is the number of genes in each chromosome which is fixed for the problem and is equal to the total number of items for the problem and i ={1,2,…,n} and j ={1,2….m} where m is the number of chromosomes. Each gene is thus expressed as a linear superposition of quantum states , it is therefore advantageous for generating wide range of diversity in the evolutionary process.

Each gene is defined as:

$$c^t_{ij} = \begin{bmatrix}\alpha^t_{ij,1} & \alpha^t_{ij,2} & \ldots & \alpha^t_{ij,k}\\ \beta^t_{ij,1} & \beta^t_{ij,2} & & \beta^t_{ij,k}\end{bmatrix} \quad (9)$$

Where i,j is as defined above and $k=\log_2 n$ is the number of Q-bits in each gene, and

$$|\alpha^t_{ij,k}|^2+|\beta^t_{ij,k}|^2=1 \quad (10)$$

VI. SOLUTION APPROACH

The procedure for developing a quantum evolutionary algorithm for bin packing problem is as described below. Here all the chromosomes in the population are used for representing a possible arrangement or solution to a problem. The number m represents a population of chromosome. Each chromosome gives an arrangement of items .In each subsequent iteration new sets of chromosomes evolve as result of the evolution from their parent chromosomes using the genetic procedures of mutation and crossover.

The fitness value or rather the optimality of all the individual chromosomes in every generation is calculated in accordance with the problem, here the cost represents the number of bins used and are retained which are then equated to the fitness value of chromosomes of other generations subsequently and the lowest valued arrangement is retained. The lowest cost thus obtained in the process gives the optimal arrangement.

The **Procedure QEA** is as follows:

Begin
For i=1 to m //**m** is the population size
t ← 0
Initialize S(t)

Make R(t) by observing the states of S(t)
Repair R(t)
EvaluateR(t) // find the cost of individuals
End for
Initialize g_bestcost
While(elimination condition not satisfied) **do** {
For i=1 to m
t← t+1
perform Crossover S(t-1)
perform Mutation S(t)
Make R(t) by observing the states of S(t)
Repair R(t)
EvaluateR(t)
End for
store g_bestcost
}
End while
End

Where, ***S(t)*** is the quantum representation of the chromosome depicting the number of items to be packed. ***R(t)*** is the measured value of the chromosome i.e. we measure each gene and find the value.

Initialize S(t) is the step in which prepares the initial population of Q-bits randomly based on the condition $|\alpha|^2+|\beta|^2 = 1$.

Make R(t) is implemented for each chromosomes as follows. When observing the state of each Q-bit of each gene $\mathbf{c^t_{ij}}$, the measured value of R(t) is 0 or 1 which is calculated from the amplitude of $|\alpha^t_{ij,k}|^2$ or $|\beta^t_{ij,k}|^2$. Measuring gives a string of binary values, which is converted into its decimal value.

Repair R(t) are used for repairing the matrices according to a specific criterion of packing the bin, such that irrelevant value produced in the above procedure are eliminated.

C**rossover S(t)** and **Mutation S(t)** are the evolutionary operators which perform crossover and mutation over the chromosomes. **Crossover S(t)** is a operator that combines two parent chromosomes of **S(t-1)** to produce a new chromosome (offspring's). The idea is that during crossover in evolution individuals with better properties evolve due to the recombination of the chromosomes of both the parents. Crossover takes place in accordance with a user-defined crossover probability.[3],[4] There are many crossover techniques available but in this algorithm we use the one point crossover technique, the description of this technique is as follows:
If the chromosome with decimal values of items and if we consider the crossover point at position 4,

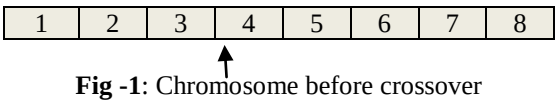

Fig -1: Chromosome before crossover

After crossover the chromosome would be

Fig -2: Chromosome after crossover

The **Algorithm** is given below:

index=ceil(n-1)*rand(); // n is the total number of items
temp=is a temporary array of size (2,n*b); //b=ceil($\log_2$n)
i=1;
While (i<k*2+1) //k=ceil($\log_2$n)
For j=1to index*b
temp(1,j)=S(1,j);

```
temp(2,j)= S (2,j);
S(1, j) = S(i, j);
S (2, j) = S(i+1,j);
S (i, j) = temp(1,j);
S (i+1,i)=temp(2,j);
End for
i=i+2;
End While
```

Mutation S(t) is followed after **Crossover S(t).** Mutation introduces randomness in the process which allows to skip a local minima to reach the global minima. Mutation occurs in accordance to a pre-defined mutation probability. Mutation modifies one or more genes in the chromosome arbitrarily. This can thus give an entirely new gene value. With the new gene values, the genetic algorithm may arrive at better solution than that was previously possible.

The pseudocode for **Mutation S(t)** is:

```
i=1;
While (i<k*2+1)            //k=log2n
   index= ceil(n*rand())    //n is the number of items
   a= S(i,index);
  S(i,point) =S (i+1,point);
  S(i+1,point)=a;
i=i+2;
End While
```

Evaluate R(t) is carried out at the end of each iteration to obtain the fitness value of the chromosome in accordance with a fitness function which is dependent on the bin packing problem. This fitness value is actually the cost of arrangement sequence of the bin. This value is retained and is further used in the next iterations.

VII. RESULT AND COMPARISON

This paper proposes an algorithm based on the concepts of Q-bits and inspired by Quantum computing for solving the bin packing problem and have obtained optimal solution for the various problem instances obtained from the various instances published by Prof. Dr. Armin Scholl and Dr. Robert Klein available online. These benchmark datasets contain the weights of items , number of items, maximum capacity of the bin and also the optimal number of bins that have been found.

There are three class of instances in the dataset, named as easy, medium and hard instances with different sizes of items and also different capacities. Here we have presented only a few results of some instances of the problem with various combination of values available in the datasets in Table-1,Table-2 and Table-3 respectively for easy, medium and hard instances along with their optimal value found respectively. Although we have applied it to various instances of the problem and have obtained near optimal values for the majority of the datasets. The algorithm has been coded in Matlab and implemented on a Intel core i3,2.4 GHZ,(2GB RAM), operating in windows 7 platform.

Table -1: Result and Comparison for easy instances

Problem Instance	Items	Capacity	Best known Solution	Result obtained by QEA
N1C1W1_C	50	100	20	25
N1C1W1_D	50	100	28	31
N1C1W1_E	50	100	26	29
N1C2W1_A	50	120	21	21
N1C2W1_B	50	120	26	27
N1C2W2_C	50	120	29	29
N1C3W1_D	50	200	19	19
N1C3W2_F	50	200	23	25
N2C1W1_A	100	100	48	50
N2C2W1_A	100	120	42	45
N2C2W2_A	100	120	52	53

N3C1W1_A	200	100	105	105
N3C2W1_A	200	120	91	91
N3C3W1_B	200	150	71	74
N3C3W2_C	200	150	82	83

Table -2: Result and Comparison for medium instances

Problem Instance	Items	Capacity	Best known Solution	Result obtained by QEA
N1W1B1R2	50	1000	19	19
N1W2B1R3	50	1000	11	13
N1W2B1R5	50	1000	10	12
N1W3B2R4	50	1000	7	9
N1W3B3R2	50	1000	7	8
N2W1B1R0	100	1000	34	34
N2W2B1R1	100	1000	20	22
N2W2B2R1	100	1000	21	21
N3W1B1R1	200	1000	67	68
N3W2B1R1	200	1000	41	41
N3W2B2R2	200	1000	39	39
N3W3B1R1	200	1000	28	31
N3W3B3R2	200	1000	29	32
N4W1B1R2	500	1000	167	170
N4W2B1R3	500	1000	100	104
N4W2B2R4	500	1000	100	100
N4W3B1R2	500	1000	71	75
N4W3B2R2	500	1000	71	71
N4W3B3R2	500	1000	72	76

Table -3: Result and Comparison for hard instances

Problem Instance	Items	Capacity	Best known Solution	Result obtained by QEA
HARD0	200	100000	56	57
HARD3	200	100000	55	56
HARD4	200	100000	57	57
HARD6	200	100000	57	59
HARD9	200	100000	56	56

The program is run with a population size of 5 and for 5000 iterations and the above results are obtained.

VIII. CONCLUSION

Thus we propose an algorithm based on the concepts of Q-bits and evolutionary algorithm for solving the bin packing problem and successfully obtained optimal solution for the various problem instances from .Moreover, Quantum inspired algorithms have already been applied to a various other NP Hard problems such as the Knapsack Problem, TSP etc.

REFERENCES

[1] Ge-Xiang Zhang , A Quantum-Inspired Evolutionary Algorithm Based on P systems for Knapsack Problem, *Fundamental Informatica 87*, 2008 , 1-24.

[2] Ajit Narayanan and Moore,Quantum inspired genetic algorithm,1996.

[3] Arther E Carter, Design and Application of Genetic Algorithms for the Multiple Traveling Salesperson Assignment Problem, April 21, 2003.

[4] Gohar Vahdati, Mehdi Yaghoubi, Mahdieh Poostchi, M.B. Naghibi, A new Approach to Solve Travelling Salesman Problem Using Genetic Algorithm Based on Heuristic Crossover and mutation Operator, 2009 *International Conference of Soft Computing and Pattern Recognition*, 2009,1-5.

[5] R. Yesodha and T. Amudha, Bio-inspired Metaheuristics for Bin Packing Problems, *International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS) 12-215, ISSN (Online): 2279-0055*, 2012, 64-67.

[6] Yehoon Kim, Jong-Hwan Kim, and Kuk-Hyun Han, Quantum-inspired Multiobjective Evolutionary Algorithm for Multiobjective 0/1 Knapsack Problems, *2006 IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada,July* , 2006,16-21.

[7] K.H. Han, J.H. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE Trans. Evolut. Comput.,6(6)* , 2002, 580–593.

[8] Aphirak Khadwilard, Sirikarn Chansombat, Thatchai Thepphakorn, Peeraya Thapatsuwan, Warattapop Chainate and Pupong Pongcharoen, Investigation of Firefly Algorithm Parameter Setting for Solving Job Shop Scheduling Problems, 2011.
[9] Ullas Mohan, Bio Inspired Computing, Division Of computer science, *SOE, CUSAT*, July 2008.
[10] David Mcmohan, *Quantum computing explained*(John Wiley & Sons,Inc.,publication,2007).