連続最適化問題への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[†] 笠井誠之[††,☆]

組合せ最適化問題における並列解法として有効な性能を示す温度並列シミュレーテッドアニーリング（TPSA）を連続最適化問題に適用する場合の問題点を抽出し，これを解決する方法を提案し，2つの異なる種類の連続最適化問題に応用した結果を示す．近傍の生成には正規分布を用い，最高温度は設計空間の大きさと近傍の大きさを基に決め，最低温度は解の精度を考えて決定した．対象とした問題は標準的なテスト関数と構造最適化問題である．用いた温度数は64で，使用した計算機は8-CPUのPCクラスタである．計算の結果，テスト関数の場合は，SAにとって難しい問題であったが，同じ計算量となる長時間の逐次的なSAと比較してTPSAはきわめて良好な性能を示した．一方，現実的な構造最適化問題では多くの制約条件が存在したにもかかわらず，TPSAは逐次SAより良好な結果を示した．また，TPSAの並列性は良好であった．

Application of The Temperature Parallel Simulated Annealing to Continuous Optimization Problems

Mitsunori Miki,[†] Tomoyuki Hiroyasu[†] and Masayuki Kasai[††,☆]

The temperature parallel simulated annealing (TPSA), which has been found so far to be very effective for combinatorial optimization problems using parallel computers, is applied to continuous optimization problems. A new treatment for determining the neighborhood, the maximum and the minimum temperature is considered based on the design space and the accuracy of the solutions. The continuous optimization problems are the minimization of one of the standard test functions and the minimum-volume design of structures. The number of temperature stages is 64, and the computer used is a PC cluster with 8 processors. The numerical experiments showed that the performance of TPSA was remarkable compared with the conventional SA with very slow cooling for the standard test function, and was very good for the realistic structural optimization problems. Further, the parallel efficiency is fairy good.

1. はじめに

最適化の分野における並列処理には主として次の3つのアプローチ，すなわち，(1) 繰返し解析の単純な並列化，(2) 解析コードの並列化，(3) 最適化アルゴリズムの並列化，がある．この中で，最適化手法の並列化と考えられるのは (3) のアプローチであり，近年多くの研究者が並列可能な最適化アルゴリズムの研究を行っている[1]．

連続設計変数の空間における最適化問題に対しては，これまで非線形数理計画法による解法が主流であった．設計空間が単峰で，しかも目的関数の勾配が連続である場合にはこのアプローチはきわめて有効であるが，そうでない場合には遺伝的アルゴリズム（GA），シミュレーテッドアニーリング（SA），あるいはタブーサーチ（TS）など，いわゆるヒューリスティックサーチ（heuristic search）[2] とよばれる方法が用いられることも多くなってきた．このことは，従来の方法で解ける問題の範囲を超えて，連続変数空間といえども目的関数がきわめて複雑な挙動を示す最適化問題が取り扱われ始めたことを意味している．なかでも，GAとSAはこのような手法の双璧であり，連続最適化問題に対しても多くの研究が行われてきた[3]．

複雑な連続最適化問題に対してGAとSAのどちらが有効かという問題は難しい．なぜなら，これらの方法はいずれもその問題に適したスキームを採用し，多くのパラメータを適切にチューニングしなければ良い結果が得られないからである．しかしながら，一般的にいえることは，設計空間内に多くのサイズの大きい局所解領域が存在する場合にはGAが有効であり，一

† 同志社大学工学部
Faculty of Engineering, Doshisha University
†† 同志社大学大学院
Graduate Student, Faculty of Engineering, Doshisha University
☆ 現在，ソニー株式会社
Presently with Sony Corporation

方，設計空間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は SA が有効である[4]．

SA[5] は，炉内の固体の熱的平衡状態をシミュレーションするための単純なアルゴリズム[6] を基本として最適化問題を解く方法であり，多くの組合せ最適化問題の解法として有用である[7]．SA の長所は，(a) ほとんど任意の非線形性を持つコスト関数を処理できる，(b) ほとんど任意の境界条件や制約条件が処理できる，(c) 他の非線形最適化アルゴリズムと比較してコード化はきわめて容易である，(d) 最適解の発見が統計的に保証されている，ことである．一方，SA の短所は，(a) 最適解を求めるのに要する時間が長い，(b) 特定の問題に対してチューニングするのが容易でない，(c) 過大評価されて用いられ，結果の解釈が間違っている場合がある，(d) 誤った利用によりエルゴード性を失う，すなわち冷却が早く，シミュレーテッドクエンチング（simulated quenching）になっている．この場合は最適解が求められる統計的な保証はない，ことである[8]．要するに，GA より単純なアルゴリズムで，計算機へのインプリメントもきわめて容易であるが，良い最適解を得るには非常に長い時間がかかる，すなわち計算負荷がきわめて高いということである．

計算時間が長いことは SA の最大の欠点である．たとえば，巡回セールスマン問題では SA で良好な近似解を得る計算量よりも完全な総当たり計算の方が計算量が少ない[9]．さらに厄介な問題は，適切な冷却スケジュール（cooling schedule，温度スケジュールともいう）が不明で，かなりの予備実験を行わなければならない点である．

SA の計算時間を短縮するには 2 つのアプローチがある．1 つはできるだけ高速の冷却スケジュールを考えることであり，もう 1 つは並列処理である．前者については Rosen らの解説[10] に詳しく述べられているが，対数型の Boltzmann アニーリングより，双曲線型の高速アニーリング，さらには指数型の超高速アニーリングを用いることにより，SA の高速化が可能となる．一方，SA の並列処理は並列計算機の発達とともに有効なアプローチとして多くの研究がなされている[11]．この中で最も典型的な方法は異なった初期値で通常の SA を並列に行い，ある時間ごとに最も良好な解を全プロセッサに渡し，並列に近傍探索するものである．一方，SA と GA を組み合わせた方法は，それぞれの方法の長所を活かし，さらに並列化が容易であることから多くの研究が行われている[12)~14)]．しかしながら，いずれの方法においても SA の冷却スケジュールが経験的にしか与えられないという問題はつねに残る．

これに対して，温度並列 SA[15] は並列処理との高い親和性を持っているだけでなく，温度スケジュールが原理的に不要であるというきわめて優れた特長を有している．このため，温度並列 SA はこれからの SA の発展に欠かせない手法と考えられ，これまでは LSI ブロック配置問題[16]，巡回セールスマン問題[17]，グラフ分割問題[18]，そして生物学的問題[19] などに応用され，逐次 SA と比較して温度並列 SA が計算時間および解の精度の点で優れていることが分かっている．しかしながら，連続最適化問題への応用はこれまでなされていなかった．

本研究は，このような背景の下に，連続変数を持つ最適化問題に温度並列 SA を応用する方法を提案し，2 つの代表的な連続最適化問題に適用した結果を基に温度並列 SA の有効性を検証することを目的とする．また，それらの結果を基に温度並列 SA の改良についても示唆する．

2. 温度並列 SA

温度並列 SA[15] は，複数のプロセッサに異なる温度を与え，各プロセッサは一定温度でアニーリングを行い，一定の間隔で隣接する温度のプロセッサ間で解の交換を行う方法である．この方法の特長は，(a) 温度を解自身が決定するので温度スケジュールの自動化が図れる，(b) 時間的に一様なので任意の時点で終了が可能であり，また，継続すれば解の改善を続けることができる，(c) 解の品質を劣化させることなく，温度数までの並列化が可能である，という点にある．

図 1 は温度並列 SA の概念図であり，通常の SA と比較している．上側に示した通常の SA では経験的に決めた単調減少の温度スケジュールを用いるのに対して，温度並列 SA では各プロセッサは一定の温度を担当し，解が自身のエネルギーを基準として適切な温度を選ぶ．このため，温度スケジュールは不要となる．しかも，最低温度での解の状態を観察していれば終了判定は容易である．すなわち，長い期間にわたって最低温度での解の更新がなければ最適解が求められたと判断できる．一方，通常の温度スケジュールを持つ SA では，最適解かどうかにかかわらず温度が下がれば解の更新がなくなる．したがって，できるだけ緩慢に冷却する必要があるが，対象としている問題に対してどの程度が緩慢な冷却かという判断は，多くの実験を行ってみなければ分からない．温度並列 SA では，このような問題はなくなる．

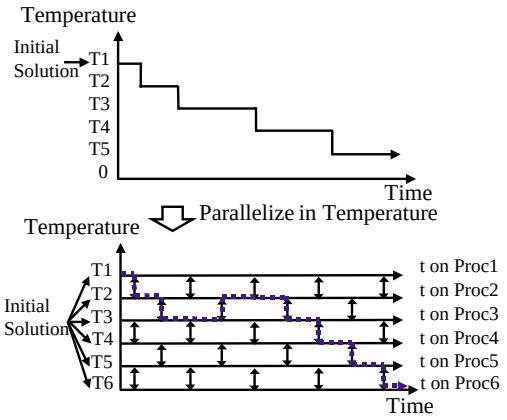

図 1 逐次 SA と温度並列 SA
Fig. 1 Sequential SA and temperature parallel SA.

温度並列 SA における隣接温度での解の交換は式 (1) を用いて確率的に行う．すなわち，隣接する温度 T と T' における解のエネルギーを E と E' とすると，高温部に低いエネルギーの解が存在した場合には無条件で解を交換し，それ以外でも温度差 $\triangle T$ とエネルギー差 $\triangle E$ などから計算される確率で解を交換する．これによって，低温部にエネルギーの低い解が集まるが，確率的にはそうでない場合もあることになる．

$$P_{EX}(T,E,T',E') = \begin{cases} 1, & \text{if } \triangle T \cdot \triangle E < 0 \\ \exp\left(-\dfrac{\triangle T \cdot \triangle E}{T \cdot T'}\right), & \text{otherwise} \end{cases} \tag{1}$$

一方，各一定温度における SA は通常の方法で行う．すなわち，近傍探索における受理確率は式 (2) で与えられる Metropolis 基準[5] を用いる．

$$P_{AC} = \begin{cases} 1, & \text{if } \triangle E < 0 \\ \exp\left(-\dfrac{\triangle E}{T}\right), & \text{otherwise} \end{cases} \tag{2}$$

$$\triangle E = E(\boldsymbol{x_{new}}) - E(\boldsymbol{x_{old}})$$

3. 連続最適化問題への温度並列 SA の適用

SA は組合せ最適化問題の有力な解法として提案され，広く用いられてきた．しかしながら，連続最適化問題においても対象とする問題の複雑度が高い場合には多く用いられている[4),8),20)~22)]．SA が有効な問題は，目的関数の勾配が連続でなく，設計空間内において大局的には単峰に近いが，局所的にはサイズが比較的小さい局所解領域がきわめて多いような問題である．

連続最適化問題と組合せ最適化問題では SA における近傍の概念およびその定義が異なる．組合せ最適化問題では，解の変更に必要な最小の操作の集合を近傍と定義することが一般的である．たとえば，巡回セールスマン問題ではグラフの 4 つのノードと 2 つの辺の関係を組み替えて，新しいグラフを作ることが可能である．これは X-交換（X-change）[23] とよばれ，これによって得られる新しいグラフの集合を X-交換近傍（X-change neighborhood）とよぶ．この近傍を基にして近似的に最適なグラフを求める解法が 2-Opt 法[24] である．こうして組合せ最適化問題では近傍は厳密に定義できる．

一方，連続最適化問題では設計変数が連続であり，上のように一般的に近傍を定義することができない．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，組合せ最適化問題でのように，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．組合せ最適化問題での近傍は目的関数とは何の関係もない．

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 Boltzmann アルゴリズムでは正規分布[22]，FSA ではコーシー分布[20]，VFSA ではペンの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[4] などが用いられている．また，分布の幅を受理確率によって適応的に変化させるという考え方もある[21]．

本研究では，温度並列 SA を連続最適化問題に対して拡張するために，近傍として正規分布を用い，しかもその幅を温度の関数とした．すなわち，現在の解からの摂動 $\triangle \boldsymbol{x}$ を式 (3) の分布で与えた．これは Boltzmann アルゴリズムで用いられる分布である．

$$g_k(\triangle \boldsymbol{x}) = \frac{1}{(2\pi T_k)^{D/2}} \exp\left(\frac{-|\triangle \boldsymbol{x}|^2}{2T_k}\right) \tag{3}$$

ここで T_k は k 番目の温度であり，k は通常の SA では温度スケジュールの番号を表し，温度並列 SA では複数の温度の 1 つを表す．正規分布を用いた理由は，この分布が確率的には無限遠の距離を推移する分布であり，近傍の幅を決める一様分布では明確な上限および下限を決める合理的理由が見当たらないからである．また，正規分布は無限の幅を持った分布であるが，現在の地点に近い地点への推移確率が高く，現在の解の情報を次のステップの解探索に利用しなければならない SA のアルゴリズムに適した分布であることも，適用した理由となっている．さらには，他のいくつか

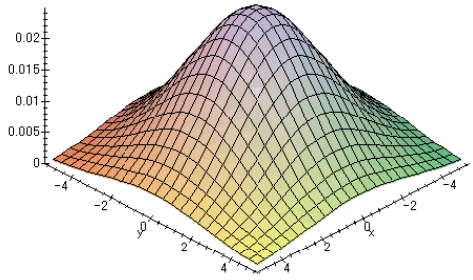

図 2 最高温度時の正規分布（2 次元）
Fig. 2 Normal distribution for the maximum temperature (2-dimensional).

の確率分布も正規分布で代用できる場合が多いからである．

次に最高温度を考える．組合せ最適化問題では一般的に，最大の改悪が生じる状態遷移がある値（たとえば 50%）の確率で受理されるという考え方が用いられる[15]．しかしながら，連続最適化問題ではそうした考え方はもはや有効ではない．その理由は，一般的に最大の改悪が不明であること，および近傍を定義した確率分布で最大を考えると非現実的な改悪を考えなければならなくなるからである．正規分布を用いれば近傍の最大は無限大の距離になってしまう．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るアプローチで最高温度を決める必要がある．

ここでは次のようにして最高温度を決定した．すなわち，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状態で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 (3) 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．これにより，現在の解が設計空間の中央にあるときに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に他の端まで移動する確率は 0.01%程度となる．最高温度における 2 次元の近傍の確率分布を**図 2** に示す．これより，最高温度では設計空間が十分に探索できていると考えられる．

温度並列 SA 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規準で考える[15]が，連続最適化問題ではそうした考え方はできない．これは最高温度と同様である．このため，最低温度は解の精度を規準に決定した．すなわち，式 (3) で与えられる近傍が十分小さくなる温度とする．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．多くの SA の応用で温度スケジュールだけを適当に決めて，かなりの繰返しを行って解が変動しなくなったのを観察して停止条件としている場合も多いが，そのときの温度を調べると非現実的に小さくなっており，最適解が見いだされたから解が変動しなくなっているのではなく，温度が下がりすぎたから変動しなくなったいるだけ，という場合も多いように思われる．これは意味のないことである．ここでは正規分布の標準偏差が設計空間の 1/10000 程度になる温度を最低温度と考えた．

次にエネルギーについて考える．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リングファクター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ー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リングファクターを乗じたものをエネルギーとすればよい．この場合のスケーリングファクターは受理確率を規準に決める．

組合せ最適化問題では，先に述べたように，最大の改悪が生じる状態遷移が 50%の確率で受理されるというような考え方でスケーリングファクターを決める．連続最適化問題ではこうした考え方はできないので，次のように考える．すなわち，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う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%で受理するようにスケーリングするのが合理的と考えられる．ここではこの考え方を用いた．

4. 並列計算機への実装

温度並列 SA では解の交換のときだけにプロセッサ間通信が発生するので，並列計算機との親和性が良好である．ここでは 8 プロセッサの PC クラスタ（Pentium II，233 MHz × 8）を用いて計算を行った．プロセッサ間通信は Fast Ethernet と PVM を用いて行った．

1 つの温度での SA を 1 つのプロセスに割り当てる．したがって，温度数が 8 までのときは 1 つのプロセッサに 1 つの温度プロセスが実行されているが，温度数が多くなると 1 つのプロセッサで複数の温度プロセスが実行される．温度数が 64 の場合は 1 つのプロセッサで 8 プロセスが実行されている．

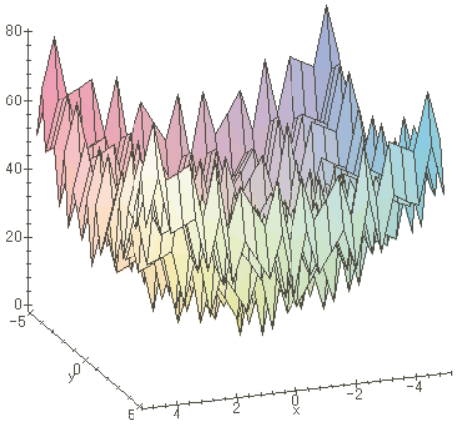

図 3 Rastrigin 関数の景観（2 次元）
Fig. 3 Landscape of Rastrigin function (2-dimensional).

5. 数学的関数の最小化問題

本研究では，連続最適化問題として種々のヒューリスティックアルゴリズムの性能検証に用いられる典型的な数学関数[25] の最小化問題を考える．ここでは式 (4) で表される 5 次元の Rastrigin 関数を用いる．

$$f(x_i|_{i=1,N}) = (N \times 10) + \left[\sum_{i=1}^{N}\left(x_i^2 - 10\cos(2\pi x_i)\right)\right] \quad (4)$$

この関数は 2 次元で示すと**図 3** のように，多くの比較的大きなサイズの局所解領域を持つ．このような関数の最小化問題は SA より GA で解くのが有効である．しかしながら，ここでは SA にとって難しい問題をあえて選んで温度並列 SA の有効性を検証することにした．

各設計変数の範囲は -5.12 から $+5.12$ とし，近傍は式 (3) で与えられる正規分布とした．先に述べたように，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最高温度（= 6.554）を決めた．一方，最低温度は，先に述べたように，解の精度で決める．ここでは正規分布の標準偏差が 0.001 となるように決めた．これで分解能は設計空間に対して 1.0E-4 となる．また，このとき最低温度は 1.0E-6 となる．

目的関数からエネルギーへの変換は次のようにした．すなわち，このような問題では設計変数の摂動と目的関数の変化には直接の関係はなく，最大の改悪は実験的に求めることになる．ここでは，式 (3) の近傍を用い，実際に乱数を用いて 100 回の試行を 5 回行い，100 回の試行における近傍での目的関数の最大の改悪の平均を実験的に得た．この値は 66.4 であり，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リングファクターを決めた．この値は 0.684 となる．

温度数は 64 とした．組合せ最適化問題では温度数は 32 程度あれば良好な結果が得られている[15]．中間の温度は最高温度と最低温度，ならびに温度数から温度が等比的に並ぶように決めた．解の交換周期は 40 とし，計算の繰返しは 64000 回および 128000 回行った．また，温度並列 SA の性能を評価するために通常の逐次 SA を行った．この場合，温度並列 SA と同じ条件とした．すなわち，冷却における温度段数は 64 とし，決めた繰返し数で最低温度に到達するように冷却率を決め，指数型の冷却を行った．ここでは冷却率は 0.78 となる．たとえば計算繰返し数が 128000 回の場合では 2000 回ごとに温度が 1 段低くなる．

通常の逐次 SA を用いる場合において，計算回数に関して 2 通りの考え方を考慮した．すなわち，温度並列 SA の並列処理における 1 つの温度が担当する計算量と同じだけの計算を行う方法と，逐次 SA の温度スケジュールをきわめて緩慢にして温度数（=64）倍の時間をかけることで総計算量を温度並列 SA と同じにする方法である．ここでは前者を SA(Short)，後者を SA(Long) とよぶ．なお，試行は温度並列 SA と SA(Long) については 5 回行い，SA(Short) については 64 回行い，その平均値，最悪値，および最良値で議論する．ここで SA(Short) のみ試行回数を 64 回としたのは，その結果の最良値が完全独立型の並列 SA [10] の結果として見ることができるという点からである．

図 4 は総計算回数が 64000 × 64 回の場合の得られた最適解の目的関数の値を示したものである．ここで温度並列 SA の平均値は最も優れた性能を示している．Rastrigin 関数の最小値は 0 であり，目的関数値が 1 となる解は 2 番目に良好な最適解である．このため，温度並列 SA は 2 番目の最適解を確実に見つけていることが分かる．これに対して SA(Long) については，最良値は温度並列 SA よりも良好な値となっているが，平均値は劣っている．一方，64 回の試行を行う SA(Short) の結果は最良値で議論を行う．その理由は，総計算量を同一として考えると，SA(Short) の 64 回試行は完全独立型の並列 SA の 1 回試行の結果と見なすことができ，その場合，最良値を結果として用いるからである．この値と，温度並列 SA の最悪値，および SA(Long) の最悪値と比較すると，SA(Short) の最良値が最も良好な値となっている．すなわち，総計算量の等しい手法として，これら 3 つの手法を比較したときに SA(Short) が最も良い性能となる場合が

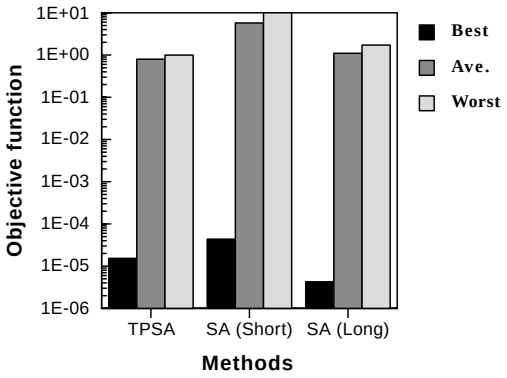

図 4 手法の比較（64000 × 64）
Fig. 4 Comparison of the methods (64000 × 64).

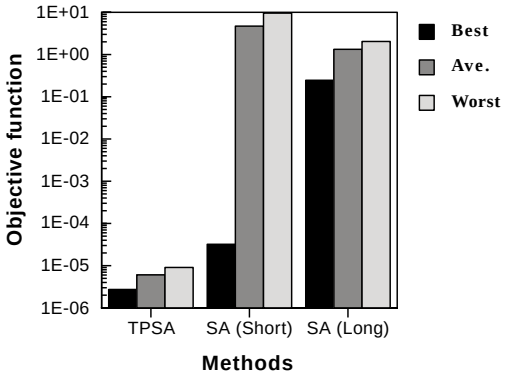

図 5 手法の比較（128000 × 64）
Fig. 5 Comparison of the methods (128000 × 64).

あることを示す．

しかしながら，総計算回数が 128000 × 64 回である**図 5** の結果は上述のものと異なる結果となっている．温度並列 SA ではすべての試行で真の最適解の近傍に達しているのに対して，長時間の逐次 SA では図 4 の結果と比較して，最良値を除いてほとんど変わらない．また，128000 回の繰返しの逐次 SA を 64 試行行った結果も図 4 と比較して目立った改善はない．一方，長時間の逐次 SA の最良値は偶然に得られたもので計算スキームや計算回数には関係ないことが分かる．

この結果より，温度並列 SA は計算回数を増やすだけで性能が上昇することが分かる．これこそ，温度並列 SA の特長である．すなわち，逐次 SA では最初に決められた温度スケジュールで行うので，総計算回数が決まってしまい，それを延長してもすでに温度が下がっているので意味はない．このため，多くの予備実験を行って温度スケジュールを最適化しなければならない．たとえばここでの結果では SA(Long)，すなわち 64 倍の時間をかけて冷却しても解の改善は図 4 および図 5 よりわずかであり，さらに長時間の別の温度スケジュールが必要と考えられる．一方，温度並列 SA では 64000 回での結果を見て，まだ不足と考えられればそれをそのまま継続するだけで図 5 の結果を得ることができる．しかも，最良値，平均値，および最悪値というすべての試行がほとんど同じになったことから真の最適解が求められたと判断できる．また，TPSA の最悪値は 64 試行行う SA（Short）の最良値よりも良好な値となっているため，TPSA は完全独立型の並列 SA よりも性能が良いということが分かる．

ここまでの結果により，温度並列 SA が解の精度の点で優れていることが分かる．しかし，次の点については注意が必要である．それは，温度並列 SA と比較した逐次 SA の温度スケジュールが適切であったかどうかである．本研究では，温度並列 SA の温度数 64 と条件を合わせるために，逐次 SA の冷却における温度段数を 64 とした．これは逐次 SA にとって最適ではなかった可能性がある．すなわち，逐次 SA では 1 つの温度において必要な繰返し数は，その温度で定常状態に達する回数であるため，それ以上の繰返しが無駄になるということである．より最適なスケジュールは，本研究で用いた温度段数が 64 ではなく，1 つの温度における繰返し数が少なくなったとしても，より多くの温度段数を用いるものであることが考えられる．しかし，そのことを考慮したうえで温度並列 SA が優れていると考えられる点は，最適な温度スケジュールというものを考慮せずに逐次 SA よりも良好な解を得たことである．これは解が低温，高温間を自ら移動する性質によるものである．

図 6 は 128000 × 64 の場合について計算時間を比較したものである．なお，TPSA(real time) は 8CPU で温度数（プロセス数）64 を行った場合の温度並列 SA の実計算時間を表し，TPSA(1 process) はこの TPSA(real time) を 1CPU に割り当てられた温度数 8 で除したものとした．こうすることで温度並列 SA における 1 つの温度（プロセス）に使用された CPU 時間の推定値が分かり，SA(Short) との比較が可能となる．TPSA(1 process) と SA(Short) を比較すると，TPSA(1 process) は解の交換という操作のための時間とプロセッサ間通信のための時間が余分にかかっているが，計算時間は SA(Short) より 48%の増加にとどまっている．また，SA(Long) は SA(Short) に比べて 64 倍の計算量であるため計算時間も 64 倍になっていることが分かる．しかし，図 5 を見ると，その長時間の計算にもかかわらず SA(Short) に比べて解の改

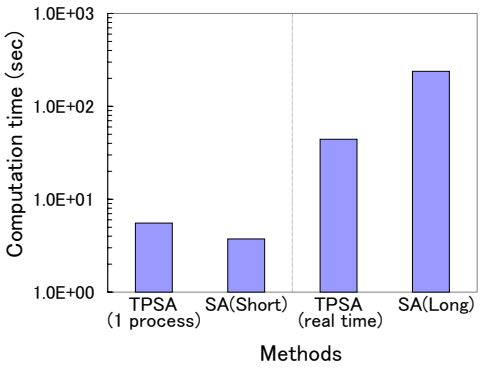

図 6 計算時間の比較（128000 × 64）
Fig. 6 Comparison of the calculation time (128000 × 64).

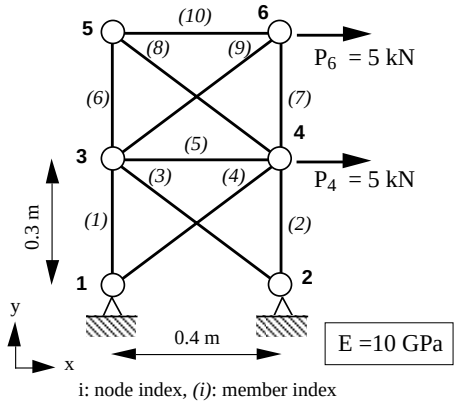

図 7 10 部材トラス構造と負荷条件
Fig. 7 A 10-member truss and the loading condition.

善は少ない．

一方, TPSA(real time)と並列化できない SA(Long) の計算時間を比較すると TPSA(real time) は 8CPU で約 19%の計算時間となっていることが分かる．このことから温度並列 SA は高い並列性で良好な最適解が出るまでの時間を非常に短縮できることが分かる．

6. 構造最適化問題

次に，上の問題とはまったく異なり，現実的な最適化問題での温度並列 SA の性能を検証する．ここでは構造最適化問題として典型的なトラス構造最小体積問題を考える．**図 7** は対象とする 10 部材トラスである．問題は，各部材の引張破壊と圧縮の座屈破損が生じないこと，および節点 6 の変位が規定値（=5.8 mm）以下であること，という制約条件の下で目的関数である構造の体積を最小化することである．設計変数は断面形状を円とする各部材の断面積とする．この問題についての詳しい説明は紙数の関係で文献 26) に譲る．

近傍の生成には前節と同様に式 (3) の正規分布を用いる．また，その標準偏差が設計領域の 1/4 となるように最高温度を決める．ここでは設計変数の範囲を 0 から 4000 mm^2 と考え，標準偏差が 1000 となるように最高温度を考えた．最高温度は 1.0E6 となる．また，最低温度は解の精度を 1 mm^2 と考えることで 1 となる．

次に，構造の体積にスケーリングファクターを乗じたものに局所制約に関するペナルティーのエネルギーと，全体制約に関するペナルティーのエネルギーを加えたものをエネルギーと考える．ここで，スケーリングファクターは可能性のある体積の改悪が最高温度においてある一定の確率（= 50%）で受理されるようにする．

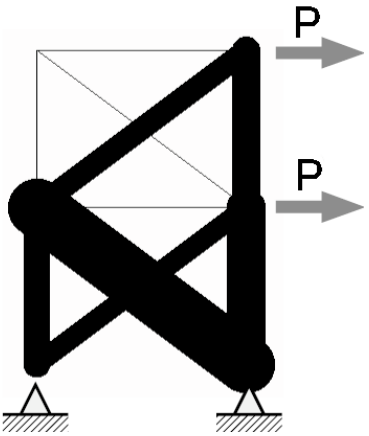

図 8 最適解の断面積分布
Fig. 8 Optimum distribution of the sectional area.

局所制約に関するペナルティーのエネルギーは局所制約を破っている部材の数の和にスケーリングファクターを乗じて求める．この場合，1 つの部材が破損するという状態遷移を最高温度において 50%の確率で受理するようにそのファクターを決めた．また，全体制約に関しては，制約を破った場合に変位の 2 乗にスケーリングファクターを乗じてペナルティーのエネルギーとした．ここでは変位制約がわずかに破られるという状態遷移を最高温度において 50%の確率で受理するようにそのファクターを決めた．

計算に用いたスキームならびに計算機は前節と同様である．温度並列 SA によって得られた最適解の一例を**図 8** に示す．初期値は乱数で与え，5 回の試行を行ったが，すべて同様の最適解が得られた．総計算量は 6400 × 64 である．通常の SA における冷却率は 0.803 となる．

図 9 に得られた構造の体積を示す．これより，温度

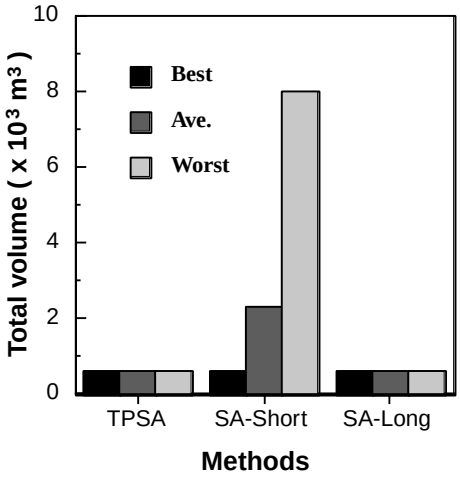

図 9 手法の比較（6400 × 64）
Fig. 9 Comparison of the methods (6400 × 64).

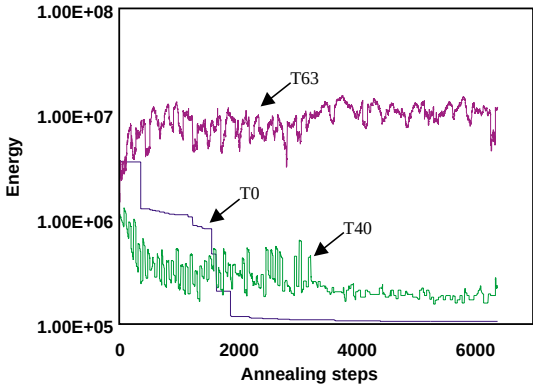

図 10 一定温度内のエネルギー変化
Fig. 10 Histories of the energy for constant temperatures.

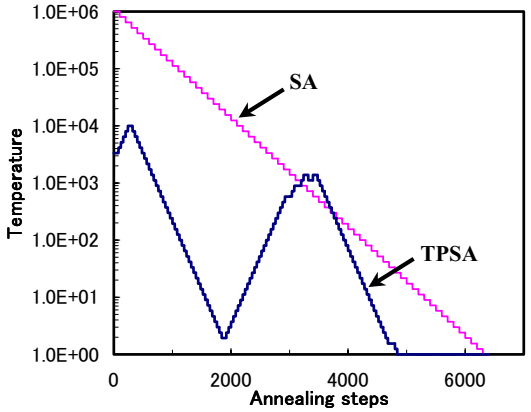

図 11 温度スケジュール
Fig. 11 Temperature schedule.

並列 SA は SA(Long) と同じ性能を有していることが分かる．一方，SA(Short) の結果より大変良好な結果となっている．この結果から，前章の数学的関数の最小化と同様に，現実的な構造最適化問題においても温度並列 SA が良好な結果を出すことが明らかとなった．なお，この問題では長時間 SA でも真の最適解を見つけているため，温度並列 SA との相違が出なかった．もうすこし少ない計算回数で比較すれば前節同様，温度並列 SA と長時間 SA の相違が明白になると予想される．いずれにしても，温度並列 SA は逐次 SA が行う繰返し計算回数を温度数で分割した短い繰返し計算回数を用いて真の最適解を出す能力があることが認められる．

図 10 は温度並列 SA における 1 つの温度内でのエネルギーの変化を示したものである．ここでは 64 温度のうち，最高温度の T63（温度 = 1.0E6），中間の T40（温度 = 6449），および最低温度の T0（温度 = 1）を考える．初期解はほぼ同じエネルギーであるが，最高温度ではエネルギーは上昇して一定値の周りを変動し，一方，中間の温度ではエネルギーは減少しつつも最適な値には到達しない．これに対して，最低温度でのエネルギーは解の交換によって着実に減少していることが分かる．温度並列 SA では最低温度での解の挙動を観察することで最適解が得られたことが分かる．ここでは 2000 回ぐらいから後はほとんど解のエネルギーに変化がなく，このため 3000 回もしくは 4000 回程度で終了判定ができる．

図 11 は温度並列 SA と SA(Short) の場合の温度スケジュールを比較したものである．ここで温度並列 SA では最も良好な解が得られた場合の，その解がたどった温度履歴を示した．これを見ると温度並列 SA では，逐次 SA にはない温度の増減を繰り返すことによって良好な解が得られていることが分かる．

7. 結　　論

本研究では，これまで組合せ最適化問題のみに用いられていた温度並列シミュレーテッドアニーリング（TPSA）法を連続最適化問題に適用する場合の問題点を抽出し，それを解決する提案を行い，2 つの異なった種類の連続最適化問題に応用し，その手法の有効性を検証した．得られた結論は以下のとおりである．

1) 連続最適化問題に TPSA を適用する場合の近傍，最高温度，および最低温度について議論を行い，近傍を正規分布で与えた場合の標準偏差を設計空間との関係で決定し，それを用いて最高温度を決定する方法を提案した．また，最低温度は必要な解の精度で決める方法を提案した．また，最大改悪となるエネルギーに

ついては決定した分布を基に実験的に求める方法を提案した．

2) 複雑な連続最適化問題として典型的な数学的関数の中で SA にとって難しい問題を選んで提案した方法に基づいて TPSA を行い，TPSA がきわめて優れた性能を示すことを明らかにした．すなわち，TPSA はきわめて長時間の逐次 SA よりも性能が高いことが確認できた．

3) 現実的な連続最適化問題として構造最適化問題を考え，これに提案した方法を用いて TPSA を適用し，この方法が長時間の逐次 SA と同様に真の最適解をきわめて短時間で見いだすことを確認した．

4) TPSA に用いた計算機は 8-CPU の PC クラスタであるが，短時間 SA と比較して TPSA は複雑な操作とプロセッサ間通信を行っているにもかかわらず，計算時間は 48%増加しただけであった．一方，解の品質は計算時間がはるかに長い長時間の SA と比較しても良好だった．これより，TPSA は連続最適化問題においてもきわめて短時間で良好な解を見つける能力があることが分かった．

5) 本研究によって，これまでの組合せ最適化問題で優位とされている TPSA の特徴が，連続最適化問題においても同様に得られるということが分かった．

参 考 文 献

1) Schnabel, R.B.: A view of the limitations, opportunities, and challenges in parallel nonlinear optimization, *Parallel Computing*, 21, pp.875–905 (1995).
2) Reeves, C.R.（編），横山，奈良ほか（訳）：モダンヒューリスティックス，日刊工業新聞社 (1997).
3) Schwefel, H.P.: *Evolution and Optimum Seeking*, John-Wiley & Sons, New York (1995).
4) Ingber, L.: Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison, *Mathematical and Computer Modeling*, Vol.16, No.11, pp.87–100 (1992).
5) Kirkpatrick, S., Gelett Jr. C.D., and Vecchi, M.P.: Optimization by Simulated Annealing, *Science*, Vol.220, No.4598, pp.671–680 (1983).
6) Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E.: Equation of State Calculation by Fast Computing Machines, *Journ. of Chemical Physics*, Vol.21, pp.1087–1092 (1953).
7) Aarts, E. and Korst, J.: *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons (1989).
8) Ingber, L.: Simulated Annealing: Practice versus Theory, *Journal of Mathl. Comput. and Modelling*, Vol.18, No.11, pp.29–57 (1993).
9) 文献 6) の p.54.
10) Rosen, B.E., 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, pp.365–372 (1994).
11) Holmqvist, K., Migdalas, A. and Pardalos, P.M.: Parallelized Heuristics for Combinatorial Search, Migdalas, A. et al. (Eds.), *Parallel Computing in Optimization*, p.269, Kluwar Academic Publishers (1997).
12) Chen, H. and Flann, N.S.: Parallel Simulated Annealing and Genetic Algorithms: A Space of Hybrid Methods, Davidor, Y. et al. (Eds.), *Parallel Problem Solving from Nature*, pp.428–438, Springer-Verlag (1994).
13) Yong, L., Lishan, K. and Evans, D.J.: The Annealing Evolution Algorithm as Function Optimizer, *Parallel Computing*, Vol.21, pp.389–400 (1995).
14) Kurbel, K., Schneider, B. and Singh, K.: Solving Optimization Problems by Parallel Recombinative Simulated Annealing on a Parallel Computer – An Application to Standard Cell Placement in VLSI Design, *IEEE Trans. Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol.28, No.3, pp.454–461 (1998).
15) 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).
16) 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関して，情報処理学会 DA シンポジウム'94, pp.223–228 (1994).
17) 小西健三，屋舗正史，瀧 和男：温度並列 SA 法による巡回セールスマン問題の解法，*Parallel Computing Workshop'96*, pp.P2-R-1–8 (1996).
18) Kimura, K. and Taki, K.: Time-homogeneous Parallel Annealing Algorithm, *13th IMACS World Congress of Computation and Applied Mathematics* (1991).
19) Hirosawa, M. and Ishikawa, M.: Temperature Parallel Simulated Annealing – Application to Biological Problems, *ICOT Technical Memorandum*, TM-1147 (1992).
20) Szu, H. and Hartley, R.: Fast Simulated Annealing, *Physics Letters A*, Vol.122, No.3, 4, pp.157–162 (1987).
21) Corana, A., Marchesi, M., Martini, C. and Ridella, S.: Minimizing Multimodal Functions of Continuous Variables with the "Simulated Annealing" Algorithm, *ACM Trans. Mathematical Software*, Vol.13, No.3, pp.262–280 (1987).

22) Rosen, B.: functional Optimization based on Advance Simulated Annealing, *IEEE Workshop on Physics and Computation*, PhysComp 92, Dallas, Texas, pp.289–293 (1992).
23) Papadimitriou, C.H. and Steiglitz, K.: *Combinatorial Optimization – Algorithms and Complexity*, p.460, Prentice-Hall (1982).
24) 情報処理学会（編）：情報処理ハンドブック，オーム社，p.119 (1997).
25) Whitley, D., Mathias, K., Rana, S. and Dzubera, J.: Evaluating Evolutionary Algorithms, *Artificial Intelligence*, Vol.85, pp.245–276 (1996).
26) 三木光範：並列分散最適化のためのアルゴリズム—資源の追加と削減に基づく方法，情報処理学会並列処理シンポジウム JSPP'98, pp.263–270 (1998).

(平成 11 年 8 月 31 日受付)
(平成 12 年 2 月 4 日採録)

三木 光範（正会員）

1950 年生．1978 年大阪市立大学大学院工学研究科博士課程修了，工学博士．大阪市立工業研究所研究員，金沢工業大学助教授を経て 1987 年大阪府立大学工学部航空宇宙工学科助教授，1994 年同志社大学工学部教授．進化的計算手法とその並列化および知的なシステムの設計に関する研究に従事．著書は「工学問題を解決する適応化・知能化・最適化法」（技報堂出版）等．IEEE，米国航空宇宙学会，人工知能学会，システム制御情報学会，日本機械学会，計算工学会，日本航空宇宙学会等会員．超並列計算研究会代表．

廣安 知之（正会員）

1966 年生．1997 年早稲田大学理工学研究科後期博士課程修了．同年早稲田大学理工学部助手．1998 年より同志社大学工学部助手．創発的計算，進化的計算，最適設計，並列処理等の研究に従事．IEEE，電気情報通信学会，計測自動制御学会，日本機械学会，超並列計算研究会，日本計算工学会各会員．

笠井 誠之

1975 年生．1998 年同志社大学工学部知識工学科卒業．2000 年同志社大学大学院工学研究科修士課程修了．同年，ソニー（株）入社．並列処理，並列最適化アルゴリズム等に興味を持つ．