温度並列シミュレーテッド・アニーリング法とその評価

小西健三[†] 瀧 和男[††] 木村宏一[†††,☆]

本論文では，新たなシミュレーテッド・アニーリング（SA）法の並行（コンカレント）アルゴリズムとして「温度並列SA法」を提案し，その評価を行う．温度並列SA法は，温度スケジュールの自動化，時間一様性（任意の時点での終了，あるいは継続による解の改善が可能），並列処理との高い親和性，という優れた性質を持つものである．本アルゴリズムは開発以来応用が先行しており，逐次SA法と比較した場合の最適化能力，実行時間の優劣については明らかでなかった．そこで本論文では，まず温度並列SAアルゴリズムについて報告し，次に逐次SA法との比較評価を実験的に行った．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，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，処理時間の短縮という観点からは，温度並列SA法は温度数まで並列処理が可能であり，また，従来の並列SA法とは異なり，並列実行しても最適化能力が劣化しないことも確認した．

Temperature Parallel Simulated Annealing Algorithm and Its Evaluation

KENZO KONISHI,[†] KAZUO TAKI [††] and KOUICHI KIMURA [†††,☆]

We propose a new concurrent simulated annealing algorithm "Temperature Parallel Simulated Annealing (TPSA)", and report its evaluation. TPSA has several advantages: automated cooling schedule, time-homogeneity, and affinity to parallel processing. Firstly, this paper represents TPSA algorithm, then compares TPSA with conventional sequential SA in optimization capability and execution time in an experimental way. As a result, TPSA is superior in optimization capability to the sequential SA under the same annealing steps, and also spending the same CPU time on 1 CPU. It is also examined that TPSA has parallelism up to the number of temperatures, keeping the same convergence quality.

1. はじめに

シミュレーテッド・アニーリング法（以下SA法）[15]は，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である．繰り返し法に属する多くのアルゴリズムが局所最適解にトラップされるという問題点を抱えているのに対し，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に到達することが保証されている[1),14)]．しかし，そのためには非現実的に長い計算時間が必要となる．一方，温度と呼ばれるSA法の振舞いを決定する制御変数のスケジュールが問題に適したものでなければ，解の品質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報告もされている[5),13)]．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．

SA法は，従来の最適化アルゴリズムと比較して多大な計算時間を要するにもかかわらず，アルゴリズムの枠組が柔軟であること，良質の解が得られやすいこともあって様々な組合せ最適化問題に適用されてきた[1)]．そこで，計算時間の短縮を目的とした並列SA法に関する研究も行われている．特に，各プロセッサで複数の状態遷移を同時に扱う方法がよく研究されている[1),3)]．しかしこの方法では，全プロセッサが最新の解の状態を把握することができないため古い状態からの処理を行うことになり，評価（目的）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[10]．この問題点がアルゴリズムの最適化能力にどのように影響を

† 神戸大学大学院自然科学研究科知能科学専攻
Division of Intelligent Science, Graduate School of Science and Technology, Kobe University
†† 神戸大学工学部情報知能工学科
Department of Computer and Systems Engineering, Faculty of Engineering, Kobe University
††† (財)新世代コンピュータ技術開発機構
Institute for New Generation Computer Technology
☆ 現在 (株)日立製作所中央研究所
Presently with Central Research Laboratory, Hitachi, Ltd.

及ぼすかについて検討されている[1]．

本論文では，従来までの並列SA法とは異なる新たなSA法の並行(コンカレント)アルゴリズムとして「温度並列SA法」[6)~8)]を提案するとともに，その最適化能力と実行時間を評価する．

温度並列SA法は，以下に示す優れた性質を持つものである．

温度スケジュールの自動化 温度スケジュールを解が自分自身で決定する．

時間一様性 任意の時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．

並列処理との高い親和性 解の品質を劣化することなしに，温度数まで並列実行が可能．

温度並列SA法は開発されて以来，最適化能力の詳しい評価が行われていないまま，LSI-CAD[2)]，遺伝子情報処理[4)]への応用が先行していた．すなわち，先に述べた利点は大いに利用されたものの，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．

そこで本論文では，はじめに温度並列SA法について述べるとともに，温度並列SA法と逐次SA法の最適化能力と実行時間を実験的に比較し，温度並列SA法の優位性を明らかにする．

以下2章では，温度並列SA法について述べるとともに，最適化能力についての理論的考察を与える．次に3章では，温度並列SA法を組合せ最適化問題に適用する際のパラメータの決定方法について，LSI配置問題を例として説明する．さらに4章では，温度並列SA法と逐次SA法との比較を，同数のアニーリングステップ数での比較に加えて，同じCPU時間を与えたときの比較をも行い，温度並列SA法の最適化能力の高さを明らかにする．最後に5章で，まとめを行う．

2. 温度並列SA法

2.1 SA法

物質を高温に熱し，徐々に低温にしていく過程(焼きなまし)で，物質がエネルギー準位の低い安定な状態になることがある．SA法はこの過程を模擬することによって，組合せ最適化問題における評価関数を最小にする系の状態を確率的に探索する近似解法の一つである．その概要を**図1**に示す[1)]．ここで，図1の，i,jは系の状態を表す変数，Xは解空間，$T(k)$はk回目の温度更新が行われた時点での温度，$L(T(k))$は系の温度を$T(k)$に更新してから，系が平衡状態に達するまでに必要な7~10行目部分のループ回数を表す．また，関数$Accept(\ )$は，受理関数と呼ばれているもので

```
   procedure SA:
 1 begin
 2   Initialize(i(0),T(0));
 3   k := 0;
 4   i := i(0);
 5   repeat
 6     for l := 1 to L(T(k)) do
 7     begin
 8       Perturb(j from i ∈ X);
 9       Accept(i,j);
10     end;
11     k := k + 1;
12     UpdateTemp(T(k));
13   until stopcriterion
14 end;
```

図1 SA法の概要
Fig. 1 Simulated Annealing Method.

様々なものが考えられるが，本論文では，メトロポリスのアルゴリズム[11)]

$$Accept(i,j)=\begin{cases}1 & E(j)-E(i)\leq 0\\ \exp\left(-\dfrac{E(j)-E(i)}{T}\right) & otherwise\end{cases} \quad (1)$$

を用いる．ここで，Eは評価関数，Tは物理系とのアナロジーから温度と呼ばれる制御変数である．

SA法の特徴は，改悪方向への解の状態遷移を，式(1)に示すように確率的に受理することによって，理論上は真の最適解に到達することが保証されていることにある[1),14)]．しかしながら，そのためには温度スケジュールが

$$T_n\leq \frac{A}{\log n}$$

(ただし，Aは関数Eの凹凸の程度を表す定数，nは温度更新回数を表す整数)に従う必要があり，非現実的に長い計算時間が必要となる．そこで通常は真の最適解への収束性を犠牲にして

$$T_{n+1}=\alpha T_n,\quad 0<\alpha<1 \quad (2)$$

の形の温度スケジュールを用いることが多い．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品質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報告もされているが[5),13)]，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．

2.2 温度並列SA法の概要

2.1節で述べた通常の逐次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，それぞれ一定温度の下で同時並列にアニーリング処理を行う．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'

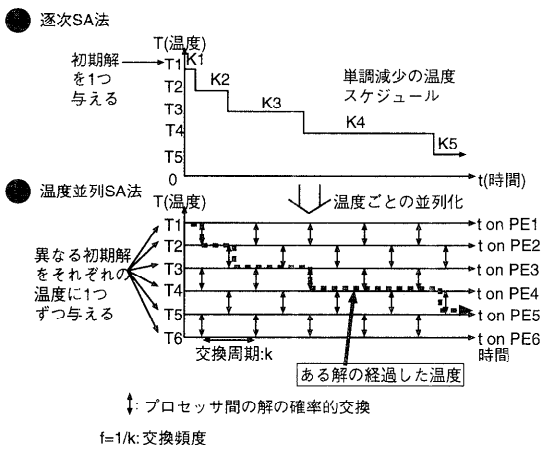


図2 温度並列 SA 法
Fig. 2 Temperature Parallel Simulated Annealing.

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2).

また，逐次 SA 法で温度スケジュールを設定することは，温度並列 SA 法ではプロセッサ間で解の交換をいつ行うか，を指定することに相当する．そこで，プロセッサ間の解の交換を確率的に行わせて温度スケジュールを自動化し，固定的な温度スケジュールを不要にしている．すなわち，確率的な解の交換によって，解自身が自分に適した温度スケジュールを選び出してくれることを期待するのである．

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うが，プロセッサをプロセスと読み替えれば，温度数分の並行プロセスを1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 プロセッサ間の解の交換確率の決め方

2.2節で述べたプロセッサ間で行う解交換の確率を以下のように決定する．以下，$p(T, E, T', E')$ は温度 T を担当するプロセッサが評価関数値 E の解を持ち，温度 T' を担当するプロセッサが評価関数値 E' の解を持つとき，それらのプロセッサ間での解の交換確率を表すものとする．

まず，次のように解の交換確率を定める．

$$(T-T')(E-E')<0 \Rightarrow p(T, E, T', E')=1. \tag{3}$$

これは，解交換時に高い方の温度を担当するプロセッサの持つ解が，低い方の温度を担当するプロセッサの解よりも良質である，すなわち評価関数値が小さい場合は，無条件に交換が行われることを示している．

次に，$(T-T')(E-E')\geq 0$ のときの $p(T, E, T', E')$ の値を定めるために以下の考察を行う．今，仮に無限時間の後に，各プロセッサ上でボルツマン分布に従う平衡状態が実現されたとする．プロセッサ間の解の確率的交換はこの平衡状態を崩してはならないので，詳細釣合の原理*から式(4)の関係が成り立つ必要がある．

$$\begin{aligned}&\frac{e^{-\frac{E}{T}}}{Z(T)}\cdot\frac{e^{-\frac{E'}{T'}}}{Z(T')}\cdot p(T, E, T', E')\\&\quad=\frac{e^{-\frac{E'}{T}}}{Z(T)}\cdot\frac{e^{-\frac{E}{T'}}}{Z(T')}\cdot 1\\&\Longleftrightarrow p(T, E, T', E')\\&\quad=\exp\left(-\frac{(T-T')\cdot(E-E')}{TT'}\right).\end{aligned} \tag{4}$$

ここで $Z(T)$ は分配関数である．以上，式(3)，(4)より，プロセッサ間の解の交換確率は次のように定めるべきであることが示された．

$$p(T, E, T', E')=\begin{cases}1 & \Delta T\cdot\Delta E<0\\ \exp\left(-\dfrac{\Delta T\cdot\Delta E}{T\cdot T'}\right) & otherwise.\end{cases} \tag{5}$$

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

2.4 解の確率的交換の効果—Kullback 情報量による検討

一般的に，離散的な確率分布 $p^+=(p_i^+)$, $p^-=(p_i^-)$ に対して，Kullback 情報量(Kullback-Leibler divergence)

$$D(p^-\|p^+)=-\sum p_i^- \log\frac{p_i^+}{p_i^-}\geq 0 \tag{6}$$

は，確率分布 p^+ と p^- とが互いにどの程度離れているかを表す測度である．この測度は非負であり，測度 $D(p^-\|p^+)$ は確率分布 p^+, p^- が等しい場合に限り0となる．

温度並列 SA 法において，ある時点における全プロセッサ上の解の集まりの分布を p とし，平衡状態におけるその分布を π とする．π は各プロセッサの温度に対する平衡分布の直積である．このとき $D(\pi\|p)$ は，その時点で系がどの程度平衡状態に近づいているかを表すことになる．

今，各プロセッサ上でそれぞれの解に対して，アニーリング処理を1ステップ進める(現在の保持している状態を1度だけ摂動する)ことを考えると，A をある推移確率行列として，解の集まりの確率分布は p から pA に変化する．また全プロセッサを幾組かのペアに分け，各ペアで一斉に解の交換を行うと，C もある

* 任意の温度において，ボルツマン分布と状態遷移確率の間で必ず成り立つ原理．

推移確率行列として，解の集まりの確率分布は p から pC に変化する．ただし，A, C の具体的な形は付録Aで与える．このとき付録Aで証明を示すように，

$$D(\pi\|p) \geq D(\pi\|pA), \quad (7)$$

$$D(\pi\|p) \geq D(\pi\|pC), \quad (8)$$

が常に成立することが判る．

従って，各プロセッサ上でアニーリング処理を1ステップずつ進めても(式(7))，プロセッサ間で解の確率的交換を行っても(式(8))，解の集まりの確率分布は平衡状態に単調に近づくと結論できる．

2.5 アルゴリズムの時間的一様性

温度並列SA法は，通常の逐次SA法における温度スケジュールのように，時間の進行とともに変化させるべきパラメータを持たない．すなわち，このアルゴリズムは時間的に一様である．

従って，本アルゴリズムの確率的な動作は時間的に一様なマルコフ連鎖として記述される．また，このマルコフ連鎖は明らかに既約かつ非周期的である．つまりこのマルコフ連鎖は，その唯一の定常状態(平衡状態)に収束することが判る．

さらにアルゴリズムの時間的一様性により，ある時点で処理を打ち切って得られた解が不満足なものであれば，処理をそのまま継続して更に最適化を図ることができる．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になる．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．

2.6 並列処理との親和性

並列処理で効率が抑制される大きな原因の一つに，プロセッサ間通信が考えられる．しかし温度並列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アルゴリズムであるといえる．

3. 温度並列SA法の組合せ最適化問題への適用

3.1 LSIブロック配置問題

本節では温度並列SA法を組合せ最適化問題に適用する際に必要となるパラメータの決定法について述べる．具体的には組合せ最適化問題の代表として，LSIブロック配置問題を対象として説明を行う．ここでLSIブロック配置問題とは，ブロックと呼ばれる部分回路を配置の対象とし，ブロックの形状，ブロック間の結線情報をもとに，チップ面積が最小となるように，チップ上のブロックの配置位置，方向を決定する問題である．

上述のようなブロック配置問題にSA法を適用するにあたり，以下のような評価関数を使用した．

$$Cost = \sum_{allnets} \text{配線長} + \text{ブロック間の隙間} + \sum \text{ブロックの重なり}. \quad (9)$$

ここで，第1項，2項は，チップ面積最小化のための項であり，第3項は最終的に得られる解を，実行可能解とするための項である．つまり，この評価関数値の値が小さくなるような解を探索することによって，良質な配置結果が得られることになる．

また，LSIブロック配置問題にSA法を適用する際の状態遷移として，1ブロックの任意の場所への移動，回転，鏡像反転，2ブロックの場所の交換を乱数によって選択することにした．

3.2 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．

最高温度と最低温度　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解交換周期内(図2の k)で1回は受理されるような温度．

ここで，温度並列SA法では最低温度よりさらに低い温度として極最低温度を設けており，極最低温度では改良方向への遷移のみを認めるとしている．そのため最低温度については，解交換周期内で1回程度，改悪方向への受理が行われれば十分であると考えた．最高温度，最低温度は，それぞれの改悪方向への受理確率 $\alpha_{max}, \alpha_{min}$ を与えることにより，式(1)より得られる以下の式，

$$T_{max} = \frac{\Delta E_{max}}{\ln \alpha_{max}}, \quad T_{min} = \frac{\Delta E_{min}}{\ln \alpha_{min}}$$

で決定した．ここに $\Delta E_{max}, \Delta E_{min}$ は，それぞれ改悪方向への最大，最小の評価関数値の変化を表す．$\Delta E_{max}, \Delta E_{min}$ は具体的には各データに対して以下のように決定した．ΔE_{max} はデータ中の最大のブロックが2個，チップ上の対角に存在して，2個のブロックの位置の交換が起こったときの評価関数値の変化とした．また，ΔE_{min} についても，データ中の最小ブロックが2個隣接して存在しているとき，それらのブロックの位置の交換が起こったときの評価値の変化とした．

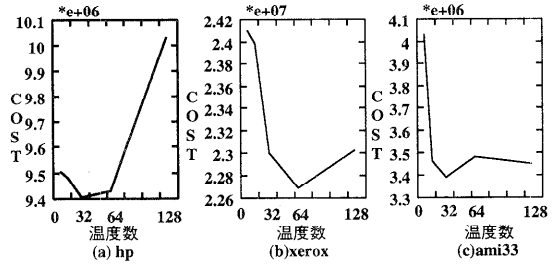


図3 温度数と評価関数値に関する実験結果
Fig. 3 The number of temperature vs. cost value.

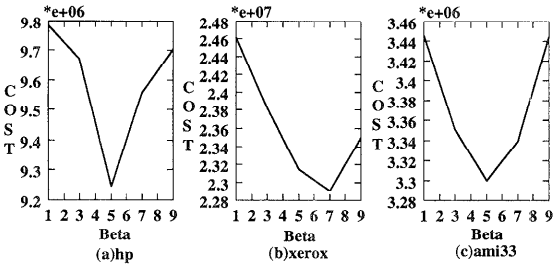


図4 β と評価関数値に関する実験結果
Fig. 4 β vs. cost value.

温度数 温度数に関して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**図3**のような結果を得た．図3(a)，(c)では，32 温度付近で最高の最適化能力を示し，図3(b)に関しては 64 温度付近で最高の最適化能力を示した。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 温度ですべての実験を行うことにする．

プロセッサへの温度の振り分け プロセッサへの温度の振り分けは最高温度と最低温度の間を，通常の SA 法で用いられる式(2)に従うように等比的に割り当てた．

解交換周期 解交換周期 k の決定法は以下の通りである．一般に，組合せ最適化問題において，現在の解から一度の状態遷移で移動可能な解を近傍と呼ぶ．逐次 SA 法の通常の温度スケジュールでは，系が平衡状態に達するまで一定温度でのアニーリングを行うが，実験的には近傍数の整数倍だけアニーリング処理を実行することが多い．この一定温度で行うアニーリングステップ数は，温度並列 SA 法の解交換周期に相当すると考えられる．そこで同様に解交換周期 k が近傍数に比例すると考え，さらに近傍数は解空間を構成する要素数(LSI ブロック配置問題ではブロック数)に比例すると考えると，解交換周期 k は式(10)のように表すことができる(ここで，n はブロック数，β はある定数)．

$$k=\beta n. \tag{10}$$

つまり解交換周期を決定するためには，式(10)中の β を決定する必要がある．そこで，後述する各データに対して β と最適化能力についての実験を行い，β を決定することにした．実験結果を**図4**に示す．

図4(a)，(c)では，$\beta=5$ 付近で，図4(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．また，正方形のモジュールを格子上に並べ，隣接するモジュール間のみ結線の存在するような人工データによるテストでも，$\beta=5$ 付近で最高の最適化能力を示しており，実データに対する良い近似を与えていると思われる．今後，さらに詳しい評価が必要とは思われるが，本論文では以上の結果を踏まえて，すべての実験において $\beta=5$ で実験を行うことにした．

3.3 解交換と温度交換

温度並列 SA 法は概念上は図2に示すようにプロセッサに温度を固定して解の確率的交換を行う．しかし，一般に組合せ最適化問題において解を表現する情報量は非常に大きなものになる．そこで，解を交換する代わりに，解をプロセッサに固定して温度交換を行うような実装を行っている[2]．温度交換を行うことで，本プログラムの例を見ても付加情報を加えて 20 byte 程度の通信量に抑えることが可能となる．

4. 温度並列 SA 法の実行と評価

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[12),17),19)]を用いて実装した．温度並列 SA 法の並行版と，比較評価に用いた逐次 SA 法のプログラムは，SPARCserver-1000(SuperSPARC+50 MHz)で実行を行った．また，温度並列 SA 法の並列版は，本研究室で開発したマルチワークステーション[17)](SPARC station-2×7 台，SPARC 40 MHz)上で実行した．

4.2 実験データ

データは，MCNC から配布されているブロックデータの中から hp, xerox, ami 33 の3種を選んだ．それぞれのデータは配置ブロック数が 11,10,33，ブロック間を結線するネット数は 83,201,123 である．

4.3 最適化性能に関する評価

温度並列 SA 法と逐次 SA 法を適用した LSI 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)の比較を行う.

4.3.1 評価プログラム

温度並列SA法の評価を行うために,4種のプログラムを実装した.2種は温度並列SA法を適用したもので,残りの2種は逐次SA法を適用したものである.それぞれのプログラムの仕様を以下に示す.

温度並列SA法1(以下,TPSA 1) 32温度の温度並列SA法で,3000回の温度交換を行う.1台の計算機上で並行に実行される.TPSA 1は,2章で述べたモデルと同様に,各プロセスで同回数のアニーリングステップ数後,解交換を行うモデルとなる.つまり,解交換部分での同期実行が行われる.

温度並列SA法2(以下,TPSA 2) 32温度の温度並列SA法で,3000回の温度交換を行う.マルチワークステーション上で並列に実行される.並列実行では,2章で述べた理論モデルとは異なり,各プロセス間の解交換は非同期に行われる.

逐次SA法1(以下,SA 1) TPSA 2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だけ,逐次SA法を実行する.

逐次SA法2(以下,SA 2) TPSA 2の全ての解が経過したアニーリングステップ数の総和だけ,逐次SA法を実行する.これはSA 1に対して,およそ(TPSAの)温度数倍だけの長いCPU時間をかけて計算したことに相当する.

測定条件 ここで,TPSA 2(並列実行版)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,各プロセス間で非同期に温度交換を行わせているため,実行結果に再現性がない.そこで,TPSA 2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.

また,SA法の解の品質は,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法ではその影響が顕著である.そこでSA 1, SA 2に対しては,温度並列SA法の温度数と同数の32種類の初期解を与えて実験を行い,それぞれのデータに対して得られた評価関数値の最小,最大,平均を測定した.一方,温度並列SA法の初期解は,全ての温度に同じものを与え,測定を行った.温度並列SA法に異なる初期解を与えた場合の最適化能力についても後述する.

さらに,逐次SA法では初期解以外に,温度スケジュールが解の品質に大きく影響するといわれている.本論文では,最高,最低温度は温度並列SA法と同様に与え,それらの間を通常のアニーリングで用いられる,式(2)に従うような温度スケジュールを採用した.また比較において公平を期すために,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SA処理を行った.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.

4.3.2 最適化能力における逐次SA法と温度並列SA法の比較

温度並列SA法と逐次SA法のアニーリング処理数,評価関数値,実行時間を**表1**に示す.

TPSA 1とSA 1の最適化能力の比較 表1のhp, xerox, ami 33それぞれのTPSA 1とSA 1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TPSA 1(並行実行)が逐次SA法SA 1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,すでに報告した別の実験結果[2),4)]と同じ傾向を示している.つまり,同程度の結果を得るという観点からは処理時間を短縮していると見ることができ,また,同程度のアニーリング処理実行後は,結果としてより良い解が得られていることを示している.

また,表1のTPSA 1とSA 1の実行時間を比較すると,いずれのデータに対してもTPSA 1はSA 1の約30倍である.これは,アニーリングを行うプロセスが温度数分だけ,1台の計算機で並行実行されているためである.

SA1とSA2の最適化能力について 表1においてSA 1とSA 2の評価関数値を比較すると,いずれのデータに対してもSA 2の方が優れていることは明らか

表1 温度並列SA法と逐次SA法の最適化能力の比較
Table 1 Comparison of TPSA and SA in optimization capability.

	SA 回数	評価関数値			実行時間(分)
Data1					
TPSA1	201006	9307161			318
TPSA2	201224	9283042			121
SA1	201224	最低	最高	平均	10
		10209584	15636722	10555674	
SA2	7000414	最低	最高	平均	389
		9822422	13623759	9949273	
Data2					
TPSA1	190930	23523006			745
TPSA2	206540	23358268			280
SA1	206540	最低	最高	平均	24
		29982674	37498112	33046949	
SA2	6617230	最低	最高	平均	918
		24130280	30808321	26692333	
Data3					
TPSA1	496290	3578236			1243
TPSA2	441650	3565491			430
SA1	441650	最低	最高	平均	34
		4141355	5553840	4779578	
SA2	14840222	最低	最高	平均	1566
		3749307	4140629	3908557	

である．この結果は，SA 法は長時間で緩やかな温度スケジュールの下で実行することで，より良質の解に到達できるという報告[1]と一致している．実際，SA 2 の実行時間は SA 1 の約 30 倍であるが，SA 1 と比較して解がどの程度改善されているかを参考までに示す．hp, xerox, ami 33 のそれぞれのデータに対して，チップ面積で 23.9%，33.5%，19.0%減，仮想配線長で 0.5%，5.9%，13.5%減となっている．これは LSI の設計品質の観点からは，非常に有意な改善であるといえる．

TPSA1 と SA2 の最適化能力の比較 まず表 1 において TPSA 1 の評価関数値と，SA 2 の平均評価関数値を比較すると，hp, xerox, ami 33 いずれの結果についても TPSA 1 の方が優れていることが判る．さらに，SA 2 で 32 種類の異なる初期解を与えた場合の最低評価関数値と比較しても，やはり TPSA 1 の方が僅かに優れている．このことは，逐次 SA 法に対して並列処理で要した計算機資源(各プロセッサの CPU 時間の総和)を与えても，TPSA 1 の最適化能力には及ばないことを示している．これは，温度並列 SA 法というアルゴリズム自身の逐次 SA 法に対する優位性を示すことに他ならない．さらに温度並列 SA 法では，2.2 節～2.6 節で述べた利点をも利用することができる．

また，TPSA 1 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 SA 2 の約 1/32 である．それにもかかわらず，温度並列 SA 法で得られた解が良い解に到達している理由は，**図 5** に示すような温度スケジュールにあると思われる．図 5 は，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である．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．また，文献 13)で示されている温度スケジュールは単調減少のものとは異なるが，特定の問題に限定した場合の温度スケジュールであり，汎用的ではない．温度並列 SA 法では，このように自動化された温度スケジュールにより，一温度当たりの実行時間を大幅に(ここでは 1/32 以下に)短縮しているためと見ることもできよう．

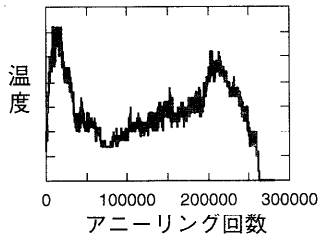


図 5 温度並列 SA 法による温度スケジュール
Fig. 5 An example of cooling schedule.

TPSA1 と TPSA2 の比較評価 TPSA 1 と TPSA 2 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である．また，この比較は，温度並列 SA 法の交換部分で同期交換を行っているモデルと，非同期交換を行っているモデルの比較に相当する．

表 1 で hp, xerox, ami 33 における TPSA 1, TPSA 2 の実行時間に注目する．TPSA 2 と TPSA 1 の実行環境を比較すると，TPSA 2 は TPSA 1 に対し，プロセッサ台数が 7 倍，CPU 性能(SPECint 92 で比較)が約 1/3 である．7 CPU の並列実行で 1 CPU に比べ 7 倍のスピードアップが得られるとすると，TPSA 1 の処理時間は約 2.3(7/3)倍となるはずである．しかし実際には，TPSA 1 の処理時間は TPSA 2 の 2.6～2.9 倍であり，7 CPU の並列実行で約 8 倍のスピードアップが得られたことになる．

これは，本プログラムで実装している非同期式の温度交換アルゴリズムによるものであり，高温域でのアニーリングステップ数を相対的に小さくして計算量を減少させているためであるが，評価は別稿に譲る．

一方，評価関数値に関しては，TPSA 2(並列実行)では動作に再現性がないため，3 回の実行の平均を表 1 に，またその元データを**表 2** に示す．表 2 から判るように評価関数値に大きなばらつきは見られない．また，表 1 の TPSA 1 の hp, xerox, ami 33 の評価関数値と比較することにより並行実行，並列実行の間で有意差は見られないことが判る．つまり，2 章で述べたような厳密な同期モデルでなくとも，実際の実行に際しては有意差は見られないことが判った．

一般に，1 章でも述べたように，従来の並列 SA 法で

表 2 TPSA 2 における 3 回の平均値
Table 2 Three times measurements of TPSA 2.

	評価関数値	平均	時間［分］
hp	9245287	9283042	423
	9283013		433
	9320826		433
xerox	24848010	23358268	294
	23707848		282
	21518946		263
ami 33	3563283	3565491	124
	3544401		118
	3588790		123

表 3 異なる初期解を与えた場合の TPSA の評価関数値
Table 3 Three times measurements of TPSA on different initial solutions.

	測定値	平均値
hp	9372278 9351132 9187860	9303756
xerox	22283192 23634191 22307694	22741692
ami 33	3599801 3519164 3565523	3561496

は並列実行したことによる解品質の劣化が問題になっている[1),10)]. しかしながら, 温度並列 SA 法では温度数に等しい CPU 台数分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 法に対して 32 個の温度にそれぞれ異なる初期解を与えた場合の実行結果(3 回の測定値と平均)を**表 3** に示す. 各温度に同一の初期解を与えた場合(表 1)と比較して, 有意差は見られなかった.

4.3.3 配置品質について

MCNC から配布されているベンチマークデータに対して, 温度並列 SA 法を用いた配置プログラムで配置を行い, 市販の配線ツールで配線を行った結果を, 従来までに報告されている他のレイアウトシステムと比較した. 比較結果は, 非常に良質なものであった[9)].

4.3.4 パラメータ決定に関する一考察

温度並列 SA 法で, 温度数, 解交換周期を決定することは, 逐次 SA 法では, 図 1 の *Update*() と $L(T(k))$ を決定することに相当する. これらのパラメータ決定に要する手間は, 逐次 SA 法と温度並列 SA 法で特に変わるものではない. 唯一温度並列 SA 法で最低温度を決める点だけが逐次 SA 法には無かった点であるが, 3.2 節で示したように簡単な計算で求められる. したがって, 温度並列 SA 法の方が優れた最適化能力を示すこと, 2.3 節～2.6 節で述べたような利点を利用できることを考慮すると, 温度並列 SA 法が実用的にも非常に優れたアルゴリズムであるといえるであろう.

5. おわりに

本論文では, 従来までの並列 SA 法とは異なる新たな SA 法の並行(コンカレント)アルゴリズムとして「温度並列 SA 法」を提案した. 温度並列 SA 法は従来の SA 法, 並列 SA 法と比較して, 以下に示す優れた性質を持つものである.

温度スケジュールの自動化 温度スケジュールを解が自分自身で決定する.

時間一様性 任意の時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.

並列処理との高い親和性 解の品質を劣化することなしに, 温度数まで並列実行が可能.

さらに温度並列 SA 法と逐次 SA 法の最適化能力, 実行時間について LSI ブロック配置問題に適用することにより, 実験的に比較を行い, 温度並列 SA 法の優位性を確認した.

まず, 最適化能力については, 従来の報告でも行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 1 CPU で並行実行を行った場合にも成立する. さらに, 上述の温度並列 SA 法の優れた利点を利用することができる.

また実行時間については, 並列処理による時間の短縮という観点から見ると, 温度並列 SA 法は温度数に等しい CPU 台数分まで並列処理が可能であり, また, 従来の並列 SA 法とは異なり, 並列実行しても最適化能力には影響を与えないことを確認した.

謝辞 本研究にあたり, 御助言とお力添えをいただ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. また本論文 2 章, 並びに付録の内容は, 木村, 瀧が(財)新世代コンピュータ技術開発機構在籍中の研究に基づく. 同財団の内田俊一研究所長をはじめとする関係諸氏に感謝する. また評価に関して有益な御助言をいただいた京都大学工学部電子工学科の小野寺秀俊助教授, (株)日立製作所日立研究所の伊達博博士に感謝する.

参考文献

1) Aarts, E. and Korst, J.: *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) Darema, F., Kirkpatrick, S. and Norton, V. A.: Parallel Algorithms for Chip Placement by Simulated Annealing, *IBM J. Res. Dev.*, Vol. 31, No. 3, pp. 391-402 (1987).
4) Hirosawa, H. et al.: Folding Simulation Using

Temperature Parallel Simulated Annealing, *Proc. Intl. Conf. on Fifth Generation Computer Systems 1992,* ICOT, Tokyo (1992).
5) Szu, H. and Hartley, R. : Fast Simulated Annealing, *Physics Letters A,* Vol. 8, No. 3, 4, pp. 122-162 (1987).
6) 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 (1990).
7) Kimura, K. and Taki, K.: Time-Homogeneous Parallel Annealing Algorithm, *Tech. Rep. 673,* ICOT (1991).
8) Kimura, K. and Taki, K.: Time-Homogeneous Parallel Annealing Algorithm, *Proc. on The 13th IMACS World Congress on Computation and Applied Mathematics* (July 1991).
9) 小西健三, 瀧 和男：温度並列シミュレーテッド・アニーリング法の評価—LSIブロック配置問題に適用して—. 情報処理学会DAシンポジウム'94, pp. 223-228 (1994).
10) Durand, M. D.: Parallel Simulated Annealing Accuracy vs. Speed in Placement, *IEEE Des. Test Comp.,* pp. 8-34 (1989).
11) Metropolis, N., Rosenbluth, A., Rosenbluth, M. and Teller, A.: Equation of State Calculations by Fast Computing Machines, *Journal of Chemical Physics,* Vol. 21, pp. 1087-1092 (1953).
12) 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, JSPP'94論文集, pp. 97-104 (1994).
13) Strenski, P. N. and Kirkpatrick, S.: Analysis of Finite Length Annealing Schedules, *Algorithmica,* Vol. 6, pp. 346-366 (1991).
14) Geman, S. and Geman, D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restration of Images, *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. 6, No. 6, pp. 721-741 (1984).
15) Kirkpatrick, S., Gelatt, C. D., Jr. and Vecchi, M. P.: Optimization by Simulated Annealing, *Science,* Vol. 220, No. 4598, pp. 671-680 (1983).
16) White, S. R.: 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 (1993).
18) 豊永昌彦, 秋濃俊郎：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) 犀鋪正史, 石原義勝, 松川 力, 小西健三, 瀧 和男：並列オブジョクト指向言語mosaicのランタイム・システム, 情報処理学会PRG18-7研究報告 (1994).

付 録

SA法を並列化して得られた上述の並列アルゴリズムをマルコフ連鎖として表現して, その推移確率行列を求め, 2.4節で与えた式(7), (8)の証明を与える.

X を探索すべき解空間とし,

$$E: X \to R \qquad i \mapsto E_i$$

を最小化すべき目的関数, すなわちエネルギーとする. 2.1節のSA法のアルゴリズムの中で述べたような, 解 x から解 x' へのランダムな遷移は, 確率行列

$$S=(s_{ij})_{i,j\in X}, \quad 0\le s_{ij}\le 1, \quad \sum_j s_{ij}=1,$$

で表現される. すなわち各 s_{ij} は条件付き確率

$$s_{ij}=Pr(x'=j|x=i)$$

を表す. ここでSA法が正しく機能するためには, Sは対称かつ既約でなければならない. このとき一定温度 T の下での逐次のSA法の挙動は, 次の推移確率行列 A をもつマルコフ連鎖として表現される.

$$A=A(\beta)=(a_{ij}(\beta)),$$

$$a_{ij}=a_{ij}(\beta)=\begin{cases} s_{ij}e^{-\beta(E_j-E_i)_+} & (j\neq i) \\ 1-\sum_{k\neq i}e^{-\beta(E_k-E_i)_+} & (j=i), \end{cases}$$

$$(E_j-E_i)_+=\max(0, E_j-E_i), \beta=1/T, \qquad (11)$$

すなわち, ある初期解 $i_0\in X$ から出発して一定温度 $T=1/\beta$ で逐次SA法を行ったとき, t ステップ後の解の確率分布を行うベクトル $p_t\in[0,1]^X$ で表すと,

$$p_0=(\delta_{ii_0})_{i\in X}, \quad p_{t+1}=p_tA$$

が成り立つ. 一方, $t\to+\infty$ の極限では, p_t は次のボルツマン分布 π に収束する.

$$\pi=\pi(\beta)=\left(\frac{1}{Z(\beta)}e^{-\beta E_i}\right)_{i\in X}, \qquad (12)$$

$$Z(\beta)=\sum_{i\in X}e^{-\beta E}.$$

X 上の確率分布 $p=(p_i)$ が平衡状態に対する分布 π からどれだけ離れているかは, 2.4節で述べた情報量によって表現される.

$$D(\pi\|p)=-\frac{1}{Z}\sum_i e^{-\beta E_i}\log p_i$$

$$-\frac{\beta}{Z}\sum_i e^{-\beta E_i}E_i-\log Z. \qquad (13)$$

これに対して次の補題が成立することから, $D(\pi\|p_t)$ は t について単調に減少することが判る. すなわち p_t は π に単調に近づいていく.

補題1 X 上の任意の確率分布 p に対して, 次式が成り立つ.

$$D(\pi\|p)\ge D(\pi\|pA). \qquad (14)$$

証明1 a_{ij} の定義より

$$e^{-\beta E_i}a_{ij}=e^{-\beta E_j}a_{ji} \tag{15}$$

が成り立つことに注意すれば，

$$\begin{aligned}&\sum_i e^{-\beta E_i}\log\Big(\sum_j p_j a_{ji}\Big)\\&\geq\sum_i e^{-\beta E_i}\sum_j a_{ij}\{\log p_j-\beta(E_j-E_i)\}\\&=\sum_{i,j}e^{-\beta E_j}a_{ji}\log p_j\\&\quad-\beta\sum_{i,j}(e^{-\beta E_j}E_j a_{ji}-e^{-\beta E_i}E_i a_{ij})\\&=\sum_j e^{-\beta E_j}\log p_j.\end{aligned}$$

よって(13)により，上の不等式(14)が成り立つ．

次に，SA法を並列化したアルゴリズムについて考える．プロセッサの台数をNとして，各プロセッサに一定温度$T_1, T_2, \cdots, T_N$を与える．ここで$T_1>T_2>\cdots T_N$とし，$\beta_n=1/T_n$とする．各プロセッサは各々1つの解を保持し，全体ではN個の解が同時並列的に探索される．従って，この並列アルゴリズムの挙動は，X^N上のマルコフ連鎖として表現できる．各プロセッサ上で，それぞれ，アニーリングを1ステップずつ進めることは，次の推移確率行列$\tilde{A}$で表現される．以下，$\otimes$はテンソル積を表す．

$$\tilde{A}=A(\beta_1)\otimes A(\beta_2)\otimes\cdots\otimes A(\beta_N)$$

また，このアルゴリズムで無限の時間をかけたときの，平衡状態の確率分布は，次の$\tilde{\pi}$で与えられる．

$$\tilde{\pi}=\tilde{\pi}(\beta_1)\otimes\tilde{\pi}(\beta_2)\otimes\cdots\otimes\tilde{\pi}(\beta_N).$$

このとき(7)に対応する次の補題が成り立つ．

補題2 X^N上の任意の確率分布$\tilde{p}$に対して，次式が成立する．

$$D(\tilde{\pi}\|\tilde{p})\geq D(\tilde{\pi}\|\tilde{p}\tilde{A}). \tag{16}$$

証明2 I_XをX上の恒等変換を表す単位行列として，

$$\tilde{A}_n=I_X{}^{\otimes(n-1)}\otimes A(\beta_n)\otimes I_X{}^{\otimes(N-n-1)}$$

とおくと，補題1の証明と同様にして，

$$D(\tilde{\pi}\|\tilde{p})\geq D(\tilde{\pi}\|\tilde{p}\tilde{A}_n)$$

が成立することが判る．一方，

$$\tilde{A}=\tilde{A}_1\cdot\tilde{A}_2\cdots\tilde{A}_N,$$

であるから帰納法により(16)が成り立つことが判る．

次に，プロセッサ間での解の確率的交換について考える．まず$N=2$で，$T_1=T$, $T_2=T'$の場合について考える．2.3節で述べたことから，これらのプロセッサ間での解の確率的交換は，次の推移確率行列で表される．

$$C=C(\beta,\beta')=(c_{(ij)(kl)}(\beta,\beta')), \tag{17}$$

$$c_{(ij)(kl)}=c_{(ij)(kl)}(\beta,\beta')=\begin{cases}c_{ij} & (i,j)=(l,k)\\1-c_{ij} & (i,j)=(k,l)\neq(j,i)\\0 & otherwise,\end{cases}$$

$$c_{ij}=c_{ij}(\beta,\beta')=e^{\min\{0,(\beta-\beta')(E_i-E_j)\}},$$

$$\beta=1/T,\quad \beta'=1/T'.$$

これに対して次の補題が成立する．

補題3 $N=2$のとき，$X^N=X^2$上の任意の確率分布$\tilde{p}$に対して，次式が成立する．

$$D(\tilde{\pi}\|\tilde{p})\geq D(\tilde{\pi}\|\tilde{p}C) \tag{18}$$

証明3 $\tilde{p}=(p_{ij})$とすると，(6)より

$$\begin{aligned}&D(\tilde{\pi}\|\tilde{p})\\&=-\frac{1}{Z(\beta)Z(\beta')}\sum_{i,j}e^{-(\beta E_i+\beta' E_j)}\log p_{ij}\\&\quad-\frac{\beta}{Z(\beta)}\sum_i e^{\beta E_i}E_i\\&\quad-\frac{\beta'}{Z(\beta')}\sum_j e^{-\beta' E_j}E_j\\&\quad-\log Z(\beta)\quad\log Z(\beta').\end{aligned} \tag{19}$$

よって(18)を示すためには，(17)，(19)より，

$$\begin{aligned}&\sum_{i,j}e^{-(\beta E_i+\beta' E_j)}\log\{(1-c_{ij})p_{ij}+c_{ji}p_{ji}\}\\&\geq\sum_{i,j}e^{-(\beta E_i+\beta' E_j)}\log p_{ij}\end{aligned}$$

を示せば良い．c_{ij}の定義より，

$$e^{-(\beta E_i+\beta' E_j)}c_{ij}=e^{-(\beta E_j+\beta' E_i)}c_{ji} \tag{20}$$

が成り立つことに注意すれば，

$$\begin{aligned}&\sum_{i,j}e^{-(\beta E_i+\beta' E_j)}\log\{(1-c_{ij})p_{ij}+c_{ji}p_{ji}\}\\&\geq\sum_{i,j}e^{-(\beta E_i+\beta' E_j)}\{-\beta E_i-\beta E_j\\&\quad+(1-c_{ij})\log(e^{\beta E_i+\beta' E_j}p_{ij})\\&\quad+c_{ij}\log(e^{\beta E_j+\beta' E_i}p_{ji})\}\\&=\sum_{i,j}e^{-(\beta E_i+\beta' E_j)}(1-c_{ij})\log p_{ij}\\&\quad+\sum_{i,j}e^{-(\beta E_j+\beta' E_i)}c_{ji}\log p_{ji}\\&\quad-\sum_{i,j}e^{-(\beta E_i+\beta' E_j)}c_{ij}(\beta-\beta')(E_i-E_j)\\&=\sum_{i,j}e^{-(\beta E_i+\beta' E_j)}c_{ij}\log p_{ij}.\end{aligned}$$

次に，一般の$N>2$の場合について考える．このとき互いに隣合う温度のプロセッサ間で解の交換を一斉に行う方法は2通りあり，それぞれ次の推移確率行列で表現できる．

$$\tilde{C}_{even}=\begin{cases}C_1\otimes C_3\otimes\cdots\otimes C_{N-1} & (N:even)\\C_1\otimes C_3\otimes\cdots\otimes C_{N-2}\otimes I_X & (N:odd),\end{cases}$$

$$\tilde{C}_{odd}=\begin{cases}I_X\otimes C_2\otimes C_4\otimes\cdots\otimes C_{N-2}\otimes I_X & (N:even)\\I_X\otimes C_2\otimes C_4\otimes\cdots\otimes C_{N-1} & (N:odd),\end{cases}$$

$$C_n=C(\beta_n,\beta_{n+1}).$$

このとき，(8)に対応する次の補題が成立する．

補題4 X^N上の任意の確率分布$\tilde{p}$に対して，次式が成立する．

$$D(\tilde{\pi}\|\tilde{p})\geq D(\tilde{\pi}\|\tilde{p}\tilde{C}_{even}), \tag{21}$$

$$D(\tilde{\pi}\|\tilde{p})\geq D(\tilde{\pi}\|\tilde{p}\tilde{C}_{odd}). \tag{22}$$

証明 4

$\tilde{C}_n = I_X{}^{\otimes(n-1)} \otimes C_n \otimes I_X{}^{\otimes(N-n-2)}$

とおくと，補題 3 の証明と同様にして，

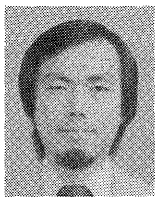
$D(\tilde{\pi} \| \tilde{p}) \geq D(\tilde{\pi} \| \tilde{p}\tilde{C}_n)$

が成立することが判る．一方 $\tilde{D}_{even}$，$\tilde{C}_{odd}$ はそれぞれ $\tilde{C}_n$ の積に分解できるから，帰納法により上式が成り立つことが判る．

(平成 6 年 8 月 3 日受付)
(平成 7 年 1 月 12 日採録)

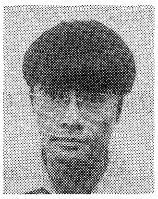
小西 健三（学生会員）

昭和 44 年生．平成 4 年神戸大学工学部システム工学科卒業．平成 6 年同大学院修士課程修了．現在，同博士後期課程在学中．並列処理，LSI-CAD 等に興味を持つ．電子情報通信学会会員．



瀧 和男（正会員）

昭和 27 年生．昭和 51 年神戸大学工学部電子工学科卒業．昭和 54 年同大学院修士課程システム工学修了．工学博士．同年(株)日立製作所入社．昭和 57 年(財)新世代コンピュータ技術開発機構に出向．遂次型および並列型推論マシンと並列応用プログラムの研究開発に従事．平成 2 年同機構第 1 研究室室長．平成 4 年 9 月より神戸大学工学部情報知能工学科助教授．並列マシンのアーキテクチャ，並列プログラミング，LSI-CAD 等に興味を持つ．電子情報通信学会，IEEE，ソフトウェア科学会，ACM 各会員．



木村 宏一（正会員）

昭和 32 年生．昭和 55 年東京大学理学部数学科卒業．昭和 58 年同大学院修士課程修了．同年(株)日立製作所中央研究所入社．昭和 63～平成 4 年に(財)新世代コンピュータ技術開発機構（ICOT）に出向．LSI レイアウト CAD，並列アルゴリズム，機械学習等の研究に従事．電子情報通信学会，IEEE 各会員．