Algorithm Dynamics Analysis Method

Wang Peng
Parallel Computing Laboratory, Chengdu University of Information Technology, Chengdu, China
University of Electronic Science & Technology of China, Chengdu, China
Email: wp002005@163.com

***Abstract*—The concepts of algorithm dynamics are proposed. Simulated annealing algorithm is analyzed by algorithm dynamics method. In the optimization progress metropolis criteria acts as an important role to construct simulated annealing algorithm. The metropolis criteria leads algorithm system to the ground state. At high temperature metropolis criteria make the algorithm system move like free particles, in this stage classic thermodynamics is suitable to analyze the algorithm. At low temperature metropolis criteria restricts the algorithm system vibration in the local area like crystal lattice vibration, in this stage quantum theory is suitable to analyze the algorithm. According to quantum dynamics the solution's distribution of algorithm is Gaussian function. This results can interpret the implicit parallelism of algorithm. Uncertainty principle of algorithm (UPA) is proposed based on algorithm dynamics too. It indicates that precision and implicit parallelism of algorithm can't achieve at the same time. Finally computational complexity is analyzed by thermodynamics. This method can get the lower bound of the algorithm easily. The computational complexity of sort problem's lower bound is analyzed by thermodynamics method. The computational complexity is decided only by initial state and final state of the problems. The algorithm's details are needless in this method.**

***Index Terms*—algorithm dynamics, simulated annealing algorithm, implicit parallelism, quantum mechanics, thermodynamics, uncertainty principle of algorithm**

I. INTRODUCTION

Everything in the world is motive including algorithms. Algorithm is a time evolution system. In physics the term dynamics customarily refers to the time evolution of physical processes. These processes may be microscopic as in particle physics, kinetic theory, and chemical reactions, or macroscopic as in the predictions of statistical mechanics and nonequilibrium thermodynamics[1]. Dynamics has a complete theoretical system. It would become a powerful tool to research the algorithm. We named this method algorithm dynamics.

Bornholdt introduced a dynamics annealing schedule for population-based optimization algorithm in 1999.[2] Then Bennett researched the mathematical modeling of genetic algorithm dynamics in 2001[3]. Thermodynamics is used to research genetic algorithm in his theory[4, 5, 6]. Quantum is also used to analyze genetic algorithm by some researchers[7]. This method is called algorithm dynamics. In fact a lot of optimization algorithms haven't complete theoretical foundation though they have been used in some fields widely [8], [9], [10], [11], [12], [13]. We need a powerful tool to help us analyze the algorithm. Quantum mechanics and thermodynamics are important dynamics theories in physics. In this paper we use them to analyze simulated annealing algorithm and computational complexity.

II. DYNAMICS ANALYSIS OF SIMULATED ANNEALING ALGORITHM

The computing progress of algorithm is a dynamic process similar to other movement process in nature. A lot of algorithms haven't set up a complete theoretical foundation especially some intelligent optimization algorithm. We think that the dynamics theory would be a useful tool to analyze characteristics of algorithms. This method is named algorithm dynamics. In this section we will analyze the simulated annealing algorithm by dynamics theory.

Simulated annealing (SA) is a generic probabilistic metaheuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global minimum of a given function in a large search space. It is often used when the search space is discrete (e.g., all tours that visit a given set of cities). For certain problems, simulated annealing may be more effective than exhaustive enumeration — provided that the goal is merely to find an acceptably good solution in a fixed amount of time, rather than the best possible solution.

This method was independently described by S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi in 1983,[14] and by V. Černý in 1985. The method is an adaptation of the Metropolis-Hastings algorithm, a Monte Carlo method to generate sample states of a thermodynamic system, invented by N. Metropolis et al in 1953.[15]

A. Simulated Annealing Algorithm's Dynamics Process

The simulated annealing algorithm's inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one. The result of competition between energy and entropy is equilibrium. The temperature decides the

doi:10.4304/jcp.5.2.178-185

weight of energy and entropy. At low temperature energy is the main factor; at high temperature entropy is the main factor. So annealing process is divided into two stages: High temperature and low temperature.

High temperature: This stage is classic thermodynamics process. According to metropolis criteria ( ) algorithm accepts the worse solution at high probability. The algorithm's dynamics behavior is random movement in search space. In this stage metropolis criteria act as the driving force of algorithm movement. We can use classic thermodynamics and Brownian motion theory to describe this stage. We named it classic dynamics stage.

Low temperature: This stage is quantum mechanics process. Metropolis criteria ( ) restricts the solution to move in the search space freely, because it accepts the worse solution at very low probability. Crystal lattice vibration theory could describe the algorithm's dynamics behavior at low temperature. In this stage energy becomes the main factor of the algorithm system. We named it quantum dynamics stage.

Figure 1 describes the two stages of simulated annealing algorithm with different temperature. Along with temperature improves the entropy of algorithm system improves too. At high temperature the algorithm system can be analyzed by classic thermodynamics; at low temperature the algorithm system can be analyzed by quantum mechanics.

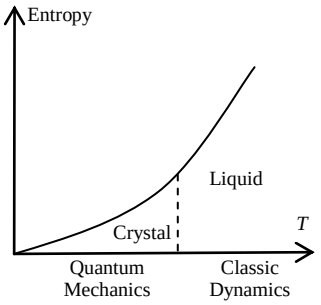

Figure 1. Two stages of simulated annealing algorithm

B. Low Temperature Dynamics Behavior of Simulated Annealing Algorithm

When the temperature T is low enough the simulated annealing algorithm's behavior like a kind of crystal lattice vibration. The algorithm system is restricted within local area at low temperature by metropolis criteria. We can use the crystal lattice vibration theory to explain the algorithm's behavior. The only problem is that the vibration is classic or quantum vibration.

In order to explain the algorithm dynamics at low temperature, we must compare the solid theory with simulated annealing algorithm. The crystal lattice can be seen as the search space of algorithm; the equilibrium position of atom is position of solution.

Assume that the equilibrium position of the n th atom is x_n^0 , Deviation from the equilibrium position displacement is u_n . The n th atom's instantaneous position is

$$x_n = x_n^0 + u_n \tag{1}$$

Let $\varphi(x_m + x_n)$ is the potential energy between two atoms. The potential energy of the entire lattice is the sum of all the pairwise potential energies:

$$\begin{aligned} V_n &= \sum_{m(m\neq n)}^{N} \varphi(x_m - x_n) \\ &= \sum_{m(m\neq n)}^{N} \varphi(x_m^0 - x_n^0 + u_m - u_n) \\ &= \sum_{m(m\neq n)}^{N} \varphi(x_{mn}^0 + u_{mn}) \end{aligned} \tag{2}$$

Where

$x_{mn}^0 = x_m^0 - x_n^0$,

$u_{mn} = u_m - u_n$

$x_{mn} = x_m - x_n = x_{mn}^0 + u_{mn}$

Exponding of potential energy V_n is

$$\begin{aligned} V_n = &\sum_{m(m\neq n)}^{N} \varphi(x_{mn}^0) + \sum_{m(m\neq n)}^{N} (\frac{\partial \varphi}{\partial x_{mn}}) u_{mn} \\ &+ \frac{1}{2} \sum_{m(m\neq n)}^{N} (\frac{\partial^2 \varphi}{\partial x_{mn}^2}) u_{mn}^2 + \cdots \end{aligned} \tag{3}$$

Where $V_n^0 = \sum_{m(m\neq n}^{N} \varphi(x_{mn}^0)$, $(\frac{\partial \varphi}{\partial x_{mn}}) = 0$

So $V_n = V_n^0 + \frac{1}{2} \sum_{m(m\neq n)}^{N} (\frac{\partial^2 \varphi}{\partial x_{mn}^2}) u_{mn}^2 + \cdots$ (4)

Suppose stubborn coefficient $K_{mn} = (\frac{\partial^2 \varphi}{\partial x_{mn}^2})$

$$V_n = V_n^0 + \frac{1}{2} \sum_{m(m\neq n)}^{N} K_{mn} u_{mn}^2 + \cdots \tag{5}$$

The n th atom's restoring force is

$$F_n = -\frac{\partial V_n}{\partial u_n} = -\frac{1}{2} \frac{\partial}{\partial u_n} \sum_{m(m\neq n)}^{N} K_{mn} u_{mn}^2 + \cdots \tag{6}$$

If we only consider harmonic oscillation and nearest neighbor atoms the restoring force F_n can be written as follows:

$$F_n = -K(u_{n+1} + u_{n-1} - 2u_n) \tag{7}$$

According to this equation, we can conclude that at low temperature the dynamics behavior of simulated annealing algorithm is similar to harmonic oscillation of crystal lattice (Figure 2). The algorithm's search space is restricted to vibrate around the solution by metropolis criteria at low temperature.

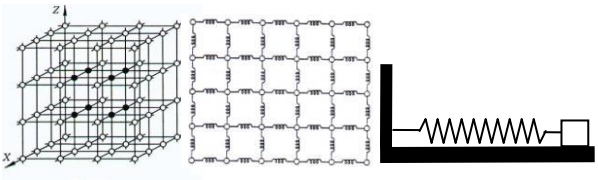

Figure 2. Dynamics behavior of simulated annealing algorithm at low temperature

C. Simulated Annealing Algorithm's One-dimensional Quantum Harmonic Oscillator Description at Low Temperature

Simulated annealing algorithm's dynamic behavior is not like classic crystal lattice vibration at low temperature. We can use quantum harmonic oscillator theory to explain algorithm dynamic. There are two reasons:

1. Though the metropolis criteria restricts the algorithm vibrate in the local area at low temperature but the metropolis criteria also allow the algorithm jump out the local area at low probability. The system may move to the new state even when it is worse than the current one.

2. The solution of simulated annealing algorithm is not exact solution. It is only a kind of probability distribution like quantum wave function.

Based on quantum harmonic oscillator theory and ensemble theory the group behavior of algorithm system can be equivalent to harmonic oscillator drived by generalized restoring force according to complex system theory. Comparing the harmonic oscillator we define F as generalized restoring force, K as generalized stubborn coefficient, x as displacement from solution. According to Hooke's law ($F=-Kx$) generalized restoring force F proportional to the displacement x . The solution of problem is just the equilibrium position of harmonic oscillator ($x=0$). The generalized potential energy of the algorithm system is $V(x)=\frac{1}{2}Kx^2$ (Figure 3).

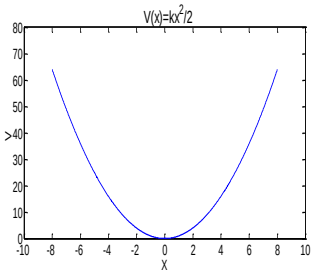

Figure 3. Generalized potential energy curve of algorithm

This is the Hooke theorem. Let denotation m represents the mass of harmonic oscillator. Then $\omega=\sqrt{K/m}$, ω is frequency of classic harmonic oscillator. The Hamiltonian for this system is

$$H=\frac{p^2}{2m}+\frac{1}{2}m\omega^2x^2 \qquad (8)$$

Where operator p is momentum operator $p=-i\hbar\frac{d}{dx}$

So stationary state Schrodinger equation of harmonic oscillator can be written as follows

$$[-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}+\frac{1}{2}m\omega^2x^2]\psi(x)=E\psi(x) \qquad (9)$$

Let $\xi=ax \quad a=\sqrt{m\omega/\hbar}$

The solution of Schrodinger equation is

$$\psi_n(x)=A_ne^{-a^2x^2/2}H_n(ax) \qquad (10)$$

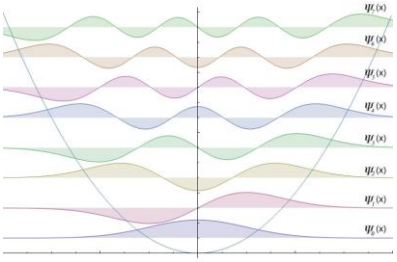

Figure 4. Wave function of quantum harmonic oscillator

Figure 4 describes the wave function of quantum harmonic oscillator at different energy levels. It is a typical quantum wave function.

$H_n(ax)$ is Hermite polynomial.

The energy levels of different wave function is

$$E_n=(n+1/2)\hbar\omega \qquad (11)$$

The wave function of ground state is $\psi_0=A_0e^{-a^2x^2/2}$. Its form is Gaussian function. According to the Copenhagen interpretation of quantum mechanics the wave function is probability. It tells us that the solution of problem is not exact. It only appears in search space according to Gaussian distribution. We know that the simulated annealing algorithm can't obtain the problem's exact solution. The quantum harmonic oscillator theory accords with this phenomenon exactly. It indicates that an algorithm system must on ground state of quantum mechanics and thermodynamics when the optimization progress stops.

We know that when the algorithm system is at ground state the problem's solution distribution is Gaussian function. The Gaussian curve shape is decided by generalized stubborn coefficient K of algorithm system.

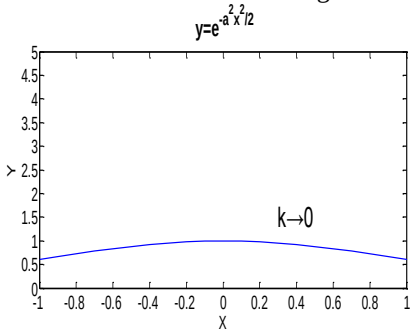

Figure 5. Wave function of ground state($k\to 0$)

If $K \to 0$ the shape of Gaussian curve is a wide curve(Figure 5). It implies that algorithm can obtain the exact solution at low probability. If $K = 0$, $\psi_0 = A_0$, A_0 is constant. It implies that algorithm hasn't information about position of solution.

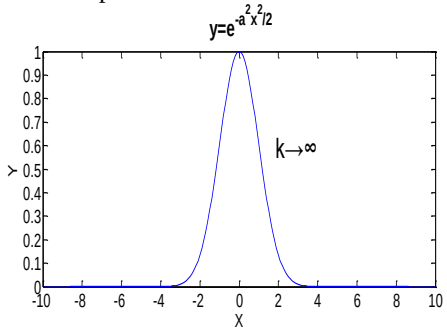

Figure 6. Wave function of ground state($k \to \infty$)

If $K \to \infty$ the shape of Gaussian curve is a narrow curve(Figure 6). It implies that algorithm can obtain the exact solution at high probability. If $K = \infty$ ψ_0 is δ function. At this condition the algorithm can obtain the exact solution.

For generalized stubborn coefficient K, $K \propto Inf$, Inf is information. If Inf =0 the wave function is constant we can't get the solution of problem by algorithm; If $Inf = \infty$ the wave function is δ function we can get the exact solution of problem by algorithm. The value of Inf is decided by heuristic rules, iteration number etc.

For classic harmonic oscillator model $|x| > a^{-1}$ is forbidden region. But for quantum harmonic oscillator model the solution of problem can appear in classic forbidden region at low probability, this phenomenon is accord with the metropolis criteria.

Based on the above discussion we consider that simulated annealing algorithm constructs a kind of quasi quantum system its dynamic behavior accords with quantum mechanics at low temperature.

D. Implicit Parallelism of Simulated Annealing Algorithm

Implicit parallelism is an important character of some intelligent optimization algorithm. The implicit parallelism of genetic algorithm has been research by Holland, Dorigo, Wright, Ding Li-xin, Zhang Guang-duo et al. [16], [17], [18], [19], [20], [21] But the essences of implicit parallelism haven't be realized by scientists. Implicit parallelism makes some algorithms more effective than exhaustive enumeration. We consider that the implicit parallelism is an important character of simulated annealing algorithm. The intelligent optimization algorithm's goal is merely to find an acceptably good solution in a fixed amount of time, rather than the best possible solution. This implies that the implicit parallelism is related to uncertainty. The quantum harmonic oscillator model of algorithm has told us this result.

The ground state of quantum harmonic oscillator dynamics model of algorithm is Gaussian wave function $\psi_0 = A_0 e^{-a^2 x^2/2}$. Gaussian wave function indicates that we can't get the exact solution of problem with 100% probability. Probability amplitude of ground state is(Figure7):

$$|\psi_0(x)|^2 = \exp(-a^2 x^2) \tag{12}$$

The solution appears at interval $|x| \le \frac{1}{a}$, so

$$\Delta x \approx \frac{1}{a} \tag{13}$$

Δx is uncertainty degree of solution. If $\Delta x = 0$ the algorithm gets the exact solution and hasn't implicit parallelism. The algorithm becomes the exhaustive enumeration algorithm.

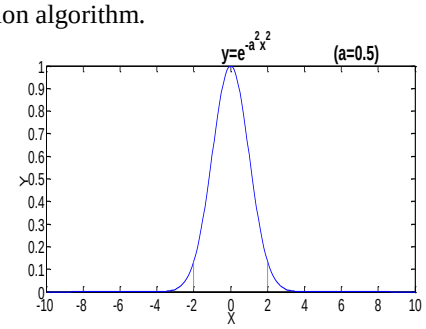

Figure 7. Gaussian wave packet

The Fourier translation of $\psi_0(x)$ is (Figure 8)

$$\varphi(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \psi_0(x) e^{-ikx} dx = \frac{1}{a} \exp(-k^2/2a^2)$$

$$|\varphi(k)|^2 = \frac{1}{a^2} \exp(-k^2/a^2) \tag{14}$$

So

$$\Delta k \approx a \tag{15}$$

Δk is named implicit parallelism degree.

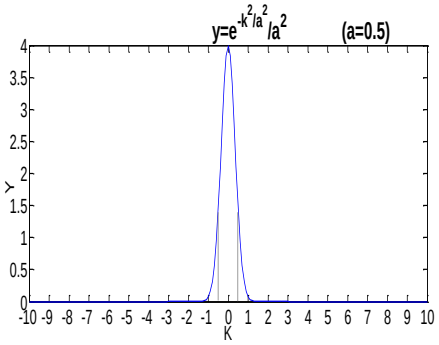

Figure 8. Fourier translation of Gaussian wave packet

According to Δx and Δk we get uncertainty principle of algorithm(UPA)

$$\Delta x \cdot \Delta k \approx 1 \tag{16}$$

This equation is the uncertainty principle of algorithm like Heisenberg uncertainty principle of quantum mechanics. [22] Uncertainty principle of algorithm applicable to analyze any algorithms for example simulated annealing algorithm, genetic algorithm and ant colony algorithm. k named to implicit parallelism operator.

Though the uncertainty principle of algorithm is deduced from simulated annealing algorithm by quantum harmonic oscillator it is a universal rule for any algorithm. Uncertainty principle of algorithm indicates that any algorithm can't get high precision of solution and high implicit parallelism degree at the same time. If we want to construct an algorithm with high implicit parallelism degree we must reduce the precision of solution. Uncertainty principle of algorithm is an important dynamics law of algorithm when we analyze the algorithms.

III. THERMODYNAMICS ANALYSIS OF COMPUTATIONAL COMPLEXITY

In 1971 S.Cook laid the foundations for the theory of NP-Completeness [23], [24], [25]. Computational complexity theory, as a branch of the theory of computation in computer science, investigates the problems related to the amounts of resources required for the execution of algorithms (e.g., execution time), and the inherent difficulty in providing efficient algorithms for specific computational problems[26]. Getting lower bound of the problem is the chief task of computational complexity though we don't know the details of the algorithm.

In theoretical analysis of algorithms it is common to estimate their complexity in asymptotic sense, i.e., to estimate the complexity function for reasonably large length of input. Big O notation, omega notation and theta notation are used to this end. For instance, binary search is said to run an amount of steps proportional to a logarithm, or in O(log(n)), colloquially "in logarithmic time". Usually asymptotic estimates are used because different implementations of the same algorithm may differ in efficiency. However the efficiencies of any two "reasonable" implementations of a given algorithm are related by a constant multiplicative factor called hidden constant.

Generally, the lower bound of computational complexity is the important target of Algorithm Optimization. The real challenge in complexity theory, and the problem that sets the theory apart from the analysis of algorithms, is proving lower bounds on the complexity of specific problems. From 1970s, some people used energy and entropy to research the computing process[27]. Energy and entropy are the important physical quantities. Entropy has been used as an important concept in information theory by Shannon[28]. In this paper we use energy method to analyze the computational complexity and entropy is also used to calculate the lower bound of energy consumption. Based on the calculation of the problem's lower bound of energy consumption energy method can get the lower bound of the computational complexity easily. It is the physical lower bound of the computational complexity and reflects the problem's essences. Energy method is a new method to calculate the computational complexity and can also be used to research NP-Completeness problems. Computational complexity is decided only by initial state and final state of problems.

Computer system (include hardware and software) is also a kind of thermodynamics system. Energy and entropy are the important physical quantities to analyze the thermodynamics system. In this paper we use energy and entropy to compute the lower bound of energy, thereby get the lower bound of computational complexity.

A. Landauer's Theorem

Classic computer's work process is an irreversible process. It is an energy-consuming process requires the input of energy.

In this section we consider the computer system as a thermodynamics system. In 1961 Landauer [29]has proved energy consumption of one bit by using thermodynamics theory.

Landauer theorem: Energy consumption of 1 bit is $kT\ln 2$.

Where k is Boltzmann constant ($k = 1.380658 \times 10^{-23}\ J/K$) , T is system temperature.

Landauer's theorem is the lower bound of energy comsumption.

B. Energy Consumption Theory

First we define Clausius entropy ∇S

$$\Delta S = S_2 - S_1 = \int_1^2 \frac{dQ}{T} \tag{17}$$

Here S_1 and S_2 are initial state entropy and final state entropy respectively, ∇S is the entropy change of system. Q is the quantity of heat.

For isothermal change process, system temperature T is constant, so

$$\Delta S = \frac{1}{T}\int_1^2 dQ = S_2 - S_1 \tag{18}$$

$$\Delta Q = Q_2 - Q_1 = T(S_2 - S_1) \tag{19}$$

We know temperature units is K (Kelvin) and entropy units is J/K . So the units of $T(S_2 - S_1)$ is energy units J (Joule).

Theorem: The system's energy consumption can be calculate using $\nabla Q = T(S_2 - S_1)$, if $\nabla Q > 0$ the system will release the energy; if $\nabla Q < 0$ the system need input the energy.

According to this theorem the energy consumption can be calculate when S_1 , S_2 and T are determined. In other words we must know initial state entropy and final

state entropy of system. The energy change is proportional to the entropy change. Usually the entropy Boltzmann expression $S = k \ln \Omega$ is used to calculate the entropy of system. Where Ω is probability of macroscopical state.

It is obvious that computational complexity is proportional to energy consumption of system. So we define the energy asymptotic notation *E*:

Definition of *E*: Given nonnegative functions $f(n)$ and $g(n)$ defined on the positive integers, we write $f(n) = E(g(n))$ if and only if there is a positive constant C and an integer N such that $C \times g(n) \le f(n)$, for all $n > N$.

There $f(n)$ and $g(n)$ represent the energy consumption function of the problem, $f(n) = |\nabla Q| = |T(S_2 - S_1)|$, n is the problem scale.

The energy consumption and the computing time have direct ratio relations so we propose the equivalent theorem.

Equivalent theorem: $E(g(n)) = \Omega(g(n))$

According to this theorem we can use lower bound of energy consumption to estimate the computational complexity. $E(g(n))$ is the lower bound of the problem and is independent of algorithm itself. This method reflects the algorithm's essences.

C. Energy Consumption of One Bit

For all algorithm the work is select the results from the results space. This process is an entropy increase process so we must input energy into this system. In this section we will prove the lower bound of energy consumption when the computer operates one bit.

One bit operate process of classic computer is shown in Figure 9.

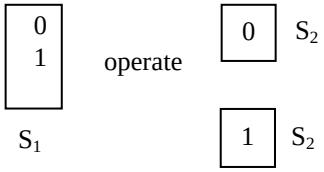

Figure9. Operator one bit of classic computer

Before the computer operates this bit the system is in the superposition state. The probability of this state is 1. We don't know whether this bit is 0 or 1. It is a kind of balance state and has the maximal probability. After the computer operates this bit the state of the system is determined. Whatever the result is 0 or 1 it has 1/2 probability.

According to the entropy Boltzmann expression $S = k \ln \Omega$ and function (2) we can get to the lower bound of one bit. Before we operator this bit Ω is equal to 1. So the entropy of the system S_1 is

$$S_1 = k \ln 1 \tag{20}$$

After we operator this bit Ω is equal to 1/2. So the entropy of the system S_2 is

$$S_2 = k \ln 1/2 \tag{21}$$

This process is a kind of irreversible process it must consume the energy. The consumption energy ΔQ is

$$\Delta Q = T(S_2 - S_1) = T(k \ln 1/2 - k \ln 1) = -kT \ln 2 \tag{22}$$

So the lower bound when computer operates one bit is $kT \ln 2$ J.

D. Thermodynamics Analysis of Sort Problem

It is well known that the computational complexity lower bound of sort problem is nlogn. In this section we will prove it using thermodynamics analysis method[30].

Sort algorithm is a kind of widely used algorithm. First we take example for ascending sort of three numbers. The example is shown in Figure 10.

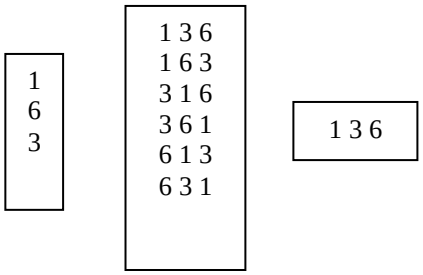

Figure 10. Ascending sort of three numbers

Before sorting the three numbers 1, 6, 3 are out of order. This is a kind of balance state and the appearance probability is 1. So $\Omega = 1$. Three numbers have 6 permutations. The ascending permutation is only one permutation of three numbers and its probability is 1/6. After sorting $\Omega = 1/6$. The energy consumption can be written as

$$Q = kT(\ln 1/6 - \ln 1) = kT \ln 1/6 = -kT \ln 6 \tag{23}$$

Sort algorithm is entropy reducing process. After sorting the order of system is increased, this process is an energy consumption process. When the sort numbers are n the permutations is $n!$. After sorting the probability of the system Ω is $1/n!$. We can get the lower bound of energy consumption.

$$Q = kT(\ln 1/n! - \ln 1) = kT \ln 1/n! = -kT \ln n! \tag{24}$$

According to the energy asymptotic the notation *E* is $E(\log n!)$. It is obvious that $\log(n!) = \Theta(n \log n)$. So the lower bound of energy consumption is $E(n \log n)$. We have proposed energy consumption and computational complexity are direct ratio $E(g(n)) = \Omega(g(n))$. So the computational complexity lower bound of sort algorithm is $\Omega(n \log n)$.

Thermodynamics analysis method reflects the algorithm's physics essences and can give physics limit of the algorithm. The thermodynamics tells us that we can't find a sort algorithm whose computational complexity is lower than $\ln n!$ and the lower bound of sort problem is independent of details of sort algorithm.

Ⅳ. CONCLUSIONS

In this paper dynamics theories are used to analyze the algorithm. Simulated annealing algorithm is divided to two stages by dynamics theories. At high temperature the simulated annealing algorithm is described as thermodynamics system. At low temperature the simulated annealing algorithm is described as quantum harmonic oscillator system. The solution distribution of simulated annealing algorithm can be described by Gaussian function if the algorithm is seen as quantum harmonic oscillator. In this theoretical framework implicit parallelism and uncertainty principle of algorithm （UPA）are proposed and analyzed. The results are accord with the fact.

Finally we use thermodynamics analysis method to get the computational complexity of the problem. The thermodynamics concept entropy is used in this method. This method reflects the problem's physics essences. We needn't known the details of the algorithm. Energy analysis is a simple method it only using the probability of initial state and final state to get the computational complexity. In this paper we calculate the computational complexity of one bit operator and sort problem by energy method. Any algorithm can't break through the lower bound limit calculated by energy method because it can't break through the energy consumption lower bound limit. It is a physics limit and the physics theory forbids the algorithms to break this limit.

Dynamics theory is an important theory. It includes quantum mechanics, thermodynamics, electrodynamics etc. If we put algorithm as a dynamics system then dynamics theory can be used to analyze the algorithm. In this paper solution distribution, implicit parallelism, computational complexity and uncertainty principle of algorithm are researched by algorithm dynamics. Algorithm dynamics would become a kind of powerful tool to analyze the algorithm. There is one important problem requiring further research: that is why the algorithm selects ground state.

ACKNOWLEDGEMENT

This work was supported in part by a grant from National Natural Science Foundation of China (60702075); China Postdoctoral Science Foundation (20070410385); Development Foundation of CUIT (KYTZ200819); Sichuan Education Office Foundation (07ZA014); Sichuan Youth Science & Technology Foundation(09ZQ026-068).

REFERENCES

[1] Niels E. Henriksen, Flemming Y. Hansen (2008). Theories of Molecular Reaction Dynamics: The Microscopic Foundation of Chemical Kinetics. Oxford University Press.
[2] Stefan Bornholdt. Annealing Schedule from Population Dynamics. Physical Review E, 1999, 59(4): 3942-3947.
[3] Prügel-Bennett, A. and Rogers, A. Modelling GA Dynamics. In: Theoretical Aspects of Evolutionary Computing, Natural Computing. 2001：59-86.
[4] Rogers, A. and Prügel-Bennett, A. The Dynamics of a Genetic Algorithm on a Model Hard Optimization Problem. Complex Systems, 2000,11 (6): 437-64.
[5] Rogers, A. and Prügel-Bennett, A. Modelling the Dynamics of a Steady State Genetic Algorithm. In: Foundations of Genetic Algorithms 1999 ,5:57-68.
[6] Prügel-Bennett, A. and Shapiro, J. L. An analysis of genetic algorithms using statistical mechanics. Physica D, 1997,104 : 75-114.
[7] Wang Peng. Quantum Model of Genetic Algorithm. 2008 International Symposium on Knowledge Acquisition and Modeling: IEEE Press 2008:576-580.
[8] J.Holland. Adaptation in Natural and Artificial System, MIT Press,1975.
[9] J. H. Holland, Genetic algorithms, Scientific American, 267(1992)1, 66-72.
[10] Colorni A, Dorigo M, Maniezzo V, et al. Distributed optimization by ant colonies. Proceedings of the 1st European Conference on Artificial Life, 1991, 134-142.
[11] Bonabeau E, Dorigo M, Theraulaz G. Inspiration for optimization from social insect behavior. Nature, 2000,406(6):39-42.
[12] Michael J B K, Jean-Bernard B, Laurent K. Ant-like task and recruitment in cooperative robots. Nature, 2000, 406(31): 992-995.
[13] Jackson D E, Holcombe M, Ratnieks F L W. Trail geometry gives polarity to ant foraging networks. Nature, 2004,432(7019):907-909.
[14] Kirkpatrick S, Gelatt Jr C D, Vecchi M P. Optimization by simulated annealing. Science, 1983, 220: 671-680.
[15] Metroplis N, Rosenbluth A, Rosenbluth M et al. Equation of state calculations by fast computing machines. Journal of Cherimal Physics, 1953, 21: 1087-1092.
[16] Darrell Whitley， Deception, dominance and implicit parallelism in genetic search， Annals of Mathematics and Artificial Intelligence, 5 (1992) 49-78 49
[17] Zhang Guang-duo, Wang Zheng-zhi. The Theoretic Research of Implicit Parallelism in the Processing of Genetic Algorithms. IEEE International Conference on Systems, Man, and Cybernetics ,Volume 3, 14-17 Oct 1996 :1922 – 1924.
[18] Ding Li-xin, Kang Lishan.Analysis of implicit parallelism in evolutionary algorithms: a stochastic version. Fourth International Conference on Computational Intelligence and Multimedia Applications,2001 :172 – 179.
[19] Alden H. Wright, Michael D. Vose and Jonathan E. Rowe. Implicit Parallelism. Proceedings of GECCO 2003, Springer Verlag Lecture Notes in Computer Science, 2003: 1003-1014.
[20] Bertoni Alberto, Dorigo Marco. Implicit parallelism in genetic algorithms. Artificial Intelligence, v 61, n 2, Jun, 1993, 307-314
[21] Wang Peng. Explaining the Implicit Parallelism of Genetic Algorithm and Computational Complexity by Quantum Theory. 2008 International Symposium on Computer Science and Computational Technology Volume 1. IEEE Press 2008:463-466.

[22] W. Heisenberg. Uber den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik, Z. Phys. 43, 172-198, (1927).
[23] Cook S.A. The complexity of theorem proving procedures, Proceedings of the Third Annual ACM Symposium on Theory of Computing, 151–158, 1971.
[24] Cook S.A. An overview of computational complexity. Communications of the ACM, 26(6):400-408, June 1983.
[25] Cook S.A. Aanderaa S.O. On the minimum computation time of functions. Trans. AMS 1969(142),291-314.
[26] Rabin M.O. Complexity of Computations. Communications of the ACM, 20(9):625-633, September 1977.
[27] Bennett C H, Logical reversibility of computation, IBM Journal of Research and Development, 1973, 17(6): 525-532.
[28] Shannon C E. A mathematical theory of communication. The Bell System Technical Journal , 1948 , 27 : 379-423 , 623-656
[29] Landauer R, Irreversibility and heat generation in the computing process, IBM Journal of Research and Development. 2000, 44(1):261-269.
[30] Knuth D.E. The art of computer programming, vol. 3 Sorting and Searching, Addison-Wesley, Reading, MA. 1973.

Wang Peng was born in Sichuan, China in 1975. He received his Doctor Diploma from the Institute of Computer Application, Chinese Academy of Science in 2004.

He holds an Associate Professor position and Director of Parallel and Quantum Computing Laboratory of CUIT since 2006. He is committeeman of HPCTC of CCF. He was appointed to chief engineer of ground station of SAR of CETC 38 in 2004. Now he has published about 60 papers in leading journals and applied 4 patents. His research interests include parallel computing, quantum computing and nuclear physics. Now he acts as project leader of some important projects such as National Natural Science Foundation of China and China Postdoctoral Science Foundation. He also acts as reviewer of some leading journals and master advisor of Chinese Academy of Sciences.