Quantum Simulated Annealing Algorithm

Dr. Rana Fareed Ghani*

Received on:2/6/2009
Accepted on:1/4/2010

Abstract

Simulated annealing (SA) has been considered as a good tool for search and optimization problems which represent the abstraction of obtaining the crystalline structure through a physical process. This algorithm works sequentially that the current state will produce only one next state. That will make the search to be slower and the important drawback is that the search may fall in local minimum which represent the best solution in only part of the solution space. In this work we present the transformation of Simulated Annealing algorithm into quantum version which will be called Quantum Simulated Annealing (QSA). This algorithm will overcome the drawbacks of slowness and local minimum falling by produce as much as possible of the neighbor states and work on in parallel by exploiting the massive parallelism feature in quantum computation. The results show that QSA can find the optimal path in smaller number of iterations than the sequential simulated annealing algorithm and the time complexity of QSA is better than any other parallel simulated annealing algorithm.

Keywords: Quantum computation, Simulated Annealing Algorithm, Grover Algorithm.

خوارزمية الـــ Simulated Annealing الكمية

الخلاصة

تعتبر خوارزمية الـــ(Simulated Annealing) وسيلة جيدة للبحث ولايجاد الحل الامثل والتي تمثل محاكاة لعملية فيزياوية تتضمن معالجة معينة للحصول على هيكل كرستالي. هذه الخوارزمية تعمل بشكل متسلسل حيث ان الحالة الحالية لا تنتج الا حالة جديدة واحدة فقط. وهذا ما يجعل عملية البحث بطيئة وقد يكون الحل الناتج لا يمثل الحل الامثل. في هذا البحث تم تعديل هذه الخوارزمية وتحويلها الى خوارزمية كمية تسمى (Quantum Simulated Annealing Algorithm) وتختصر الى (QSA). حيث تتجاوز الخوارزمية المعدلة معظم المشاكل الموجودة في الخوارزمية الاصلية والتي تتمثل ببطئ العمل والوقوع بمشكلة الوصول الى الحل شبه الامثل. حيث ان الخوارزمية المعدلة تعالج عدد من الحالات والتي تمثل النقاط المجاورة للحالة الحالية وبشكل متوازي من خلال خاصية المعالجة الكمية المتوازية. نتائج البحث اثبتت ان ال(QSA) تسيتطيع ايجاد الحل الامثل بعدد دورات اقل من الخوازمية الاصلية كما ان تعقيد الوقت في الخوارزمية المعدلة افضل مما هو عليه في الخوارزمية الاصلية واية محاولة سابقة لتحويل الـــ(SA) الى خوارزمية متوازية.

***Computer Science Department, University of Technology/ Baghdad**

https://doi.org/10.30684/etj.28.10.18
2412-0758/University of Technology-Iraq, Baghdad, Iraq

1. Introduction

Simulated annealing (SA) has been considered a good tool for search and optimization problems which represent the abstraction of obtaining the crystalline structure through a physical process which consists of two steps. At the first step a solid will be heated up by increasing the temperature until the liquid phase will be reached. The first step will be followed by the second step which includes the cooling down of the temperature until a minimum energy crystalline structure will be reached. This algorithm works sequentially that the current state will produce only one next state. That will make the search to be slower and the important drawback it will make the search to be fall into local minimum which represent the best solution in only part of the solution space [7] [4].

There have been many attempts to develop parallel versions of the algorithm. These attempts can be categorized into two main approaches. The first one is called massively parallel simulated annealing. This approach works by running the same version of the sequential algorithm on P processors independently as a parallel version of multistart technique. This is well suited for SIMD computers which perform the same instruction in parallel but on different data stream [3]. The second approach is called parallel neighborhood. The essential idea of this approach is to supply the population with spatial structure which may be defined as any connected graph. On each node there is an individual which communicates with its nearest neighbors in the graph. These nearest neighbors are regarded to be the neighborhood of the individual. Most implementations in this approach have been developed for genetic algorithm in order to solve combinatorial optimization problems [9] [6].

The previous work in the field of parallel simulated annealing aims to improve the simulated annealing algorithm either by reducing the time complexity or overcoming the problem of local minimum. Also, some parallel simulated annealing exists for achieving close to ideal speedup on small processor arrays. The work with parallel neighborhood improves the simulated annealing by finding the optimal solution without falling in local minimum but with the drawback of increasing the time complexity of the step that choose the next state from the current state. This step in the sequential algorithm takes O(1) of time complexity but in parallel neighborhood algorithm will take O(N) of time complexity, but this approach gets the optimal solution of the problem in less iterations than the sequential simulated annealing algorithm .

This work presents the transformation of Simulated Annealing algorithm into quantum version which will be called Quantum Simulated Annealing (QSA). This algorithm will overcome the above mentioned drawbacks of slowness and local minimum falling by produce as much as possible of the neighbor states and work on the in parallel by exploiting the massive parallelism feature in quantum computation. All these states will be seen and processed simultaneously to choose the best one of them. This part of the algorithm will have time complexity equals to $O(\sqrt{N})$ while it is equals to O(N) in all the previous

parallel simulated annealing algorithms.

The outline of the paper is as follows. In Section 2, some basic concepts of simulated annealing algorithm are present, Section 3, introduces the concepts of Quantum information processing in particular Grover's algorithm and its variations, Section 4, introduces the idea of the proposed quantum Simulated annealing Algorithm and a brief explanations of the quantum version based on Grover's algorithm that we are exploiting in order to speed-up the algorithms. Then, section 5 presents the results of applying the QSA on the shortest path problem. Finally, section 6 introduces the conclusions.

2. Simulated Annealing Algorithm

Simulated annealing (SA) has been considered a good tool for complex nonlinear optimization problems. Simulated annealing (SA) is a random search technique which exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system; it forms the basis of an optimization technique for combinatorial and nonlinear problems. The technique has been widely applied to a variety of problems. The main advantages and robustness over other search methods are its flexibility and its ability to approach global optimality. On the other hand, the major drawbacks of the technique are its very slow convergence, also the tradeoff between the quality of the solutions and the time required to compute them. The work with SA requires accounting for different classes of constraints and to fine-tune the parameters of the algorithm [7] [4].

In SA, first an initial solution is randomly generated, and a neighbor is found and is accepted with a probability of min(1, exp(2d/T)), where d is the cost difference and T is the control parameter corresponding to the temperature of the physical analogy and will be called temperature. On slow reduction of temperature, the algorithm converges to the global minimum, but the time taken increases drastically. Simulated Annealing is inherently sequential and hence very slow for problems with large search spaces. Several attempts have been made to speed up this process, such as development of parallel simulated annealing techniques and special purpose computer architectures. Algorithm (1) shows the steps of the traditional simulated annealing algorithm.

3. Quantum Information Processing

Quantum information processing draws its power from three quantum resources that have no classical counterpart. Quantum parallelism harnesses the superposition principle and the linearity of quantum mechanics in order to compute a function simultaneously on arbitrarily many inputs [1] [2].

Quantum computer is quite similar to that of a classical computer, both of which have registers. The register in quantum computer initially is set to some initial state. As the quantum computation progresses, the register will be transformed via a quantum mechanical transformation. The output of any process is produced through the observation of the register. This cycle will be repeated

several times before the computation is complete [8].

A classical computer operates on strings of 0's and 1's, such as 1110010101011000, converting them into other strings. Each position in such a string is called a bit, and contains either a 0 or a 1. The only thing a step in a classical computation can do to the state of a bit is to leave it alone, or flip it. A quantum computation can do much more to a quantum bit because quantum bits can have a much broader variety of possible states. In a quantum computer, the fundamental unit of information is represented as the state of quantum subsystems, so the electric potential can be replaced by some quantum state: the quantum bit (qubit for short). Just as a bit has a state 0 or 1, a qubit also has a state $|0\rangle$ or $|1\rangle$. This is called the Dirac notation and it is the standard notation for states in quantum mechanics. The difference between bits and qubits is that a qubit $|\psi\rangle$ can also be in a linear combination of states $|0\rangle$ and $|1\rangle$ [8]:

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

This state is called a superposition of the states $|0\rangle$ and $|1\rangle$. with amplitudes α and β (α and β may be complex numbers). Thus, the state $|\psi\rangle$ is a vector in a two-dimensional complex vector space, where the states $|0\rangle$ and $|1\rangle$ form an orthonormal basis, called the computational basis.

To achieve any computation we need an operator to change a state to another one that will lead to make a progress in the computation process. In quantum computers these operators represent an interaction with the quantum state. These operators are called unitary transformation. Most unitary transformations are linear operator; therefore these operators are applied simultaneously to all the base vectors of a quantum state, this unique feature of quantum computers is called quantum parallelism. Also, all the operations allowed by quantum mechanics are reversible [1] [8].

3.1 Grover's Algorithm

Grover's algorithm is the most famous quantum algorithm that is invented in 1996. In the original version of Grover's algorithm, it is suppose that we have an unstructured database with N elements. Classically, each element is tested at a time, until the searched element is found. This takes an average of N/2 attempts and N in the worst case, therefore the complexity is O(N). Using Quantum Mechanics only O ($\sqrt{N}$) trials are needed to achieve this task using Grover's Algorithm. Grover's algorithm has n qubits register. The first step is to create a superposition of all 2n computational basis states {|0>... |2n − 1>} of the register. Figure (1) shows the diagram of Grover's algorithm steps [10][1].

3.2 Dürr-Hoyer Algorithm

The Dürr-Hoyer algorithm is a quantum algorithm for finding the minimum within an unsorted table of N items. This algorithm is an application of Grover's algorithm. The core of the algorithm is a procedure which returns an index of items smaller than the item determined by a particular threshold.

This procedure is iterated until the minimum is reached [5].

4. Quantum Simulated Annealing

Simulated Annealing algorithm is the one that is used in search and optimization problems, has been improved by adding some of the concepts of quantum computing. The added quantum concepts will improve the time complexity and execution time of some steps within Simulated Annealing.

This work presents a modification of the simulated annealing algorithm that applies Dürr-Hoyer algorithm to the step:

Generate y randomly from N(x);

in algorithm (1) to find the neighbor node to the current node, here after the modified algorithm is called Quantum Simulated Annealing algorithm (QSA).

The Dürr-Hoyer algorithm is used here to find the neighboring node in the algorithm QSA that represent the best nighbour to the node x according to the fitness value of the problem. In the beginning of applying the Dürr-Hoyer in QSA, all the nodes that are in N(x) are stored in the input register with their cost value. The first one of these nodes in input register is stored in the output register, which at the end of the algorithm will contain the best node. Then, the Hd transform distributes the same amplitude to each node in the input register before the application of the oracle function. The oracle function is that the amplitude of the node which has better cost value than the cost of the node in the output register is marked with minus sign. Therefore the oracle function is as follows:

$$f(xi) = \begin{cases} -1 & fitness(xi) < fitness(x0) \\ 1 & otherwise \end{cases} \quad \ldots\ldots\ldots(2)$$

where |xi> is any sub state in the input register and |x0> is the state of the node in the output register. At the end of this step all nodes that have cost value better than the node in the output register will be with negative amplitude.

The diffusion function is used to increase the amplitude of the items that are marked with the minus sign because of the application of the oracle function and decrease in the amplitude of those which are not marked, by the traditional diffusion equation of Grover's algorithm:

$$|\psi_k\rangle = \sin\left[\left(\frac{2*k+1}{2}\right)*\theta\right]|xf\rangle + \cos\left[\left(\frac{2*k+1}{2}\right)*\theta\right]|x1\rangle \quad \ldots\ldots\ldots(3)$$

Where k is the iteration and θ is the rotation angle.

The rotation angle is computed according to the number of nodes in the input register since this algorithm considers that only one solution per iteration (even there are more solutions). Then, if the state of the input register $|\psi\rangle$ is:

$$|\psi\rangle = \frac{\sqrt{N-1}}{\sqrt{N}}|u\rangle + \frac{\sqrt{N-1}}{\sqrt{N}}|1\rangle \quad \ldots\ldots\ldots(4)$$

then the angle θ is computed as follows:

$$\theta = 2arccos\frac{\sqrt{N-1}}{\sqrt{N}} \quad \ldots\ldots\ldots(5)$$

the number of iterations is computed to be smaller than or equal to a maximum number of iterations m which must be determined in the beginning of this algorithm. The value of m must be a small value in its initialization (in most cases its being 1 or 2).

The time complexity of this step in the traditional Simulated algorithm is O(1), while the time complexity of this step in the previous work to implement parallel simulated annealing algorithm is O(N) where N is the number of involved nodes, while the time complexity of this step after using the quantum approach will be O($\sqrt{N}$), but the parallel approach and the QSA will speed up the work of the algorithm and overcome the problem of local minimum. Algorithm (2) shows the steps of the QSA.

5. Experimental Results

In this work the modified algorithm is simulated on a traditional computer to solve the traveling sales man problem (TSP) with different number of cities. Figure (2) shows the relation between the number of cities and the number of iterations in the traditional simulated annealing algorithm and QSA.

Figure (2) shows that QSA can find the optimal solution in smaller number of iterations than the sequential simulated annealing algorithm. The figure no. shows that the number of iteration increases proportional to the increasing in the number of cities in TSP problem using the traditional simulated annealing algorithm, while the number of iteration increase very much slowly according to the increasing in the number of the cities.

Although the time complexity of the traditional simulated annealing is O(1) for the step of choosing the next state which seems to be better than QSA which have a time complexity equals to O($\sqrt{N}$) for this step, But the improvement in the number of iteration can make the convergence of the QSA algorithm to be much better than the traditional simulated annealing algorithm due to the massive parallelism in quantum computation. Also, the QSA chooses the next state as the best neighbor states of current state, therefore; QSA can overcome the problem of falling in local minimum. Also, the QSA is better than any other parallel simulated annealing algorithms since they have O(N) time complexity for the step of choosing the next state from the current state.

6. Conclusions

In this work a transformation of Simulated Annealing algorithm into quantum version which will be called Quantum Simulated Annealing (QSA) is presented. This algorithm will overcome the drawbacks of slowness and local minimum falling by produce as much as possible of the neighbor states and work on in parallel by exploiting the massive parallelism feature in quantum computation. All these states will be seen and processed simultaneously to choose the best one of them. Quantum computation will be applied into the part of choosing the next state from the neighbors of the current state. This part of the algorithm in QSA will have time complexity equals to O ($\sqrt{N}$) while it is equals to O(N) in all the previous parallel simulated annealing algorithms. Also the number of iterations is tested according to the number of cities in the travelling sales man problem. The results show that QSA can find the optimal path in smaller number of iterations than the sequential simulated annealing algorithm. The modified algorithm

can be applied for any optimization or search problem that can be solved using the traditional simulated annealing algorithm.

References

[1] Aimeur E., Brassard G., Gambs S., "Quantum Clustering Algorithms", Proceedings of the 24th International Conference on Machine Learning, Corvallis, 2007.

[2] Chi D., Kim J., "Quantum Computation", Technical paper, Department of Mathmatics, Seoul National University, 1997.

[3] Chu K., Deng Y, Reinitzy J., "Parallel Simulated Annealing by Mixing of States", Journal of Computational Physics 148, 646–662 , 1999.

[4] De Weck O., Jilla C.," Simulated Annealing, A Basic Introduction", Massachusetts Institute of Technology,2005.

[5] Durr C., Hoyer P., "A Quantum algorithm for Finding the Minimum", quanta-ph/9607014 V2, 1999.

[6] Janaki Ram D., Sreenivas T. H., Subramaniam K. G, "Parallel Simulated Annealing Algorithms", Journal Of Parallel And Distributed Computing 37, PP207–212, 1996.

[7] Kirkpatrick S, Gelatt D., Vecchi M. P., "Optimization by Simulated Annealing", Science, 220(4598):671–680, 1983.

[8] Mermin N. D., "Lecture Notes on Quantum Computation and Information Theory", Cornell University, Physics 481-681, CS 483, Fall, 2000.

[9] Rudolph G., "Massively Parallel Simulated Annealing and its Relation to Evolutionary Algorithms", technical paper, University of Dortmund, 1997.

[10] Viamontes G.F., Markov I. L., Hayes J. P., "Is Quantum Search Practical ?", IEEE Computing in Science and Engineering, pp. 22-30, May/June 2005.

Algorithm (1) Traditional Simulated Annealing Algorithm

Input: x = x0 {randomly chosen initial point }
T = T0 {Initial temperature value}
Output: S {the optimal solution}

```
Begin
x = x0 ; T = T0 ;
do While (termination-criterion has not been met),
    For i: = 1 to L do
        Generate the neighbor of x, N(x);
        Generate y randomly from N(x);
        lf (fitness (y) - fitness (x) ≤ 0) Then
          x: = y
        Else_if (exp [- (fitness(y )- fitness(x) )/ T] > random [0,1])
then
          x: = y
    End {for}
    Decrement T;
End. {while}
S=x;
End. {algorithm}
```

Algorithm (2) Quantum Simulated Annealing Algorithm

Input: x = x0 {randomly chosen initial point }
T = T0 {Initial temperature value}
Output: S {the optimal solution}

Begin
x = x0 ; T = T0 ;
do While (termination-criterion has not been met),
For i: = 1 to L do
generate all the neighbor states N(x)={x1,x2,…..,xn};
Repeat
Put the nodes N(x)=(x1,x2,…,xn) in input quantum register and x1 in output register.
Apply H transformto all the nodes inN(x) to compute the initial amplitude Node(xi).

$$\psi(xi) = \frac{1}{\sqrt{N}}$$

Compute the rotation angle $\vartheta = 2arccos\frac{\sqrt{N-1}}{\sqrt{N}}$

Compute the number of iterations $k = \frac{\pi - \theta}{2\theta}$

Apply the Oracle Function to sign with minus the node xj, that satisfies the following:

$$f(xi) = \begin{cases} -1 & fitness(xi) < fitness(xo) \\ 1 & otherwise \end{cases}$$

Apply the diffusion function

$$|\psi_k\rangle = \sin\left[\left(\frac{2*k+1}{2}\right)*\theta\right]|xj\rangle + \cos\left[\left(\frac{2*k+1}{2}\right)*\theta\right]|x1\rangle$$

Measure Input-Register to find a solutions Sx
Put Sx in Output-Register, y=Sx;
Until Sx = Ø

lf (fitness (y) - fitness (x)≤ 0) Then
x: = y
Else_if (exp [- (fitness(y)- fitness(x))/ T] > random [0,1]) then
x: = y
End {for}
Decrement T;
End. {while}
S=x;
End. {algorithm}

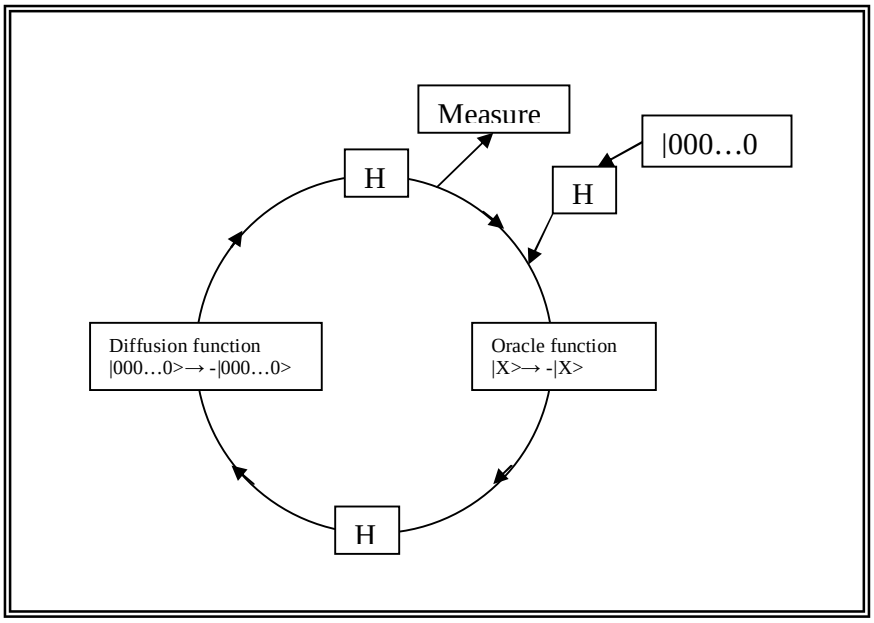

Figure (1) Grover's Algorithm Diagram

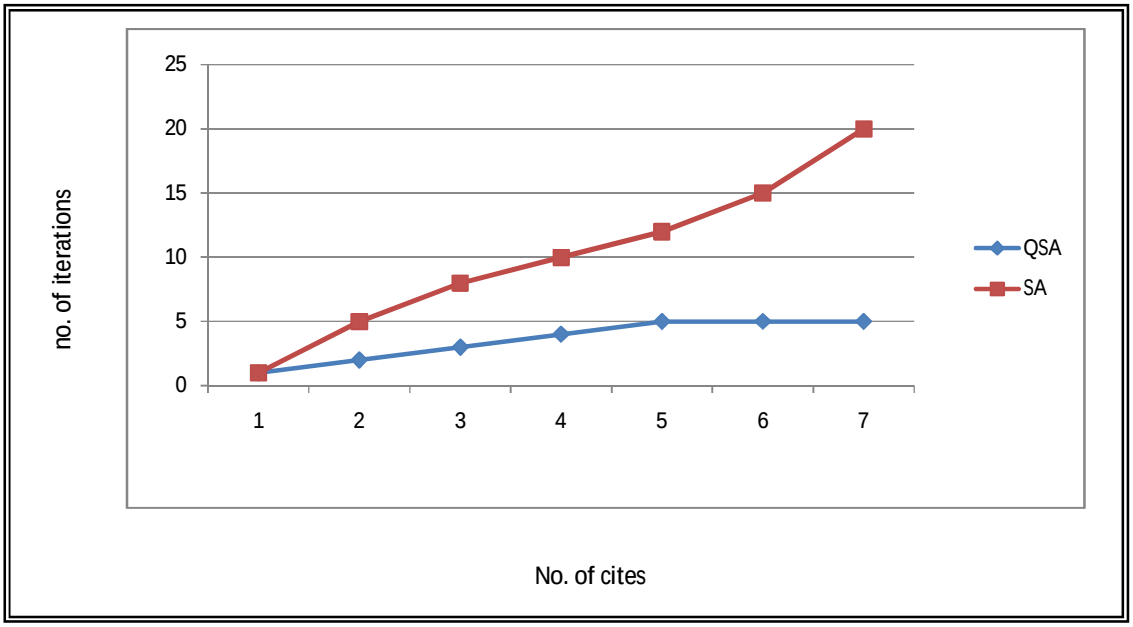

Figure (2) The Relation Between No. Of Cities And No. Of Iterations In SA And QSA