

# 温度並列シミュレーテッド・アニーリング法の応用 - ブロック配置問題への適用と評価 -\*

小西 健三<sup>1</sup>      瀧 和男<sup>2†</sup>

{konishi,taki}@seg.kobe-u.ac.jp

<sup>1</sup>神戸大学大学院自然科学研究科 知能科学専攻<sup>‡</sup>

<sup>2</sup>神戸大学工学部情報知能工学科<sup>§</sup>

## 概要

温度並列 SA 法は新しい並列化 SA 法であるが、提案後、応用が先行したままで実行時間や最適化能力についての評価は行なわれていなかった。本論文では温度並列 SA 法を LSI ブロック配置問題に適用することにより、温度並列 SA 法の評価を実験的に行なった。最適化能力における温度並列 SA 法と逐次 SA 法の比較では、同じアニーリングステップ数での比較に加えて、同じ CPU 時間を与えた場合の比較においても、温度並列 SA 法の方が優れていることが判明した。つまり、一台の CPU で同じ計算時間をかける場合でも、逐次 SA 法より温度並列 SA 法の方が良質の解が得られることを示しており、温度並列 SA 法のアルゴリズム自体の優位性を確認した。また、温度並列 SA 法を適用した LSI ブロック配置プログラムで配置を行ない、市販の配線ツールで実配線を行なった結果、設計品質に関しても従来までに報告されている他のシステムと比較して良質の結果を得ることができた。

## Abstract

"Temperature-Parallel Simulated Annealing(TPSA)" is a new concurrent simulated annealing(SA) algorithm. Although several applications of TPSA have been reported, optimization capability and execution time have not been evaluated precisely. In this paper, we apply TPSA to LSI macro block placement problem, and evaluate the optimization capability of the algorithm, execution time, and placement quality comparing with conventional sequential SA. It is also examined that TPSA can get the linear speed-up without degrading the convergence quality. Measurement results shows that TPSA(concurrent execution with 1CPU) is superior in optimization capability to the sequential SA, spending the same CPU time. Furthermore, placement results for MCNC standard benchmarks show good chip area and wire length to other systems that has been reported.

## 1 はじめに

広範囲の組合せ最適化問題に適用可能な汎用アルゴリズムとして、シミュレーテッド・アニーリング法(以下、SA法)[15]がある。SA法は、改悪方向への状態遷移を確率的に認めることによって、理論上は真の最適解への収束が保証されている[1]。しかしながら、そのためには非現実的に長い時間が必要となる。またSA法では、温度と呼ばれる制御変数のスケジュールが問題に適したものでないと、得られる解のクオリティに影響を与えることが報告されている[16]。しかし、様々な問題に適用可能な万能

の温度スケジュールは報告されていない。このような欠点にもかかわらず、SA法はその柔軟な枠組と、比較的良質な解が得られやすいということもあって、現在までにLSI設計など様々な組合せ最適化問題に適用されてきた。

一方、計算時間短縮を目的としたSA法の並列化に関する研究も行なわれてきた。特に、プロセッサ台数に比例したスピードアップを得ることができる手法として、各プロセッサで複数の状態遷移を同時に扱う方法が最もよく研究されている[1]。しかしながらこの方法では、評価関数値の変化を正確に計算できなくなる等の問題点が生じる[11]。このことがアルゴリズムの最適化能力にどのように影響を及ぼすかについて検討されている[1]。

そこで、上述の問題点を解決するような新しい並列化SA法として、温度並列SA法が提案された[8,9]。しかし提案後、最適化能力の詳しい評価

\*An Application of Temperature-Parallel Simulated Annealing - Block Placements and Their Evaluations -

<sup>†</sup>Kenzo KONISHI<sup>1</sup> and Kazuo TAKI<sup>2</sup>

<sup>‡</sup>Division of Intelligent Science, Graduate School of Science and Technology, Kobe University

<sup>§</sup>Department of Computer and Systems Engineering, Faculty of Engineering, Kobe University

が行なわれていないまま応用が先行していた [2,6]. すなわち, その利点は利用されているものの, 逐次 SA 法と比較した場合の実行時間や最適化能力の優劣については不明であった.

本論文では, 温度並列 SA 法を LSI ブロック配置問題に適用し, 実行時間や解品質についての評価を行なう. 評価は逐次 SA 法と実験的に比較し, その優位性を示すものである. また, 温度並列 SA 法を適用した配置プログラムを用い, ベンチマークデータに対して設計を行なった結果も示す.

## 2 温度並列 SA 法

### 2.1 SA 法

SA 法は, 組合せ最適化問題における評価関数値  $E$  を最小にする解を, 確率的に探索する近似解法の一つである. その概要を以下に示す.

```

procedure SIMULATED ANNEALING:
begin
  Initialize( $i(0), T(0)$ );
   $k := 0$ ;
   $i := i(0)$ ;
  repeat:
    for  $l := 1$  to  $L(k)$  do
      begin
        Generate( $j$ );
        if  $f(j) \leq f(i)$  then  $i := j$ ;
        else
          if  $\exp(-\frac{f(i)-f(j)}{T(k)}) > \text{random}[0,1)$ 
            then  $i := j$ ;
        end;
       $k := k + 1$ ;
      UpdateTemp( $T(k)$ );
    until stopcriterion
  end;

```

ここで制御変数  $T$  と評価関数  $E$  はそれぞれ統計物理学とのアナロジーから温度, エネルギーと呼ばれる. SA 法は, 保持している解に微小変形を施し, 新しい解が以前の解よりも評価関数値において良質であれば受理, そうでなければ確率  $\exp(-\Delta E/T)$  に従って受理する. 一方, 温度スケジュールに関しては, 通常は最適解への収束性を犠牲にして, 式 (1) の温度スケジュールを適用することが多い.

$$T_{n+1} = \alpha T_n, \quad 0 < \alpha < 1 \quad (1)$$

しかし, 温度スケジュールが問題に適したものでなければ解のクオリティに大きく影響することも報告されており [16], 温度スケジュールに関する研究も行なわれている. しかしながら, 多様な問題に適用可能な万能の温度スケジュールはまだ報告されていない.

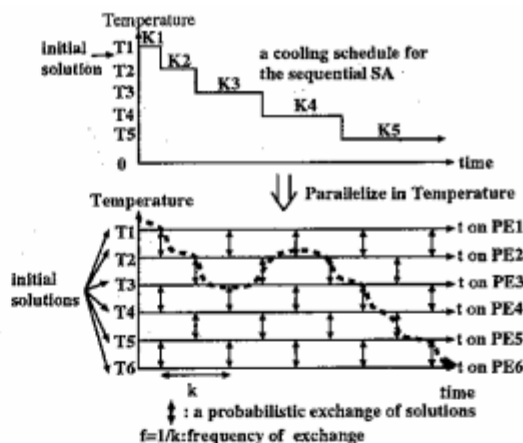


Fig.1: Temperature Parallel Simulated Annealing

図 1: 温度並列 SA 法

また, スピードアップを目的とした SA 法の並列化についての研究も行なわれている [3,14]. 並列 SA 法については, 大きく「同期式」と「非同期式」の 2 つに分類可能であり [3], 特に並列処理でスピードアップの効果が得られやすいのは非同期式 SA 法である. しかし, 非同期式 SA 法で最も良く研究されている各プロセッサが互いに同期をとることなく状態遷移を評価する手法においては, 別のプロセッサは古い状態からの処理をしていることになり, 遷移は不当であり, 最適解への漸近収束を示す理論も保証されなくなる. この手法における不当な状態遷移は, 特に高温部で顕著なものとなるが, 低温部ではほとんどの遷移が受理されないため, この誤差による影響は小さいと考えられる. そのため, 非同期式 SA 法に関する評価は賛否両論である [1,3]

### 2.2 温度並列 SA 法

2.1. で述べた通常の SA 法では, 温度  $T$  を温度スケジュールに従って単調減少させる. 一方, 温度並列 SA 法では, それぞれ相異なる温度を担当するプロセッサに異なる初期解を与え, 一定温度の下で同時並列にアニーリングを行なう. このとき, 逐次 SA 法で温度  $T$  から  $T'$  に冷却することは, 温度並列 SA 法では温度  $T$  を担当するプロセッサと温度  $T'$  を担当するプロセッサの間で解を交換することに相当する (図 1). このプロセッサ間の交換確率は式 (2) に従う [8,9].

$$P(T, E, T', E') = \begin{cases} 1 & \Delta T \cdot \Delta E < 0 \\ \exp(-\frac{\Delta T \cdot \Delta E}{T T'}) & \text{otherwise} \end{cases} \quad (2)$$

ただし,  $\Delta E = E' - E$ ,  $\Delta T = T' - T$  である.

ここで, 温度並列 SA 法と同様に, 相異なる温度で一定温度の逐次 SA を行ない, 異なる温度間で確

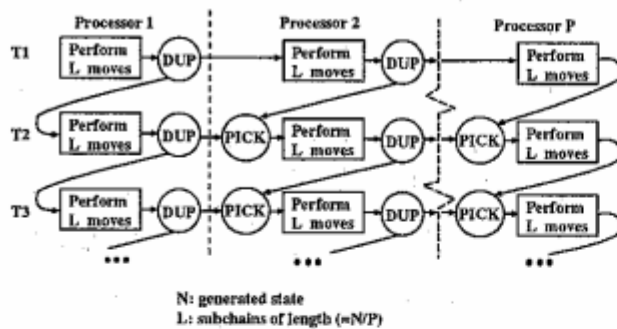


Fig.2: Systolic algorithm [3,4]

図 2: [4] のアルゴリズムの動作

率的に解の交換を行なうモデルとして報告されている, 文献 [4] に挙げられている手法 (図 2) と, 温度並列 SA 法の違いについて述べる. 手法 [4] では, 2 組の解, 温度をそれぞれ,  $(E, T), (E', T')$  であるとしたとき, 2 組の解が交差する部分で, 前者である  $(E, T)$  が選択される確率は

$$p = \frac{p_0}{p_0 + p_1}, \text{ where } p_0 = \frac{1}{Z(T)} \exp\left(-\frac{E}{T}\right)$$

$$\text{and } p_1 = \frac{1}{Z(T')} \exp\left(-\frac{E'}{T'}\right)$$

となる. このとき, 選択確率を決定するためには, 分配関数  $Z(T), Z(T')$  を計算する必要があるが, 一般的に分配関数を計算することは統計物理学における非常に困難な問題でもあり, 厳密解は現在でも限られたものに対してしか得られていない. 温度並列 SA 法では, 交換確率に分配関数  $Z(T)$  が含まれていないため, 理論面だけでなく実装面においても, 手法 [4] と比較して優位であると思われる. また, 手法 [4] では, 解が辿る温度スケジュールは, 単調減少となる.

以下の説明では各温度における SA 処理を異なるプロセッサが実行すると仮定して説明を行なうが, プロセッサをプロセスと読み替えば, 温度数分の並行プロセスを一台の CPU で時分割実行させると考えても差し支えない.

### 2.3 温度並列 SA 法の利点

温度並列 SA 法は通常の逐次 SA 法と比較して, 以下のような利点を持つ.

**温度スケジュールの自動化** 逐次 SA 法を実装する上で問題となる温度スケジュールは, 温度並列 SA

法においてはプロセッサ間で解の確率的交換を行なうことにより自動化される.

**アルゴリズムの時間的一様性** 温度並列 SA 法は時間的に一様なアルゴリズムである. このことは, プログラムを任意の地点で終了することができ, また継続すれば解の改善を続けることを示している. これに対して通常の逐次 SA 法では, 得られた解が不満足なときは温度を再び上げることが必要となる. しかも, このとき温度をどの程度上げるべきかが問題となる. [10] では, 再加熱 (Re-Annealing) についての報告もなされている.

**並列処理との親和性** 並列処理で効率を下げる大きな原因の一つとして, プロセッサ間通信が考えられる. 温度並列 SA 法では, 各プロセッサ上で独立に一定温度のアニーリング処理が行なわれるため, プロセッサ間通信が必要となるのは解交換の瞬間のみである. よって, 温度並列 SA 法は並列処理と非常に親和性の高い並行アルゴリズムであるといえる.

## 3 ブロック配置問題への適用

### 3.1 ブロック配置問題の定式化

本論文で扱うブロック配置問題では, 次のようなモデルを考える.

ブロックは形状, 入出力端子位置が固定で, 寸法は任意のものを扱う. ブロックの向きは, 90 度ごとの回転と, 鏡像反転が許されるものとする. また, 今回の実装においては, クリティカルネットや, 電源線に関する制約は一切取り扱っていない.

### 3.2 状態遷移

SA 法における解の微小変形 (すなわち, 解の状態遷移候補の生成) は, 1 つのブロックの移動, 回転, 裏返しと, 2 ブロック間の場所の交換を, 乱数で選択するものとする.

### 3.3 評価関数

評価関数  $E$  は, 式 (3) を用いた.

$$E = w_o O + w_s S + w_l L \quad (3)$$

ここで,  $O, S, L$  はそれぞれ, 重なりペナルティ, 隙間ペナルティ, 仮想配線長和を表し,  $w_o, w_s, w_l$  はそれぞれの項の重みである. 第 1 項の重なりペナルティは, LSI 配置問題で評価関数として頻繁に用いられる項である. 第 2 項の隙間ペナルティはチップ面積最小化のための項で, 式 (4) に従う.

$$S = S_{outer} - S_{block} + O \quad (4)$$

ここで、 $S_{outer}$ は全てのブロックを囲む最小矩形の面積、 $S_{block}$ はブロック面積の総和、 $O$ は式(3)の重なりペナルティである。第3項の仮想配線長も、LSI配置問題で頻繁に用いられる項である。従来の多くの仮想配線長は、結線すべき端子を全て囲む最小矩形のマンハッタン距離が用いられている。しかし、配置対象がブロックのように大きい場合には、実配線との誤差が大きくなる。

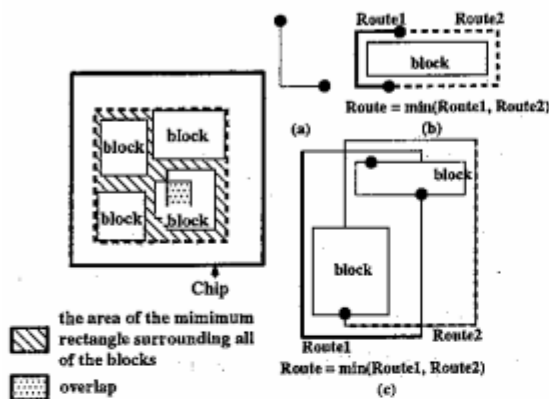


Fig.3:Space penalty Fig.4:Wire length  
図3: 隙間ペナルティ 図4: 仮想配線長

そこで、本プログラムの仮想配線長は、図4のように結線すべき端子間にブロックが存在する場合にはネットの迂回を考慮した最小矩形を考え、そのマンハッタン距離とした。

### 3.4 パラメータの決定

温度並列SA法では、時間の関数としての温度スケジュールは自動的に決定されるが、最高温度、最低温度、温度数、温度の振り分け、解交換周期は指定する必要がある。

まず、最高温度、最低温度は次のように考えた。

最高温度 最大の改悪となる状態遷移が50%の確率で受理されるような温度 [18].

最低温度 最小の改悪となる状態遷移が一定温度のアニーリング処理における解交換周期内に1回は受理されるような温度。

次に、温度数は実験的に決定した。具体的には8,16,32,64,128温度で、3000回の温度交換分だけ温度並列SA法を実行し、後述する各データに対して

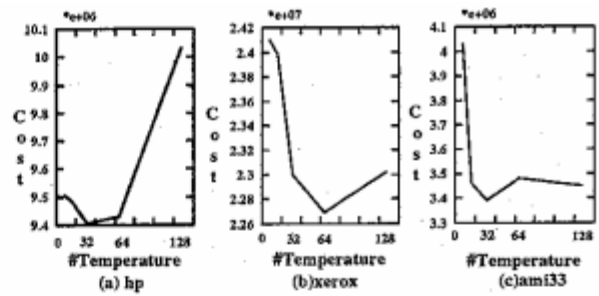


Fig.5:#Temperature vs. cost value

図5: 温度数と評価関数値に関する実験結果

図5のような結果を得た。図5.(a),(c)では、32温度付近で最高の最適化性能を示し、図5.(b)に関しては64温度付近で最高の最適化性能を示した。今後、温度数と問題規模などの関係についてさらに考察が必要であると思われるが、本論文における比較実験では、比較的良質な解が得られている32温度で全ての実験を行なうことにする。

また、プロセッサへの温度の割り当ては最高温度と最低温度の間を式(1)に従うように等比的に割り当てた。また、最低温度の下に、改良方向への遷移のみ受理する「極最低温度」を設けている。

最後に、解交換周期  $k$  は逐次SA法で一定温度で生成する状態遷移数に相当するものと考えられることができる。本実験では  $k$  はブロック数に比例する量であると考え、

$$k = \beta n$$

とした。ここで  $\beta$  は定数である。本論文では、後述する各データに対して  $\beta$  と最適化能力の関係についての実験を行ない、 $\beta$  の値を決定することにした。実験結果を図6に示す。

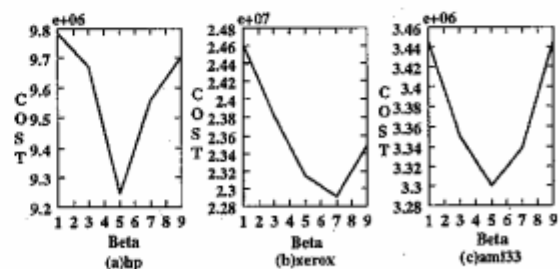


Fig.6: $\beta$  vs. cost value

図6:  $\beta$ と評価関数値に関する実験結果

図6.(a),(c)では、 $\beta = 5$ 付近で、図6.(b)に関しては  $\beta = 7$ 付近で最高の最適化能力を示していることが判る。以上の結果を踏まえて、以下、本論文に

おける実験では、すべて $\beta = 5$ で実験を行なうことにした。

## 4 実験結果と評価

### 4.1 実行環境

温度並列 SA 法を用いたプログラムを、並列オブジェクト指向言語 mosaic [13,17,19] を用いて実装した。また、比較評価に用いた逐次 SA 法のプログラムは C 言語を用いて実装を行なった。温度並列 SA 法の並行版と逐次 SA 法のプログラムは、SPARCserver-1000(SuperSPARC+ 50MHz) 上で実行した。また、温度並列 SA 法の並列版は、本研究室で提案したマルチワークステーション (SPARC station-2  $\times$  7 台, SPARC 40MHz) [17] 上で実行した。

### 4.2 実験に用いたデータ

評価を行なうために、MCNC から配布されているベンチマークデータの中から 3 種のブロックデータを選んだ。実験に用いたデータを表 1 に示す。

Table.1:MCNC benchmark data

表 1: 実験に用いたデータ

	hp	xerox	ami33
#Block	11	10	33
#Net	83	201	123

### 4.3 評価プログラム

温度並列 SA 法と逐次 SA 法を適用した LSI ブロック配置プログラムを作成し、アニーリングステップ数を等しくした場合の最適化能力の比較に加えて、計算機資源を等しくした場合 (詳細は下記参照) の比較を行なう。評価には、以下に示す 4 種のプログラムを用いた。

**温度並列 SA 法 1(以下, TPSA1)** 32 温度の温度並列 SA 法で、3000 回の解交換を行なう。1 台の計算機上で並行に実行される。

**温度並列 SA 法 2(以下, TPSA2)** 32 温度の温度並列 SA 法で、3000 回の解交換を行なう。マルチワークステーション (7CPU) 上で並列に実行される。

**逐次 SA 法 1(以下, SA1)** TPSA2 で得られた最良の解が経過したアニーリングステップ数 (およそ  $(3000 \times (5 \times \text{ブロック数}))$ ) だけ、逐次 SA 法を実行する。

**逐次 SA 法 2(以下, SA2)** TPSA2 の全ての解が経過したアニーリングステップ数の総和 (およそ  $(3000 \times (5 \times \text{ブロック数}) \times 32)$ ) だけ、逐次 SA 法を実行する。

ここで、温度並列 SA 法の実装においては、プロセッサ資源を最大限に利用することを目的として、各プロセッサ間で解交換を非同期に行なう方法をとった。そのため、実行結果は再現性を持たない。そこで、TPSA2 のデータに関しては 3 回の実行結果の平均値を採用している。ここで、3 回の実行結果のそれぞれは表 3 に示すとおりであり、有意なばらつきは認められなかった。

また、逐次 SA 法では温度スケジュールが問題となってくるが、最高、最低温度は温度並列 SA 法と同様に与え、それらの間を式 (1) に従うような温度スケジュールを採用した。ただし、ここでは式 (1) の  $n$  は、アニーリングステップ数を表している。さらに、公平を期すため、最低温度まで等比的に冷却した後、温度並列 SA 法における極最低温度で問題規模 (本プログラムではブロック数) の 10000 倍だけアニーリング処理を行なった。これは、温度並列 SA 法における 2000 回分の解交換に相当する。

SA 法の解のクオリティは、初期解の影響を大きく受けるといわれている。特に一つの初期解から状態遷移を繰り返す逐次 SA 法では影響が顕著である。そこで SA1, SA2 に対して、温度並列 SA 法の温度数と同パタン (つまり 32 パタン) の異なる初期解に対して測定を繰り返し、最大、最小、平均の評価関数値を求めた (表 2)。

### 4.4 最適化能力の比較評価

温度並列 SA 法と逐次 SA 法のアニーリングステップ数 (表中の SA 回数)、評価関数値、実行時間を表 2 に示す。ただし、ここで SA 回数は、一つの解が経過したアニーリングステップ数を表す。

**TPSA1 と SA1 の最適化能力の比較** 表 2 の hp, xerox, ami33 それぞれの TPSA1 と SA1 の評価関数値より、1CPU で同数のアニーリングステップ数の下では温度並列 SA 法 TPSA1 (並行実行) が逐次 SA 法 SA1 より明らかに優れていることがいずれの結果についても判る。またこの結果は、すでに報告されている別の実験結果 [2,6] と同じ傾向を示している。

Table.2: Comparison of TPSA and SA in optimization capability and execution Time

表 2: 温度並列 SA 法と逐次 SA 法の最適化能力の比較

	#SA	Cost Value			Minutes
		hp			
TPSA1	201006	9307161			318
TPSA2	201224	9283042			121
SA1	201224	Min.	Max.	Avg.	10
		10209584	15636722	10555674	
SA2	7000414	Min.	Max.	Avg.	389
		9822422	13623759	9949273	
xerox					
TPSA1	190930	23523006			745
TPSA2	206540	23358268			280
SA1	206540	Min.	Max.	Avg.	24
		29982674	37498112	33046949	
SA2	6617230	Min.	Max.	Avg.	918
		24130280	30808321	26692333	
ami33					
TPSA1	496290	3578236			1243
TPSA2	441650	3565491			430
SA1	441650	Min.	Max.	Avg.	34
		4141355	5553840	4779578	
SA2	14840222	Min.	Max.	Avg.	1566
		3749307	4140629	3908557	

図 7 に温度並列 SA 法で得られた最良の解が経過した温度スケジュールの一例を示す。温度スケジュールは何度かの加熱、冷却過程を繰り返した後、最低温度に落ち着くという特徴を示している。この温度スケジュールは逐次 SA 法で用いられる単調減少の温度スケジュールとは全く異なるタイプの温度スケジュールである。このような温度スケジュールが、温度並列 SA 法の優れた最適化能力の原因であると考えられる。

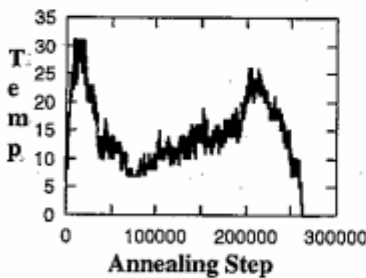


Fig.7: Cooling schedule

図 7: 温度スケジュールの例

また、表 2 の TPSA1 と SA1 の実行時間を比較すると、いずれのデータに対しても TPSA1 は SA1 の約 30 倍である。これは、アニーリングを行なうプロセスが温度数分だけ、一台の計算機で並行実行されているためである。

Tab.3: Three times measurements of TPSA2

表 3: TPSA2 の 3 回の実行による平均評価値

	Cost Value	Avg.
hp	9245287	9283042
	9283013	
	9320826	
xerox	24838010	23358268
	23707848	
ami33	21518946	3565491
	3563283	
	3544401	
	3588790	

TPSA1 と SA2 の最適化能力の比較 これは、ともに .1CPU 実行で、ほぼ同じ CPU 時間をかけたときの解品質の比較である。まず表 2 において TPSA1 の評価値と、SA2 の平均評価値を比較すると、hp, xerox, ami33 いずれの結果についても TPSA1 の方が優れていることが判る。さらに、SA2 の最低評価関数値と比較してもやはり TPSA1 の方が僅かに優れている。このことから、1 台の CPU で同じ計算時間をかける場合でも、逐次 SA 法より温度並列 SA 法の方が良質の解が得られることを示している。これは、温度並列 SA 法というアルゴリズム自身の優位性を示すことに他ならない。さらに 2.3. で述べた利点をも利用することができる。

また、TPSA1 では 32 温度分を並行実行しているため、一温度分の CPU 時間は SA2 の約 1/32 である。それにもかかわらず、より良い解に到達しているのは、図 7 のような温度スケジュールにより、一温度あたりの実行時間を大幅に (1/32 以下に) 短縮しているためと見ることもできよう。

TPSA1 と TPSA2 の比較 TPSA1 と TPSA2 の比較は、温度並列 SA 法を並行に実行した場合と、並列に実行した場合の比較になる。表 2 より、評価関数値に関しては、並行実行 (TPSA1)、並列実行 (TPSA2) の間で有意差は見られない。

一般に、1.でも述べたように、従来の並列SA法では並列実行したことによる解の劣化が問題になっている。しかしながら、温度並列SA法では温度数に等しいCPU台数まで並列処理が可能であり、並列実行しても最適化能力には影響を与えないといえる。

**初期解の与える影響** 温度並列SA法において、同一の初期解から出発した場合と、それぞれのプロセッサに異なる解を与えた場合の最適化能力に関して比較を行なう。後者について3回の測定を行なった結果を表4に示す。表2のhp,xerox,ami33におけるTPSA1,TPSA2の評価関数値と比較することにより、温度並列SA法の最適化能力は初期解の影響を受けないことが確認された。

Table.4: Three times measurements of TPSA on different initial solutions

表 4: 異なる初期解の与える影響

Data	Cost Value			
hp		9372278	9351132	9187860
	Avg.	9303756		
xerox		22283192	23634191	22307694
	Avg.	22741692		
ami33		3599801	3519164	3565523
	Avg.	3561496		

#### 4.5 並列処理によるスピードアップ

TPSAの実行時間、特に並列計算機上で実行した場合のスピードアップについて述べる。

本節での測定は、全てSun Microsystems社の共有メモリ型並列計算機SPARCserver1000 (Super-SPARC+ 50MHz) 上で行なった。実行時間と並列処理によるスピードアップを表5に示す。

Table.5: Speed-up on SPARCserver1000

表 5: 並列処理によるスピードアップ

#PE	#SAproc /PE	Execution time[min.](Speed-up)		
		hp	xerox	ami33
1	32	143	283	585
2	16	77(1.86)	151(1.87)	296(1.98)
4	8	38(3.76)	73(3.88)	154(3.80)

表5より分かるように、ほぼPE台数に比例したスピードアップが得られている。今後、さらに温度数までPE台数を増加させた場合のスピードアップに関する実験、評価を行なう必要がある。

#### 4.6 配置結果に関する評価

温度並列SA法で得られた配置結果に対して、市販の配線ツール\*を用いて実配線を行なった。配線結果を図8~図10に示す。また、現在までに発表されている他のレイアウトシステムとの比較を表6に示す。それぞれのシステムの制約条件、実行環境などが異なるため直接的な比較は難しいが、本プログラムによる配置結果が良質なものであることが判る。

Table.6: Comparison with other systems

表 6: 他システムとの比較

Data	System	area[mm <sup>2</sup> ]	wire length[mm]
hp	TPSA	12.31	236
	BB [7]	12.15	278
	GEN [5]	12.89	-
xerox	TPSA	25.99	617
	BB [7]	26.17	628
	MBP [12]	25.79	601
	GEN [5]	29.33	-
ami33	TPSA	2.20	97.7
	BB [7]	2.24	109
	MBP [12]	2.42	91

#### 5 おわりに

SA法の新しい並列化アルゴリズムである温度並列SA法は、提案後、実行時間や最適化能力の詳しい評価が行なわれていないまま応用が先行していた。そこで本論文では、温度並列SA法をLSIブロック配置問題に適用し、従来の逐次SA法と比較を行なった。比較は従来の報告でも行なわれていたアニーリングステップ数を等しくした場合の比較に加え、計算機資源を等しくした場合の比較も行なった。実験結果より、最適化能力において、温度並列SA法は逐次SA法より優れていることが判明した。この結果は1CPUで並行実行を行なった場合にも成立する。言い換えれば、1CPUで同じ計算時間をかける場合でも、逐次SA法より温度並列SA法の方が良質の解を得ることができ、さらに2.3で述べた利点を利用することができる。

また、温度並列SA法をLSIブロック配置問題に適用する手法を示し、実装したプログラムによる配置結果は、従来までに報告されていた配置結果と比べても、良質の結果が得られることが判った。

今後は、1) 大規模並列マシンへの移植、2) 一定温度内の並列処理、3) 解交換部分の非同期アルゴリ

\* (株) CADENCE デザインシステムズの Block Ensemble を使用

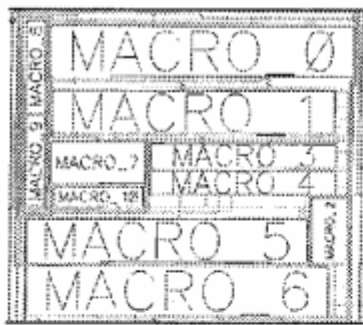


Fig.8:Routing result : hp

図 8: hp のレイアウト結果

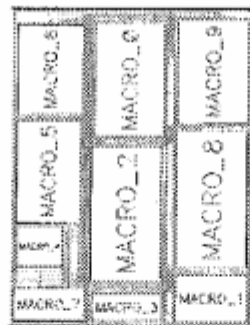


Fig.9:Routing result : xerox

図 9: xerox のレイアウト結果

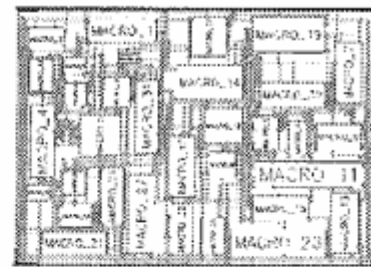


Fig.10:Routing result : ami33

図 10: ami33 のレイアウト結果

ズムの正確な評価, 3) 最適解既知の他の組合せ最適化問題 (TSP, Graph-Partitioning etc.) への適用と他手法との厳密な比較評価, を行なう予定である.

### 謝辞

本研究にあたり, 御助言とお力添えを頂いた神戸大学工学部情報知能工学科の金田悠紀夫教授に感謝する. また, 評価に関して有益な御助言を頂いた京都大学工学部電子工学科の小野寺秀俊教授, (株) 日立製作所日立研究所の伊達博博士に感謝する. また, 配置結果の評価に御協力頂いた (株) Cadence デザインシステムズに感謝する.

### 参考文献

- [1] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines*. Wiley, NY, 1989.
- [2] 伊達博, 瀧和男. 温度並列シミュレーテッド・アニーリング法に基づくスタンダードセル配置プログラム. 情報処理学会 DA シンポジウム'93, pp. 173-176, August 1993.
- [3] D.R.Greening. Parallel simulated annealing techniques. *Physica*, Vol. D-42, pp. 293-306, 1990.
- [4] E.H.L.Aarts, F.M.J.de Bont, J.H.A.Habers, and P.J.M.van Laarhoven. A parallel statistical cooling algorithm. *Proc.STACS'86, Lecture Notes in Computer Sciences*, Vol. 210, pp. 87-97, 1986.
- [5] H.Esbensen. A genetic algorithm for macro cell placement. *Proc.Euro.Des.Auto.Conf.*, pp. 52-57, September 1992.
- [6] H.Hirosawa, et al. Folding simulation using temperature parallel simulated annealing. *In Proc.Intl.Conf. on Fifth Generation Computer Systems 1992, ICOT, Tokyo, 1992.*
- [7] H.Onodera, Y.Taniguchi, and K.Tamaru. Branch-and-bound placement for building block layout. *Proc.ACM/IEEE Des.Auto.Conf.(DAC)*, pp. 433-439, 1991.
- [8] 木村宏一, 瀧和男. 時間的一様な並列アニーリングアルゴリズム. 信学技報, Vol. NC90-1, May 1990.
- [9] K.Kimura and K.Taki. Time-homogeneous parallel annealing algorithm. Tech.Rep 673, ICOT, 1991.
- [10] L.Ingber. Very fast simulated reannealing(VFSR). *Mathematical and Computer Modelling*, Vol. 12, No. 8, pp. 967-973, 1989.
- [11] M.D.Durand. Parallel simulated annealing accuracy vs. speed in placement. *IEEE Des.Test Comp.*, pp. 8-34, June 1989.
- [12] M.Upton, K.Samil, and S.Sugiyama. Integrated placement for mixed macro cell and standard cell. *Proc.ACM/IEEE Des.Auto.Conf.(DAC)*, pp. 32-35, 1990.
- [13] 小倉毅, 瀧和男. 並列オブジェクト指向言語とマルチワークステーション上の実装. JSP'94 論文集, pp. 97-104, May 1994.
- [14] P.Banerjee. *Parallel Algorithms for VLSI Computer-Aided Design*. Prentice-Hall, Inc., 1994.
- [15] S.Kirkpatrick, C.D.Gelatt Jr., and M.P.Vecchi. Optimization by simulated annealing. *SCIENCE*, Vol. 220, No. 4598, pp. 671-220, May 1983.
- [16] S.R.White. Concepts of scale in simulated annealing. *Proc.IEEE Intl.Conf.Comp.Des.(ICCD)*, pp. 646-651, 1984.
- [17] 瀧和男, 小倉毅, 小西健三. ワークステーション複合体による並列処理システム— 中・小粒度オブジェクト指向並列処理の実現 —. 情報処理学会 PRG13-7 研究報告, Vol. 93, No. 73, August 1993.
- [18] 豊永昌彦, 秋邊俊郎. 高速アニーリング・シミュレーション法:FAST. 情報処理学会 DA 研究報告, Vol. 70, pp. 33-38, 1986.
- [19] 屋鋪正史, 石原義勝, 松川力, 小西健三, 瀧和男. 並列オブジェクト指向言語 mosaic のランタイム・システム. 情報処理学会 PRG18-7 研究報告, July 1994.