

適応的近傍を持つ温度並列シミュレーテッドアニーリング

三木 光範^{*} , 廣安 知之^{*} , 笠井 誠之^{**}

E-mail: mmiki@mail.doshisha.ac.jp

^{*} 同志社大学工学部 ^{**} 現 ソニー株式会社

本論文では、並列計算機を用いて組み合わせ最適化問題を解く場合に有効な並列アルゴリズムの1つである温度並列シミュレーテッドアニーリング (TPSA) を、連続変数最適化問題に応用するために、適応的近傍を持つ TPSA を提案する。この手法は、連続設計変数空間に適用された Corana の SA と TPSA のハイブリッドな手法であり、各プロセス (プロセッサ) が担当する各温度において、受率率が一定となるような解摂動を行う。この仕組みによって、各プロセッサにおいて関数の景観に適応した探索が行われ、温度並列 SA の探索性能を大幅に向上させる。並列計算機として PC クラスタを用い、3つの代表的なテスト関数に適用した結果、適応的近傍のメカニズムは逐次 SA では効果が少なかったが、TPSA では非常に有効に働くことがわかった。

Temperature Parallel Simulated Annealing with Adaptive Neighborhood

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

^{*}Doshisha University ^{**}Presently with Sony Corporation

In this paper, a Temperature Parallel Simulated Annealing with Adaptive Neighborhood (TPSA/AN) for continuous optimization problems is introduced. TPSA/AN is based on the temperature parallel simulated annealing (TPSA), which is suitable for parallel processing, and the SA that Corana developed for continuous optimization problems. The moves in TPSA/AN are adjusted to have equal acceptance rates. Because of this mechanism, the proposed method provides global search in the processors of parallel computers for high temperatures and local search in the processors for low temperatures. Therefore, all the processors are used for searching very efficiently. The TPSA/AN is evaluated for the standard test functions, and it is found that adopting the adaptive neighborhood range increases the searching ability of TPSA remarkably.

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) 繰り返し解析の単純な並列化、(2) 解析コード部分の並列化、および (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]。

近年の最適化アルゴリズムの並列化において最も注目すべきことは、並列処理によって計算時間の短縮が行われるだけでなく、解の精度が向上するという点である。たとえば、遺伝的アルゴリズム (GA) の並列化に関する研究では、並列計算機の各プロセッサに割り当てられた分割母集団において独自の良好な個体が生み出されることによって、良好な解が得られている [2][3]。また、シミュレーテッドアニーリング (SA) の並列化に関する研

究においては、逐次型 SA よりも良好な解が得られる温度並列 SA と呼ばれる並列アルゴリズムが提案されている [4]。

シミュレーテッドアニーリング (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 にはほとんど任意の非線形性を持ったコスト関数を処理できるという大きな利点があるが、2つの欠点が存在する [8]。1つは、解探索のふるまいを制御する冷却スケジュールの決定が困難であるということであり、もう1つは解を得るまでの計算時間が長いことである。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] ことが報告されている。

そこで SA の計算時間を短縮するための並列処理は、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 [10] . 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 [11][12][13] . 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常に残る .

これに対して、温度並列 SA [4] 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極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 [14]、巡回セールスマン問題 [15]、グラフ分割問題 [16]、そして生物学的問題 [17] 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わかっている . 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 .

このような現状に対して、著者らは連続空間に温度並列 SA を応用する新しい手法を提案した [18] . ここでは、連続空間上の解摂動のための近傍として正規分布を用い、その正規分布の分散を温度パラメータに割り当てる方法で、連続空間における解探索を実現している . このことは、従来の温度並列 SA に Boltzmann アニーリング [19] のメカニズムを導入したことになる . その結果、連続最適化問題に拡張された温度並列 SA は、同じ近傍を持つ逐次 SA と比較して高い性能を持つことがわかった . しかしながら、摂動する近傍の大きさが解探索の精度に与える影響については議論しなかった . その理由は、温度パラメータと正規分布の広がりに関係付けたことにより、解の精度は温度のみで規定されることになったからである .

本研究では、解の摂動近傍を温度とエネルギー関数の景観をもとに適応的に決めることを考える . この考え方は、逐次 SA に対して Corana が提唱した手法 [20]、すなわち受率率をもとにする摂動近傍の調節メカニズムを温度並列 SA に導入することで可能となる . このアルゴリズムを適応的近傍を持つ温度並列 SA (TPSA/AN: Temperature Parallel Simulated Annealing with Adaptive Neighborhood) と呼ぶ .

そして、摂動のための近傍レンジ (近傍の大きさ) の違いが逐次 SA と温度並列 SA の解探索のふるまいに異なる影響を与えることを示し、代表的な数学関数最小化問題を用いて本提案手法の評価を行う . また、著者らが前報で提案した手法 [18]

との比較も行う .

2 温度並列 SA (TPSA)

温度並列 SA [4] 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) 時間的に一樣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 .

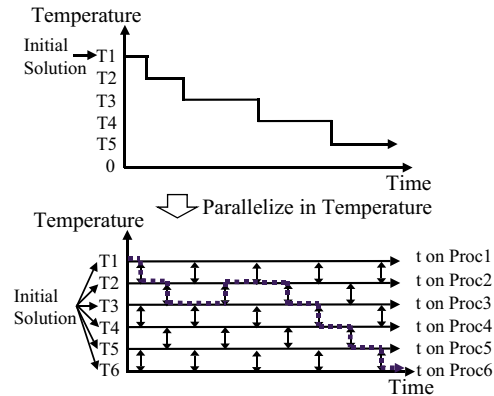


図 1: 逐次 SA と温度並列 SA

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 .

温度並列 SA における隣接温度での解の交換は、隣接温度間の温度差とエネルギー差を用いて確率的に行う . これによって、低温部にエネルギーの低い解が確率的に集まる . 一方、各一定温度における SA は通常の方法で行う . なお、温度並列 SA の詳細なアルゴリズムについては、文献 [4] を参照されたい .

3 適応的近傍を持つ温度並列シミュレートッドアニーリング (TPSA/AN)

3.1 連続最適化問題のための SA

連続空間で SA を実行する場合、解摂動のための近傍レンジ (近傍の大きさ) を考えることが重要となる . 解摂動のための近傍レンジを一定とした場合、探索空間の大きさに対して適切な大きさにしなければならない . 近傍レンジが大きすぎる場合は、得られる解の精度は良好なものとはなりにく

い。また、近傍レンジが小さすぎる場合は、探索の進行が遅すぎる [20]。

この問題に対して、温度の高低が直接に近傍レンジを決定するアルゴリズムがいくつか考えられている。それらは、正規分布型の近傍構造を持ち、温度の平方根をその標準偏差としている Boltzmann アニーリング [19]、コーシー分布の FSA [21] などである。これらのアルゴリズムでは、温度が高ければ近傍レンジが大きく、温度が低ければ近傍レンジが小さくなる。このメカニズムによって、高温では大域探索、低温では局所的な探索を実現し、それによって探索の効率向上と収束の高速化を行っている。しかしながら、これらの方法は近傍レンジと受理確率を決定する温度が密接に関係しているため、目的関数のランドスケープを考慮することができない。

これを解決するため、SA の実行の各局面で解摂動のための近傍レンジを関数のランドスケープに従った適応的なものとするアルゴリズムとして VFSR [22] と Corana の SA [20] が提案されている。前者の方法では、温度が適応的に変化するため、温度が一定である温度並列 SA には、そのままの形で用いることはできない。そのため、ここでは温度並列 SA に後者のアルゴリズムを組み込むことを考える。

3.2 適応的近傍を持つ温度並列 SA (TPSA/AN)

Corana は、近傍レンジ (近傍の大きさ) が大きすぎることにより受理率が低くなりすぎることや、近傍レンジが小さすぎることにより受理率が高くなりすぎることによって、無駄な探索が生じることを指摘した。そして、より効果的な探索を行うために、解摂動に用いる近傍レンジを受理率が 50% になるように適応的に調節する SA のアルゴリズムを提案した [20]。本研究では、その考え方を温度並列 SA に組み込んだ適応的近傍を持つ温度並列 SA (TPSA/AN: Temperature Parallel Simulated Annealing with Adaptive Neighborhood) を提案する。

本手法と従来の温度並列 SA との違いは、近傍レンジを受理率にしたがって適応的に調節する処理を付け加えたことである。近傍レンジの調節は解のある摂動回数毎に行うものとする。

このアルゴリズムにおいて、解摂動は式 (1) で表す一様分布の近傍を考え、現在の各設計変数 x_i から、次状態の各設計変数 x'_i を生成することで行う。

$$x'_i = x_i + rm \quad (1)$$

ここで、 r は $[-1, 1]$ の一様乱数である。また、 m は近傍レンジを決定するパラメータである。このパラメータ m を解のある摂動回数毎に変化させることで、適応的な解摂動が行われることを期待す

る。なお、Corana の SA では各設計変数の近傍レンジを異なるものとしているが、ここでは簡単のために各変数の近傍レンジは同じものとしている。

近傍レンジを調節する処理では、式 (1) のパラメータ m を式 (2) のように受理率 p によって変化する関数 $g(p)$ を用いて決定する。また、この $g(p)$ は式 (3) を用いて決定する。

$$m' = m \times g(p) \quad (2)$$

$$\begin{cases} g(p) = 1 + c \frac{p-0.6}{0.4}, & \text{if } p > 0.6 \\ g(p) = (1 + c \frac{0.4-p}{0.4})^{-1}, & \text{if } p < 0.4 \\ g(p) = 1, & \text{otherwise} \end{cases} \quad (3)$$

ここで p は、近傍レンジを変更する間隔 N の間に解摂動が受理された回数 n から、 $p = n/N$ と計算される。また、 c は調節の度合いを決定するパラメータである。本研究では Corana と同様に $c = 2$ としている。

4 対象問題

提案した TPSA/AN の性能を評価するために 3 つの標準テスト関数を用いる。式 (4) に示す Rastigrin 関数 [23]、式 (5) に示す Griewangk 関数 [23]、および式 (6) に示す Shekel 関数 [20] である。設計変数の数はそれぞれ 2 変数とした。

$$f_R(\mathbf{x}) = (N \times 10) + \left[\sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i)) \right] \quad (4)$$

定義域: $-5.12 < x_i \leq 5.12$,
最適解: $(x_1, x_2) = (0, 0)$,
最適値: $f = 0$

$$f_G(\mathbf{x}) = 1 + \sum_{i=1}^N \left(\frac{x_i^2}{4000} \right) - \prod_{i=1}^N \left(\cos \left(\frac{x_i}{\sqrt{i}} \right) \right) \quad (5)$$

定義域: $-600 < x_i \leq 600$,
最適解: $(x_1, x_2) = (0, 0)$,
最適値: $f = 0$

$$f_S(\mathbf{x}) = - \sum_{j=1}^m \left(\left(\sum_{i=1}^n (x_i - a_{ij})^2 + c_j \right)^{-1} \right) \quad (6)$$

定義域: $0 < x_i \leq 10$,
最適解: $(x_1, x_2) = (4, 4)$,
最適値: $f = -10.3012$

ただし、Shekel 関数においては、 $m=5$ 、 $\mathbf{x} = (x_1, \dots, x_n)^T$ 、 $\mathbf{a}_j = (a_{1j}, \dots, a_{nj})^T$ であり、 a_{ij} と c_j は表 1 に示す行列の各要素である。

表 1: a_{ij} と c_j

i	a_{ij}	c_j
1	4	0.1
2	1	0.2
3	8	0.2
4	6	0.4
5	3	0.4

5 パラメータ設定について

5.1 温度パラメータ

温度並列 SA では、温度パラメータとして最高温度と最低温度をあらかじめ決定しなければならない。ここでは、その決定方法について述べる。

本手法においては、近傍レンジの適応的調節によって、高温時には近傍レンジは大きく、低温時には近傍レンジは小さく調節される。そこで、最高温度で調節される近傍レンジによって問題空間を大域的に探索することが可能であれば、その最高温度は問題に適したものであると考えられる。具体的には、予備実験によって、最高温度でのアニーリング時の解探索領域と問題空間の定義域との比較を行い、その差が最小になるように最高温度を決める。なぜなら、最高温度がこれより高ければ、大域的探索に支障はないが、解摂動が定義域をはみ出す確率が高くなり、計算効率が下がる。ところが、最高温度がこれより低ければ、定義域に探索されない領域が生じる。

一方、最低温度では解の精度を向上させるための探索が行われなければならない。そのため、最低温度での近傍レンジは十分に小さなものが望まれる。ただし、必要とする精度以上に小さな近傍レンジである必要もない。そこで、予備実験によって、求める精度と同じオーダーの近傍レンジが得られる温度を最低温度とする。

5.2 各パラメータ

計算に用いた 3 つの対象問題のために設定したパラメータを、表 2, 表 3, および表 4 に示す。最高温度、最低温度は前節に示す方法で決定し、逐次 SA と温度並列 SA でそれぞれ同様のものを用いる。また、逐次 SA におけるクーリングで用いる温度数は温度並列 SA の温度数と同じ温度を用いた。逐次 SA のクーリング率は最高温度、最低温度、および温度数を決定すれば自動的に決定される。また、繰り返し数は逐次 SA の総計算量と、温度並列 SA における各温度の計算量の総和が等しくなるように設定した。他のパラメータについては、経験的に決定した。

表 2: Rastrigin 関数のためのパラメータ

Algorithms	SA	TPSA
Number of Processes	1	32
Max.(Initial) temperature	10	
Min.(Final) temperature	0.01	
Number of iterations	10240 × 32	10240
Markov length	10240	-
Cooling rate	0.80025	-
Exchange interval	-	32
Neighborhood adjust interval	8	

表 3: Griewangk 関数のためのパラメータ

Algorithms	SA	TPSA
Number of Processes	1	32
Max.(Initial) temperature	20	
Min.(Final) temperature	0.001	
Number of iterations	30720 × 32	30720
Markov length	30720	-
Cooling rate	0.726	-
Exchange interval	-	32
Neighborhood adjust interval	8	

表 4: Shekel 関数のためのパラメータ

Algorithms	SA	TPSA
Number of Processes	1	16
Max.(Initial) temperature	0.8	
Min.(Final) temperature	0.001	
Number of iterations	80 × 16	80
Markov length	80	-
Cooling rate	0.640414	-
Exchange interval	-	4
Neighborhood adjust interval	8	

6 並列計算機への実装

本研究では 8 プロセッサの計算用ノード (Pentium 233MHz×8) と 1 プロセッサの NFS サーバ用ノード (Pentium 266MHz) からなる PC クラスタ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。

アルゴリズムの実装は、一つの温度での SA を一つのプロセスに割り当てることで行う。したがって、温度数が 8 までのときには 1 つのプロセッサに 1 つの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プロセスが実行される。温度数が 32 の場合は一つのプロセッサで 4 プロセスが実行されている。

7 実験結果および考察

7.1 Rastrigin 関数

近傍レンジの設定が、逐次 SA および温度並列 SA に与える影響を見るために 4 種類のアルゴリズムで比較を行う。その 4 つとは、適応的近傍を持つ温度並列 SA (TPSA/AN), 適応的近傍を持つ逐次 SA (SA/AN), 固定近傍レンジを持つ温度並列 SA (TPSA/FN: TPSA with Fixed

Neighborhood), および固定近傍レンジを持つ逐次 SA(SA/FN)である。

Rastrigin 関数を解いたときの結果を図 2 に示す。縦軸がエネルギーであり、横軸が固定近傍レンジの各値を列挙したものを示している。固定近傍レンジでは、全温度で同じ近傍レンジを使用し、それぞれを固定したままでアニーリングを行う。それぞれの結果は、10 回試行の中央値で表した。中央値を用いた理由は、複数の局所最適解が存在し、しかもそれらの関数値に大きな差がある場合には、平均値は最悪値に大きく影響され、正しい評価とはならないからである。

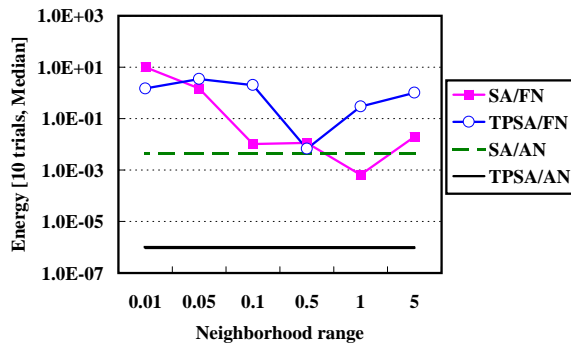


図 2: Rastrigin 関数の結果

この結果において、固定近傍レンジのアルゴリズムについて見ると、逐次 SA では近傍のレンジが 1 のときに、また、温度並列 SA では近傍のレンジが 0.5 のときに、最も良い解が得られていることがわかる。これによって、近傍レンジの設定は探索性能に影響を与え、その中でも適切な値が存在するということがわかる。

さらに、固定近傍レンジのアルゴリズムにおいては、温度並列 SA よりも逐次 SA の方が全体的に良好な解が得られるということがわかる。これは、SA を並列化すると、各マルコフ連鎖の長さが探索に十分な長さでなくなるためであり、したがって SA は並列化の困難なアルゴリズムであるといえる。このことは文献 [24] でも指摘されている。

しかしながら、ここで提案した適応的な近傍調節を行う TPSA/AN は、最も適切な近傍レンジを与えた逐次 SA と比べても極めて良好な解を得ることができる。一方、適応的解振動を行う逐次 SA/AN は、最も適切な近傍レンジに設定したものよりも解の品質が劣化している。すなわち、適応的近傍のメカニズムは、逐次 SA よりも、温度並列 SA に

おいて極めて有効に機能しているといえる。

7.2 Griewangk 関数

図 3 は、図 2 と同様に Griewangk 関数を解いた結果である。この結果より、この問題に適切な近傍レンジの値は逐次 SA と TPSA 共に 5 であることがわかる。さらに、この問題においても、固定近傍レンジを与えた場合、ほとんどの近傍レンジの設定において温度並列 SA の性能が逐次 SA のそれよりも劣化している。しかし、前述の Rastrigin 関数の時と同様に、TPSA/AN によって得られた解の精度は最も良好な値になっている。そして、この問題においても SA/AN の性能はそれほど良好ではない。

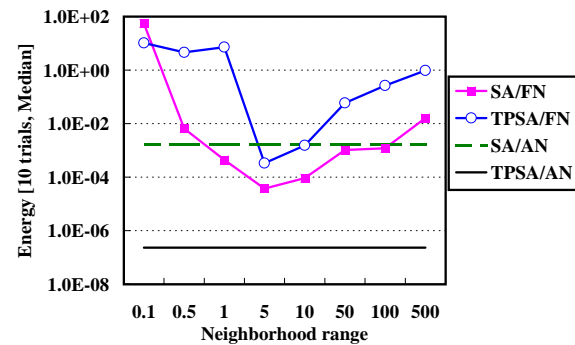


図 3: Griewangk 関数の結果

7.3 Shekel 関数

図 4 は、Shekel 関数を解いた結果である。この問題においても、TPSA/AN は全ての設定の中で最も良好な解を得ることがわかる。一方、この問題においてのみ、SA/FN は平均的に TPSA/FN より良好ではなかった。これは、他の 2 つの関数が非常に多くの局所解を持っていたのに対し、この関数は局所解の数が少なかったためであると考えられる。すなわち、温度並列 SA において、並列化は局所解の数を超えており、このような条件では温度並列 SA は有効な性能を示すと思われる。

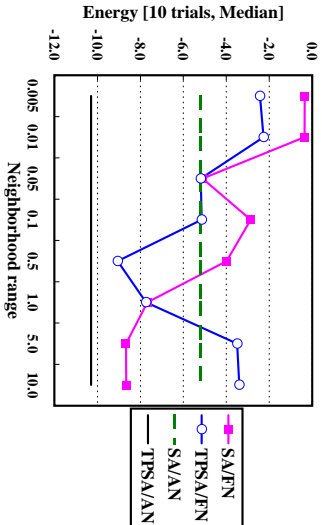


図 4: Shekel 関数の結果

これまでの結果により，TPPSA/AN が良好な解探索性能を持つことと，TPPSA/AN が解の精度に影響を与える近傍レンジの設定を不要にするアルゴリズムであることがわかった．

7.4 TPPSA/FN の性能について

これまでの結果によって，アニーリング回数を固定し，しかも近傍レンジが固定された条件では，逐次 SA が温度並列 SA よりも良好な結果を得る場合が多いことがわかる．これは次のように考えることができる．

SA の並列化は，SA が持つ一連のマルコフ連鎖をプロセッサ数あるいは並列プロセッサ数に分割して速度向上を図るというものである．すなわち，温度並列 SA ならば 1 つの温度がたどることができるマルコフ連鎖は，逐次 SA の全マルコフ連鎖長を温度数で割った長さとなる．これは，各温度間を移動しているある 1 つの解に着目すると，その解が逐次 SA の全マルコフ連鎖長を温度数で割った回数の摂動しか行えないことを意味する．この場合，当然，解の精度が劣化すると考えられる．

7.5 適応的近傍の効果

TPPSA/FN では良好な結果が得られないにも関わらず，TPPSA/AN において良好な結果が得られる理由は，図 5 を見ることでより明らかになる．

図 5 は，対象問題を前述の Rastrigin 関数としたときの，いくつかの固定近傍レンジと適応的近傍レンジの温度並列 SA において，最終的に最低温度に到達した解のエネルギーの履歴を示している．横軸はアニーリングステップであり，縦軸はエネルギーを示す．相対的に小さな近傍 (0.1) を持った解は，最適化の初期の段階で解の改善が行われず，局所解に到達している．また，相対的に大きな近傍 (5.0) を持った解は，最適化の初期の段階で比較

的良好な値に到達しているが，後半において改善されなくなっている．これは大きすぎる近傍では精度の高い摂動ができないことによるものである．一方，中程度の近傍 (0.5) を持った解は，良好な解に収束している．

ところが，適応的近傍 (adaptive) を持った解は，その適応的な近傍調節によって最も良好な解に到達している．特に 5000 ステップ目と 6500 ステップ目あたりで大域的な探索による局所解からの脱出が行われており，一方，6000 ステップ目と 7000 ステップ目以降には局所的な解への収束が加速されていることがわかる．このように適応的近傍持つ温度並列 SA (TPPSA/AN) は，大域的探索能力と局所解への収束という相反する利点をあわせ持つことがわかる．ただし，これは温度並列 SA のみ当てはまることであり，温度の上昇メカニズムを持たない逐次 SA には当てはまらない．すなわち，適応的近傍のメカニズムは，温度並列 SA に組み込んで始めてその効果が発揮されるものと考えられることができる．

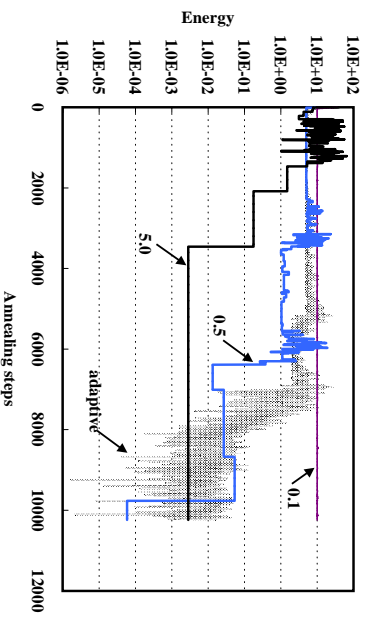


図 5: 最低温度に到達したある解のエネルギー履歴

8 正規分布を近傍に持つ温度並列 SA との比較

著者らは前報で，連続変数を持つ最適化問題に対する温度並列 SA の拡張について議論した [18]．ここでは，正規分布を解摂動のための近傍として用いる温度並列 SA を提案している．ここでは，そのアルゴリズムと本手法の比較を行う．対象問題は前節でも扱った Rastrigin 関数であり，5 次元のものとする．本論文で提案する手法の各パラメータは，表 5 に示すものにし，また正規分布を近傍として用いる温度並列 SA (TPPSA/NN: TPPSA with

Normal distribution Neighborhood) の各パラメータは、前報 [18] と同様の方法で決定する。

表 5: TPSA/AN と TPSA/NN のパラメータ

Algorithms	TPSA/AN	TPSA/NN
Number of Processes	128	
Max.(Initial) temperature	20	6.5536
Min.(Final) temperature	0.0005	0.0001
Number of iterations	102400	
Exchange interval	32	
Neighborhood adjust interval	8	-
Scaling factor[18]	-	0.068413

図 6 の左のグラフにそれぞれのアルゴリズムによって得られた最適解のエネルギーを示す。図中の値は、10 回試行で得られたエネルギーの中央値を示している。これを見ると本論文で提案する手法の方が解の精度が良好であることがわかる。解の精度が良好となる原因は、正規分布が無窮遠の摂動距離を持つものに対して、一様分布は摂動距離に上限があるからと考えられる。すなわち、無窮遠の移動距離を持つ近傍を用いた場合は、摂動近傍が十分に小さなものでないため最適化処理の終盤における収束が緩慢になると考えられる。一方、一様分布は摂動距離に上限があり、また、その上限は本手法の適応的なメカニズムによって、適切な値となると考えられる。

図 6 の右のグラフは計算時間を比較したものである。それぞれは 10 回試行の平均値を示している。計算時間についても、TPSA/AN の方が良好な値となっている。これは、TPSA/NN では最高温度において、探索領域外への摂動が多いためである。

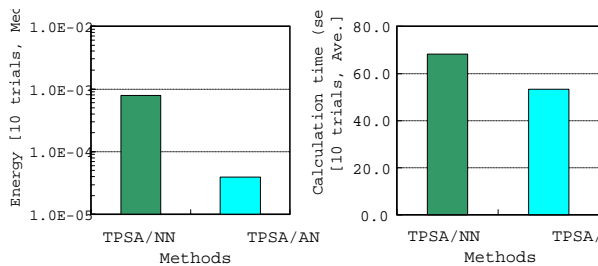


図 6: TPSA/NN と TPSA/AN の比較

9 並列処理による計算時間の短縮

図 7 は、表 2 の設定で実行したときの SA/AN に対する TPSA/AN の速度向上率 (10 回試行の平均値) を表している。対象問題は 2 次元の Rastrigin 関数である。TPSA/AN(case1) は、エネルギー関数として、Rastrigin 関数をそのまま用いたもので

あり、8CPU で約 1.8 倍の速度向上しか得られていない。これは、計算の粒度が小さすぎ、解交換に関するプロセッサ間通信のオーバーヘッドが大きいためである。

一方、TPSA/AN(case2) は、計算粒度を高めるため関数評価において 100 倍の負荷をかけたものである。この時、8CPU で約 6 倍の速度向上が得られた。しかしながら、これ以上計算粒度を大きくしても、この速度向上はほとんど変化しなかった。この原因については、現在のところ不明である。

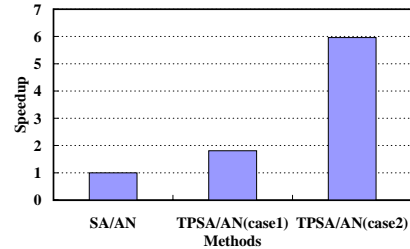


図 7: 並列計算による速度向上

10 おわりに

本研究では、温度並列 SA を連続最適化問題に適用するための拡張アルゴリズムとして適応的近傍を持つ温度並列 SA を (TPSA/AN) を提案した。得られた結論は、以下の通りである。

1) 連続空間上の探索において、近傍レンジの設定は逐次 SA および温度並列 SA の双方にとって非常に重要であり、その設定は解の精度に大きな影響を与える。

2) 最適な近傍レンジに固定した逐次 SA(SA/FN) は、最適な近傍レンジに固定した温度並列 SA(TPSA/FN) の解探索性能を上回ることが多い。

3) 適応的近傍を持つ温度並列 SA(TPSA/AN) は、3 つの代表的なテスト関数において、固定近傍を持つ逐次 SA(SA/FN)、固定近傍を持つ温度並列 SA(TPSA/FN)、および適応的近傍を持つ逐次 SA(SA/FN) よりも極めて良好な性能を示した。

4) 連続最適化問題において、近傍レンジは解探索性能に影響を与えるものであるが、TPSA/AN はそれらの設定を不要とすることができる。

5)TPSA/ANは，大域的探索能力と，局所解への収束の加速化という相反する利点を，共存させる手法であるということがわかった．

6)TPSA/ANは8プロセッサのPCクラスタを用いることで，SA/ANに比べて約6倍の速度向上を得ることができた．

以上の結論より，TPSA/ANは，連続最適化問題に対する温度並列SAの有効な拡張アルゴリズムであるといえる．

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] Tanese, R.: Distributed genetic algorithms, *Proc. 3rd International Conference on Genetic Algorithms*, pp. P.434-439 (1989).
- [3] Belding, T. C.: The distributed genetic algorithm revisited, *Proc. 6th International Conference on Genetic Algorithms*, pp. P.114-121 (1995).
- [4] 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評価, *情報処理学会論文誌*, Vol. 36, No. 4, pp. 797-807 (1995).
- [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 [7] の p. 54.
- [10] Holmqvist, K., Migdalas, A. and Pardalos, P. M.: Parallelized Heuristics for Combinatorial Search, in *Parallel Computing in Optimization*, Migdalas, A. et al. eds, Kluwer Academic Publishers, p. 269 (1997).
- [11] Chen, H. and Flann, N. S.: Parallel Simulated Annealing and Genetic Algorithms: a Space of Hybrid Methods, in *Parallel Problem Solving from Nature*, Davidor, Y. et al. eds., Springer-Verlag, pp. 428-438 (1994).
- [12] Yong, L., Lishan, K. and Evans, D. J.: The Annealing Evolution Algorithm as Function Optimizer, *Parallel Computing*, Vol. 21, pp. 389-400 (1995).
- [13] 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. on Systems, Man, and Cybernetics - Part B: Cybernetics*, Vol. 28, No. 3, pp. 454-461 (1998).
- [14] 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).
- [15] 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).
- [16] Kimura, K. and Taki, K.: Time-homogeneous Parallel Annealing Algorithm, *The 13th IMACS World Congress of Computation and Applied Mathematics* (1991).
- [17] Hirosawa, M. and Ishikawa, M.: Temperature Parallel Simulated Annealing - Application to Biological Problems, *ICOT Technical Memorandum TM-1147* (1992).
- [18] 三木光範, 廣安知之, 笠井誠之: 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, *情報処理学会並列処理シンポジウム JSP'99*, pp. 135-142 (1999).
- [19] Rosen, B.: functional Optimization based on Advance Simulated Annealing, *IEEE Workshop on Physics and Computation, PhysComp 92* (Dallas, Texas), pp. 289-293 (1992).
- [20] Corana, A., Marchesi, M., Martini, C. and Ridella, S.: Minimizing Multimodal Functions of Continuous Variables with the "Simulated Annealing" Algorithm, *ACM Trans. on Mathematical Software*, Vol. 13, No. 3, pp. 262-280 (1987).
- [21] Szu, H. and Hartley, R.: Fast Simulated Annealing, *Physics Letters A*, Vol. 122, No. 3,4, pp. 157-162 (1987).
- [22] Ingber, L.: Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison, *Mathematical and Computer Modeling*, 16(11), pp. 87-100 (1992).
- [23] Whitley, D., Mathias, K., Rana, S. and Dzubera, J.: Evaluating Evolutionary Algorithms, *Artificial Intelligence*, Vol. 85, pp. 245-276 (1996).
- [24] Boissin, N. and Lutton, J. L.: A Parallel Simulated Annealing Algorithm, *Parallel Computing*, Vol.19, pp. 859-872 (1993).