QUANTUM COMPUTER AND QUANTUM ALGORITHM FOR TRAVELLING SALESMAN PROBLEM

Utpal Roy, Sanchita Pal Chawdhury, Susmita Nayek

ABSTRACT

Depending upon the extraordinary power of Quantum Computing Algorithms various branches like Quantum Cryptography, Quantum Information Technology, Quantum Teleportation have emerged [1-4]. It is thought that this power of Quantum Computing Algorithms can also be successfully applied to many combinatorial optimization problems.

In this article, a class of combinatorial optimization problem is chosen as case study under Quantum Computing. These problems are widely believed to be unsolvable in polynomial time. Mostly it provides suboptimal solutions in finite time using best known classical algorithms. Travelling Salesman Problem (TSP) is one such problem to be studied here. A great deal of effort has already been devoted towards devising efficient algorithms that can solve the problem [5-18]. Moreover, the methods of finding solutions for the TSP with Artificial Neural Network and Genetic Algorithms [5-8] do not provide the exact solution to the problems for all the cases, excepting a few. A successful attempt has been made to have a deterministic solution for TSP by applying the power of Quantum Computing Algorithm.

Keywords: Quantum Computing, Travelling Salesman Problem, Quantum algorithms.

1. INTRODUCTION

A quantum computer is a device that can arbitrarily manipulate the quantum state of a part or itself. The field of quantum computation is largely a body of theoretical promises for some impressively fast algorithms which could be executed on quantum computers. The field of quantum computing has advanced remarkably in the past few years since Shor [1] presented his quantum mechanical algorithm for efficient prime factorization of very large numbers, potentially providing an exponential speedup over the fastness on known classical algorithm. As much of today's cryptography [2] relies on the presumed difficulty of factoring large numbers, the Shor's algorithm has important implication for data encryptions technology. It was Deutsch who first suggested [3] that quantum superposition might allow evaluation to perform many classical computations in parallel. R. P. Feynman [4] considered the question of how well classical computer can simulate quantum states whereas that quantum system could, in principle simulate each other without any slow down. In a traditional computer, information is encoded in a series of bits, and these bits are manipulated via Boolean Logic Gates arranged in succession to produce an end result. Similarly a Quantum Computer can manipulate qubits by executing a series of operations over quantum gates, (each a unitary transformation acting on a single qubit or a pair of qubits). While using these gates in succession, a quantum computer can perform a complicated unitary transformation to a set of qubits occupying previous states. The qubits can then be evaluated to provide the final computational result. This similarity in calculation between a classical and quantum computer affords that at least in theory, a classical computer can accurately simulate a quantum computer. However, this simulation in classical computer is grossly inefficient and effectively incapable of performing many tasks that quantum computer model can perform at ease. This is simply because the correlation between the quantum bits is qualitatively different from correlation among classical binary bits. This feature leads to the source of massive power of computation using quantum algorithm.

It is needless to say that the Transportation and Assignment problems have both practical and theoretical importance. These problems are well known since a long time and have been studied with classical algorithms. A case study of Travelling Salesman problem (hereafter referred as TSP) with Hopfield net [6] produces the following results as function of D, a parameter penalizing the total length of the tour as follows:

D	Convergence rate	Found Best Tour
1.0	100	3
1.5	96	8
2.0	88	11
3.0	41	10

This clearly indicates the inadequacy of Hopfield net for the solution of TSP. Further, the Artificial Neural Network (ANN) does not produce satisfactory results for the solution of TSP [5-8] and similar problems. The classical algorithms produce a few sub-optimal solutions and ultimately problems are said to belong to `P' class i.e. `intractable'. Some of these problems under TSP offer complexity of class `NP'[19]. The extraordinary power of quantum algorithms may be exploited for the solution of these classes of problems.

Our aim is to develop some efficient quantum computing algorithms for some well known optimization problems with special reference to TSP. These algorithms will be implemented to run in classical computers, as quantum computing hardwires are not available till date. Later, the nature and output of the algorithms will be compared with those available from the classical algorithms. In essence, the problems believed to remain unsolved even with best known classical algorithms will have a way out.

Optimization problems are daily routine problems, drawing enormous theoretical interests. To study the nature and efficiency of the proposed quantum algorithm a toolbox for implementing quantum gates and quantum registers through C programming, and simulation of the quantum algorithms. As the quantum hardware is not yet available, the actual performance of quantum algorithms in the proper environment is not possible. Actually we are interested to have the realization of quantum algorithms in classical computers. The properties of quantum registers will be exploited to implement the quantum algorithms for the problems of interest.

The organization of the paper is as follows: Section 2 presents the survey of the literature, Section 3 deals with underlying physics of the Quantum Computer, The properties and nature of the Quantum Memory Register is described in Section 4. The Section 5 presents the brief history and application of TSP. The Quantum Algorithm of the TSP is depicted in Section 6. Section 7 contains the Analysis of the Results and Discussion. Conclusion of the study is placed in Section 8. The Acknowledgement and the exhaustive, up-to-date bibliography are placed in respectively in Sections 8 and Section 9.

2. SURVEY OF LITERATURE

In the international level, a variety of theoretical work in the field of Quantum Computing [20, 21] and its application to various other fields is under rapid progress [22, 23]. In 1994 Peter Shor [1] published a factoring algorithm for quantum computer, that finds the prime factors of a composite integer N (N is a product of two odd prime numbers), more efficiently than is possible with the best known algorithms with a classical computer. After that quantum computation suddenly blossomed, quantum computing has emerged as a future tool for solution of many unsolved problems. The experimental progress has been rapid with several schemes yielding two [24, 25] and three quantum bits manipulations [26]. In addition with the discovery of quantum error correction schemes [27], such machines have the promise of providing long term storage of quantum information. Quantum computer was first discussed by Benioff [28] in the context of simulating classical Turing machines (primitive computer model). Feynman [4] posed the converse question, as how well the classical computer can simulate quantum systems. He concluded that classical computers invariably suffer from an exponential slow-down in trying to simulate quantum systems, whereas that quantum system could in principle, simulate each other without this slow-down. It is Deutsch [29] who first suggested that quantum superposition might allow quantum evolution to perform many classical computations in parallel.

Different disciplines related to quantum computing algorithms have emerged. The few important are Quantum Cryptography, Quantum Information and Quantum Teleportation.

At the national level, no such serious effort has been paid in the field of Quantum Computer and Quantum Computing Algorithms. Very few groups are working in this field together with its applications in different aspects of theoretical and experimental physics.

3. PHYSICS IN QUANTUM COMPUTERS

Digital Computers are built out of circuits that have definite, discrete state, on or off, zero or one, high voltage or low voltage. It is sure that circuit never settles in between intermediate state. Quantum mechanical system, for example, spin orientation of an electron, is always up or down; never in between. Likewise an atom emits or absorbs energy in discrete way not in continuous form. Letting particle spins or energy levels of atom stand for binary unit of information. So one can think in gross that a quantum computer out of quantum mechanical system can be built. In a quantum computer fundamental unit of information is called a quantum bit or qubit, is not binary rather quaternary in nature. The nature of qubit is very much different from that of classical bits. Classical bits can have values between 0 and 1 at all moments but qubit can occupy a superposition of 0^s and 1^s during certain stages of computation. This is not to say that the qubit has some intermediate values between 0^s and 1^s, rather the qubit is in both the 0 state and 1 state at the same time to varying extents. When the states of the qubits are eventually measured it is invariably either 0 and 1. This is known as superposition principle or quantum entanglement. This property of qubit arises from the direct consequences of the laws of quantum mechanics that differ from the laws of classical physics. More generally, where a string of n classical bits exists in any of the 2^n Boolean states, $x = 00000.......0$ through $11111........1$ a string of n qubits can exist in any of the form:

$$\Phi = \sum_{x=00...0}^{111...1} C_x |x\rangle \qquad (1)$$

Where C_x the complex numbers are satisfies the condition

$$\sum_x |C_x|^2 = 1 \qquad (2)$$

The physical implication of the expression is that a quantum state of n qubits represented by a complex vector Φ of unit length in Hilbert space of 2^n dimensions, one for each possible classical states. Exponentially large dimensionality of this space distinguishes quantum computer from classical computer. The states of the digital or analogue computers are described by a number of parameters that grows linearly with the size of the system. That is why a classical system whether digital or analogue can be completely described by describing separately the states of each part. The vast majorities of quantum states, by contrast, are entangled and permits no such description. The ability to preserve and manipulate the entangled quantum states is the distinguishing feature of quantum computers, responsible both for their power and for the difficulty of building them. An isolated quantum system evolves in such a way as to preserve superposition and distinguishability;

4. QUANTUM MEMORY REGISTER

The history of computer technology has involved a sequence of changes from one type of physical realization to another. Today's advanced lithographic techniques can squeeze fraction of micron wide logic gates are so small that they are made out of only handful of atoms. On the atomic scale the matters obeys the rules of quantum mechanics, which are quite different from the classical rules. That determines the properties of conventional logic gates. So computers are likely to become smaller in the future, new quantum technology must replace of supplement what we have now. The point is, however, that quantum technology can offer much more that cramming more and more bits to silicon and multiplying the clock speed of microprocessors. It can support entirely a new kind of computation with qualitatively new algorithms based on quantum principles. However if we choose an atom a physical bit then quantum mechanics tells us that apart from the two distinct electronic states the atom can also be prepared in a coherent superposition of the two states. It means that atom lies in both the states with different probability.

Now we push the idea of superposition of numbers a bit further. Consider a register composed of three physical bits. Any classical register of that type can store in a given moment of time only one out of eight different number, other way one can say that the register can be only in one out of eight possible configuration such as 000, 001, 010, 011, 100, 101, 110, 111. A quantum register composed of three qubits can store in a given moment of time all eight numbers in a quantum superposition with different probability. This is quite remarkable that all eight numbers are physically present in the register but it should be o more surprising than a qubit being both in the state 0 and 1 at the same time. If quits are keep adding on adding into the register we increase its storage capacity exponentially i.e. three qubits can store eight numbers at once four qubits can store 16 such different numbers at once, and so on; in general L qubits can store 2^L numbers at once. Once the register is prepared in super position of different numbers we can perform operation on all of them. For, example if qubits are atoms then suitably tuned laser pulses affect atomic electronic states and evolve initial superpositions of encoded numbers into different superpositions. During such evolution each number in the superposition is affected and as a result we generate a massive parallel computation albeit in one piece of quantum hardware. This altogether means that a quantum computer can in only one computational step perform the same mathematical operation on 2^L different input numbers encoded encoded in coherent superpositions of L qubits. In order to accomplish the same task any classical computer has to repeat the same computation 2^L times or one has to use 2^L numbers of processors working in parallel. In other words a quantum computer offers an enormous gain in the use of computational resources such as time and memory. But this, after all, sounds as yet another purely technological progress. It looks like classical computers can do the same computation as quantum computer but simply need more time or more memory. The catch is that the classical computers need exponentially more time or memory to match the power of quantum computers and this is asking for too much because an exponential increase is really fast and we run out of available time or memory quickly. The schematic diagram of a three qubits quantum register is shown in Figure 1.

Information is something that is encoded in the state of a physical system, a computation is something that can be carried out in a actual physically realizable device. Given that quantum information has many unusual properties, it might have been expected that quantum theory would have a profound impact on our understanding of computation. Shore[1] demonstrated that, at least in principle, a quantum computer can factor a large number efficiently.

Factoring (finding the prime factors of a composite number) is an example of an intractable problem with the property :

- The solution can be easily verified once found.
- But the solution is hard to find.

That is if p and q are large prime numbers, the product $n = pq$ can be computed quickly. But given n it is hard to find p and q. The time required to find the factors is strongly believed (though this has never been proved) to be superpolynomial in $\log(n)$. The best known factoring algorithm[6] requires

$time \cong \exp[c(\ln n)^{1/3}(\ln \ln n)^{2/3}]$, where $c = 1.9$.

The current state of the art is that the 65 digit number can be found in order of one month by a network of hundreds of work stations. Using this to estimate the prefactor we can estimate that factoring a 400 digit number would take about 10^{10} years, the age of the universe. So even with vast improvements in technology, factoring a 400 digit number will be out of reach for a while.

The factoring algorithm is interesting from the prospective of complexity theory, as an example of a problem presumed to be intractable; that is, a problem that can not be solved in time bounded by polynomial in the size of the input, in this case $\log n$. But it is also of practical importance, because the difficulty of factoring is the basis of schemes for public key cryptography, such as widely used in RSA scheme.

Shore [1] found an exciting new result that a quantum computer can factor in polynomial time, e.g. in time $O[(\log n)^3]$. Shore results indicates that even 400 digit number can be factor in reasonable time with quantum computer. The Shore's result is fascinating to contemplate the implications for complexity theory, for quantum theory, for technology. A quantum computation can be described this way. The N qubits are assembled and prepared them a standard initial state such as $|0>|0>...........|0>$, or $|x=0>$. Then a unitary transformation is applied to the N qubits. The transformation U is constructed as product of standard quantum gates, unitary transformations that act on just a few qubits at a time. After U is applied, we measure all of the qubits by projecting onto the $\{|0>,|1>\}$ basis. The measurement outcome is the output of the computation.

It is of worth mentioning that the algorithm performed by the quantum computer is probabilistic algorithm. That is we could run exactly the same programme twice and obtain different results, because of the randomness of the quantum measurement process. The quantum algorithm actually generates a probability distribution of possible outputs. In fact Shor's factoring algorithm is not guaranteed to succeed in finding the prime factors; it just succeeds with a reasonable probability.

It should be clear from this description that a quantum computer, though it may operate according to different physical principles that a classical computer, cannot do anything that a classical computer can not do. Classical computer can store vectors, rotate vectors, and can model the quantum measurement process by projecting a vector onto mutually orthogonal axes. So a classical computer can simulate the quantum computer to an arbitrarily good accuracy. But quantum computers in principle can cope many things those are merely impossible to compute with classical computers.

But we should also consider how long the simulation will take. Suppose we have a computer that operates on modest number of qubits, like N=100. Then to represent the typical quantum state of computer, we would need to write down $2^N = 2^{100} \approx 10^{30}$. No existing or foreseeable classical computer will be able to do it. And performing a general rotation of vector in space of dimension 10^{30} is far beyond the computational capacity of the foreseeable classical computer.

So it is true that a classical computer can simulate a quantum computer in principle but the simulation becomes extremely inefficient as the number of qubits N increases. Quantum mechanics is hard computationally because we must deal with huge matrices – there is too much room in the Hilbert space. This observation led Feynman to speculate that a quantum computer would be able to perform certain tasks that are beyond the reach of any conceivable classical computer. In the end, our simulation should provide a means of assigning probabilities to all the possible outcomes of the final measurement.

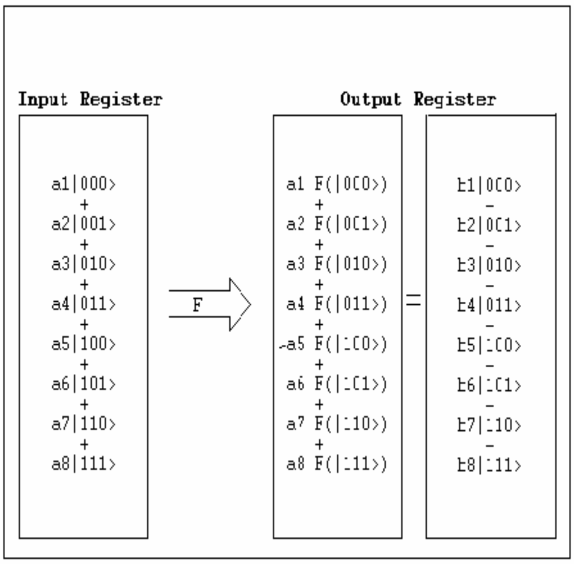

Fig. 1. A Quantum Register with three qubits capacity

5. TRAVELLING SALESMAN PROBLEM

Mathematical problems related to the Travelling salesman problem were treated in the 1800s by the Irish mathematician Sir William Rowan Hamilton and by the British mathematician Thomas Penyngton Kirkman. A nice discussion of the early work of Hamilton and Kirkman can be found in the book Graph Theory 1736-1936 by N. L. Biggs, E. K. LLoyd, and R. J. Wilson, Clarendon Press, Oxford, 1976.

The general form of the TSP appears to be have been first studied by mathematicians starting in the 1930s by Karl Menger in Vienna and Harvard. The problem was later promoted by Hassler Whitney and Merrill Flood at Princeton.

The Travelling salesman problem is easy to state: given a collection of cities and the cost of travel between each pair of them, the TSP, is to find the cheapest way of visiting all of the cities and returning to your starting point. In this study we concentrate on symmetric TSP in which travel costs are symmetric.

The travel costs are symmetric in the sense that Travelling from city X to city Y costs just as much as Travelling from Y to X; the "way of visiting all the cities" is simply the order in which the cities are visited. To put it differently, the data consist of integer weights assigned to the edges of a finite complete graph; the objective is to find a Hamiltonian cycle (that is, a cycle passing through all the vertices) of the minimum total weight. In this context, Hamiltonian cycles are commonly called tours.

The simplicity of the statement of the problem is deceptive -- the TSP is one of the most intensely studied problems in computational mathematics and yet no effective solution method is known for the general case. Indeed, the resolution of the TSP would settle the P versus NP problem..

Although the complexity of the TSP is still unknown, for over 50 years its study has led the way to improved solution methods in many areas of mathematical optimization.

APPLICATION OF TSP

Much of the work on the TSP is motivated by its use as a platform for the study of general methods that can be applied to a wide range of discrete optimization problems. This is not to say, however, that the TSP does not find applications in many fields. Indeed, the numerous direct applications of the TSP bring life to the research area and help to direct future work.

The TSP naturally arises as a subproblem in many transportation and logistics applications, for example the problem of arranging school bus routes to pick up the children in a school district. This bus application is of important historical significance to the TSP, since it provided motivation for Merrill Flood, one of the pioneers of TSP research in the 1940s. A second TSP application from the 1940s involved the transportation of farming equipment from one location to another to test soil, leading to mathematical studies in Bengal by P. C. Mahalanobis and in Iowa by R. J. Jessen. More recent applications involve the scheduling of service calls at cable firms, the delivery of meals to homebound persons, the scheduling of stacker cranes in warehouses, the routing of trucks for parcel post pickup, and a host of others.

Although transportation applications are the most natural setting for the TSP, the simplicity of the model has led to many interesting applications in other areas. A classic example is the scheduling of a machine to drill holes in a circuit board or other object. In this case the holes to be drilled are the cities, and the cost of travel is the time it takes to move the drill head from one hole to the next. The technology for drilling varies from one industry to another, but whenever the travel time of the drilling device is a significant portion of the overall manufacturing process then the TSP can play a role in reducing costs.

Researchers at the National Institute of Health have used Concorde's TSP solver to construct radiation hybrid maps as part of their ongoing work in genome sequencing. The TSP provides a way to integrate local maps into a single radiation hybrid map for a genome; the cities are the local maps and the cost of travel is a measure of the likelihood that one local map immediately follows another. This application of the TSP has been adopted by a group in France developing a map of the mouse genome. The mouse work is decribed in Nature Genetics "A Radiation Hybrid Transcript May of the Mouse Genome", Nature Genetics 29 (2001), pages 194--200.

A semi-conductor manufacturer has used Concorde's implementation of the Chained Lin-Kernighan heuristic in experiments to optimize scan chains in integrated circuits. Scan chains are routes included on a chip for testing purposes and it is useful to minimize their length for both timing and power reasons.

An old application of the TSP is to schedule the collection of coins from payphones throughout a given region. A modified version of Concorde's Chained Lin-Kernighan heuristic was used to solve a variety of coin collection problems. The modifications were needed to handle 1-way streets and other features of city-travel that make the assumption that the cost of travel from x to y is the same as from y to x unrealistic in this scenario

Concorde library is currently being incorporated into the Worldwide Airport Path Finder web site to find shortest routes through selections of airports in the world. The author of the site writes that users of the path-finding tools are equally split between real pilots and those using flight simulators

The travel itinerary for an executive of a non-profit organization was computed using Concorde's TSP solver. The trip involved a chartered aircraft to visit cities in the 48 continental states plus Washington, D.C. (Commercial flights were used to visit Alaska and Hawaii.) It would have been nice if the problem was the same as that solved in 1954 by Dantzig, Fulkerson, and Johnson, but different cities were involved in this application (and somewhat different travel costs, since flight distances do not agree with driving distances).

An early version of Concorde's tour finding procedures was used in a tool for designing fiber optical networks at Bell Communications Research (now Telcordia). The TSP aspect of the problem arises in the routing of sonet rings, which provide communications links through a set of sites organized in a ring. The ring structure provides a backup mechanism in case of a link failure, since traffic can be rerouted in the opposite direction on the ring.

Modules from Concorde were used to locate cables to deliver power to electronic devices associated with fiber optic connections to homes.

6. QUANTUM ALGORITHM FOR TSP

TSP can be modeled as a graph where the vertices correspond to cities and the edges correspond to connections between cities, the length of an edge is the corresponding distance from one city to another city. A TSP tour is now a Hamiltonian cycle in the graph and an optimal TSP tour is the shortest Hamiltonian cycle.

Input: A completely connected n cities where each link is assigned by a weight
OR, A complete weighted graph.

Output: Find the shortest circuitous path connecting n cities.
Or, Find a minimal Hamiltonian circuit.

Procedure:

Step1: Let n be the total no. of cities where each city is connected to all other cities. Therefore the network has $\frac{(n^2-n)}{2}$ links whereas each link is assigned by a non-negative weight. So, there are $(n-1)$ weights from one city to other $(n-1)$ cities.

Step2: n quantum registers R_i for $i=1$ to n be considered for storing all the weights initially. i.e., register R_i for $1 \le i \le n$ is initialized with $(n-1)$ weights { $\{(v_i, v_j)\} \to w_{ij,}$ for $j \ne i$ and $\forall j$ } for city i where v_i is the starting city and v_j is the next city.

Initialization

for $i=1$ to n
  for $j=1$ to n
    if $i \ne j$
      {
      Initialize R_i with $(n-1)$ weights w_{ij} such that
      $w_{ij} = w_{ji}$ for $j < i$
      and w_{ij} is the user input if $j > i$
      }

Step 3: The quantum register R_i for $i=1$ to n is sorted in parallel and in ascending order. A city is chosen arbitrarily as a starting city from n cities. In quantum register $R_i(v_j, v_k, w_{jk}, null, flag)$ suppose v_j is the arbitrarily chosen starting city. Then v_k is assigned with the next starting city in a tour iff w_{jk} is minimum $\forall k \ne j$. The null indicator changes to 1, immediately by applying CNOT gate [21].

Step4: Now traversing is started from city v_k that is connected in the tour in step3, to the remaining $(n-2)$ cities $v_l, \forall l \ne k$ for obtaining minimum weight w_{kl} from city k to city l.

If v_l is assigned in the tour, the status of the 'null' indicator is changed and the 'flag' is also changed to indicate that v_l is already been traversed.

Repeat this step until $(n-1)$ cities are assigned in the tour.

Repeat this step until cities are assigned in the tour.

Step5: Now all n cities are in the tour. The last city means the city that is assigned recently in the tour automatically connect to the starting city to obtain a Hamiltonian circuit. But this circuitous path is not always satisfied by the minimality criterion. For most of the cases, the algorithm shows a minimal Hamiltonian circuit, but for some cases the results are not yet satisfactory.

We look here at the Travelling Sales man Problem which is a well known NP-Complete problem. Any NP-complete problem including TSP can be solved with exhaustive search. Unfortunately, when the size of the instances grows the running time for exhaustive search soon becomes forbiddingly large, even for instances for fairly small size. For some problem it is possible to design algorithms that are significantly faster than exhaustive search, though still not polynomial time. In the recent years there has been growing interest in the field of NP-complete problems. The interest has many causes. It is now commonly believed that $P \neq NP$, and the super-polynomial time algorithms are best we can hope for when we are dealing with an NP-complete problem. There is a handful isolated scattered results available in literature but we are far from developing a general theory. In fact, it has been observed that we have not even started a systematic investigation of the worst case behavior of such super-polynomial algorithms.

Some NP-complete problems have better and faster exact algorithms than others. There is a wide variation in the worst case complexities of known exact (super-polynomial time) algorithms. Classical theory cannot explain these differences [30]. It is a matter of interest whether there exists any relationship among the worst case behavior of various problems.

With the increased speed of modern computers large instances of NP-complete problems can be solved effectively. For example it is now a day's routine to solve the TSP instances with 2000 cities. And if the data is nicely structured, then instances with upto 13000 cities can be handled in practice [30]. But there is huge gap between the empirical results from testing implementation and the known theoretical results on exact algorithms.

The most straight forward and accurate algorithm to solve an instance of TSP is to examine all possible Hamiltonian circuits and select one of possible total length. How many circuits we have to examine to solve the problem if there are n vertices in the graph? Once a starting point is chosen there are $(n-1)!$

different Hamiltonian circuits to examine, since there are $(n-1)!$ choices for the second vertex, $(n-2)!$ choices for the third vertex and so on. For the symmetric Travelling Salesman Problem we need only examine $\frac{(n-1)!}{2}$ to answer the question. Note that $\frac{(n-1)!}{2}$ grows extremely rapidly. Trying to solve the TSP in this way when there are only a few dozen vertices is impractical. For example with 25 vertices, a total of $\frac{24!}{2} \approx 3.1 \mathrm{x}\, 10^{23}$ different Hamiltonial circuits would have to be considered. It took just one nanosecond (say) to examine each Hamiltonian Circuit; a total of approximately 10 million years would be required to minimum length Hamiltonian Circuit in this graph. So whatever may be the speed of the modern computer it is extremely beyond imagination to find a workable algorithm for the solution of the TSP. But TSP, in its own merit has both practical and theoretical importance, till date no algorithm with polynomial worst case time complexity is known for solving the problem. Following NP-completeness theory, if a polynomial worst case time complexity algorithm were discovered for solving the TSP many other difficult problems would also be solvable using polynomial worst case time complexity algorithm.

Analysis of Results and Discussion

In the above algorithm, step2, Step 3 and Step 4 are more crucial for time complexity. It is observed that $O(n^2)$ is the required time complexity to execute step2. Step3 needs $O(n \log n)$ time whereas step4 requires $O(n^2)$ time. As the complexities of the different steps are additive in natures of the overall time complexity of the algorithm is $O(n^2)$.

The proposed quantum algorithm has been tested in a compiler libquantum 0.5 in linux environment freely available from the internet [22,23]. The novelty of the algorithm resides in the property of the quantum register described in the Section 4. This compiler basically is a toolbox in which various quantum operations in quantum gates and quantum registers are built in and can be called as a library function and some other important quantum operations(like sorting, storing, initialization of quantum register etc) are implemented through C programming and as whole it provides an environment for simulation of the quantum algorithms. As the quantum hardware is not yet available, the actual performance of quantum algorithms in the proper environment is not possible. Actually we have here an interesting realization of the of quantum algorithms in classical computers. Mainly the parallelism property and as well as the enormous storage capacity of the quantum registers have been exploited in great extent to implement the quantum algorithms for the problems of interest, TSP in this case. So far the solutions of the TSP are concerned there is handful scattered references are available in the literature but no general theory has been developed so far, to the best of our knowledge. It is not our intension to go details of the development of the TSP algorithms using various techniques (heusristic search, dynamic programming method, genetical algorithm and many others). Out of many other published algorithms almost exact one is from Held and Karp [32] which is based on dynamic programming method. They have achieved an overall time complexity of $O(n^2 2^n)$. The result was published in 1962 and from nowadays point of view looks very trivial. Still it provides a best time complexity for TSP that is known today. A trivial but accurate algorithm for TSP checks all $O(\frac{(n-1)!}{2})$ permutations as described in Section 6. The obtain time complexity available from the present algorithm have been compared with those two algorithms.

Table 1. Comparison of the present algorithm with other popularly known established algorithm for TSP.

No of Cities n	Present Quantum Algo. $O(n^2)$	Algorithm $O(\frac{(n-1)!}{2})$ (trivial algo)	Algorithm[32] $O(n^2 2^n)$
3	$O(9)$	$O(1)$	$O(72)$
4	$O(16)$	$O(3)$	$O(256)$
5	$O(25)$	$O(12)$	$O(800)$
6	$O(36)$	$O(60)$	$O(36 \times 64)$
100	$O(100^2)$	$O(\frac{99!}{2})$	$O(100^2 \times 2^{100})$

The obtained results are presented in Table1. For lower values of n it is obvious that the present quantum algorithm can not supersede the time complexity of the other algorithms but with the increasing values of n the result is very much interesting and the time complexity remarkably gets lower and lower in comparison to the time complexities available from other two as mentioned earlier. In practical cases where the TSP becomes applicable the number of cities n always assigned to large values. So the proposed algorithm can easily cope any larger number of cities for the solution of the TSP. The Figure 2 represents an undirected weighted graph for 6 cities of TSP problem. With the help of the present algorithm the shortest Hamiltonian Path and the cost of the tour have been computed . Two minimum Hamiltonian paths are available here but a close look to the Hamiltonian paths will reveal that the two paths are basically the same and for the closed shortest Hamiltonian paths the starting city is immaterial.

Path I: $A \rightarrow B \rightarrow D \rightarrow C \rightarrow E \rightarrow F \rightarrow A$.

Path II: $B \rightarrow D \rightarrow C \rightarrow E \rightarrow F \rightarrow A \rightarrow B$.

The cost for both the paths is same and is 170.9.

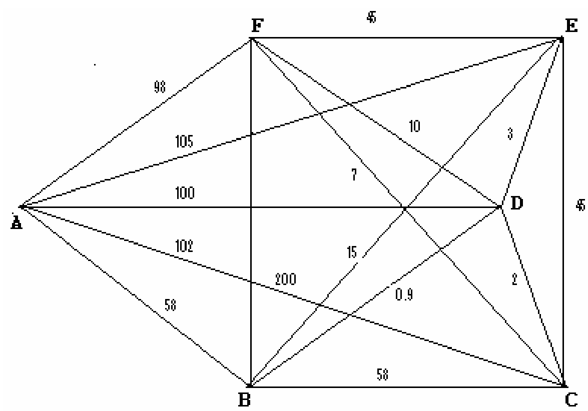

Fig. 2. The TSP Graph with six cities.

7. CONCLUSION

Due to the virtue of massive parallelism of quantum computer the sorting operation will be performed within the quantum register within a single step. In essence the quantum complexity of the TSP problem according to the present algorithm is much smaller than the well established available results in literature[32,33]. But till date no quantum hardwires are available for the commercial purpose. The only available quantum hardwires are merely within the laboratory use and in the experimental state. So, if quantum hardwires are, at all, be available then the problem like TSP could be coped with the quantum Computer, easily. Here we have designed the algorithm for TSP and have run it in the classical computers (Intel PIV in Linux environment) with the compiler libquantum 0.5 available in various websites.

There remain many open problems and challenging questions around the worst case analysis of the exact algorithms for NP-hard problems. This seems to be a rich and promising area. We only handful of techniques available, and there is an ample space for improvements and applications in the field of quantum computer.

9. Acknowledgement:

One of us (UR) is interested to acknowledge the financial support from UCG project Ref. No. F33-437(SR)/2007, dated 28.02.2008.

REFERENCES:

1. P.W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring,",Proc. Of the 35th Annual Symposium on the Foundation of Computer Science, IEEE Computer Society Press, Los Alamitos, California, 1994), pp.124-134.
2. R. Rivest, A. Shamir and L. Adelman, Communication ACM 21, 120(1978).
3. D. Deutsch, ``Quantum Theory is a universal physical theory," Int. J. Theor. Phys. 24, 1-41(1985); ``Quantum Theory the Crunch turning Principle and the universal quantum computer," Proc. Roy. Soc. Lond. A 400 97-117(1985)
4. R.P. Feynman, ``Simulating physics with computers," Int. J. Theor. Phys. 21, 467-488(1982);``Quantum Mechanical Computers," 507-531.
5. S. Haykin, 1999, Neural Networks-a Comprehensive foundation, Prentice Hall.
6. The TSP home page, Mathematics Department, university Princeton, http://www.math.princeton.edu/tsp/index.html.
7. J.J.Hopfield, D.W.Tank, 1985. Neural Computation of decisions in Optimization Problem, Biological cybernetics, vol.52, pp.141-152".
8. G.V.Wilson, G.S.Pawley, 1988. On the Stability of the Travelling Salesman problem Algorithm of Hopfield and Tank, Biological Cybernetics, vol.58, pp.63-70.
9. Gee, S.V.B.Aiyer, R.Prager, 1993. An Analytical framework for Optimizing Neural Networks, vol.6, pp.79-97.
10. S.V.B.Aiyer, M.Niranjan, F.Fallside, 1990. A theoretical investigation into the performance of the Hopfield model, IEEE Transactions on Neural Networks, vol.1, pp.204-215.
11. N.Ansari, E.Hou, 1997. Computational Intelligence For Optimization, Norwell, MA:Kluwer Academic publishing Group.

12. H. Crowder and M. Padberg, ``Solving large-scale symmetric Travelling salesman problems to optimality," Mgmt. Sci. 26 (1980),495-509.
13. M.Padberg and G.Rinaldi, ``Optimization of a 532-city symmetric Travelling salesman problem by branch and cut," Operations Res. Lett. 6 (1987), 1-7.
14. D.L.Applegate and W. Cook, ``Solving large-scale matching problems," in {\it Network Flows and Matching: First DIMACS Implementation Challenge}, American mathematical Society,Providence,RI,1993,557-576.
15. M.Padberg and G.Rinaldi, ``A branch-and-cut algorithm for the resolution of large-scale symmetric Travelling salesman problems," {\it SIAM Review} 33 (1991),60-100.
16. M.Gr\"otschel and O. Holland, ``Solution of large-scale symmetric Travelling salesman problems," Math. Programming 141-202 (1991).
17. D.Applegate, R.Bixby, V.Chv\'atal, and W. Cook, private communication(1994).
18. E.L.Lawler,J.K.Lenstra, A.H.G.Rinnooy Kan, and D.B.Shmoys,{\it The Travelling Salesman Problem}, John Wiley \& Sons,Chichester, 1985.
19. M.R.Garey, and D.S.Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness}, W.H.Freeman, San Francisco, 1979.
20. M.A. Nielson and I.L. Chang; Quantum Computation and Quantum Information, Cambridge, U.K. Cambridge Univ. Press, 2000
21. Josef Gruska; Quantum Computing, The McGraw-Hill Companies, 1999.
22. http://cam.qubit.org//links.php
23. B.Oemer(2003), http://tph.tuwien.ac.at/~oemer/qcl.html.
24. Q. A. Turchette, C. J. Hood, W. Lange, H. Mabuchi and H. J. Kimble, “ Measurement of Conditional phase shifts for quantum logic”, Phys. Rev. Lett. 75, 4710-4713.
25. C. Monroe, D.M. Meekkhof, B.E. King, W.M. Itano and D.J. Wineland, “ demonstration of Fundamental of logic gates”.
26. N. Gershenfeld and I. Chuang, “ Bulk spin-resonance quantum computation”, Science 275, 350-356(1997).
27. P.W. Shor, “Scheme for reducing decoherence in quantum computer memory”, Phys. Rev. A52, R2493-R2496(1995).
28. P. Benoiff, “ The computer as a physical system: microscopic quantum mechanical Hamiltonian model of Computers as rpresented by Turing Mechines”, J. Stat. Physics, 22 563-591(1980); “Quantum Mechanical Hamiltonian models of discrete processes”, J. Math. Phys. 22, 495-507(1981); “ Quantum Mechanical Model for Turing Machines that dissipate no energy,” Phys. Rev. Lett. 48, 1581-1585(1982); “Quantum Mechanical Hamiltonian models of Turing machines,” J. Stat. Phys. 29, 515-546(1982).
29. D. Deutsch, “ Quantum Theory as a Universal Physical Theory”, Int. J. Theor. Phys. 24, 1-41(1985); “Quantum theory, the Church-Turing principle and the Universal Quantum Computer,” Proc. Roy. Soc. London. A 400, 97-117(1985).
30. M.R. Garey and D.S. Johnson(1979). Computer and Intractability. A Guide to the NP-Completeness. Ney York, NY:W.H Freeman and Company.
31. D. Applegate, R. Bixby, V. Chavatal and W. Cook [1988] on the solution of the Travelling Salesman Problem, Documenta Mathematica 3, 645-756.

32. Held and R. M. Karp[1962]. A Dynamic programming Approach to sequencing Problem.
33. R.Z. Hwang, R.C. Chang and R.C.T. Lee[1993]. The searching over separator strategy to solve some NP-Hard problems in sub exponential time. Algorithemica 9, 398-423.