MAXIMUM TOLERABLE ERROR BOUND IN DISTRIBUTED SIMULATED ANNEALING

Chul-Eui Hong, Hee-Il Ahn and Bruce M.McMillin

CONTENTS

I. Introduction
II. Error Occurrence
III. Previous Work on Error Tolerance
IV. Analysis of Cost Error
V. New Error Tolerance Method
VI. Experimental Results
VII. Conclusion
References

ABSTRACT

Simulated annealing is an attractive, but expensive, heuristic method for approximating the solution to combinatorial optimization problems. Attempts to parallel simulated annealing, particularly on distributed memory multicomputers, are hampered by the algorithm's requirement of a globally consistent system state. In a multicomputer, maintaining the global state *S* involves explicit message traffic and is a critical performance bottleneck. To mitigate this bottleneck, it becomes necessary to amortize the overhead of these state updates over as many parallel state changes as possible. By using this technique, errors in the actual cost *C(S)* of a particular state *S* will be introduced into the annealing process. This paper places analytically derived bounds on this error in order to assure convergence to the correct optimal result. The resulting parallel simulated annealing algorithm dynamically changes the frequency of global updates as a function of the annealing control parameter, i.e. temperature. Implementation results on an Intel iPSC/2 are reported.

I. INTRODUCTION

The simulated annealing algorithm is based on the analogy between simulation of the annealing of solids and the problem of solving large combinatorial optimization problems (Figure 1) [1]. The ground states (global optima) of a complex physical system can be reached by heating the system up to some high temperature (melting point) and then cooling it down slowly keeping the equilibrium condition so we can search all possible states. At each temperature value T, the system is allowed to reach thermal equilibrium, characterized by a probability of being in a state with energy E given by the Boltzmann distribution [2].

$$Prob\,[\mathbf{E}=E]=\frac{1}{Z(T)}\cdot\exp\left(-\frac{E}{\kappa_B T}\right) \qquad (1\text{-}1)$$

where $Z(T)$ is the normalization factor depending on the temperature T and κ_B is the Boltzmann constant. The factor $\exp(-E/\kappa_B T)$ is known as the Boltzmann factor.

Physical Systems	Optimization Problems
State(Structure)	Configuration
Energy	Cost
Phase Transition	Move Generation
Ground State	Optimal Solution
Quick Cooling(Quenching)	Iterative Improvement
Slow Cooling(Annealing)	Simulated Annealing

Fig. 1. Analogy between physical systems and optimization problems.

Metropolis et al. proposed a Monte Carlo method [3], which simulates the evolution to thermal equilibrium of a solid for a fixed value of the temperature T. The Pascal-like pseudo code for the simulated annealing algorithm is in Fig. 2. Implementing the simulated annealing, the initial temperature is set sufficiently high so that all moves are accepted. With a small perturbation of the current state space, we can reach a new state. Let ΔC be the difference of the energy (cost) of current state and new state. The probability that a candidate move is accepted or rejected in simulated annealing is determined by *the Metropolis criterion*:

$$Prob[\Delta C \text{ is accepted }]=\min\left(1,\exp\left(-\frac{\Delta C}{T}\right)\right) \qquad (1\text{-}2)$$

If a candidate move is accepted, then the new state becomes the current state; if the candidate

```
PROCEDURE SIMULATED ANNEALING
begin
 INITIALIZE;
 k:= 0;
 repeat
   repeat
     PERTURB (config.i → config.j, ΔC_ij);
        /* evaluation of the cost change*/
     if ΔC_ij ≤ 0 then accept
     else if exp(-ΔC_ij /T_k) > random[0,1) then accept;
     if accept then
       UPDATE (configuration j);
   until equilibrium is approached sufficiently
        closely;
   T_k+1 := f(T_k);
   k:= k + 1;
 until stop criterion == true (system is 'frozen');
end.
```

Fig. 2. The Metropolis Procedure Probabilistically samples states of the configuration.

move is rejected, the current state remains unchanged. We iterate the above procedures until the system gets in the thermal equilibrium, i.e. the probability distribution of the states approaches the Boltzmann distribution.

Evaluation of the annealing procedure, which calculates the cost change (ΔC), is expensive due to the large number of state parameters that need to be evaluated. Parallelization of the annealing procedure is an attractive option. In particular, distributed memory multicomputers provide the best promise in massive performance speedup. A multicomputer consists of individual processors with local memory that communicate by message passing over an interconnection network. Thus, there is no shared memory available for maintaining the global system state. However, the lack of shared memory causes the inconsistent states among the processors.

The model problem in this paper is the stock-cutting problem [4]. The stock-cutting problem is to allocate regular and/or irregular patterns onto a large stock sheet of finite dimensions in such a way that the resulting scrap will be minimized. This problem is common to many applications in aerospace, shipbuilding,VLSI design, steel construction, shoe manufacturing, clothing and furniture. Stock-cutting problem is commonly known as the 2D bin packing problem. The cost function is made up of the affinity relation between patterns, the distance from the origin, and overlap penalty between patterns. Consider the cost function C as

$$C = -\alpha \sum \frac{a_{i,j}}{d_{i,j}} + \beta \sum d_{i_0} + \gamma \sum O_{i,j},$$

where α, β, and γ are positive real numbers that indicate the contribution of each of the components in the cost function. $a_{i,j}$ is the affinity relation between pattern i and j. $d_{i,j}$ is the distance between pattern i and j. d_{i_0} represents the distance of pattern i from the origin. $O_{i,j}$ is the overlap between pattern i and j.

The stock cutting problem yields a straightforward parallel decomposition, and, thus is an interesting model problem. A data parallel domain decomposition of the stock cutting problem gives each node approximately the same number of patterns. Each node performs internal move, rotate and exchange operations as well as participating in moves between nodes. The distance from origin, Σd_{i_0}, is calculated correctly without the global information of location of all object since the origin is fixed. The overlap penalty, $O_{i,j}$, is also calculated correctly by making two adjacent processors cooperate when the pattern lies in boundary. However, in calculating the affinity relation between two patterns i and j, $\sum(a_{ij}/b_{ij})$,we need global information of correct location of two patterns. So a cost error occurs in calculating the affinity relation cost.

Definition 1-1: When the new cost is larger than the current cost, this proposed move is called a *hill climbing move*.

The simulated annealing algorithm can be looked upon as a random iterative improvement algorithm, with a certain probability of making mistakes by accepting hill climbing moves that increase the cost to get out of the local minima. Since simulated annealing randomly selects hill climbing moves, it can tolerate some cost errors. Thus, an approximate calculation, instead of an exact calculation, which uses old state information from other nodes can be used to evaluate the cost function. This modified procedure is an asynchronous algorithm, whereas, a straightforward implementation of parallel simulated annealing is strictly synchronous (and sequential!). Under the proper conditions, annealing algorithms can evaluate the cost using old state information, and still converge to a reasonable solution. So it is important to find an upper bound on the cost error at a particular temperature to maximize speedups in the parallel implementation. Herein these two algorithms will be differentiated as Sequential Simulated Annealing (even for a parallel version since the sequence of state updates is the same as for a sequential version) for the former and Error-Present Simulated Annealing for the latter.

Cost error tolerance plays a useful role in multiprocessing. When processors independently operate on different parts of the problem, they need not synchronously update other processors. A processor can save several changes, and then send a single block to the other processors. Asynchronous algorithms require a minimum of synchronization. However, at low temperatures, the cost error may degrade the final result unless corrected by a later move. So, simulated annealing does not have unlimited tolerance for cost error.

Previous work of the cost-error-tolerant schemes cannot measure the cost error correctly, and cost error has been tolerated empirically. In this paper, we define maximum bound of tolerable cost error as a function of the global update frequency.

In Section II, we explain how a cost error can occur in multicomputers. Section III discusses previous work on cost-error-tolerant schemes. In Section IV, we analyze three interesting phenomena of cost-error-present algorithm. In Section V, we classify the error model by case study, and present our cost-error measurement scheme and error-tolerant method. This cost-error-tolerant method is applied to the stock-cutting problem using an asynchronous parallel spatial decomposition Simulated Annealing algorithm. Finally, Section VI presents and discusses some experimental results.

II. ERROR OCCURRENCE

Consider a system with two state variables x and y in Figure 3, so some state $s=<x,y>\in S$. Let the cost function be $f(x+y)$. Now put x and y on two separate processors. Each processor proposes a move: processor 0 generates $x \leftarrow x-1$, while processor 1 generates $y \leftarrow y-1$ simultaneously.

In both cases, $\Delta C < 0$, so each move will be accepted. However, the cost function error causes the state to jump to a high local minimum. At low temperatures, the annealing algorithm probably will not escape this trap because there is no hill climbing move in low temperatures. So the maximum bound of the tolerable error is proportional to the temperature.

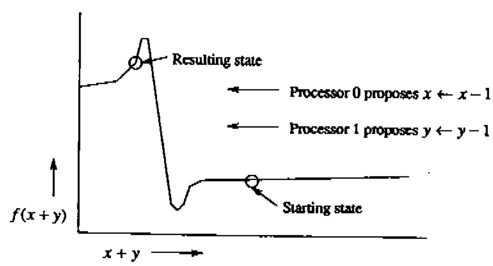

Fig. 3. Errors can cause annealing failure [5].

III. PREVIOUS WORK ON ERROR TOLERANCE

Jayaraman, et al. describe the characteristics of cost errors at different temperatures [5]. The error in the cost function is defined to be the difference between the real change in cost from initial to final states and the estimated change in cost, which is equal to the sum of the changes in cost (ΔC_i for processor i) at each processor.

Definition 3-1: The *cost error* (ΔE_{tr}) is defined as the difference between the actual (real) cost change and the estimated (measured) cost change. That is, due to the local copy of the outdated information, the actual cost change calculation may be different from the estimated cost change.

$$\Delta E_{tr} = \Delta C_a - \Delta C_e = \left(C_{af} - C_{ai}\right) - \sum_{i=1}^{P} \Delta C_i \qquad (3\text{-}1)$$

where ΔC_a is the actual cost change, and ΔC_e is the estimated cost change. C_{af} is the actual final cost and C_{ai} is the actual initial cost. ΔC_i is the estimated cost change in processor i, and P is the total number of processors.

This cost error measurement scheme will be referred as *the traditional error measurement scheme*. There are shortcomings in this traditional error measurement scheme. These are discussed in the next section.

Definition 3-2: An *optimistic error* occurs when the cost error (ΔE_{tr}) is positive from Definition 3-1, i.e. the estimated cost change (ΔC_e) is less than the actual cost change (ΔC_a), $\Delta C_a > \Delta C_e > 0$. Since the Metropolis criterion (equation 1-2) is used for the cost changes, $\exp(-\Delta C_e/T) > \exp(-\Delta C_a/T)$ the acceptance ratio is increased in the case of an optimistic error. In other words, the candidate move is accepted while this move may be rejected in sequential (error-free) simulated annealing since $\exp(-\Delta C_e/T) > \exp(-\Delta C_a/T)$. This kind of error is called an optimistic error and this move an optimistic move.

Definition 3-3: A *pessimistic error* occurs when the cost error (ΔE_{tr}) is negative, i.e. the actual cost change is less than the estimated cost change, $\Delta C_e > \Delta C_a > 0$. This being the reverse case of an optimistic error, the acceptance ratio is decreased in the case of a pessimistic error.

Definition 3-4: The *stream length, s,* is defined as the number of continuous moves before the global update where all local information is broadcasted and updated.

Jayaraman, et al. [6] observe that the average cost error in the high temperature region increases with an increase in the stream length. However, the average cost error reduces and finally drops to 0 in the low temperature region, because in the low temperature region the acceptance ratio of moves is small and consequently, there are very few interacting moves causing cost errors.

Grover [7] presents an cost-error-tolerant scheme based on the analogy with statistical mechanics to show that cost errors which are much smaller than the temperature do not change the results of the algorithm. In statistical mechanics, all macroscopic properties of a material can be derived from the partition function z, which is defined as the sum of the Boltzmann factors over all possible states, $z = \sum_{i \in S} \exp(-C(i)/T)$. With this method, the maximum stream length in a fixed temperature can be probabilistically predicted based on the expected magnitude of a cost error.

Banerjee, et al. [8] suggest an adaptive stream length control. The goal is to find an upper bound on the maximum permissible cost error at a particular temperature. By adjusting the stream length dynamically, the average cost error can be limited to a specific range. This method is based on the move acceptance curve in which the acceptance ratio P is given by:

$$P = Prob\,[\text{move accepted} \mid \Delta C > 0] \cdot Prob[\Delta C > 0] + Prob\,[\text{move accepted} \mid \Delta C \leq 0] \cdot Prob[\Delta C \leq 0] \quad (3\text{-}2)$$

where ΔC is the proposed cost change. By considering the induced cost error and the Metropolis criteria, the acceptance ratio with cost error can be rewritten as

$$P_E \approx e^{-(\Delta C \pm E)/T} \cdot Prob[\Delta C > 0] + Prob[\Delta C < 0\,] \quad (3\text{-}3)$$

where E is the total amount of cost error at a fixed temperature.

If the acceptance ratio with cost error (P_E) is held to within 5 percent of a normal distribution, a pessimistic cost error bound B_+ and an optimistic cost error bound B_- are approximated as follows:

$$\begin{aligned} B_+ &\leq -T \cdot \ln(1-0.05) \approx T/20 \\ B_- &\leq T \cdot \ln(1+0.05) \approx T/21. \end{aligned} \quad (3\text{-}4)$$

If the average cost error after a stream length is higher than (T/21), the stream length is reduced commensurate with that excess. If average cost error is lower than (T/42), the stream length is increased slowly. A five percent deviation in composite acceptance is set experimentally to maintain convergence.

IV. ANALYSIS OF COST ERROR

There are two different types of cost errors: *Temporary errors* and *cumulative errors* [9]. *Temporary errors* occur when two processors simultaneously consider interacting moves. For example, in the stock cutting problem, if two processors attempt to move an object simultaneously to the same location which is empty, the objects will overlap. If the processors investigate the overlap of the moved objects after each move, the system gets a single, consistent and correct state. *Cumulative errors* develop when local state information used to compute the cost becomes increasingly out of date as the annealing process continues. In the stock cutting problem, the affinity relation cost becomes incorrect as the stream length increases, because as the stream length increases, the local information gets out-dated. Durand [9] observes that a temporary error has only a minor effect on the convergence of simulated annealing, while a cumulative error appears to have a strong effect on convergence.

When a cost error affects the annealing process, there are some interesting phenomena. First, from Definition 3-1, Jayaraman et al.[6] and Casotto et al. [10] say that the cost error is mostly negative and the absolute value of the cost error is large at high temperatures, but goes to zero as the temperature decreases. The ratio of accepted moves versus attempted moves tends to be very small at low temperatures, even if the range limiter tries to keep it large. With very few moves accepted, the probability of accepting parallel moves is also very small. Furthermore, even if moves generated in parallel are actually accepted, they are range-limited, so that the error cannot be arbitrarily large at low temperatures.

Figure 4 through 9 are drawn from the stock cutting of 16 irregular patterns in 4 processors. The Markov Chain length is 500, and the temperature decrement ratio is 0.98. 125 is the maximum bound of the stream length. That is, stream length 125 parallel simulated annealing has a large cost error from Definition 3-4.

In the stock cutting problem, Fig. 4 indicates that the cost error measured by the traditional scheme (Definition 3-1) is not mostly negative. However, the cost error only for the accepted hill climbing moves is mostly negative. Figure 5 indicates that the absolute value of the average cost error of pessimistic moves and that of optimistic moves are almost the same in the high temperature region. However, the absolute value of the average cost error of pessimistic moves is a little smaller than that of optimistic moves in the critical and the low temperature regions. Fig. 6 shows that the number of accepted pessimistic moves is greater than that of accepted optimistic moves. From Figures 5 and 6, it is expected that the total pessimistic cost error is greater than the total optimistic cost error. This corresponds to Fig. 4.

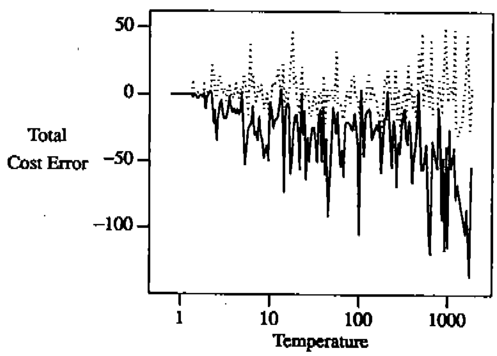

Fig. 4. Total cost error for all accepted hill climbing moves.(Stream length is 125)
(···: ΔE_{tr} for all accepted moves,
—: ΔE_{tr} only for the accepted hill climbing moves)

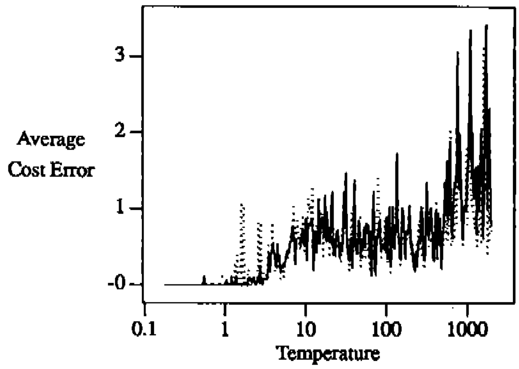

Fig. 5. Average cost error for one accepted hill climbing move (Stream length is 125)
(··· : $|\langle \Delta E_{tr} \rangle|$ for optimistic moves,
— : $|\langle \Delta E_{tr} \rangle|$ for pessimistic moves)

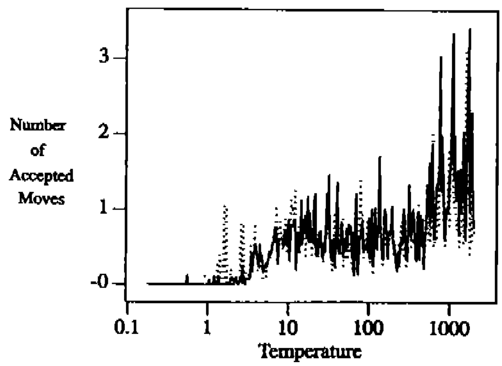

Fig. 6. Total number of accepted hill climbing moves.(Stream length : 125)
(··· : for optimistic moves,
—: for pessimistic moves)

Secondly, when a cost error is present, Rose et al. [11] states that the average acceptance ratio increases in the low temperature region as the number of processors increases. This is due to the misinformation causing some moves that would not have been made in sequential simulated annealing. Further moves are then necessary to make up for these "wrong" moves, thus increasing the acceptance ratio. In the stock cutting problem, Fig. 7 depicts the acceptance ratio of hill climbing moves. The acceptance ratios of the optimistic and pessimistic hill climbing moves are almost the same. So from Figure 6 and 7, it can be expected that the pessimistic hill climbing moves occur more frequently than the optimistic moves. This can be explained by noting that the hill climbing move tends to be estimated higher than the actual cost by using the out-of-date local information.

The acceptance ratio of the error-present algorithm is smaller than that of the sequential simulated annealing algorithm, because a pessimistic move occurs more frequently than an optimistic move and a pessimistic move decreases the acceptance ratio. In the critical (middle) temperature region, a decreasing number of hill climbing moves occur. Since only hill climbing moves affect the acceptance ratio, the acceptance ratios of the error-present algorithm and the sequential simulated annealing algorithm are nearly the same. In the low temperature region, most accepted moves have negative cost change. However, with incomplete information, some moves are accepted, which would not have been accepted in the sequential simulated annealing algorithm. So the acceptance ratio of the error-present algorithm is increased slightly (Fig. 8).

The third phenomenon in the presence of cost error is the reduced fluctuation of the average change in cost as a function of temperature($<\Delta C>$ vs. T) at high and intermediate temperature regions [12]. Since pessimistic moves occur more frequently, the fluctuations in cost are

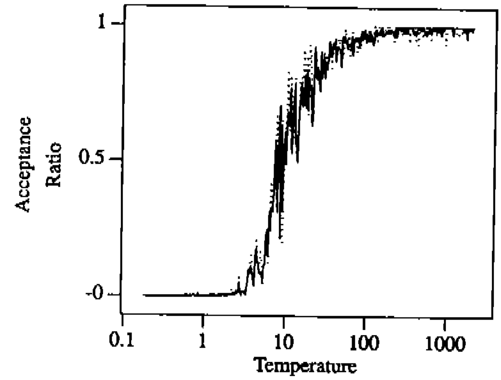

Fig. 7. Acceptance Ratio of Hill Climbing Moves.(Stream length: 125)
(··· : for optimistic moves,
— : for pessimistic moves)

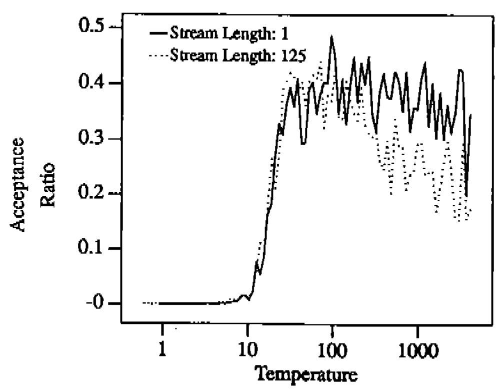

Fig. 8. Acceptance Ratio of All Moves.

reduced in the high temperature region. So the system is likely to be kept in the high local minimum. The average cost using the error-present algorithm is less than that of sequential simulated annealing algorithm because the hill climbing moves are rejected more frequently in the error-present algorithm(Fig. 9).

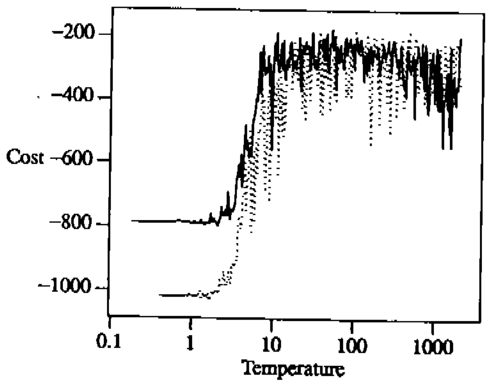

Fig. 9. Fluctuations of Costs
(··· : sequential annealing,
— : stream length is 125)

There are shortcomings in the traditional cost error measurement scheme (Definition 3-1). Since there is no way to calculate the actual cost without global information, the traditional error measurement scheme is to calculate the cost error after a global update as a difference between the actual cost change (ΔC_a) and the estimated cost change (ΔC_e) using Definition 3-1. However, this method has inherent problems.

This method counts only the accepted moves, i.e. if the candidate move is a pessimistic hill climbing move, $\Delta C_e > \Delta C_a > 0$, and this move is rejected, this move may be accepted in sequential simulated annealing because $\exp(-\Delta C_e/T) > \exp(-\Delta C_a/T)$. This kind of cost error cannot be included with this method. In other words, this method cannot calculate the cost error of rejected moves.

The second problem is that when both the actual cost change (ΔC_a) and the estimated cost change (ΔC_e) are negative, regardless of the cost error, the candidate move is accepted. However, the difference in cost, $\Delta C_a - \Delta C_e$, is added to the total amount of cost error, even though the acceptance of the move is correct, i.e. there is no error in the move decision.

Finally, the optimistic error ($\Delta C_a > \Delta C_e > 0$) and the pessimistic error ($\Delta C_e > \Delta C_a > 0$) are compensated during a stream length. Only the rough average error can be conjectured. So this traditional error measurement scheme can be used only qualitatively. It indicates at which temperature large cost errors occur.

These three problems are corrected by a new cost error measurement scheme with some assumptions (see Section V).

V. A NEW ERROR TOLERANCE METHOD

In this section, a new cost error measurement scheme is presented. Using the measured amount of cost error, an optimal stream length is derived based on the hill climbing nature of simulated annealing. Bounds on the cost error are proved analytically to be a function of global update frequency, or stream length s (Definition 3-4).

Erroneous move decisions due to the cost error (ΔE) will be proved to be exponentially distributed with respect to fixed temperature ($T > 0$). Here ΔE is used to differentiate the traditional cost error, ΔE_r (Definition 3-1) . That is, ΔE is measured by the improved cost measurement method correcting the shortcomings of the traditional method discussed in Section IV. With this known distribution, the probability of an erroneous move decision and the amount of cost error due to the erroneous move decision can be determined in s parallel moves without global updating, or in stream length s.

Fig. 9 shows one possible interpretation that as the stream length increases, the hill climbing power decreases since the fluctuations in costs reduce in the error-present annealing process. The decreased hill climbing power can be compensated by an increased additional Markov chain length. That is, the additional move generations provide a greater chance of a hill climbing move. Since the cost error increases as the stream length increases, the optimal stream length and the additional Markov chain length are proportional to keep the convergence as in sequential (error-free)annealing process because as the stream length increases the cost error increases too.

When the stream length is fixed, the generated cost error must be tolerated by changing the additional Markov chain length dynamically. Meanwhile, when the additional Markov chain length is fixed, the tolerable amount of the cost error, bounds of the cost error, is fixed, so the stream length is varied according to the bounds of the cost error.

With the increment of the Markov chain length, the annealing process converges to the good results with reasonable speedups. Since the additional Markov chain length is fixed in the experiment, the amount of cost error must be controlled by increasing or decreasing the stream length. By adjusting the global update frequency, the convergence property is maintained to the same degree as in the sequential annealing process.

Here the distribution of the move acceptance and the erroneous move decision are defined.

Theorem 5-1: The acceptance move decision is exponentially distributed with respect to the parameter $T > 0$.

$$Prob[\text{Move accepted with cost change}[0,\Delta C]] = 1-\exp(-\Delta C/T)$$

Proof

Define the continuous random variable x to be a function which associates a positive real number, the hill climbing cost change (ΔC) with each possible outcome of an accepted move decision. The probability of move acceptance is $\exp(-\Delta C/T)$ when the cost change is $(\Delta C, \infty)$. So the cumulative probability of move acceptance is $1-\exp(-\Delta C/T)$ when the cost change is $[0, \Delta C]$, which is the exponential cumulative distribution function.

$$\begin{aligned} & Prob[X \leq \Delta C] \\ &= 1-Prob[X > \Delta C] \\ &= 1-Prob[\text{Move accepted with cost change } \Delta C] \\ &= 1-\exp(-\Delta C/T) \end{aligned}$$

So the continuous random variable x has an exponential distribution with respect to the parameter $T > 0$. □

Since the estimated and actual cost changes are different, erroneous moves can result. Consider two possible cost changes, ΔC_1 and ΔC_2 where $\Delta C_1 < \Delta C_2$ where it is not known which is the actual and which is the estimated cost change. If a move is accepted with a smaller cost change, ΔC_1, while the move is rejected with a larger cost change, ΔC_2, then an erroneous move of error, $\Delta E = \Delta C_2 - \Delta C_1$, occurs.

Theorem 5-2: The erroneous move decision is exponentially distributed with respect to the parameter $T > 0$, given that the candidate move is accepted with smaller cost change, ΔC_1, between the actual and the estimated cost changes.

Prob[The erroneous move decision
with cost error $[0,\Delta E]$]
=*Prob*[Move rejected with cost change ΔC_2|
Move accepted with cost change ΔC_1]
$=1\text{-exp}(-\Delta E/T)$

Proof

Define a continuous random variable $\gamma_{\Delta C_1}$ to be a function which associates a positive real number, the cost error of the hill climbing move (ΔE), with each possible outcome of the erroneous move decision. Consider two cost change values ΔC_1 and ΔC_2 with $\Delta C_2 \geq \Delta C_1$. Then $\Delta E \equiv \Delta C_2 - \Delta C_1$. The erroneous move decision is the event that the candidate move with the smaller cost change, ΔC_1, is accepted, while the candidate move with the larger cost change, ΔC_2, is rejected. The random variable $\gamma_{\Delta C_1}$ represents the excess life of the move acceptance, i.e. $\gamma_{\Delta C_1} = S_{N(\Delta C_1)+1} - C_1$, where $N(\Delta C)$ is the number of acceptances with cost change $[0, \Delta C]$, and S_n is the sum of the cost change when the move is accepted n times. S is the sum of the random variable X in Theorem 5-1. So, $\gamma_{\Delta C_1}$ represents how long the acceptance move decision is maintained given that the candidate move with the smaller cost change, ΔC_1, is accepted. In other words, $S_{N(\Delta C_1)+1}$ is the cost change of the move rejection given that the candidate move with cost change ΔC_1 is accepted.

$$
\begin{aligned}
&Prob[\gamma_{\Delta C_1} \leq \Delta E] \\
&= Prob[S_{N(\Delta C_1)+1} - \Delta C_1 \leq \Delta E] \\
&= Prob[S_{N(\Delta C_1)+1} \leq \Delta E + \Delta C_1] \\
&\quad (\textit{from } N(C) \geq n \Leftrightarrow S_n \leq C\,) \\
&= Prob[N(\Delta E + \Delta C_1) \geq N(\Delta C_1) + 1] \\
&= Prob[N(\Delta E + \Delta C_1) - N(\Delta C_1) \geq 1] \\
&= 1 - Prob[N(\Delta E + \Delta C_1) - N(\Delta C_1) \leq 0] \\
&= 1 - Prob[N(\Delta E + \Delta C_1) - N(\Delta C_1) = 0] \\
&\quad \text{(Since the number of rejections is non-negative)} \\
&= 1 - Prob[N(\Delta E) = 0] \\
&\quad \text{(From the memoryless property of exponential distribution.(Theorem 5-1))} \\
&= 1 - \exp(-\Delta E/T) \qquad \square
\end{aligned}
$$

In Section V.1, the move decisions are classified according to the actual cost change (ΔC_a) and the estimated cost change (ΔC_e), and the probability of the erroneous move decision is calculated from Theorem 5-2 by a case-by-case analysis. In Section V.2, the amount of cost error is measured probabilistically regardless of whether the move is accepted or rejected. This cost error measurement method is unlike the traditional cost error measurement scheme (Definition 3-1). This method includes the cost error due to the rejected moves. In Section V.3, since cost error can be tolerated by hill climbing moves, the measured amount of cost error is used to derive the optimal stream length.

1. CASE-BY-CASE STUDY OF ERROR MODEL

There are four possible cost change cases (*Case 5-1* through *Case 5-4*) each with four possible subcases, that is:

1) A move is accepted based on an estimated cost change and also based on an actual cost change.
2) A move is accepted based on an estimated cost change, however will be rejected based on an actual cost change.
3) A move is rejected based on an estimated cost change, yet will be accepted based on

cost change, yet will be accepted based on an actual cost change.

4) A move is rejected based on an estimated cost change and also based on an actual cost change.

In sub-cases 1) and 4), the move decision is correct regardless of the cost error used. However, in sub-cases 2) and 3), an erroneous move decision occurs due to the cost error.

Since the actual cost change can not be calculated at run time, the estimated cost change is used in an acceptance decision using the Metropolis criteria (equation 1-2).

Case 5-1: $\Delta C_e \geq \Delta C_a > 0$ (Pessimistic move).

The first case is that the actual cost change (ΔC_a) and the estimated cost change (ΔC_e) for one move are positive and the estimated cost change is greater than or equal to the actual cost change.

Define one move error $\Delta E_1 = \Delta C_e - \Delta C_a$, where $\Delta E_1 \geq 0$.

1) The move is accepted with the estimated cost change ΔC_e, where the probability of a move acceptance is $\exp(-\Delta C_e/T)$.

1-1) This move will be accepted with the actual cost change ΔC_a as well because the probability of a move acceptance with the actual cost change is greater than or equal to the probability of a move acceptance with the estimated cost change, i.e. $\exp(-\Delta C_e/T) \leq \exp(-\Delta C_a/T)$. So the move decision is correct regardless of the cost error.

2) The move is rejected with the estimated cost change ΔC_e, where the probability of a move rejection is $1-\exp(-\Delta C_e/T)$.

2-1) This move can be accepted with the actual cost change ΔC_a with a probability P_1, where

$$
\begin{aligned}
P_1 &= Prob[\text{Move accepted with } \Delta C_a \cap \\
&\qquad \text{Move rejected with } \Delta C_e] \\
&= Prob[\text{Move rejected with } \Delta C_e \mid \\
&\qquad \text{Move accepted with } \Delta C_a] \\
&\quad \times Prob[\text{Move accepted with } \Delta C_a] \\
&= Prob[\text{The erroneous move decision} \\
&\qquad \text{with cost error} \Delta E_1] \\
&\quad \times Prob[\text{Move accepted with } \Delta C_e - \Delta E_1] \\
&= (1-\exp(-\Delta E_1/T)) \cdot \exp(-(\Delta C_e - \Delta E_1)/T) \\
&\qquad\qquad \text{from Theorem 5-2} \\
&= \exp(-\Delta C_e/T) \cdot (\exp(\Delta E_1/T)-1)
\end{aligned}
$$

2-2) When this move is rejected with the actual cost change ΔC_a, there is no erroneous move decision.

Case 5-2: $\Delta C_a \geq \Delta C_e > 0$ (Optimistic move).

The second case is that the actual cost change and the estimated cost change for one move are positive, however, the actual cost change is greater than or equal to the estimated cost change.

We define one move error $\Delta E_2 = \Delta C_a - \Delta C_e$, where $\Delta E_2 \geq 0$.

1) The move is accepted with the estimated cost change ΔC_e, where the probability of a move acceptance is $\exp(-\Delta C_e/T)$

1-1) If this move is accepted with the actual cost change ΔC_a, then there is no error.

1-2) This move can be rejected with the actual cost change ΔC_a with probability P_2, where

$$
\begin{aligned}
P_2 &= Prob[\text{Move accepted with } \Delta C_e \cap \\
&\qquad \text{Move rejected with } \Delta C_a] \\
&= Prob[\text{Move rejected with } \Delta C_a \mid \\
&\qquad \text{Move accepted with } \Delta C_e]
\end{aligned}
$$

$\times Prob$[Move accepted with ΔC_e]
$= Prob$[The erroneous move decision with cost error ΔE_2]
$\times Prob$[Move accepted with ΔC_e]
$= \exp(-\Delta C_e/T) \cdot (1-\exp(-(\Delta E_2/T))$
from Theorem 5-2

2) The move is rejected with the estimated cost change ΔC_e.

2-1) This move will be rejected with the actual cost change ΔC_a as well because $\Delta C_a \geq \Delta C_e$. So there is no erroneous move decision.

Table 1. Probability of the Cost Error in a Hill Climbing Move.

	Pessimistic Move ($\Delta C_e \geq \Delta C_a$)		Optimistic move ($\Delta C_a \geq \Delta C_e$)	
Move	acc w/ΔC_e	rej w/ΔC_a	acc w/ΔC_e	rej w/ΔC_a
acc w/ΔC_a	0	$\exp(-\Delta C_e/T) \times(\exp(\Delta E/T)-1)$	0	-
rej w/ΔC_a	-	0	$\exp(-\Delta C_e/T) \times(1-\exp(-\Delta E/T))$	0

Case 5-3: $(\Delta C_e > 0 \cap \Delta C_a \leq 0) \cup (\Delta C_a > 0 \cap \Delta C_e \leq 0)$

In Case 5-3, the cost error is greater than the absolute value of the estimated cost change, so the signs of the estimated and actual cost changes are different. Computing these two probabilities is somewhat complex, thus any error control scheme will be complex. These two cases happen rarely since the cost error is much smaller than the cost change in real experiments. The portion of Case 5-3 to the total moves is less than 4% in the maximum stream length implementation. Thus, the occurrence of these events is ignored.

Case 5-4: $\Delta C_e \leq 0 \cup \Delta C_a \leq 0$

In Case 5-4, a move will be always accepted and there is no cost error, since the decision of move is correct, i.e. move is accepted also with the actual cost change ΔC_a.

The summary of the above cases are in Table 1, where ΔE is the amount of the cost error.

2. IMPROVED ERROR MEASUREMENT SCHEME

In improving the cost error measurement scheme, the total amount of cost error is calculated throughout the given stream length, s. Unlike previous methods (see Section III) which ignore cost errors from rejected moves and find the optimal stream length heuristically, this new method calculates the cost error analytically based on the results in the previous section.

Lemma 5-1: The actual cost change (ΔC_a) is represented as the sum of the estimated cost change (ΔC_e) and the cost error throughout an iteration i.

$$\Delta C_a \equiv \Delta C_e \pm \Delta E = \Delta C_e \pm i \cdot \alpha \cdot |<E>|$$

Proof

ΔE is the cost error of any iteration i. In a spatial decomposition stock cutting problem (Fig. 10), the absolute value of the average cost error ($|<E>|$) is calculated when one move is supposed to be accepted with a half distance of the range-limiter, and the other processors do not know this acceptance. Actually, this is the maximum bound of $|<E>|$ because as the move distance is shorter, the move is accepted more easily. For example, any one pattern of Processor P_1 is assumed to be moved to a half distance of the range-limiter and Processor P_2 does not know it. Then the absolute value of the average cost error $|<E>|$ for P_2 can be calculated when the Processor P_2 tries the move generation. The cost error ΔE at an iteration i, can be represented as an average error ($|<E>|$) times the total number of accepted moves throughout an iteration i. The total number of accepted moves is the acceptance ratio (a) times the iteration i. So $\Delta E = i \cdot a \cdot |<E>|$. □

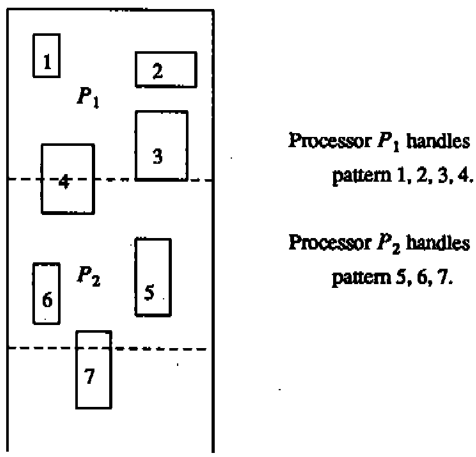

Fig. 10. Spatial Decomposition.

In the spatial decomposition stock-cutting problem, the stock sheet is nearly equally divided in x-direction. Each processor governs a space and handles the pattern whose reference coordinate belongs to its own space. The reference coordinate of the pattern is the smallest (x,y) of the bounding box surrounding the pattern. Then each processor does move generations asynchronously when the move does not affect the space of the other processors (Intra-Move). When the move affects the space of the other processor (Inter-Move), all the affected processors cooperate the move generation.

Since a cost error occurs only with a positive cost change in this analysis, to calculate the cost error, it is necessary to compute a probability for the conditions of *Case 5-1* and *Case 5-2*.

Theorem 5-3:

$$Prob[\Delta C_e \geq \Delta C_a > 0] + Prob[\Delta C_a \geq \Delta C_e > 0] = Prob[\Delta C_e > 0]$$

Proof

$$\begin{aligned} Prob[\Delta C_e \geq \Delta C_a > 0] &= Prob[\Delta C_e \geq \Delta C_a \mid \Delta C_e > 0, \Delta C_a > 0] \times Prob[\Delta C_e > 0, \Delta C_a > 0] \\ &= Prob[\Delta C_e \geq \Delta C_a \mid \Delta C_e > 0, \Delta C_a > 0] \times Prob[\Delta C_a > 0 \mid \Delta C_e > 0] \times Prob[\Delta C_e > 0] \\ &= Prob[\Delta C_e \geq \Delta C_a \mid \Delta C_e > 0] \cdot Prob[\Delta C_e > 0]. \end{aligned}$$

Since $Prob[\Delta C_a > 0 \mid \Delta C_e > 0] = 1$ from the assumption that the cost error is not greater than the absolute value of the estimated cost change, i.e. the estimated cost change and the actual cost change have the same signs.

Similarly,

$$Prob[\Delta C_a \geq \Delta C_e > 0] = Prob[\Delta C_a > \Delta C_e \mid \Delta C_e > 0] \times Prob[\Delta C_e > 0].$$

So

$$\begin{aligned} Prob[\Delta C_e \geq \Delta C_a > 0] + Prob[\Delta C_a \geq \Delta C_e > 0] &= Prob[\Delta C_e \geq \Delta C_a \mid \Delta C_e > 0] \cdot Prob[\Delta C_e > 0] \\ &\quad + Prob[\Delta C_a \geq \Delta C_e \mid \Delta C_e > 0] \cdot Prob[\Delta C_e > 0] \\ &= Prob[\Delta C_e > 0]. \end{aligned}$$ □

The next task is to estimate the $Prob[\Delta C_e > 0]$. Since $Prob[\Delta C_e > 0]$ is a function of state configuration, i.e. in a maximum cost, $Prob[\Delta C_e > 0]$ is zero in a move generation, while in a local minimum cost, $Prob[\Delta C_e>0]$ is one in a move generation.

Lemma 5-2: The probability of positive estimated cost change is

$$Prob[\Delta C_e>0] = \frac{\sum_{i \neq j} I_{\{\Delta C_e(i,j)>0\}}(i)}{N(i)}, \quad \text{for any state } i$$

Proof

State j is any neighbor of state i. $\Delta C_e(i,j)$ is the estimated cost change of a move from state i to state j. $N(i)$ is the number of neighbor states from state i. The proof is obvious by using the specified indicator, I. □

Since $N(i)$ and $\Delta C_e(i,j)$ are not known in advance, it is difficult to estimate the $Prob[\Delta C_e>0]$. However, during the running of the algorithm, the estimated cost change can be calculated. So only when the estimated cost change is greater than zero, the total probability of cost error (P_T) is counted.

The probabilities of optimistic and pessimistic errors are calculated, when the estimated cost change (ΔC_e) and cost error (ΔE) are fixed, i.e. experimentally the optimistic and pessimistic cost errors are the same (Figure 5) and the actual cost changes are calculated from the predefined estimated cost change and cost error (Lemma 5-1).

Corollary 5-1: The probability of optimistic cost error

$$P_{opt}=Prob\,[\Delta C_a \geq \Delta C_e > 0] \cdot \left| \exp\left(-\frac{\Delta C_e}{T}\right) \cdot \left(\exp\left(-\frac{\Delta E}{T}\right) - 1\right) \right|$$

Corollary 5-2: The probability of pessimistic cost error.

$$P_{pes}=Prob\,[\Delta C_e \geq \Delta C_a > 0] \cdot \left| \exp\left(-\frac{\Delta C_e}{T}\right) \cdot \left(\exp\left(\frac{\Delta E}{T}\right) - 1\right) \right|$$

In Section IV, it is shown that the total pessimistic cost error is greater than the total optimistic cost error. This can be explained by the next theorem with some assumptions.

Theorem 5-4: The probability of the pessimistic cost error is greater than that of the optimistic cost error with the following four assumptions.

1. The cost error (ΔE) is less than the estimated cost change in the hill climbing move.
2. The amount of the pessimistic and optimistic cost error are the same, $\Delta E_1 = \Delta E_2$ from *Cases 5-1* and *5-2*.
3. Only the estimated cost change is measured and the actual cost change can be expected, or calculated, using Lemma 5-1.

Proof

The assumption 1 is reasonable because the cost error is much smaller than the cost change in most cases from *Case 5-3*. Figure 5 proves the assumption 2. So only the occurrence numbers of the pessimistic and optimistic cost errors are different. Since there is no way of calculating the actual cost change directly, the assumption 3 is inevitable. Figs. 6 and 7 tell that the pessimistic hill climbing moves occur more frequently than the optimistic moves, i.e. $Prob[\Delta C_a \geq \Delta C_e>0] \leq Prob[\Delta C_e \geq \Delta C_a>0]$ With the above assumptions, and Corollaries 5-1 and 5-2, it is obvious that the probability of the pessimistic cost error is greater than that of the optimistic cost error because $|\exp(-\Delta E/T)-1| < |\exp(\Delta E/T)-1|$. So the large pessimistic cost error is more likely to be accepted comparing with the optimistic cost error. □

The probability of optimistic cost error (P_{opt}) is always in [0, 1] from Corollary 5-1. The probability of pessimistic error (P_{pes}) must be checked to be in [0, 1]. Pessimistic errors happen only in *Case 5-1*.

$$\begin{aligned} \text{So, } & \Delta C_e \geq \Delta C_a > 0 \\ & \Rightarrow \Delta C_e \geq \Delta C_a,\ \Delta C_a > 0 \\ & \Rightarrow \Delta C_e \geq \Delta C_e - \Delta E,\ \Delta C_e - \Delta E > 0 \\ & \Rightarrow \Delta E \geq 0,\ \Delta E < \Delta C_e \\ & \Rightarrow \Delta C_e > \Delta E \geq 0 \\ & \Rightarrow 1 \geq P_{pes} \geq 0 \qquad \text{from Corollary } 5-2 \end{aligned}$$

So the range of the probability of pessimistic error is well defined.

Theorem 5-5: Since a cost error occurs only in a positive cost change, the total probability of cost error, P_T, is given by

$$P_T = Prob[\Delta C_e > 0] \times \exp(-\Delta C_e/T) \cdot (\exp(\Delta E/T) - 1)$$

Proof

$$\begin{aligned} P_T &= P_{pes} + P_{opt} \\ &= Prob[\Delta C_e \geq \Delta C_a > 0] \\ &\quad \times |\exp(-\Delta C_e/T) \cdot (\exp(\Delta E/T) - 1)| \\ &\quad + Prob[\Delta C_a \geq \Delta C_e > 0] \\ &\quad \times |\exp(-\Delta C_e/T) \cdot (\exp(-\Delta E/T) - 1)| \\ &\leq Prob[\Delta C_e \geq \Delta C_a > 0] + Prob[\Delta C_a \geq \Delta C_e > 0] \\ &\quad \times \exp(-\Delta C_e/T) \cdot (\exp(\Delta E/T) - 1) \\ &= Prob[\Delta C_e > 0] \cdot \exp(-\Delta C_e/T) \cdot (\exp(\Delta E/T) - 1) \end{aligned}$$

□

Now the cost error can be determined using the probability of cost error (P_T).

Theorem 5-6: The amount of cost error in the hill climbing move ($\Delta C_e > 0$) is

$$E = \Delta C_e \cdot \exp(-\Delta C_e/T) \cdot (\exp(\Delta E/T) - 1)$$

Proof

Since $Prob[\Delta C_e > 0] = 1$, the probability of cost error in the hill climbing move is given by

$$P_T = \exp(-\Delta C_e/T) \cdot (\exp(\Delta E/T) - 1) \qquad \text{from Theorem } 5-5$$

So,

$$\begin{aligned} E &= \Delta C_e \cdot P_T \\ &= \Delta C_e \cdot \exp(-\Delta C_e/T) \cdot (\exp(\Delta E/T) - 1), \\ &\qquad \text{given that } \Delta C_e > 0 \end{aligned}$$

□

Durand [9] suggested that some problems or algorithms of Jayaraman, et al. [13] are more resistant to the cost error than those of Rose, et al.[11][14]. This robustness to the cost error can be explained by Theorem 5-7.

Theorem 5-7: The total amount of the cost error (E) depends on the portion of the cost error (ΔE or $i \cdot \alpha \cdot |\langle E \rangle|$) in the estimated cost (ΔC_e).

Proof

The cost function is made up of the cost-error-dependent terms which result in the cost error in calculating the cost function and the cost-error-independent terms which do not result in the cost error. For example, in the stock cutting problem, the cluster term and the overlap penalty term are the cost-error-independent terms, while the affinity relation term is the cost-error-dependent term(see Section I). From Theorem 5-6, when the cost error, ΔE, has only a small portion of the estimated cost error, ΔC_e, i.e. $\Delta E \ll \Delta C_e$, the total probability of the cost error goes to 0.

$$\begin{aligned} E &\propto \exp(-\Delta C_e/T) \cdot (\exp(\Delta E/T) - 1) \\ &= \exp(-(\Delta C_e - \Delta E)/T) - \exp(-\Delta C_e/T) \end{aligned}$$

Since $\Delta C_e > \Delta E > 0$ in the hill climbing move, the total amount of cost error (E) is always positive. So robustness to the cost error depends on the portion of the cost error (ΔE) in the estimated cost (ΔC_e). □

It was shown that the traditional cost error measurement scheme (Definition 3-1) has three shortcomings (see Section IV). These three shortcomings of the traditional method are corrected with the assumptions in Theorem 5-4. The shortcomings are corrected as follows. First, The new cost error measurement method includes the cost error of the rejected moves (Corollary 5-2 and Theorem 5-5). Second, this method does not include the cost error of the negative cost change moves, because the move decision is always correct regardless of the cost error used. Finally, there is no compensated cost error between the pessimistic and optimistic cost errors because this method adds the probabilities of the pessimistic and optimistic cost error (Theorem 5-5).

3. MAXIMUM BOUND OF TOLERABLE ERROR

In this section, the optimal stream length is derived for the measured amount of cost error. Since a cost error is tolerated by hill climbing moves, a maximum bound on the cost error can be defined using a maximum bound on the hill climbing move.

Theorem 5-8 [15]: Let $d(s)$ be the maximum amount (or depth) of cost which can be hill-climbed at a given temperature T and stream length s. Then

$$\exp(-d(s)/T) \geq \frac{1}{s} \Rightarrow d(s) \leq T \ln s \qquad (5\text{-}1)$$

This means that there is possibility to choose $d(s)$ hill climbing move in s moves [15]. The maximum hill climbing depth is the function of temperature and log of the stream length.

The error-present simulated annealing has a small hill climbing power than sequential simulated annealing, so the error-present algorithm is likely to be kept in a local minimum due to cost error (Fig. 9). Hill climbing power is the degree of accepting the hill climbing move. In order to get out of the local minimum and converge to the optimal result, the error-present algorithm must have the same hill climbing power as the sequential simulated annealing algorithm. Since the decreased hill climbing power is due to the cost error, the following theorem is derived.

Theorem 5-9: The hill climbing depth of the error-present algorithm (d_e) is less than that of the sequential algorithm (d_a) by at most the amount of error (E).

$$d_a \leq d_e + E$$

where d_a is the hill climbing depth of sequential simulated annealing for one hill climbing move, d_e is the hill climbing depth of the error-present algorithm for one hill climbing move, and E is the hill climbing error derived from Theorem 5-6.

Proof

A loss of hill climbing power is introduced only by pessimistic errors ($\Delta C_e > \Delta C_a > 0$). Hill climbing power, probabilistically, is $d_a = \Delta C_a \cdot \exp(-\Delta C_a/T)$ in the error-present annealing process and $d_e = \Delta C_e \cdot \exp(-\Delta C_e/T)$ in the sequential annealing process. Using E (from Theorem 5-6) and pessimistic condition ($\Delta E = \Delta C_e - \Delta C_a$ where $\Delta C_e > \Delta C_a > 0$), we have

$$\begin{aligned} d_a - (d_e + E) &= \Delta C_a \cdot \exp\left(-\frac{\Delta C_a}{T}\right) \\ &\quad - \left(\Delta C_e \cdot \exp\left(-\frac{\Delta C_e}{T}\right)\right. \\ &\quad \left. + \Delta C_e \cdot \exp\left(-\frac{\Delta C_e}{T}\right) \cdot \left(\exp\left(\frac{\Delta E}{T}\right) - 1\right)\right) \\ &= \Delta C_a \cdot \exp\left(-\frac{\Delta C_a}{T}\right) \end{aligned}$$

$$-\left(\Delta C_e \cdot \exp\left(-\frac{\Delta C_e - \Delta E}{T}\right)\right)$$

$$= \Delta C_a \cdot \exp\left(-\frac{\Delta C_a}{T}\right) - \Delta C_e \cdot \exp\left(-\frac{\Delta C_a}{T}\right)$$

from Lemma 5 – 1

$$= \exp\left(-\frac{\Delta C_a}{T}\right) \cdot (\Delta C_a - \Delta C_e) \leq 0$$

from pessimistic condition

So, $d_a \leq d_e + E$. □

Next, an extra stream length (u) is required for the decreased amount of hill climbing depth, $E(s)$, throughout the stream length s.

Corollary 5-3: The extra move (u) to tolerate the cost error $E(s)$ is given by

$$\exp\left(-\frac{E(s)}{T}\right) \geq \frac{1}{u},\ so\ u \geq \exp\left(\frac{E(s)}{T}\right) \quad \text{from Theorem } 5-8$$

For a given temperature T, at least u moves have a hill climbing power $E(s)$; and with stream length s, there is a hill climbing power $d_e(s)$ in an error-present algorithm.

Corollary 5-4:

$$\exp\left(-\frac{d_e(s)}{T}\right) \geq \frac{1}{s} \qquad \text{from Theorem } 5-8$$

Now the stream length s_a can be calculated for the error-tolerable algorithm having a regular hill climbing depth $d_a(s)$. That is, in order to increase the hill climbing depth to match that of sequential simulated annealing, the stream length s_a is required.

Theorem 5-10: When a total amount of cost error ($E(s)$) occurs during stream length s, $s_a = s \cdot u$ stream length is needed to tolerate the cost error.

Proof

s_a is defined as $\exp(-d_a/T) \geq 1/s_a$

$$\exp\left(-\frac{d_a}{T}\right) \geq \exp\left(-\frac{d_e + E}{T}\right) \qquad \text{from Theorem } 5-9$$

$$\exp\left(-\frac{d_a(s)}{T}\right) \geq \exp\left(-\frac{d_e(s) + E(s)}{T}\right)$$

$$= \exp\left(-\frac{d_e(s)}{T}\right) \cdot \exp\left(-\frac{E(s)}{T}\right) \qquad \text{from ergodicity}$$

$$\geq \frac{1}{s} \cdot \frac{1}{u} \qquad \text{from Corollary 5-3 and Corollary 5-4}$$

So $s_a = s \cdot u$ stream length is needed for the error present algorithm to have the same hill climbing depth as the sequential annealing process has in the stream length s. □

The next task is to solve how to define the extra stream length factor u, considering the time for global update. From Corollary 5-3,

$$E(s) \leq T \cdot \ln u, \qquad \text{since } \exp\left(-\frac{E(s)}{T}\right) \geq \frac{1}{u} \qquad (5\text{-}2)$$

In order to decrease the extra stream length factor u for speedups, the maximum tolerable cost error $E(s)$ must be decreased as well. So, the extra stream length factor u and the maximum tolerable cost error $E(s)$ are proportional. That is, as the stream length increases, the cost error increases, so the extra stream length factor (u) must increase to keep the convergence.

For example, if a 10% increase of Markov chain length is allowed, i.e. u= 1.1, then the maximum error bound $E(s)$ can be calculated using equation (5-2). If the measured amount of cost error in a given stream length s is greater than the maximum bound cost error $E(s)$, the stream length will be decreased. If the measured amount of cost error in a given stream length s is less than the maximum bound cost error $E(s)$, the stream length will be increased. When the stream length is changed, the Markov chain length M is kept fixed, i.e. Markov chain length M =stream length (s) × # of global updates in a given temperature. The pseudo code for the error tolerant scheme is in Figure 11.

PROCEDURE ERROR - TOLERANT SIMULATED ANNEALING

begin
 INITALIZE;
 k:= 0;
 repeat
  calculate $|<E>|$ in T_k
   /*Average cost error in one accepted move*/
  $E(s) = 0$
  /* $E(s)$ is the total amount of cost error in a given stream length*/
  repeat
   PERTURB(config. $i \rightarrow$ config. $j, \Delta C_{ij}$);
   if $\Delta C_{ij} \leq 0$ then accept
   else

$$E(s) = E(s) + \Delta C_{ij} \cdot \exp\left(-\frac{\Delta C_{ij}}{T_k}\right) \times \left(\exp\left(\frac{i \cdot \alpha \cdot |<E>|}{T_k}\right) - 1\right)$$

    /* i is the i - th iteration in the stream length*/
    /* α is the acceptance ratio*/
   if $\exp(-\Delta C_{ij}/T_k) >$ random [0.1) then accept;
   if accept then
    UPDATE(configuration j);
  until **equilibrium is approached sufficiently closely;**
  $T_{k+1} := f(T_k)$;
  k:= k + 1;
  if $(E(s) \leq T_k \cdot \log u)$ /* $u = 1.1$ */
    increase stream length;
  else
    decrease stream length;
 until **stop criterion == true(system is 'frozen');**
end.

Fig. 11. The Error-Tolerant Simulated Annealing

VI. EXPERIMENTAL RESULTS

The new adaptive error-tolerance method (see Section V), which will be referred to as the *adaptive method*, was implemented on a 16-node Intel iPSC/2. The target problem was the composite stock cutting problem (see Section I), which was decomposed specially along the space of stock sheet.

The parallel space-decomposition simulated annealing algorithm was implemented in 4 nodes. A total of 16 irregular patterns were used. The Markov chain length was 500. To track the behavior of the cost error, the weight of the affinity relation term was set much greater than that of the cluster term, since the cost error occurs only in the affinity relation term of the cost function. The fixed stream length method, which will be referred as the *static method*, was implemented twelve times on each stream length. The stream length was varied to note its effect on the cost error.

Fig. 12 shows that the average final cost starts to increase above stream length 10. Using 4 patterns per node and an optimal stream length of 10, the herein algorithm is much more robust with respect to the cost error than the floor planning algorithms of Jayaraman [6] and Durand [9], where the optimal stream length was set equal to the number of patterns per node. This can be explained by Theorem 5-7. The portion of the cost error (ΔE) in the estimated cost (ΔC_e) of the composite stock cutting problem may be smaller than that of floor planning problems. This can be due to more inter-processor move generations where the correct location of pattern is broad-casted to the neighbor processors. And more we can guess that the cost error-free term (cluster term of the cost function) is weighted greater than the floor planning algorithm. From Theorem 5-7, this reduces the this reduces the total amount of the cost error, so the stream length can be increased keeping the convergence to the optimal results.

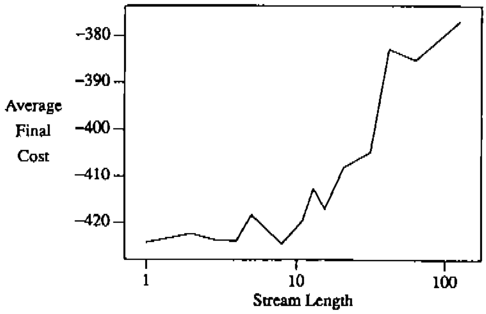

Fig. 12. Final Cost vs. Stream Length

Comparing the stream length at each temperature (Figure 13) with the annealing curve (Figure 14), the stream length reduces to 2 in the critical region where specific heat is very high. However, the stream length increases to 125 far from the critical region, i.e. the global update is done only once at the end of each temperature. The stream length varies dynamically according to the annealing curve. This means the cost error has little effect on the annealing process away from the critical region, but affects it greatly in the critical region. This corresponds to the fact that the annealing process proceeds rapidly away from the critical region, but much more slowly in the critical region.

In Table 2, Adp means the adaptive method, and Static-10 means that the stream length was fixed at 10. Since the average final cost starts to increase above the stream length 10, the stream length of 10 was selected for the static method. The average final costs was almost the same. However, the standard deviation of the adaptive method was smaller than that of the static method, as expected. The average stream length of the adaptive method is larger than that of the static method. Since the number of global update was inversely proportional to the stream length, the average number of global updates was reduced 6.3 times in the adaptive method, compared to the static method. From the above data (Figs. 13, 14, and Table 2), the adaptive method adapts the stream length dynamically, with comparable final results.

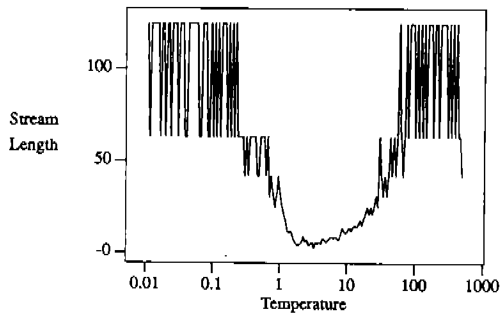

Fig. 13. Stream Length vs. Temperature.

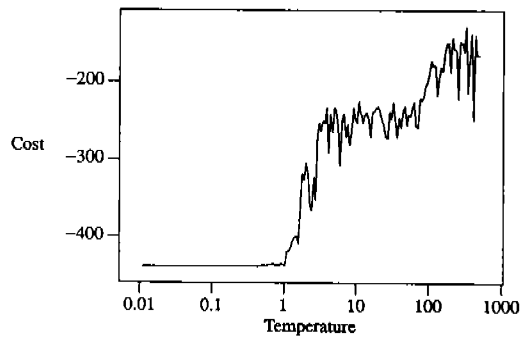

Fig. 14. Annealing Curve.

Table 2. Final Cost of Adaptive and Static Methods.

	Mean	Std. Dev	Worst	Best
Adp	-424.5	8.29	-414.4	-436.2
Static-10	-424.3	11.70	-402.9	-436.2

As the second experiment, 16 different sets of patterns were implemented to observe the results of the adaptive method. The number of patterns varied from 128 to 160. A cooling schedule was set such that the initial temperature was about 200,000, the temperature decrement ratio was 0.98 to 0.99, and the Markov chain length was 5,000 to 20,000. In other words, the weight of the cluster term in the cost function is balanced with that of the affinity relation term.

Table 3. Speedups of Adaptive and Static Methods.

node size		Mean	Std.Dev.	Max	Min
2	Adp	1.47	0.16	1.67	1.15
	Static-5	1.18	0.09	1.28	1.02
4	Adp	3.55	0.50	3.95	2.11
	Static-5	2.76	0.29	3.16	2.05
8	Adp	6.98	0.94	7.64	4.13
	Static-5	4.89	0.47	5.46	3.97
16	Adp	11.94	1.34	13.04	8.10
	Static-5	7.63	0.84	9.45	6.35

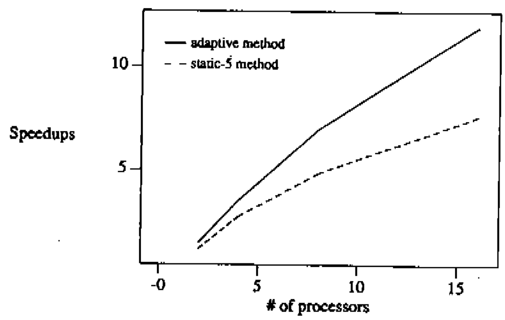

Fig. 15. Speedups of Adaptive and Static Methods.

In Table 3, Adp means the adaptive method and Static-5 represents the static method, where the stream length was fixed at 5. Table 3 indicates the speedups of both the adaptive method and the static method comparing with the sequential annealing process. The mean of speedup of the adaptive method was greater than that of the static method over the entire processor range. However, the standard deviation of the adaptive method was always greater than that of the static method. Since the experiment was done on different sets of patterns, the run time may have varied according to the problem, in order to maintain the convergence in the adaptive method.

Figure 15 plots Table 3. Figure 15 indicates that as the number of processors increased the adaptive method was much better than the static method in speedups. As the number of processors increased, the global update time increased as well. Thus, the speedups of the parallel implementation was reduced as the number of processors increased.

Since the adaptive method reduced the global update frequency, the adaptive method achieved better speedups than the static method for a large number of processors.

Table 4. Final Cost of Adaptive and Static Methods.

node size		Mean	Std.Dev.
Seq.		583848	91797.9
2	Adp	587628	83453.7
	Static-5	588608	83138.0
4	Adp	578001	77181.2
	Static-5	575734	79882.2
8	Adp	574773	76009.6
	Static-5	575499	75534.0
16	Adp	589618	75242.4
	Static-5	591113	76073.4

Table 4 compares the final cost of the adaptive method with that of the static method. The stream length of the static method varied from 5 to 100 where the convergence was assumed to be maintained. The mean final cost of the adaptive method was smaller than that of the static method for the entire processor range. In 16 nodes, the final cost of the parallel implementation was greater than that of the sequential annealing process. This may result from restricted mobility in move generation. Let the cost deviation of the parallel implementation be defined as:

$$\text{Cost Deviation} = \frac{\text{Cost of parallel - Cost of Sequential}}{\text{Cost of Sequential}}$$

The cost deviation of the parallel implementation using 16 nodes was less than 1% for the adaptive method and 1.2% for the static method. The standard deviation of the adaptive method was smaller than that of the static method for all node ranges. This corresponds to the previous experiments (Table 2)

As the third experiment, the stream length of the static method was varied to determine the cost error behavior using a different set of the stream lengths. This experiment was similar to the the first experiment (Table 2). However, the weight of the cluster term was balanced with that of the affinity relation term in order to consider the packing density. The experiment was done 3 times using 128 regular patterns and 16 nodes. The initial temperature was set around 20,000; the decrement ratio was 0.98 to 0.985; and the Markov chain length was 10,000.

In Table 5, the performance is defined as the inverse of the product of run time and cost.

$$\text{performance} = \frac{1}{\text{cost} \times \text{run time}}$$

Table 5. Performance of Static and Adaptive Methods.

Stream Length	Time(Avg)	Cost(avg)	Performance(avg)
625	6.96378e+06	797019	1..80075e-13
208	7.0591e+06	791015	1.78955e-13
125	7.31633e+06	790672	1.72831e-13
89	7.38138e+06	784053	1.72784e-13
69	7.48652e+06	794623	1.68088e-13
57	718042e+06	791005	1.76246e-13
45	7.52592e+06	787917	1.68743e-13
35	7.83123e+06	786564	1.62355e-13
26	8.44653e+06	784167	1.50949e-13
21	8.56733e+06	787604	1.48084e-13
16	9.43514e+06	788615	1.34336e-13
10	1.09276e+07	788676	1.16026e-13
5	1.74531e+07	783721	7.31384e-14
Avg	8.73649e+06	788896	1.45192e-13
Adaptive	7.20755e+06	786792	1.76375e-13
Sequential	9.34506e+07	753421	1.42030e-14

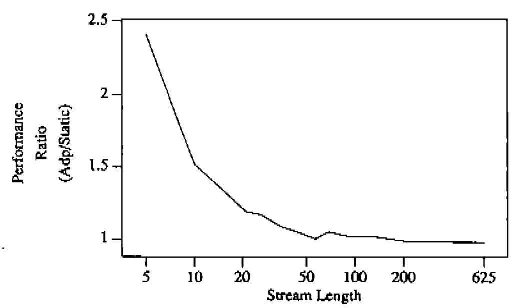

Fig. 16. Performance of Adaptive vs. Static Methods.

The performance was used as a parameter of a kind of goodness test for the trade-off between run time and cost. The weights of cost and run time are set to be equal for convenience. The cost deviation of the cost of the sequential process was 4.4% for the adaptive method and 4.7% for the static method.

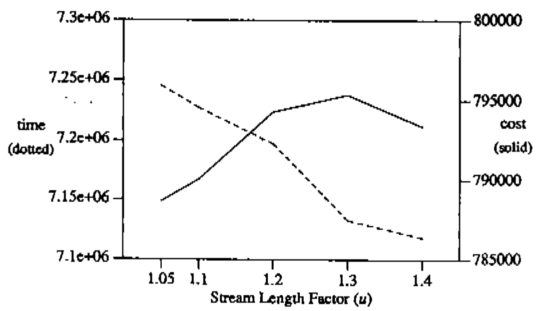

Fig. 17. Time and Final Cost for Various Stream Length Factor.

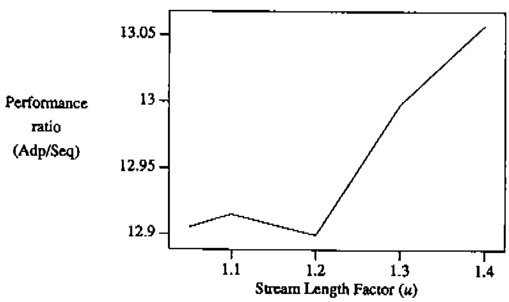

Fig. 18. Performance for Various Stream Length Factor.

In Fig. 16, the *performance ratio* is defined as the performance of the static method over that of the adaptive method (performance of Adaptive/performance of Static). Fig. 16 indicates that the static method can get a fairly good performance around the fixed stream length 100. In other words, considering the trade-off between the run time and the cost, 100 is a desirable stream length. When the stream length was larger than 200, the run time reduces with a relatively small increase of the cost. So the performance defined here can not represent the trade-off properly in large stream length regions. The performance of the adaptive method was lower than that of the static method in the proper regions. The average performance ratio of the static method to that of the adaptive method was 0.82.

In Fig. 17 and 18, stream length factor (u) is varied without the additional stream length, so we can see the final cost change and run time. The final costs are almost the same and high when the stream length factor is above 1.2, u= 1.2. This means that the final cost diverges fully when $u \geq 1.2$. The run-time decreases as the stream length factor (u) increases because the additional stream length according to u is not introduced. In Fig. 18, when the stream length factor is above 1.2, the run-time decreases with almost the same final cost, so performance increases too. That is, the performance definition does not represent the goodness test.

From the experimental results, the adaptive method is well suited for relaxing the frequency of the global updates, i.e. for increasing the stream length while maintaining the quality of the final results comparatively. Aside from the improved speedups, the adaptive method has an advantage over the static method in that, in the latter, much implementation is needed to determine the optimal stream length.

VII. CONCLUSION

Simulated Annealing is a general purpose algorithm that can be applied to the broad range of NP-complete problems such as the traveling salesman problem, graph theory, VLSI cell placement, and composite stock cutting.

Since Simulated Annealing is a stochastic process, the real disadvantage is the massive computing time required to converge to a near optimal solution. One of the promising approaches for speeding up the Simulated Annealing algorithm is parallelization. Distributed memory multicomputers show the most promise in achieving large parallel speedups. However, in a distributed memory architecture such as a hypercube, there is no globally available, centrally located system state. Updating the entire global state S thus involves explicit message traffic and is a critical bottleneck. To mitigate this bottleneck, it becomes necessary to amortize the cost of these state updates over as many parallel move evaluations as possible by using an approximate cost calculation. Thus, error in maintenance of the cost function $C(S)$ is inevitable and bounds must be placed on this error in order to assure convergence to the correct result.

The Simulated Annealing algorithm can be looked upon as a random iterative improvement algorithm with a certain probability of making mistakes by accepting hill-climb moves that increase the cost to get out of local minima. Since Simulated Annealing randomly selects hill-climbing moves, it can tolerate some degree of cost error. Previous work on cost-error-tolerant schemes is mainly based on experimental results [8].

In this paper, we prove, analytically, bounds on the cost error as a function of global update frequency, or stream length s. The erroneous move decision is exponentially distributed with respect to the parameter $T > 0$, i.e. *Prob* [The Erroneous Move Decision With Cost Error [0, ΔE]] is 1-exp ($-\Delta E/T$) where ΔE is the cost error in one move at temperature T from Theorem 5-2. With a cost error distribution, we can find the total probability of cost error in s parallel moves without global updating

$$P_T = Prob\,[\Delta C_e > 0] \cdot \exp\left(-\frac{\Delta C_e}{T}\right) \cdot \left(e^{\frac{s \cdot \alpha \cdot |<E>|}{T}} - 1\right)$$

where s is the stream length, α is the acceptance rate, $|<E>|$ is the average cost error in one acceptance move, T is a fixed temperature from Theorem 5-5. So $\Delta E = s \cdot \alpha \cdot |<E>|$.

With a total probability of cost error (P_T), we can calculate the amount of cost error in the hill climbing move,

$$E = \Delta C_e \cdot \exp\left(-\frac{\Delta C_e}{T}\right) \cdot \left(\exp\left(\frac{\Delta E}{T}\right) - 1\right).$$

We assume that hill-climb power is decreased proportional to the amount of cost error from Fig. 9. To recover the decreased hill-climb power, we introduced 10% increase of Markov Chain length and calculated the optimal stream length. So, in the presence of cost error, we need $s \cdot u$ stream length in order to have the same hill-climb power as much as the original (error-free) algorithm has in stream length s.

$$\exp\left(\frac{d_a(s)}{T}\right) \geq \frac{1}{s} \cdot \frac{1}{u}$$

where $d_a(s)$ is the hill-climb power of error-free simulated annealing during stream length s, and u is the extra stream length increase, i.e. in the experiment we set u to 1.1.

We applied the adaptive cost-error-tolerant scheme on the stock-cutting problem in an Intel iPSC/2. We saved 6.3 times of number of global update comparing a fixed stream length scheme keeping same solution quality. In comparing the speedups of running time in 16 processors, the speedup of fixed stream length is 7.6 and the speedup of adaptive error control scheme is 11.9. Aside from the improved speedups, the adaptive error control scheme has an advantage over the static stream length (original) method in that, in the latter, to find out the optimal fixed stream length, we have to run many experiments. However, in our adaptive error control scheme, we vary the stream length by choosing large stream length in high and low temperature regions, and in the critical temperature region, choosing small stream length dynamically.

REFERENCES

[1] Kirkpatrick, S., Gellatt Jr., C.D. and Vecci, M.P., "Optimization by Simulated Annealing," Science, Vol. 220, 1983, pp 975-986.

[2] van Laarhoven, P.J.M., and Aarts, E.H.L., "Simulated Annealing: Theory and Applications," D.Reidel Publishing Co., 1988.

[3] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. "Equation of State Calculations by Fast Computing Machines," Chem. Physics, Vol. 21, 1953, pp 1087-1092.

[4] Lutfiyya H., McMillin B., Poshyanonda P., and Dagli C.,"Composite Stock Cutting Through Simulated Annealing," Mathl. Comput. Modeling, Vol. 16, No. 1, 1992, pp 57-74. Pergamon Press, New York.

[5] Greening, D.R., "A Taxonomy of Parallel Simulated Annealing Techniques," Computer Science Technical Report, University of California, Los Angeles, CSD-890050, August 1989.

[6] Jayaraman, R. and Darma, F., "Error Tolerance in Parallel Simulated Annealing Technique," Proc. the International Conference on Computer Design, 1988.

[7] Grover, L.K., "A New Simulated Annealing Algorithm for Standard Cell Placement," Proc. the International Conference on Computer-Aided Design, November 1986.

[8] Banerjee, P., Jones, M.H., and Sargent, J.S., "Parallel Simulated Annealing Algorithm for Cell Placement on Hypercube Multiprocessors," IEEE Trans. on Parallel and Distributed System, Vol. 1, No. 1, January 1990.

[9] M.D. Durand, "Parallel Simulated Annealing: Accuracy versus speed in placement," IEEE Design and Test, June 1989, pp 8-34.

[10] Casotto, A., Romeo, F. and Sangiovanni-Vincentelli, A.,"A Parallel Simulated Annealing Algorithm for the Placement of Macro-Cells," IEEE Transactions on Computer-Aided Design, September 1987, pp 838-847.

[11] J.S. Rose, D.R. Blythe, W.M. Snelgrove, and Z.G. Vranesic, "Fast, High Quality VLSI Placement on an MIMD Multi-processor," Proc. International conference on Computer-Aided Design, 1986, pp 42-45.

[12] F. Darema and G.F. Pfister, "Multipurpose Parallelism for VLSI CAD on the RP3," IEEE Design & Test of Computers, pp 19-27, Oct. 1987.

[13] Jayaraman, R. and Rutenbar, R.A., "Floorplanning by Annealing on a Hypercube Multiprocessor," Proc. the International Conference on Computer-Aided Design, 1987.

[14] Jonathan Rose, Wolfgang Klebsch, and J.

Wolf, "Temperature Measurement and Equilibrium Dynamics of Simulated Annealing Placement," IEEE Trans. on Computer-Aided Design, Vol. 9. No. 3. March 1990.

[15] White, S.R., "Concepts of Scales in Simulated Annealing," Proc. IEEE Int. Conference on Computer Design, Port Chester, November 1984, pp 646-651.

Chul-Eui Hong received a B.S. Degree in metallurgical engineering from Hanyang University in 1985, an M.A. in computer and information science from New Jersey Institute of Technology in 1989, and a Ph.D. in computer science from University of Missouri-Rolla in 1992. He is currently a senior researcher in computer H/W Section of ETRI.

Bruce M. McMillin received a Ph.D. degree in computer science from Michigan State University in 1987. He is currently a professor of University of Missouri-Rolla. His research interests include fault tolerance algorithm in parallel computing, communications, and distributed systems.

Hee-Il Ahn received a B.E. degree in electronics from Seoul National University in 1973. He worked in KIST from 1976 to 1978 where he developed Hangul display terminal using microprocessor. He is working in ETRI as the head of computer H/W Section since 1978 and developing TICOM, or multiprocessor system.