

## 問題に適応する摂動近傍を持つ温度並列シミュレーテッドアニーリング

笠井 誠之, 三木 光範, 廣安 知之  
同志社大学 工学部 知識工学科

本研究では、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、問題に適応する摂動近傍を持つ 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、各プロセッサの計算資源を有効に利用することができる。並列計算機として PC クラスタを用い、テスト関数として Rastrigin 関数を使用した結果、適応的な解摂動を行うことによって温度並列 SA の求解性能が大幅に向上した。

### Temperature parallel simulated annealing with adaptive neighborhood

Masayuki KASAI, Mitsunori MIKI, Tomoyuki HIROYASU  
Knowledge Engineering Dept., Doshisha University, Kyoto, Japan

In this study, a temperature parallel simulated annealing with adaptive neighborhood (TPSA/AN) for continuous optimization problems is proposed. 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 in TPSA/AN. Therefore, all the processors are used for searching very efficiently. The TPSA/AN is evaluated for a standard test function, and it is found that adopting the adaptive neighborhood range increases the searching ability of TPSA.

#### 1 はじめに

本研究では、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)[1]を、連続変数最適化問題に応用するために、問題に適応する摂動近傍を持つ温度並列シミュレーテッドアニーリング(Temperature Parallel Simulated Annealing with Adaptive Neighborhood: TPSA/AN)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[2]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、各温度プロセスにおいて関数の景観に適応した探索が行われ、各プロセスが計算資源を無駄にしない探索を行うと考えられる。本論文では、代表的な数学関数最小化問題に本手法を適用した結果を報告し、その有効性を検証する。

#### 2 問題に適応する摂動近傍を持つ温度並列シミュレーテッドアニーリング(TPSA/AN)

Corana の提案した SA[2]は、受率率が低すぎたり、受率率が高すぎることによって無駄な探索が生じることを防ぐために、解摂動に用いる近傍の範囲を受率率が 50%になるように適応的に調節するアルゴリズムである。本研究では、その考えを温度並列 SA に組み込んだ問題に適応する摂動近傍を持つ温度並列 SA(TPSA/AN)を提案する。そのアルゴリズムをFig. 1に示す。従来の逐次 SA との違いは、温度並列 SA の特徴である解交換フェーズをクーリングの代わりに付け加えたことと、近傍の範囲を適応的に調節するフェーズを付け加えたことである。近傍の範囲の調節は解の推移  $N$  回毎、解交換は推移  $L$  回毎に行うものとする。なお、解摂動の受率は従来の SA[3]で用いられる Metropolis 基準を用いている。また、解の交換方法などの温度並列 SA の詳しいアルゴリズムについては文献[1]を参照されたい。

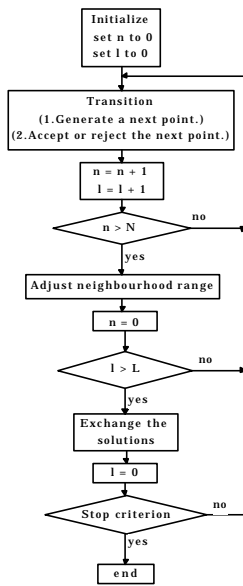


Fig. 1 TPSA/AN のアルゴリズム

ここで提案する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、 $m$  は近傍の範囲を決定するパラメータである。このパラメータ  $m$  を解の推移  $N$  回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の範囲を異なるものとしているが、本研究でのアルゴリズムでは各変数の近傍の範囲は同じものとする。さらにこのパラメータ  $m$  は、式(2)のように受理率  $p$  によって変化する関数  $g(p)$  に従い決定する。また、この  $g(p)$  は式(3)、(4)、および(5)に従って決定する。

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

$$\left\{ \begin{array}{l} \text{もし } p > 0.6 \text{ なら, } g(p) = 1 + c \frac{p - 0.6}{0.4} \quad (3) \\ \text{もし } p < 0.4 \text{ なら, } g(p) = \left( 1 + c \frac{0.4 - p}{0.4} \right)^{-1} \quad (4) \\ \text{それ以外なら, } g(p) = 1 \quad (5) \end{array} \right.$$

ここで  $p$  は、近傍の範囲を変更する間隔  $N$  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。本研究では経験的に  $c = 2$  とした。

### 3 TPSA/AN における温度パラメータの決定

温度並列 SA では、温度パラメータとして最高温度プロセスの温度と最低温度プロセスの温度をあらかじめ決定しなければならない。そこで以下のように決定した。

まず、前章で述べた近傍の範囲の適応的調節を用いて、高温時には近傍の範囲は大きく、低温時には近傍の範囲は小さく調節される。そこで、最高温度で調節される近傍の範囲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。一方、最小温度時に調節される近傍の範囲が最終的に得ようとする解の精度のオーダーと等しくなっていれば、その温度が最小温度としては適切なものであると考えられる。以上を考慮して、今回行った最高温度と最低温度の具体的な決定プロセスを次に述べる。

最高温度時に要求されることは、そのときの解摂動が問題空間全域に渡っているということである。そこで予備実験として、ある回数のアニーリングを最高温度で行ったときの設計変数の定義域を与える各境界線(面)に最も接近した解と境界線(面)との距離を記録する。そして、それらの値の平均値が問題空間に対して十分に小さな値となっていれば、十分な大域探索が行われているということである。そうでなければ、最高温度の設定が低くすぎるということであり、より高い値に最高温度を設定しなければならない。ただし、高温になりすぎると問題空間の領域外への解摂動が生じる。本手法の解摂動アルゴリズムは、解摂動によって問題空間の領域外に出てしまった解のエネルギー評価は行わず、再度解摂動をやり直すというアルゴリズムであるため、それはそのやり直しの分だけ無駄な乱数発生計算が生じ、計算時間がかかる。そこで、最大温度は大域的な探索が行える温度であり、かつ、計算時間がかからない程度の温度にする必要がある。そのトレードオフから決定する。

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，最低温度での近傍の範囲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の範囲である必要もない．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の範囲が得られる温度を最低温度とする．

#### 4 対象問題

対象問題は，式(6)に示す Rastrigin 関数[4]である．設計変数の数は 2 変数とした．また，この関数の最適解はともに原点であり，そのときの関数値は 0 である．

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

#### 5 実験結果

問題に適應する摂動近傍を持つ温度並列 SA (TPSA/AN) の性能を評価するために，Table. 1 に示すの 2 つのアルゴリズムについて比較を行った．ここに示す SA は逐次 SA，TPSA は温度並列 SA を示している．そして，この表には各アルゴリズムの，並列温度プロセス数，最高温度 (逐次 SA の場合，初期温度)，最低温度 (逐次 SA の場合，最終温度)，アニーリング回数，マルコフ連鎖の長さ，冷却率，解交換周期を示している．温度については，前章で述べた方法により決定し，他のパラメータについては経験的に

に決定した．また，アニーリング回数では，温度並列 SA において全プロセッサが行う総アニーリング回数を，逐次 SA のアニーリング回数としている．そこで，逐次 SA は温度並列 SA の 32 倍の長時間の SA となっている．

Table. 1 各手法のパラメータ

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	0.8	
Exchange interval		32

まず，それぞれのアルゴリズムで Rastrigin 関数を解いたときの結果を Fig. 2 に示す．横軸が固定された近傍の範囲の各値と適應的近傍 (AN: Adaptive Neighborhood) を列挙したものであり，縦軸がエネルギーを示している．それぞれの結果は，10 回試行の結果である．近傍の範囲固定のアルゴリズムについて見ると，逐次 SA では近傍の範囲が 1 のときに，温度並列 SA では近傍の範囲が 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，近傍の範囲の設定は求解性能に影響を与えるということがわかり，その中でも適切な値が存在するということになる．

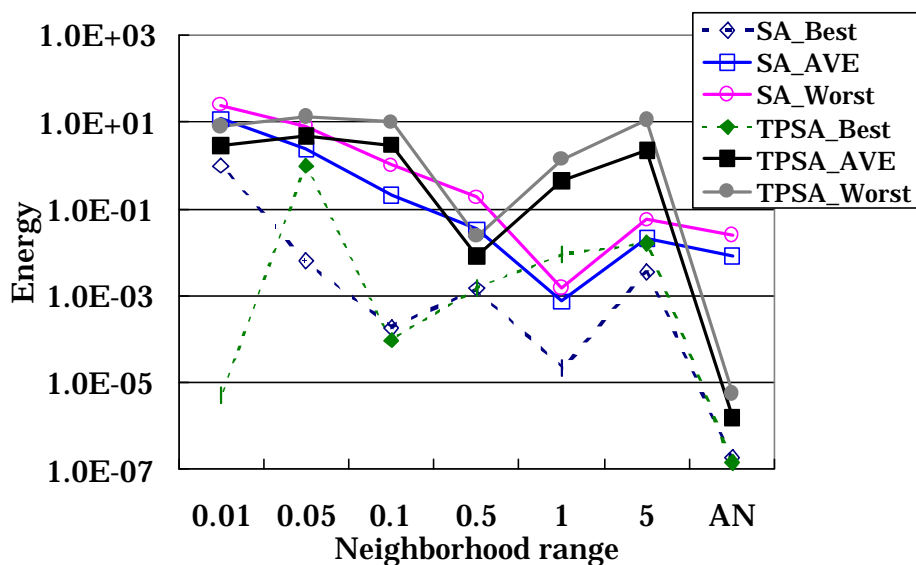


Fig. 2 計算結果 (10 回試行)

さらに、近傍の範囲が固定されたアルゴリズムにおいては、温度並列 SA よりも長時間の逐次 SA の方が平均的に良好な解が得られるということがわかる。これは、SA の並列化は、おのこのマルコフ連鎖の長さが探索に十分な長さでなくなってしまうため、並列化の困難なアルゴリズムであるという報告[5]に従う結果である。ところが、本研究で提案する適応的な近傍調節を行う TPSA/AN は、最も適切な近傍の範囲を与えた逐次 SA と比べても極めて良好な解を得ることができた。また一方で、適応的摂動を行う逐次 SA は、最も適切な近傍の範囲に設定したものよりも解の品質が劣化している。

また、Fig. 3に TPSA/AN のエネルギー履歴を示す。これを見ると、局所解への収束の加速化と、局所解からの脱出という相反する効果の共存が実現できていることがわかる。さらにFig. 4を見ることで、局所解からの脱出と、各局所解への収束の加速化を 2 変数の問題空間上で確認できる。

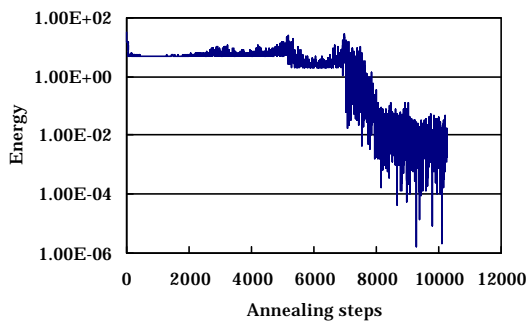


Fig. 3 TPSA/AN の設計解のエネルギー履歴

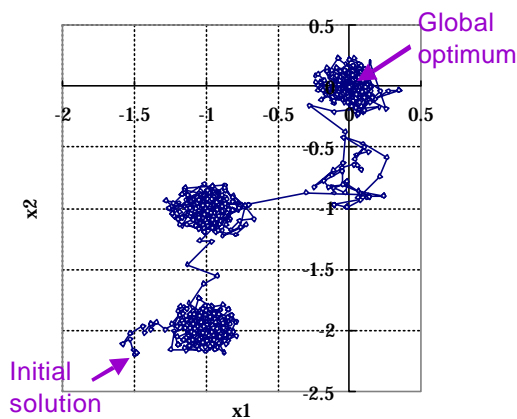


Fig. 4 TPSA/AN の設計解の推移履歴

この局所解からの脱出は、温度並列 SA の高温プロセスの大域探索の効果であり、収束の高速化は適応的な近傍調節の効果であると考えられる。この収束の高速化という効果によって、長時間の逐次 SA の 32 分の 1 という短いアニーリング回数で良好な解を得ることができている。

## 6 おわりに

本研究では、これまで組み合わせ最適化問題にしか用いられてこなかった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ための拡張アルゴリズムとして、問題に適應する摂動近傍を持つ温度並列 SA(TPSA/AN)を提案した。そして、本手法が温度並列 SA の拡張アルゴリズムとして、極めて有効であるということがわかった。

## 7 参考文献

- [1] 小西健三, 瀧 和男, 木村宏一: 温度並列シミュレーテッド・アニーリング方とその評価, 情報処理学会論文誌, Vol.36, No.4, pp.797-807 (1995)
- [2] Corana,A., Marhesi,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)
- [3] Kirkpatrick,S., Gelett Jr. C.D., and Vecchi,M.P.: Optimization by Simulated Annealing, Science, Vol.220, No.4598, pp.671-680(1983)
- [4] Whitley,D., Mathias,K., Rana,S. and Dzubera,J.: Building Better Test Function, Proc. 6th Int. Conf. on Genetic Algorithms, Morgan Kaufmann, pp.239-246 (1995)
- [5] Boissin,N. and Lutton,J.,L.: A Parallel Simulated Annealing Algorithm, Parallel Computing, Vol.19, pp.859-872 (1993)