

Application of Temperature Parallel Simulated Annealing to Continuous Optimization Problems

Mitsunori MIKI^{*}, Tomoyuki HIROYASU^{*}, Motonori IKEUCHI^{**} and Masayuki KASAI^{***}

(Received January 11, 2000)

The simulated annealing (SA) based on probabilistic search algorithms is a method of solving optimization problems. There are several types of SA and among SAs, the temperature parallel simulated annealing (TPSA) is very suitable for parallel processing and automatically decides a cooling schedule, which is very difficult to determine in conventional SA. However, the TPSA has been applied only for some combinatorial optimization problems, and therefore its application to continuous optimization problems has been a subject to be investigated. Newly developed methods are introduced for determining the neighborhood using normal distribution, considering the maximum and the minimum temperatures based on the design space and accuracy of the solution, respectively and tuning the scale factor according to the evaluation using the Metropolis criterion. The continuous optimization problem is the minimization of one of the standard test functions. The number of temperature stages is 32, and the computer used is a PC cluster with 8 processors. As a result, the TPSA with the proposed scheme is found to be also superior to the sequential SA for continuous optimization problems.

Key words : Simulated annealing, Parallel processing, Temperature parallel, Continuous optimization problems, Normal distribution

キーワード : シミュレーテッドアニーリング, 並列処理, 温度並列, 連続最適化問題, 正規分布

連続設計変数空間における 温度並列シミュレーテッドアニーリングの応用

三木光範・廣安知之・池内智悟・笠井誠之

1. はじめに

ある制約条件下において, 与えられた状態空間で定義された関数 (目的関数) の最大値 (または, 最小値) を与える状態空間の要素を求める問題が, 最適化問題である. 最適化問題において, 設計変数が連続値であ

る場合の問題を連続最適化問題, 意志決定変数が離散値である場合, すなわち, 解が整数などの集合, あるいは, 数列で与えられる場合の問題を組合せ最適化問題と呼んでいる.

^{*} Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6434, Fax:+81-774-65-6796, E-mail:mmiki@mail.doshisha.ac.jp, tomo@is.doshisha.ac.jp

^{**} Graduate Student, Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6716, Fax:+81-774-65-6716, E-mail:tt0706@mail4.doshisha.ac.jp

^{***} Graduate Student, Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6716, Fax:+81-774-65-6716, E-mail:mksai@mikilab.doshisha.ac.jp

連続設計変数最適化問題に対してはこれまで非線形数値計画法による解法が主流であった。設計空間が単峰であり、目的関数の勾配が連続である場合にはこのようなアプローチは極めて有効であるが、そうでない場合には発見的手法（ヒューリスティック法）と呼ばれる方法が多く用いられるようになってきた。シミュレーテッドアニーリング (Simulated Annealing; 以下 SA)¹⁾ もその一つの手法として報告されている。一般に、設計空間が全域的には単峰に近いが、サイズの小さい局所解領域が無数に存在する場合に SA は有効である。SA の基礎となる考えは、Metropolis らが 1953 年に発表した焼なましと呼ばれる過熱炉内の固体の冷却過程をシミュレートするアルゴリズム²⁾ に端を発し、この SA は最適化問題、特に組合せ最適化問題を解く汎用近似解法の一つとして用いられている。

SA は、局所探索をランダムに行いながら、更に解に改良が見られない場合でも、新しい解に移る可能性を残すことで局所解に陥ることを防ぐことができる点に特徴がある。この特徴により、現在では組合せ最適化問題にだけでなく、複数の局所解を持つ連続変数最適化問題のための多くの SA アルゴリズム^{1, 3, 4)} も提案されている。しかし、組合せ最適化問題においても、また、連続最適化問題においても、温度と呼ばれる制御パラメータを持っており、この温度を下げすぎると局所解から脱出できる確率が低くなってしまいうため、温度を徐々に下げる冷却過程（クーリングスケジュール）の制御が重要となってくる。

SA には良質な解を得ようとすればするほど、膨大な量の計算が必要になってくるという問題が存在する。この非効率性を克服するために考え出された手法として温度並列シミュレーテッドアニーリング⁵⁾ (Temperature Parallel Simulated Annealing; 以下 TPSA) がある。TPSA は膨大な量の計算による計算機への負荷（計算時間）を複数のプロセッサを使用して並列処理を行うので軽減することができ、並列処理の問題となるプロセッサ間通信も解交換の瞬間にのみ行われるだけであるので、並列処理との親和性が高いといえる。また、確率的に温度を上げたり下げたりするため、解の精度を左右するクーリングスケジュールを自動でできる。しかし、現在までに TPSA は巡回セールスマン問題、ブロック配置問題、グラフ分割問題などのわずかな離散問題にのみ適用されただけであり、その有効性は広く確認されておらず、特に連続設計変数空間には適用されていない。そこで、本研究では連続設計変数空間に TPSA を適用する手法を提案し、また、逐

次 SA との比較も行う。

2. シミュレーテッドアニーリング (SA)³⁾

2.1 逐次型シミュレーテッドアニーリング

2.1.1 シミュレーテッドアニーリングの概要

材料を適当な温度まで加熱し、その温度を保持した後、徐々に冷却していく操作、それが焼なまし（アニーリング⁶⁾）である。焼なましは鉄鋼の熱処理でよく用いられ、その目的は以下のようなものである。

- (1) 鋼の軟化による塑性加工性や切削加工性の改善
- (2) 鍛造、鋳造、溶接、機械加工で生じた残留応力の除去

焼なましを行うことによって、エネルギーがもっとも少ない状態に分子が配列し、結晶構造を形成させることができる。そこで、この焼なましを結晶組織のエネルギー準位の最小化の技術とみなすことができる。その焼なましのアナロジーから、システム最適化のアルゴリズムとしてシミュレーテッドアニーリング (SA) 法⁷⁾ が考え出された。

SA が適用可能な対象は以下の通りである。すなわち有限の状態からなる状態空間 Ω があって、各状態 x に対してコスト $f(x)$ が定義されているとする。コストは統計力学の用語としてはエネルギー $E(x)$ にあたる。SA では、最小のコストを与える状態、すなわち真の最適状態 x^* を求める事が目的である。SA には、与えられた初期状態から出発して、次々と状態を推移させ、最終的には最適状態に行き着くことが期待される。

SA のアルゴリズムでは、生成、受理判定、クーリングの三つの処理が重要な役割を担う。まず、温度 T を初期設定した後、与えられた状態 x_0 から出発して、次の状態 x' を生成処理で生成し、そのエネルギー E' を計算する。エネルギーの差分 $\Delta E (= E' - E)$ と温度 T_k に応じて受理するか否かを受理判定処理で計算し、受理の場合は次の状態に推移する。この処理を繰り返して現在の温度での平衡状態が実現されるまで”十分な探索”を行う。このとき、”十分な探索”としては、定めた回数だけ繰り返す、定めた受理（または棄却）回数になるまで繰り返す、あるいは温度幅を十分小さくする場合は一回受理するまで繰り返す、などの実装方法がある。温度 T_k で平衡状態に達したら、クーリングを行い、次の温度 T_{k+1} を求め、再びその温度で平衡状態に達するまで”十分な探索”を進める。十分温度が冷えて停止条件に達すれば、その時の状態と工

エネルギーをおのおの最適状態と最適値として出力する。
SAの基本アルゴリズムを Fig.1 に示す。

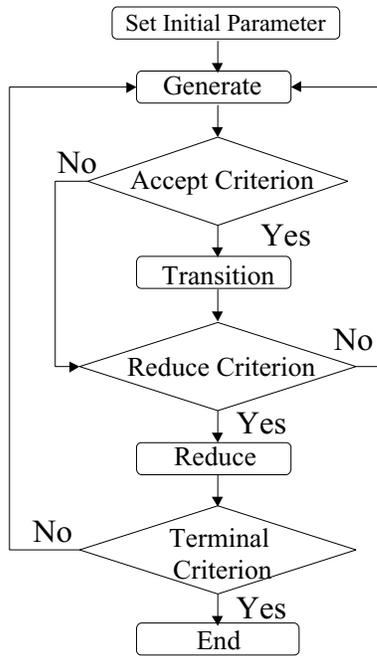


Fig. 1. Simulated annealing algorithm.

2.1.2 生成処理

生成処理は、現在の状態 x を与えられて、次に推移すべき状態 x' を返す。この生成処理には式 (1) のような、確率分布 $G(x, x')$ を用いる。組み合わせ最適化問題の場合は、状態 x' は状態 x の近傍にあたり、推移に優先性を考えない場合は、以下の等確率推移となる。

$$G(x, x') = \frac{1}{n(x)} \quad (1)$$

ここで、 $n(x)$ は、状態 x の近傍を構成する状態の数を表す。また、状態空間が連続の場合は、状態の確率分布として正規分布などを仮定するが、これに関しては 4 章で述べる。

2.1.3 受理判定処理

受理判定は、次の状態 x' のエネルギー E' と現在の状態 x のエネルギー E との差分 $\Delta E (= E' - E)$ 、および、温度パラメータ T によって、次の状態への推移を受理するか否かの判定をする。通常は、以下の Metropolis の基準²⁾ が採用される。

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

温度 T は、コスト (エネルギー) が増大する方向への推移確率に重大な影響を与えるパラメータである。温

度が高い場合は改善方向への推移確率も大きくなり、反対に温度が低い場合は専ら改善方向に推移することになる。しかし、どんな温度でも改善方向への推移確率がゼロになるわけではない。

2.1.4 クーリング

クーリングはアニーリングの第 k ステップの温度 T_k を与えて、次のステップの温度 T_{k+1} を返す。最適解への漸近収束性を保証するには、対数型アニーリング

$$T_{k+1} = \frac{T_k}{\log k} \quad (3)$$

以上に急速に冷やしてはならない。しかし、それではあまりにも遅いため、運用上は、真の最適解への収束を犠牲にして、指数型アニーリング

$$T_{k+1} = T_k(0.8 \geq < 1) \quad (4)$$

がよく用いられる。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、温度スケジュールに関する報告もなされている^{8, 9)} が、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。

また、徐冷に関して、これ以上徐冷をする事が無意味だと分かれば、つまり、現在の谷を探索することに進捗がないと分かれば、今度は再び加熱¹⁾ するアルゴリズムも提案されているが、その場合どれだけ温度を上げるかが問題となってくる。

2.1.5 終了条件

終了条件には、アニーリングを定めた回数だけ繰り返して停止、受理がほとんど起こらなくなったら停止、同じ状態が何度も生成されるようになったら停止、温度が十分小さくなったら停止、エネルギーの変化またはエネルギーそのものが十分小さくなったら停止、などの実装方法がある。

2.1.6 シミュレーテッドアニーリングの長所と短所

SA 法は、結晶構造の状態を組合せ最適解に、また、エネルギー値を最小化する関数値に対応させた同様のアルゴリズムで、最適解を探索する方法である。

SA の長所として、以下のようなものが上げられる。

(a) 多くの最適化解法が局所最適解に補足される欠点を持つのに対し、SA は容易に捕捉されず、理論上は真の最適解に、実際には準最適解に到達できる。これは、解品質が改良方向のみに探索を進めるのではなく、時折、改悪する方向も選ぶ仕組みによる。

(b) 枠組み自体が極めて汎用にてできているので、ほと

んど任意の境界条件,あるいは制約条件を処理できる.
(c) 目的(コスト)関数に対する制約がほとんどない.
滑らかさ,連続性,決定性はいずれも満たされなくてもよい.つまり目的関数は微分可能でなくても,複雑な式で求まるものであっても,確率的であってもよい.
(d) 他の非線形最適化アルゴリズムと比較して,コード化が極めて容易である.

また,SAには以下のような欠点が存在する.

- (a) 最適解を得るために膨大な計算量を要する.
- (b) 汎用解法であるため,特定の問題を解く場合には,パラメータをチューニングする努力が必要となる.特に,温度と呼ばれる制御パラメータのチューニングが極めて困難である.

2.2 シミュレーテッドアニーリング(SA)の並列化

SAには,良質な解を得ようとすればするほど,膨大な量の計算が必要となってくるという問題がある.こういった非効率性を克服するための方法は二つ上げられる.一つは逐次処理のまま高速なアニーリングを導入する高速化の研究^{8, 10, 11)},つまり,より速く処理できるアルゴリズムの研究である.もう一つは並列化して高速化を図る並列処理の研究⁷⁾である.

逐次SAの非効率性を解決するためのアプローチとして,AartsとKorstは以下のような並列SAを明らかにしている⁷⁾.

- (a) 定期同期型
- (b) 受理時同期型
- (c) 非同同期型
- (d) 完全独立型³⁾

上に示すものは逐次SAの欠点の一つである膨大な計算量を克服するための手法として考えられた並列SA手法である.しかし,これらの並列SA手法では,逐次SAのもう一つの欠点である温度パラメータのチューニングの困難さを克服することができない.そこで考え出された手法が温度並列シミュレーテッドアニーリング(Temperature Parallel Simulated Annealing:以下TPSA)⁵⁾である.上に示す4つの並列SA手法では,逐次SAと同様に,各々のプロセッサにおいてクーリングを行う必要があるため,クーリングスケジュールを設定しなければならないのに対し,温度並列シミュレーテッドアニーリングは各々のプロセッサにおいて温度が一定であるのでクーリングスケジュールを設定する必要がなく,また温度スケジュールを自動化できる.

3. 温度並列シミュレーテッドアニーリング(TPSA)⁵⁾

SAはマルコフ連鎖を次々とたどる処理であるため,本来強い逐次性があり,並列化は概して容易ではない.そのため,2.2に示す4つの並列SA手法では高速化,解の品質のいずれかを犠牲にすることが多く,また,温度スケジュールを決定しなければならないという欠点が残っている.そこで望まれる手法として考え出された手法に並列処理によるスピードアップが得られやすく,解の品質の劣化が少ない,また,温度スケジュールを自動化できる並列SAである温度並列シミュレーテッドアニーリング⁵⁾がある.

通常の逐次SAでは,温度 T を温度スケジュールに従って減少させる.効率的な最適化を行うためには,温度スケジュールをうまく設定することが重要な課題である.TPSAでは,各プロセスに解を一つずつ与え,互いに相異なる一定温度で同時並列的にアニーリングを行うことを考える.

3.1 基本アルゴリズム

逐次SAでは温度 T を温度 T' に減少させるのに対して,TPSAのアルゴリズムでは,温度 T のプロセスから温度 T' のプロセスへ解(または温度)を渡すことを考える.更に,プロセッサ間の解の受け渡しは,互いの解の交換によって行うものとする.このときの交換は確率的に行う.

逐次SAのアルゴリズムで温度スケジュールを設定することは,TPSAのアルゴリズムではプロセッサ間の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自動化し,温度スケジュールを不要にすることが考えられる.すなわち,確率的な解の交換が,良い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.

二つのプロセッサ間で解の交換を行うとは,実数 $0 \leq p \leq 1$ を定めて,確率 p で互いの解を交換し,確率 $1 - p$ で何もしないことと定義する.このときの確率を解交換確率と呼ぶ.解の交換は互いに異なる一定温度で同時並列的にアニーリングを行い,ある一定のアニーリングステップごとに解交換確率に従って行う.最終的に最低温度のプロセッサで得られる状態とエネルギーをおのおの最適状態と最適値として出力する.

TPSAのアルゴリズムをFig.2に示す.

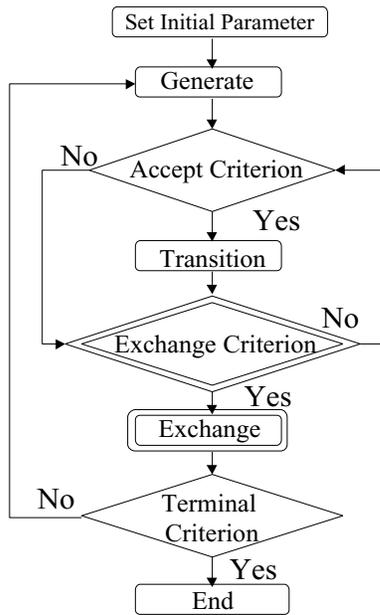


Fig. 2. Temperature parallel simulated annealing algorithm.

逐次 SA のクーリングスケジュールと TPSA のクーリングスケジュールの概念図をそれぞれ, Fig.3 と Fig.4 に示す. 図は共に, 横軸は時間の経過を表し, 縦軸は温度を表す.

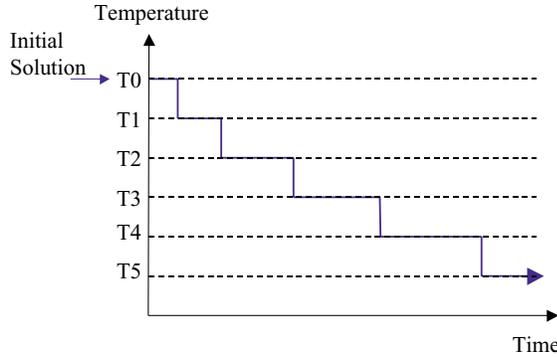


Fig. 3. Cooling schedule for sequential simulated annealing.

Fig.3 に示されている T_0 から T_5 は T_0 を最高温度として等比的に温度を減少させたものである. 逐次 SA の場合は, はじめに最高温度である T_0 に初期解を与え, その温度においてある程度の時間アニーリングを行う. 次に等比的に減少させた温度である T_1 において, ある程度の時間アニーリングを行う. これを繰り返し行い, 最終的に最低温度である T_5 が最適解を出力する. Fig.3 に示す矢印は逐次 SA がたどった温度

スケジュールである. この動作は 1 プロセッサが初めから順に行う.

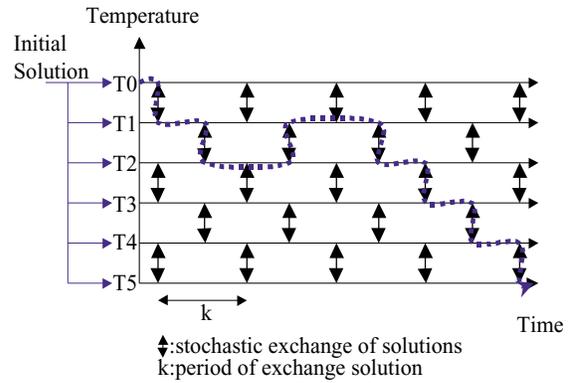


Fig. 4. Cooling schedule for temperature parallel simulated annealing.

Fig.4 に示されている T_0 から T_5 は T_0 を最高温度, T_5 を最低温度として等比的に温度を減少させたものである. TPSA の場合は各 T_0 から T_5 までの温度をそれぞれ異なったプロセッサに割り当て, それぞれの温度には異なった初期解が与えられる. すなわち, 各プロセッサの温度はシミュレーションの間一定であり, 解候補は複数存在する. そして, 各プロセッサごとに同時並列にアニーリングを行い, 隣り合う温度同士で解交換を行う. 解交換は解交換周期に達した時に解交換確率を満たしていれば行われる. 次の解交換周期に達した時には先ほど交換した相手とは異なった隣り合う温度と解交換を行う. 最終的には最低温度である T_5 が最適解を出力する. Fig.4 に示される点線の矢印は TPSA がたどった温度スケジュールである.

3.2 解交換

TPSA を行うにあたり, 各プロセッサごとに互いに相異なる温度でアニーリングを行った場合, 高温のプロセッサでは比較的自由に解空間が探索されて最適化はあまり進まないのに対し, 低温のプロセッサではエネルギー E の減少する方向ばかりに探索が進むので逆に局所最適解に陥り易い. このことは, 単にプロセッサごとに独立にアニーリングを行っただけでは, 最低温度のプロセッサで最良の解が得られるとは限らないということを意味している. そこで, プロセッサ間で解の交換を行う必要が出てくる. この解の交換には次のような解交換確率が用いられる.

$$P_{EX}(T, E, T', E') = \begin{cases} 1 & \text{if } \Delta T \Delta E < 0 \\ \exp\left(-\frac{\Delta T \Delta E}{T T'}\right) & \text{otherwise} \end{cases} \quad (5)$$

この交換確率では、交換相手プロセッサの解のエネルギー E' および温度 T' と、自分の解のエネルギー E および温度 T とのそれぞれの差分 $\Delta E(E' - E)$ と $\Delta T(T' - T)$ を用いて、解を交換するか否かを判定する。

式 (5) は、交換相手が低温 (または高温) であるが自分より悪い解 (または良い解) を持っていれば全て交換し、交換相手が低温 (または高温) で自分よりも良い解 (または悪い解) の場合でもある確率で交換するということを意味している。TPSA では、逐次 SA というクーリングの代わりに、隣り合う温度のプロセッサ間で解を確率的に交換することにより、温度スケジュールを自動化している。

3.3 TPSA の利点

TPSA は通常の逐次 SA と比較して、以下のような利点を持つ。

(a) 逐次 SA を実装する上で問題となる温度スケジュールは、TPSA においては異なった温度を担当するプロセッサ間で、解の確率的交換を行うことにより自動化できる。すなわち、逐次 SA においてもっとも重要な温度スケジュールの設定を考えなくてもよい。一方、逐次 SA においてはアニーリングの途中で温度を冷却していくとき、あまりにも急激に冷却しすぎると局所解に捕らわれ易くなり、また逆にあまりにもゆっくりと冷却すると最適解を得るまでにかなりの時間を要してしまう。

(b) 並列処理で効率を下げる大きな要因の一つとして、プロセッサ間通信が考えられる。TPSA では、各プロセッサで独立に一定温度のアニーリング処理が行われており、プロセッサ間通信が必要となるのは初めに定めたアニーリングステップ数に達したときに行う解交換の瞬間のみである。常にお互いが通信しているわけではないので TPSA は並列処理と非常に親和性の高い並列アルゴリズムであるといえる。

(c) TPSA は、通常の逐次 SA 法における温度スケジュールのような、処理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、アルゴリズムが時間的に一様である。このことは、ある地点で処理をうち切って得られた解が不満足なものならば、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ということを意味する。一方、通常の逐次 SA 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り、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。

4. 連続設計変数空間における TPSA の応用

この章では、連続設計変数の最適化問題に TPSA を応用する方法を述べる。

4.1 対象問題

本研究で対象とした連続問題は式 (6) 示す Rastrigin 関数である。

$$f(x_i|_{i=1,N}) = N * 10 + \left[\sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i)) \right] \quad (6)$$

N は設計変数の数であり、 x_i は設計変数の値である。ただし、 x_i の値は - 5.12 以上 5.12 の範囲である。この関数は座標 $(0, \dots, 0)$ で最小値 0 をとり、その周辺に格子状に複数の準最適解をもつ。Fig.5 から分かるように、極めて局所解を多く有する問題である。Fig.5 に 2 変数の場合の Rastrigin 関数の景観を示す。

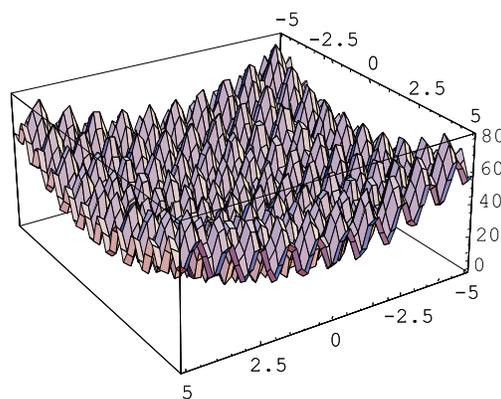


Fig. 5. The rastrigin function (Design value 2).

SA の探索の特徴としては、一般的に言えることは、アニーリングの初期の段階では関数の”大域的な景観に従う探索”を行い、最適解が存在する領域に到達することができ、その後、その到達した領域内で”より細かな探索”を行うということである。しかし、本研究で用いた連続最適化問題である Rastrigin 関数は大域的最適解がある領域と局所的最適解がある領域が似ているため、SA で解くことは困難な問題である。

4.2 連続空間での近傍の構造

連続最適化問題では、設計変数が連続であり、離散問題のように簡単に近傍を定義することはできない。連続設計変数空間における解の変更は、近くであろうと遠くであろうと数値を変更するだけであり、組合せ最適化問題のように、操作的に近傍を定義することはできない。そのかわり、物理的に意味のある、すなわ

ち目的関数の連続性における近傍を考えることは容易である。

そこで、連続最適化問題では一般に現在の解を中心とし、移動距離に関する確率分布を与え、近傍を定義する。この確率分布としては以下のようなものが上げられる。

(a) Boltzmann アニーリングは、真の最適解に至ることが保証されている対数型アニーリング⁸⁾であり、正規分布¹¹⁾が用いられる。

(b) FSA では、双曲線型の高速アニーリングを実現した。この分布には、裾野が正規分布に比べるとより肉厚の形をした Cauchy 分布¹⁰⁾が用いられる。

(c) 特に高次元の問題に対しては、際だった効果を発揮すると期待される VFA はペン先の形のように、中央部で尖っており、両側で分布を切断した特殊な形の確率分布¹⁰⁾を用いる。

(d) 適応的な一様分布¹²⁾。

これらの研究から明らかになったことは、用いる確率分布は近傍探索が十分に行えるように、中央部で厚く、離れたところで急速に減少する、あるいは確率をゼロにするものが良いとされ、しかもその分布の幅を受理確率によって適応的に変化させるものが良いとされるということである。すなわち、近傍探索において受理の頻度が高すぎると、近傍が小さすぎて無駄な探索が多くなり、一方受理が少なすぎると、近傍が大きすぎて無駄な探索が多く行われることになり、受理の頻度を基に近傍の大きさを適応的に変化させることが重要となる。

本研究では、近傍としてはもっとも単純な正規分布を用いた。しかも、標準偏差に用いたものは温度の平方根である。すなわち、現在の状態 x からの摂動 $\Delta \vec{x} (= \vec{x}' - \vec{x})$ を次の式 (7) の分布で与えた。このとき D は設計変数の数であり、 T は現在の温度である。

$$g_k(\Delta \vec{x}) = \frac{1}{(2\pi/T_k)^{D/2}} \cdot \exp\left(-\frac{|\Delta \vec{x}|^2}{2T_k}\right) \quad (7)$$

温度が高ければより遠くまで探索できる確率が増え、温度が低ければ近くを探索することになる。

4.3 温度パラメータの設定方法

この節では TPSA に必要な最高温度と最低温度の決定方法を説明する。

一般に組合せ最適化問題では、最大の改悪が生じる状態遷移が、ある値 (例えば 50%) の確率で受理される温度を最高温度とするという考え方が用いられる¹³⁾。しかしながら、連続最適化問題では、一般的に最大の

改悪が不明であるため、そうした考え方は有効ではない。したがって、連続最適化問題では組合せ最適化問題とは異なった方法で最高温度を決める必要がある。本研究では次のような方法で、最高温度を決定した。

設計変数がとりうる値で決まる設計空間は工学的問題では現実的な制約条件から通常はある程度の大きさで与えることが可能であり、この設計空間を最高温度の状態です十分に探索可能であるという条件を基に決定する。ここでは近傍の分布として式 (7) の正規分布を用いているため、各変数の標準偏差がその変数の設計領域の 1/4 となるようにした。よって最高温度は 6.5536 となる。これにより、現在の解が設計空間の中央にあるときは、それが端まで移動する確率は 4.6% であり、現在の解が設計空間の端にあるときは、他の端まで移動する確率は 0.01% 程度となる。設計変数の数が 2 の場合の最大温度のときの分布を Fig.6 に示す。

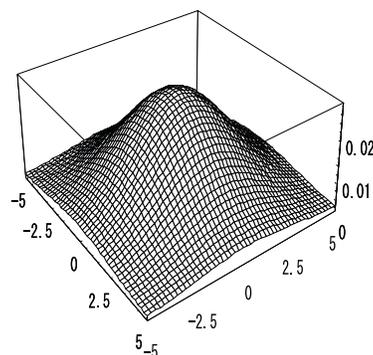


Fig. 6. Normal distribution (High temperature).

また、TPSA では温度スケジュールが不要であるかわりに最低温度を決める必要がある。組合せ最適化問題では最小の改悪がある確率で受理されるという基準で考えるが、連続最適化問題ではそうした考え方は最高温度と同様にできない。このため、最低温度は解の精度を基準に決定する。すなわち、式 (7) で与えられる近傍が十分小さい温度ということである。工学的には温度が下がりすぎて近傍が非現実的に小さくなっていることは意味がない。そこで本研究では、それとは違った方法として、正規分布の標準偏差が設計空間の 1/100 程度になる温度を最低温度と考えた。よって、このときの最低温度は 0.01 ということになる。最低温度のときの分布を Fig.7 に示す。

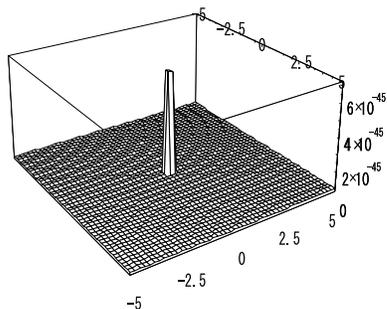


Fig. 7. Normal distribution (Low temperature).

上で述べたように、最高温度と最低温度が決定されると、その間を等比的に分割したものが各プロセッサに割り当てられることになる。

4.4 エネルギー計算

制約条件のない最適化問題では、目的関数に適当なスケールファクタを乗じたものをエネルギーとすればよい。制約条件がある最適化問題では、目的関数に制約条件に関するペナルティ関数を加えた疑似目的関数を作成し、それに適当なスケールファクタを乗じたものをエネルギーとすればよい。この場合のスケールファクタは受理確率を基準に決める。それには2つの考え方がある。まず、解の摂動と目的関数の関係が分かっている場合には、ある程度の値で起こり得る可能性のある目的関数の改悪が評価できるので、その改悪を受理する確率が50%になるように決める。一方、そうでない場合は、近傍を定義した確率分布を用いて実際に試行を行い、ある試行回数の中で生じる目的関数の最大の改悪の平均値を実験的に求め、それを50%の確率で受理するようにスケールするのが合理的と考えられる。本研究では後者の考え方をを用いた。予備実験として、最高温度のときに式(7)の分布を用い、乱数を用いて、100回の試行を10回行った。このときの目的関数の最大の改悪の平均値は、49.2となり、この改悪が50%の確率で受理されるようにエネルギーのスケールファクタを決めた。その値は0.0923である。

5. 実験結果

5.1 本手法を実装した計算機

本研究では、TPSAが並列処理との親和性が高いということを考慮し、8プロセッサのPCクラスタ(Pentium, 233MHz x 8)を使用した。プロセッサ間通信はFastEthernetとPVMを用いた。

このとき、一つの温度を一つのプロセスに割り当てる。よって、温度数が8のときは、一つのプロセッサに一つの温度であるが、温度数が32のときは、一つのプロセッサで4プロセスが実行されていることになる。

5.2 最適化性能

ここでは設計変数の数が5の場合のRastrigin関数を用いた。TPSAにおける温度数は32とし、各プロセスが担当する温度は最高温度と最低温度を等比的に分割したものを割り当てた。解の交換周期は20とし、繰り返し計算回数は32000回と決定した。また、このTPSAの性能を評価するため、通常の逐次SAとの比較を行った。このときの逐次SAはTPSAと同じ条件になるように設定した。すなわち、逐次SAにおいて、冷却における温度段数は32とし、32000回の計算で最低温度に達するように徐冷率を決め、指数型アニーリングを用いた。このときの冷却率は0.811となる。このことは32000回の計算で1000回の計算ごとに一段数温度が下がるということである。通常の逐次SAを用いる場合、計算回数に関して二通りの考え方がある。すなわち、TPSAの計算量と同じになるように温度数32だけ行う方法と、通常のSAの温度スケジュールを極めて緩慢にして、温度数32倍の時間をかける方法である。ここで、前者をSSAshort、後者をSSAlongと呼ぶ。なお、試行は全て10回を行い、その最良値、平均値、最悪値を比較する。

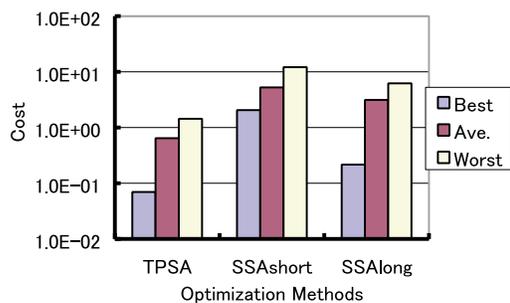


Fig. 8. Comparison of the performance of simulated annealing methods.

上のFig.8には、縦軸は目的関数のコストを、横軸にTPSA, SSAshort, SSAlongの最良値、平均値、最悪値をとったものを示した。

最適化性能に関しては、平均値を比較したときを考えてみると、TPSAは最も優れた性能を示していることが分かる。Rastrigin関数の最小値は0であり、1はその次に良い解とされている。よって、Fig.8よ

り、最適解に一番多く到達しているのは TPSA であることが分かる。TPSA に比べ他の逐次 SA は、まれに SSAlong が最適解に到達しているだけで、そのほとんどが局所解に補足されてしまっていることが分かる。SSAshort に関しては一度も最適解の近傍に達していない。また、逐次 SA では、冷却が緩慢であればあるほど、良好な解が得られることが予想されるが、TPSA は 32 倍の時間をかけた SSAlong よりも優れた性能を持っていることが分かる。ここでは、通常の SA よりも TPSA の方が連続設計変数空間において、極めて優れた性能を持っていることが分かった。

5.3 計算時間

この節では 5.3 節で用いたステップ数 32000 で交換周期 20 の TPSA、ステップ数 32000 の SSAshort、ステップ数 32000 × 32 の SSAlong のそれぞれの計算時間の最良値、平均値、最悪値を比較する。

Fig.9 は縦軸に計算時間 (sec) を、横軸に TPSA、SSAshort、SSAlong の最良値、平均値、最悪値をとったものを示す。

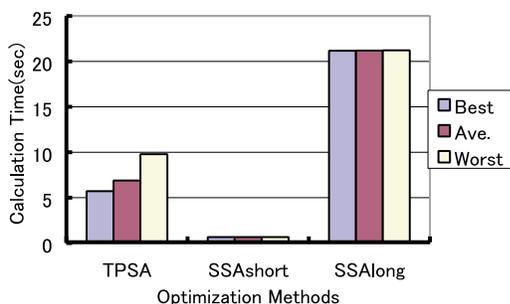


Fig. 9. Comparison of the calculation time of simulated annealing methods.

Fig.9 より、TPSA の場合の計算時間に少しの差が見られるが、逐次 SA の場合の計算時間で、最良、平均、最悪、に変わりが見られないことが分かる。よって、ここではそれぞれの平均について見てみることにする。

まず、SSAshort と SSAlong について見てみると、SSAlong は SSAshort に温度数 32 を乗じた計算回数 (1024000 回) を行っており計算時間も約 32 倍の長さになっているのが分かる。5.2 節でも考察したが、解の性能はよりじっくりと長い時間をかけて、冷却を行った SSAlong の方が良い結果を示している。

次に TPSA と SSAshort について見てみると、TPSA は SSAshort の約 9.9 倍の計算時間を要してい

ることが分かる。この要因として考えられることは、並列処理には欠かすことのできないプロセッサ間通信である。本研究において解交換は 20 回の計算ごとに行っているため、かなりの通信回数に達していると思われる。よって、この通信の分だけ、TPSA では処理に要する時間が長い。

最後に、TPSA と SSAlong について見てみると、全体としては計算回数が同じにもかかわらず、TPSA の計算時間に関しては SSAlong に比べ、1/3 より少ない時間しか要していないことが分かる。ここで本来ならば、並列温度数 32 で行っているため、効率的には 1/32 となるはずであるが、本研究で用いた並列計算機は 8 プロセッサであるため、一つのプロセッサが 4 つのプロセスを担当しているため、計算時間としては 1/8 になるのが一番効率の良い結果である。しかし、ここでも、並列処理に特有のプロセッサ間通信が行われているため、計算時間は結果として 1/3 になっている。しかしながら最適化性能を示す Fig.8 では、計算時間が TPSA の 3 倍以上要している SSAlong に比べ、TPSA は SSAlong より極めて良好な解を示しているため、これらのことより TPSA は計算時間においても解の品質においても、最も効率よく行われていると言える。

5.4 温度変化の履歴

この節では、総計算回数が 32000 回で、温度数が 32 の場合の逐次 SA と TPSA の温度の履歴を比較する。

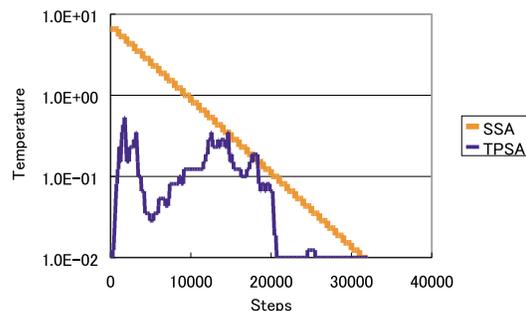


Fig. 10. Record of temperature of temperature simulated annealing and sequential simulated annealing.

Fig.10 は、1000 ステップごとに冷却を行った逐次 SA の温度の履歴と最終的に一番良い値に到達した解がプロセスが通過した温度の履歴を示している。横軸がステップ数で縦軸が温度を示す。

Fig.10 の TPSA の温度履歴を見てもわかるように、逐次 SA の単調減少とは違い、複雑に温度が変化して

いるのが分かる。また、このことより、最適解を得るために冷却だけでなく、再加熱を行った方が良いということが分かる。

逐次 SA では初期温度 (最大温度) で大域探索をし、中間温度で最適解の谷を探索し、最低温度で最適解へ収束していく。このとき、中間温度で局所解の谷へ補足された場合、その谷から抜け出せるだけの温度がなければ、出力される解は局所解ということになってしまう。

TPSA では局所解の谷へ入ってしまっても、再加熱する事により、局所解の谷を抜け出すことができ、再び冷却することにより、最適解へ収束する事ができる。しかも、TPSA の場合、冷却、再加熱という、クーリングスケジュールは自動で行われている。

6. おわりに

本研究では、これまでの研究の中でそのほとんどが組合せ最適化問題に適用され、連続最適化問題には適用されていなかった温度並列シミュレーテッドアニーリングを連続最適化問題に適用する方法を提案した。また、その問題に適用するときの問題点を抽出し、それを解決する方法を提案した。このとき用いた連続最適化問題は数学的関数であり、その最小化問題に本手法を応用し、その有効性を検証した。本研究によって得られた結論は以下の通りである。

(1) 連続最適化問題に温度並列シミュレーテッドアニーリングを適用する場合に必要な、温度パラメータの設定の方法を提案した (最高温度は設計空間の大きさを考慮して、設計変数の範囲の $1/4$ が近傍生成における正規分布の標準偏差になるように、また、最低温度は解の精度を基準に解の遷移が小さくなりすぎないように決定した)。

(2) この手法により、連続空間における最適化問題においても温度スケジュールを自動化することができる TPSA を実現できた。

(3) 逐次 SA では良好な性能が出にくい連続最適化問題である数学的関数の最小化技法に、(1) で提案した方法を用いた TPSA を適用した。その結果、TPSA は逐次 SA よりも、極めて良好な最適化性能を持っていることを示した。

(4) TPSA の計算時間は、8 台の PC を用いて同じ計算回数である逐次 SA (SSAlong) の 3 分の 1 の時間で計

算できることを確認できた。その上、最適化性能においても極めて優れた性能を持っていることを示した。

参考文献

- 1) S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi, "Optimization by simulated annealing", Science, vol.220, no.4598, pp.671-680, May (1983).
- 2) N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of State Calculation by Fast Computing Machines", Journ. of Chemical Physics, Vol. 21, pp.1087-1092 (1953).
- 3) B.E. Rosen, 中野良平, "シミュレーテッドアニーリング - 基礎と最新技術 - ", 人工知能学会誌, Vol.9 No.3 (1994).
- 4) N.E. Collins, R.W. Eglese and B.L. Golden, "Simulated annealing-an annotated bibliography", American J. Math. & Management Sci., Vol.8, No.3-4, pp.209-307 (1988).
- 5) 小西健三, 屋鋪正史, 瀧和男: "温度並列 SA 法の TSP への適用と分散メモリ型並列計算機上での効率の良い実現手法", 並列処理シンポジウム JSP'96, (1996).
- 6) 日本熱処理技術協会編, 「入門・金属材料と組織」, (大河出版, 東京, 1997)
- 7) E.H.L. Aarts and J.H.M. Korst, 「Simulated annealing and Boltzmann machines」, (Wiley, Chichester, 1989).
- 8) H. Szu and R. Hartley, "Fast Simulated Annealing", Physics Letters A, Vol. 122, No. 3,4, pp.157-162 (1987).
- 9) P.N. Strenski and S. Kirkpatrick, "Analysis of Finite Length Annealing Schedules", Algorithmica, Vol. 6, pp. 346-366 (1991).
- 10) L. Ingber, "Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison", Mathematical and computer Modeling, 16(11), pp. 87-100 (1992).
- 11) G.E. Hinton, T.J. Sejnowski and D.H. Ackley, "Boltzmann machines: constraint satisfaction networks that learn", Technical Report CMU-CS-84-119, Carnegie-Mellon Univ. (1984).
- 12) A. Corana, M. Marchesi, C. Martini, and S. Ridella, "Minimizing Multimodal Function of Continuous Variables with the "Simulated Annealing" Algorithm", ACM Trans. on Mathematical Software, Vol. 13, No. 3 pp.262-280 (1987).
- 13) 小西健三, 瀧和男, 木村宏一, "温度並列シミュレーテッドアニーリング法とその応用", 情報処理学会論文誌, Vol.36, No.4, pp.797-807 (1995).