

TR-0887

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 (神戸大学)、瀧 和男 (神戸大学)、
木村 宏一 (日立)

August, 1994

© Copyright 1994-8-23 ICOT, JAPAN ALL RIGHTS RESERVED

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191~5

Institute for New Generation Computer Technology

論文

温度並列シミュレーテッド・
アニーリング法とその評価

Temperature Parallel Simulated Annealing
Algorithm and Its Evaluation

学生会員 (9204371) *小西 健三[†],
正員 (7902947) 瀧 和男^{††},
正員 (8819381) 木村 宏一^{†††},
Kenzo KONISHI [†], Kazuo TAKI ^{††},
and Kouichi KIMURA ^{†††}

[†]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

^{†††} (株) 日立製作所 中央研究所
Central Research Laboratory, Hitachi, Ltd.

連絡先：小西健三
〒6 5 7 神戸市灘区六甲台町 1
神戸大学工学部 情報知能工学科 瀧 研究室
Tel.078(803)1185
konishi@seg.kobe-u.ac.jp

論文作成手段： $\text{\LaTeX}$

和文アブストラクト

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、その評価を行なう。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。本アルゴリズムは研究会報告以来、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なった。最適化能力の比較では、同じアニーリングステップ数での比較に加えて、逐次 SA 法に温度並列 SA 法で要した各プロセッサの CPU 時間の総和を与えても、逐次 SA 法は温度並列 SA 法に及ばないことを確認した。この結果は温度並列 SA 法を 1CPU で並行実行した場合にも成立し、1CPU で同じ計算時間をかけた場合でも、逐次 SA 法より温度並列 SA 法の方が優れた最適化能力を示すことを確認した。また、並列処理による時間の短縮という観点からは、温度並列 SA 法は温度数までの並列実行分（CPU 台数分）のスピードアップが得られることを確認した。さらに従来の並列 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も示した。

abstract

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 a 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 1CPU. It is also examined that TPSA can get the linear speed up using up to the number of processors equal 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）¹³⁾ 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^{1,12)}。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¹⁴⁾、温度スケジュールに関する報告もされている^{5,11)}。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。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なわれている。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最も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なうことになり、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⁹⁾、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」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。なお本アルゴリズムについては、研究会等で先に報告している⁶⁻⁸⁾。

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 法は最適化能力の詳しい評価が行なわれていないまま、LSI-CAD²⁾、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 法についてまとめるとともに、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節では、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節では、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説明する。さらに4節では、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ない、温度並列 SA 法の最適化能力の高さを実証する。最後に5節で、まとめを行なう。

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（目的）関数 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。その概要を図1に示す。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、現時点の解 x の評価関数値 $E(x)$ に対して、新しく生成した解 x' の評価関数値 $E(x')$ を計算し、評価関数値が良くなれば新しい解を無条件に受理し、そうでなければ確率 $\exp(-\Delta E/T)$ に従って受理する。このように改悪方向への解の状態遷移を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

1,12). しかし, 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) 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

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

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¹⁴⁾, 温度
スケジュールに関する報告もされている^{5,11)}. 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.

2.2 温度並列 SA 法の概要

2.1. 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
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

つ行なうか、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な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なうが、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 プロセッサ間の解の交換確率の決め方

2.2. で述べたプロセッサ間で行な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。

$$(T - T')(E - E') < 0 \implies p(T, E, T', E') = 1 \quad (2)$$

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、無条件に交換が行な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、以下の考察を行なう。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、詳細釣り合いの原理*から、式 (3) の関係が成り立つ必要がある。

$$\begin{aligned} \frac{e^{-\frac{E}{T}}}{Z(T)} \cdot \frac{e^{-\frac{E'}{T'}}}{Z(T')} \cdot p(T, E, T', E') &= \frac{e^{-\frac{E'}{T}}}{Z(T)} \cdot \frac{e^{-\frac{E}{T'}}}{Z(T')} \cdot 1 \\ \iff p(T, E, T', E') &= \exp\left(-\frac{(T - T') \cdot (E - E')}{TT'}\right). \end{aligned} \quad (3)$$

ここで $Z(T)$ は分配関数である。以上、式 (2),(3) 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。

$$P(T, E, T', E') = \begin{cases} 1 & \Delta T \cdot \Delta E < 0 \\ \exp\left(-\frac{\Delta T \cdot \Delta E}{T \cdot T'}\right) & otherwise \end{cases} \quad (4)$$

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 $p = (p_i)$, $q = (q_i)$ に対して、Kullback 情報量 (Kullback-Leibler divergence)

$$D(q \parallel p) = -\sum q_i \log \frac{p_i}{q_i} \geq 0 \quad (5)$$

*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。

は、確率分布 p と q 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 $D(q \parallel p)$ は確率分布 p, q 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 $D(\pi \parallel p)$ 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、各プロセッサ上でアニーリング処理を 1 ステップ進めることを考えると、 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なうと、 C も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 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、

$$D(\pi \parallel p) \geq D(\pi \parallel pA), \quad (6)$$

$$D(\pi \parallel p) \geq D(\pi \parallel pC), \quad (7)$$

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 (6))、プロセッサ間で解の確率的交換を行なっても (式 (7))、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 パラメータの決定

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なパラメータの決定法について述べる。例として、温度並列 SA 法を代表的な組合せ最適化問題である LSI ブロック配置問題に適用し、具体例に沿って説明を行なう。ここで LSI ブロック配置問題とは、複数の設計済みブロックをチップ上に総配線長、チップ面積が最小となるように配置する問題である。また評価関数としては、「ブロックの重なりペナルティ」、「チップ面積を最小化するためのペナルティ」、「仮想配線長和」の重み付き和を用いた。

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。

最低温度 最小の改悪となる状態遷移が一定温度の A

ニーリング処理の間（図 2 の k ）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 温度で温度交換 3000 回分だけ実行し、表 1 のような結果を得た。hp, xerox は小規模なデータで、ami33, ami49 の順にデータ規模は大きくなる。小規模データでは、16 温度以上では最適化能力に大きな変化は見られないが、ami33, ami49 では 32 温度で最高の最適化能力を示している。また、ami49 の結果からは温度数を増すことで、さらなる最適化が予想される。以下、本論文の実験では、全て 32 温度で実験を行なうことにした。

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、通常の SA 法で用いられる式 (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 は問題規模 n （本プログラムではブロック数）に依存すると考え、

$$k = \beta n$$

とした。ここで β は定数である。本論文では、ブロック配置問題用の人工データを用意し、 β と最適化能力の関係について実験を行ない、図 3 の結果を得た。この結果から $\beta = 5$ 付近で最高の最適化能力が得られていることが判る。本論文の以下の実験では全て $\beta = 5$ とした。

3.2 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なう。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なうような実装を行なっている²⁾。温度交換を行なうことで、本プログラムの例を見ても付加情報を加えて 20byte 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^{10,15,16)}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+50MHz)で実行を行なった。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¹⁵⁾(SPARC station-2 × 7 台, SPARC 40MHz)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,ami33 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）の比較を行なう。

4.3.1 評価プログラム

温度並列 SA 法の評価を行なうために、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、TPSA1） 32 温度の温度並列 SA 法で、3000 回の温度交換を行なう。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 回の温度交換を行なう。マルチワークステーション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だけ, 逐次 SA 法を実行する. これは SA1 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, TPSA2(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なわせているため, 実行結果に再現性がない. 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.

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 SA1, SA2 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実験を行ない, 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, 測定を行なった. 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 (1) 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。

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 2 に示す。

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、すでに報告した別の実験結果^{2,4)}と同じ傾向を示している。

図 4 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。

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。

SA1 と SA2 の最適化能力について 表 2 において SA1 と SA2 の評価関数値を比較すると、いずれのデータに対しても SA2 の方が優れていることは明らかである。一般的に 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、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、SA2 の実行時間は SA1 の約 30 倍であるが、SA1 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,ami33 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 2 において TPSA1 の評価関数値と、SA2 の平均評価関数値を比較すると、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 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 TPSA1 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、TPSA1 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。

言い替えると、1 台の CPU で同じ計算時間にかかる場合でも、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 2.2. で述べた利点をも利用することができるということを示している。

TPSA1 と TPSA2 の比較評価 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である。

表 2 で hp, xerox, ami33 における TPSA1, TPSA2 の実行時間に注目する。TPSA2 と TPSA1 の実行環境を比較すると、TPSA2 は TPSA1 に対し、プロセッサ台数が 7 倍、CPU 性能 (SPECint92 で比較) が約 1/3 である。7CPU の並列実行で 1CPU に比べ 7 倍のスピードアップが得られるとすると、TPSA1 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、TPSA1 の処理時間は TPSA2 の 2.6~2.9 倍であり、7CPU 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、TPSA2（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 2 に、またその元データを表 3 に示す。表 3 からわか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 2 の TPSA1 の hp,xerox,ami33 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。

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品質の劣化が問題になっている^{1,9)}。しかし、以上のことから、温度並列 SA 法においては、温度数までの並列実行分 (CPU 台数分) のスピードアップを必ず得ることができ、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。

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 4 に示す。各温度に同一の初期解を与えた場合（表 2）と比較して、有意差は見られなかった。

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，実行時間について実験的に比較を行ない，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，逐次 SA 法に対して温度並列 SA 法と同じ計算機資源（温度並列 SA 法で要したプロセッサの CPU 時間の総和）を与えても，温度並列 SA 法の最適化能力に及ばないことを確認した。この結果は温度並列 SA 法を 1CPU で並行実行した場合にも成り立つことを確認した。つまり，1CPU で同じ計算時間をかけた場合でも，逐次 SA 法より温度並列 SA 法の方が最適化能力において優れており，さらに上述の優れた性質をも利用することができるということを示している。

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 法は温度数までの並列実行分（CPU 台数分）のスピードアップを必ず得ることができ，さらに従来の並列 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も確認した。

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節、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] 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] F.Darema, S.Kirkpatrick, and V.A.Norton. Parallel algorithms for chip placement by simulated annealing. *IBM J.Res.Dev.*, Vol. 31, No. 3, pp. 391-402, May 1987.
- [4] H.Hirosawa, et al. Folding simulation using temperature parallel simulated annealing. *In Proc.Intl.Conf. on Fifth Generation Computer Systems 1992, ICOT, Tokyo, 1992.*
- [5] H.Szu and R.Hartley. Fast simulated annealing. *Physics Letters A*, Vol. 8, No. 3,4, pp. 122-162, June 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, , May 1990.
- [7] K.Kimura and K.Taki. Time-homogeneous parallel annealing algorithm. Tech.Rep 673, ICOT, 1991.

- [8] K.Kimura and K.Taki. Time-homogeneous parallel annealing algorithm. *Proc. on The 13th IMACS World Congress on Computation and Applied Mathematics*, July 1991.
- [9] M.D.Durand. Parallel simulated annealing accuracy vs. speed in placement. *IEEE Des.Test Comp.*, pp. 8-34, June 1989.
- [10] 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.
- [11] P.N.Strenski and Scott Kirkpatrick. Analysis of finite length annealing schedules. *Algorithmica*, Vol. 6, pp. 346-366, 1991.
- [12] S.Geman and D.Geman. Stochastic relaxation, Gibbs distributions, and the bayesian restration of imeages. *IEEE Trans. Pattern Analysis and Machine Inteligence*, Vol. 6, No. 6, pp. 721-741, November 1984.
- [13] S.Kirkpatrick, C.D.Gelatt Jr., and M.P.Vecchi. Optimization by simulated annealing. *SCIENCE*, Vol. 220, No. 4598, pp. 671-220, May 1983.
- [14] S.R.White. Concepts of scale in simulated annealing. *Proc.IEEE Intl.Conf.Comp.Des.(ICCD)*, pp. 646-651, 1984.

- [15] 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.
- [16] 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.

A 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. で与えた式 (6),(7) の証明を与える。

X を探索すべき解空間とし、

$$E : X \rightarrow R \quad i \mapsto E_i$$

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. 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

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

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。

$$\begin{aligned} A = A(\beta) &= (a_{ij}(\beta)), \\ a_{ij} = a_{ij}(\beta) &= \begin{cases} s_{ij} e^{-\beta(E_j - E_i)_+} & (j \neq i) \\ 1 - \sum_{k \neq i} s_{ik} e^{-\beta(E_k - E_i)_+} & (j = i), \end{cases} \\ (E_j - E_i)_+ &= \max(0, E_j - E_i), \quad \beta = 1/T, \end{aligned} \quad (8)$$

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 法を行なったとき、 t ステップ後の解の確率分布を行ベクトル $p_t \in [0, 1]^X$ で表すと、

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

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

$$\begin{aligned}\pi &= \pi(\beta) = \left(\frac{1}{Z(\beta)} e^{-\beta E_i} \right)_{i \in X}, \\ Z(\beta) &= \sum_{i \in X} e^{-\beta E_i}.\end{aligned}\quad (9)$$

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. で述べた Kullback 情報量によって表現される.

$$\begin{aligned}D(\pi \parallel p) &= -\frac{1}{Z} \sum_i e^{-\beta E_i} \log p_i \\ &\quad - \frac{\beta}{Z} \sum_i e^{-\beta E_i} E_i - \log Z.\end{aligned}\quad (10)$$

これに対して次の補題が成立することから, $D(\pi \parallel p_t)$ 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.

$$D(\pi \parallel p) \geq D(\pi \parallel pA). \quad (11)$$

証明 1 a_{ij} の定義より

$$e^{-\beta E_i} a_{ij} = e^{-\beta E_j} a_{ji} \quad (12)$$

が成り立つことに注意すれば,

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

よって (10) により, 上の不等式 (11) 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 $T_1, T_2, \dots, T_N$ を与える。ここで $T_1 > T_2 > \dots > T_N$ 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。

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

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。

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

このとき (6) 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。

$$D(\tilde{\pi} \parallel \tilde{p}) \geq D(\tilde{\pi} \parallel \tilde{p}\tilde{A}). \quad (13)$$

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、

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

が成立することが判る。一方、

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

であるから帰納法により (13) 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. 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。

$$C = C(\beta, \beta') = (c_{(ij)(kl)}(\beta, \beta')), \quad (14)$$

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

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

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

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。

$$D(\tilde{\pi} \parallel \tilde{p}) \geq D(\tilde{\pi} \parallel \tilde{p}C) \quad (15)$$

証明 3 $\tilde{p} = (p_{ij})$ とすると、(5) より

$$\begin{aligned} & D(\tilde{\pi} \parallel \tilde{p}) \\ &= -\frac{1}{Z(\beta)Z(\beta')} \sum_{i,j} e^{-(\beta E_i + \beta' E_j)} \log p_{ij} \\ &\quad -\frac{\beta}{Z(\beta)} \sum_i e^{\beta E_i} E_i \\ &\quad -\frac{\beta'}{Z(\beta')} \sum_j e^{-\beta' E_j} E_j \\ &\quad -\log Z(\beta) - \log Z(\beta'). \end{aligned} \quad (16)$$

よって (15) を示すためには、(14)、(16) 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、

$$e^{-(\beta E_i + \beta' E_j)} c_{ij} = e^{-(\beta E_j + \beta' E_i)} c_{ji} \quad (17)$$

が成り立つことに注意すれば、

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

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なう方法は2通りあり、それぞれ次の推移確率行列で表現できる。

$$\begin{aligned} \tilde{C}_{\text{even}} &= \begin{cases} C_1 \otimes C_3 \otimes \cdots \otimes C_{N-1} & (N : \text{even}) \\ C_1 \otimes C_3 \otimes \cdots \otimes C_{N-2} \otimes I_X & (N : \text{odd}), \end{cases} \\ \tilde{C}_{\text{odd}} &= \begin{cases} I_X \otimes C_2 \otimes C_4 \otimes \cdots \otimes C_{N-2} \otimes I_X & (N : \text{even}) \\ I_X \otimes C_2 \otimes C_4 \otimes \cdots \otimes C_{N-1} & (N : \text{odd}), \end{cases} \\ & C_n = C(\beta_n, \beta_{n+1}). \end{aligned}$$

このとき、(7) 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。

$$D(\tilde{\pi} \parallel \tilde{p}) \geq D(\tilde{\pi} \parallel \tilde{p} \tilde{C}_{\text{even}}), \quad (18)$$

$$D(\tilde{\pi} \parallel \tilde{p}) \geq D(\tilde{\pi} \parallel \tilde{p} \tilde{C}_{\text{odd}}). \quad (19)$$

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 A.3. の証明と同様にして，

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

が成立することが判る．一方 $\tilde{C}_{even}, \tilde{C}_{odd}$ 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．

```

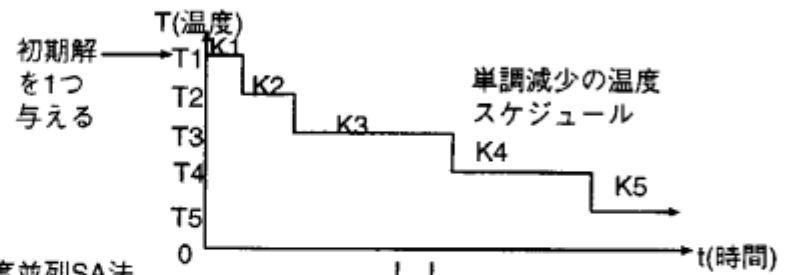
begin
  初期解  $x$ , 初期温度  $T$  の設定
  repeat
    begin
      新しい解  $x'$  を生成
       $\Delta E = E(x') - E(x)$ 
      if( $\Delta E < 0$ ) then  $x=x'$ 
      else  $\exp(-\Delta E/T)$  に従って  $x=x'$ 
    end
    温度  $T$  の更新
  until 終了条件
end

```

Fig.1: Simulated Annealing Method

図 1: SA 法の概要

● 逐次SA法



● 温度並列SA法



Fig.2: Temperature Parallel Simulated Annealing

図 2: 温度並列 SA 法

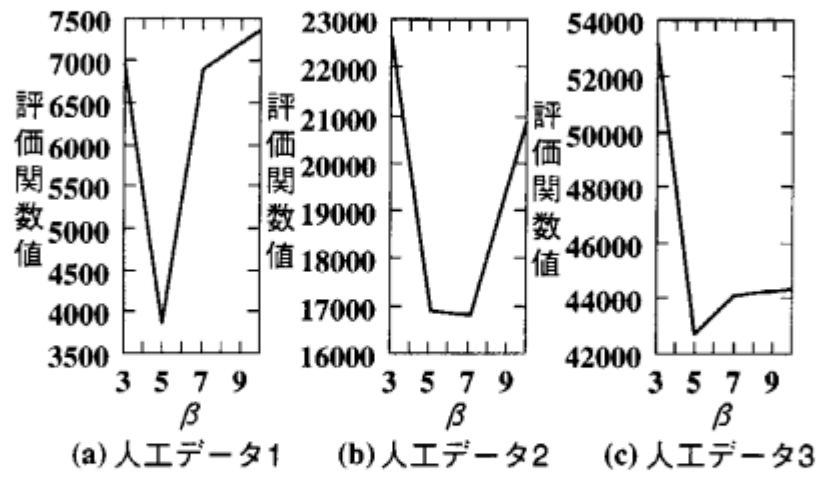


Fig 3: Cost values vs. inner loop

図 3: 内側のループと最適化性能

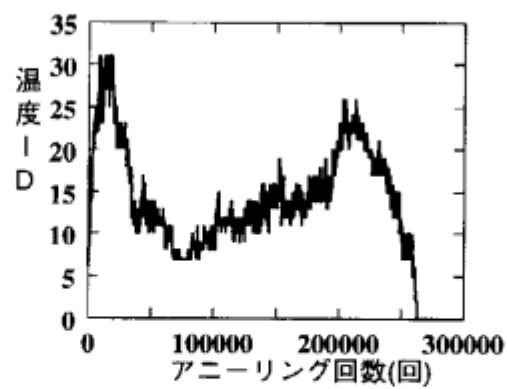


Fig 4: An Example of Cooling Schedule
 図 4: 温度並列 SA 法による温度スケジュール

Table.1:Cost values vs. temperature numbers

表 1: 温度数と最適化能力

	8 温度	16 温度	32 温度
hp	10839869	9344239	9283042
xerox	31103958	23066144	23358268
ami33	4492974	3770362	3565491
ami49	71471861	61920002	55335692

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

表 2: 温度並列 SA 法と逐次 SA 法の最適化能力の比較

	SA 回数	評価関数値			実行時間 (分)
Data1					
TPSA1	201224	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	

Table.3: Three times measurements of TPSA2

表 3: TPSA2 における 3 回の平均値

	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
ami33	3563283	3565491	124
	3544401		118
	3588790		123

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

表 4: 異なる初期解を与えた場合の TPSA の評価関数値

	測定値	平均値
hp	9372278	9303756
	9351132	
	9187860	
xerox	22283192	22741692
	23634191	
	22307694	
ami33	3599801	3561496
	3519164	
	3565523	