

連続最適化問題における
近傍並列シミュレーテッドアニーリング

同志社大学工学部知識工学科卒業論文

2003年3月

学籍番号 990064

知的システムデザイン研究室
及川雅隆

目 次

1	序 論	1
2	シミュレーテッドアニーリング (SA)	3
2.1	SA の特徴	3
2.2	SA の基本アルゴリズム	3
2.3	SA の並列化	6
2.4	並列 SA の分類	7
3	連続最適化問題のための SA	9
3.1	連続空間での近傍の定義	9
3.2	連続最適化問題のテスト関数	10
3.3	最適な近傍の設定	10
4	近傍並列 SA	14
4.1	近傍並列 SA の提案	14
4.2	近傍並列 SA のアルゴリズム	14
4.3	近傍並列 SA の探索性能	16
5	複雑な多峰性関数への近傍並列 SA の適用	21
5.1	複雑な多峰性関数	21
5.2	正規分布と温度スケジュールの設定	21
5.3	数値実験	25
6	結 論	30
6.1	結 論	30
6.2	今後の課題	30

1 序 論

近年、我々の社会を構成するシステムは大規模化、複雑化が進み、それにとまって最適なシステムの設計や設定を行なうことが不可欠になっている。最適化問題とは、与えられた制約条件のもとでその評価関数を最大または最小にする最適解を求める問題のことをいい、実世界の多くの場面で見られる。最適化問題は、設計変数が連続である連続最適化問題と設計変数が離散的である組合せ最適化問題に分類できる。前者は、主として目的関数の勾配情報をもとに連続的な探索を行なうことで最適解を求めることが多い。一方、後者は全探索を行なうと組合せ爆発が起こり、最適解を求めることは困難である。よって一般的には、汎用的な確率的アルゴリズムによって準最適解を求めることが多い¹⁾。それらの代表的な手法には、遺伝的アルゴリズム (Genetic Algorithm : GA) やシミュレーテッドアニーリング (Simulated Annealing : SA) などがある。

SA²⁾は、金属の焼き鈍しという物理現象にヒントを得て開発され、組合せ最適化問題に有効なアルゴリズム³⁾である。SA では目的関数をエネルギー関数と呼び、現在の解から決められた範囲内に確率的に次の解候補を生成する。現在の解と比べてエネルギーが減少する場合はその解に遷移し、そうでない場合も、温度というパラメータにより決定される確率でその解に遷移する。このように、改悪方向への遷移も確率的に認めることで大域的最適解を得ることが可能である。SA は主に組合せ最適化問題を解く汎用的近似解法⁴⁾であるが、複雑な連続最適化問題にも広く用いられてきた。たとえばタンパク質の立体構造の予測を系のエネルギー最小化という観点から解く場合⁵⁾には、SA は最も多く利用されている。その場合、設計変数は分子結合における回転角となり、連続最適化問題となる。このように、複雑な連続最適化問題にも SA は有効な解法として用いられている。しかし、SA にもパラメータのチューニングの難しさと、膨大な計算量を要するという欠点が存在する。

SA で重要となるパラメータは、近傍と温度である。組合せ最適化問題では、近傍は解摂動に用いる方法を決定すると一意に定まる。そのため、温度パラメータの調節が重要になる^{2,6-12)}。一方、連続最適化問題に SA を適用する場合、近傍はユークリッド空間内での距離に相当し、任意に決めることが可能である。近傍が大きければエネルギーの変化は大きく、大域探索になる。しかし、収束に要する時間は膨大となり、効率的ではない。これに対して近傍が小さい場合はエネルギーの変化は小さく、収束に要する時間は短縮されるが、得られる解精度を犠牲にする。このように、近傍の設定は探索に要する時間と得られる解精度とのトレードオフの関係にあり、適切な近傍の設定が重要である。一般的に、連続最適化問題では温度を受理処理だけでなく、生成処理にも用い、温度より決定されるある分布によって次状態を生成する¹³⁻¹⁵⁾。つまり、高い温度では大域探索を行い、低い温度では現在の解から比較的近い局所探索を行なう。一方、分布を用いて近傍を決定しない場合は、あるルールを用いて適応的に近傍を調節する^{16,17)}。

SA のもう一つの問題点として膨大な計算量を要することが挙げられる。SA は理論的に大域的最適解に到達することが保証されているが、膨大な探索が必要になる²⁾。そのため、問題の大規模化と複雑化に応じて、並列化による効率的な探索が不可欠である。従来より提案されている SA の並列化手法には、独立的に SA の探索を行なう独立並列 SA、ある周期で同期を取る同期並列 SA、各並列 SA に別々の温度パラメータを与え、適応的に温度調節を行なう温度並列 SA (Temperature Parallel Simulated Annealing : TPSA)^{18,19)} などがある。しかし、これまで連続最適化問題に特化した SA の

並列化の研究は行なわれてこなかった．

本論文では，これら 2 つの問題を解決する手法を提案する．すなわち，連続最適化問題で重要となる近傍に着目し，近傍を並列化し，その調節を行なう近傍並列 SA (Neighborhood Parallel Simulated Annealing : NPSA) を提案する．NPSA は，近傍の並列化と交換によって，近傍の適応的調節が期待できる．本論文ではこの手法を提案したのち，最適な近傍に設定した SA と比較し，さらに複雑な多峰性関数へ適用して NPSA の有効性を検証する．

本論文の流れは，まず第 2 章で SA を概説し，第 3 章で連続最適化問題における SA の適用と近傍について述べる．第 4 章では，近傍を適応的に調節する手法として，NPSA を提案する．最適な近傍に設定した SA と比較することにより，提案手法の近傍の推移について検証する．第 5 章で NPSA を複雑な多峰性関数に適用し，その解探索性能の検証を行い，第 6 章で結論を述べる．

2 シミュレーテッドアニーリング (SA)

2.1 SA の特徴

物質を融解状態になるまで加熱し、徐々に冷却する操作を焼きなまし (アニーリング) という²⁾。この焼きなましにより、エネルギーが最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。この物理プロセスに着想を得て、これを計算機上で模擬することにより最適化問題を解こうとする手法をシミュレーテッドアニーリング (SA)²⁾と呼んでいる。SA は構成が簡単で理解しやすく、適用範囲の広い組み合わせ最適化手法である。

SA には以下のような長所がある。

- 頑強性

多くの最適化解法が局所最適解に捕捉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、解品質が改良する方向のみ探索を進めるのではなく、改悪する方向も選ぶ仕組みによる。

- 汎用性

枠組み自体が極めて汎用にできているので、実に広範囲の問題に適用できる。

- 柔軟性

目的関数 (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。さらに、問題に複雑な境界条件があってもよい。

- 簡便性

アルゴリズムは極めて簡単で誰でも容易に作ることができる。

また、SA には以下の欠点もある。

- 非効率性

最適解を得るのに非常に多くの計算量を要する。この問題を克服するため、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、および並列化して高速化を図る並列化の研究が近年見られる。

- 操作性

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必要がある。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となる。

2.2 SA の基本アルゴリズム

はじめに、SA が適用可能な対象について考える。有限の状態からなる状態空間 Ω があって、各状態 x に対してエネルギー $E(x)$ が定義されているとする。SA では目的関数をエネルギー関数と呼ぶ。

SA は、最小のエネルギーの状態、すなわち真の最適状態 x^* を求めることが目的である。与えられた初期状態から出発して状態を推移させ、最終的には最適状態に行きつくことが期待される。

SA の基本アルゴリズムを Fig. 2.1 に示す。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。温度 T_k で平衡状態に達したら、徐冷処理を行って次の温度 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。十分温度が冷えて終了条件に達すれば、そのときの状態とエネルギーをおののの最適状態、最適値として出力する。

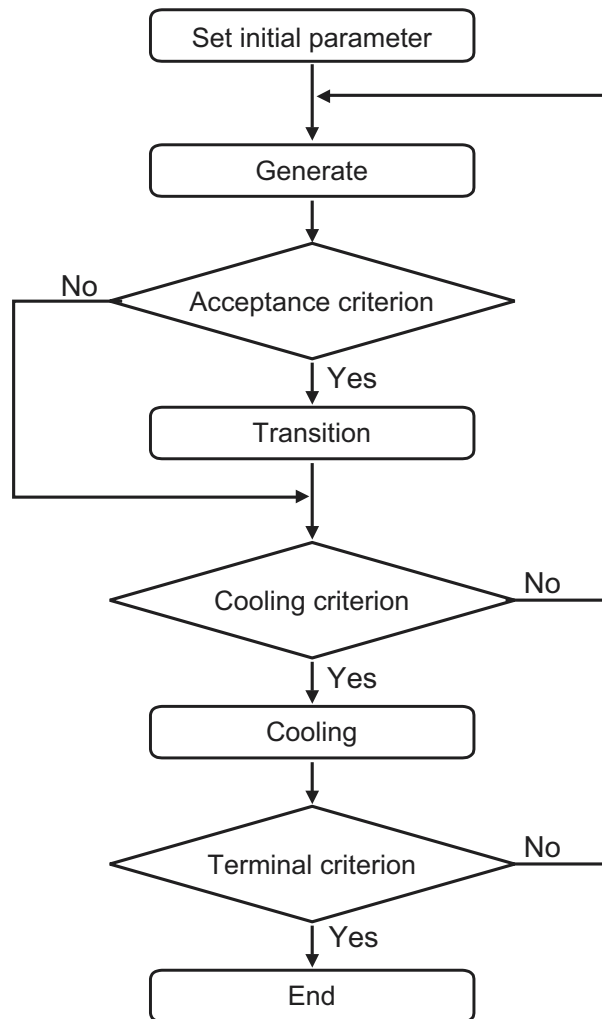


Fig. 2.1 Algorithm of simulated annealing

2.2.1 生成処理 (Generate)

生成処理では、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、状態 x が与えられて状態 x' が生起する確率分布 $G(x, x')$ を用いる。組合せ最適化問題の場合は、状態 x' は状態 x の近傍にあたり、推移に優先性を与えない場合は式 (2.1) のような等確率推移となる。 $n(x)$ は、状態 x の近傍を構成する状態の数を表す。

$$G(x, x') = \frac{1}{n(x)} \quad (2.1)$$

2.2.2 受理判定 (Accept criterion)

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。通常は式 (2.2) の Metropolis の基準³⁾ が採用される。

$$A(E, E', T) = \begin{cases} 1 & \text{if } \Delta E < 0 \\ \exp\left(-\frac{\Delta E}{T}\right) & \text{otherwise} \end{cases} \quad (2.2)$$

温度 T は、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。温度が高い場合は悪い状態への推移確率も大きくなり、反対に温度が低い場合はよい方向に推移することになる。しかし、どんな温度でも悪い方向への推移確率がゼロになるわけではない。

2.2.3 クーリング (Cooling)

クーリングでは、アニーリング (徐冷) 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、式 (2.3) に示す対数型アニーリング以上に急速に冷やしてはならない。

$$T_{k+1} = \frac{T_1}{\log k} \quad (2.3)$$

しかし、式 (2.3) を用いたのではあまりにも計算スピードが遅いため、運用上は真の最適解への収束を犠牲にして、式 (2.4) に示す指数型アニーリングがよく使われる。

$$T_{k+1} = \gamma T_k \quad (0.8 \leq \gamma < 1) \quad (2.4)$$

式 (2.4) に示す指数型アニーリングを適用した場合、温度スケジュールは Fig. 2.2 のようになる。

2.2.4 終了条件²⁰⁾ (Terminal criterion)

終了条件には以下のような実装方法などがある。

- ・ アニーリングを定めた回数だけ繰り返して終了
- ・ 受理がほとんど起こらなくなると終了
- ・ 同じ状態が何度も生成されるようになると終了
- ・ 温度が十分低くなると終了
- ・ エネルギーの変化、またはエネルギー自体が十分小さくなると終了

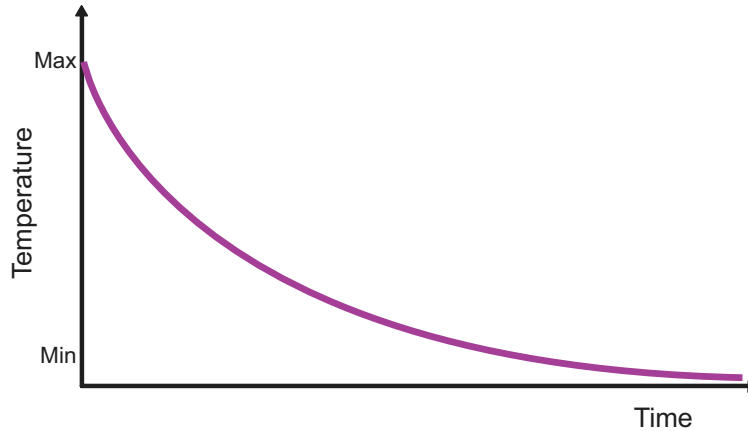


Fig. 2.2 Temperature schedule

2.3 SA の並列化

SA の最大の欠点は非効率性で、最適解を得るのに非常に多くの計算量を要する．この欠点は特に式 (2.4) に示すクーリング方法を用いた場合に顕著である．このような非効率性を克服するためには 2 つのアプローチがある．1 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であり、もう一つは並列処理を用いることによって高速化を図る研究である²¹⁾．

逐次処理の高速化に関する研究では、最適解への収束性を保証した上で、より高速なアニーリング法が望まれる．これには温度スケジュールを変化させ解の収束を早める方法や^{2,6,7)}、生成処理にボルツマン分布以外の分布を用いる方法^{14,15,22,23)}などが考案されており、良好な結果を示している．しかし、アニーリングの高速化によって解品質の劣化がみられ⁴⁾、また様々な対象問題に対する汎用性も検証されていない．

一方、並列計算機の普及に伴い、SA を並列化する研究も行われてきた^{4,24)}．しかし、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ない．SA の基本の処理は (1) 次状態の生成、(2) 次状態のエネルギー計算、(3) 受理の判断、(4) 受理の場合は次状態への推移である．この 4 つの処理から、SA の並列化を考える際に重要なポイントとして、次の 2 点が考えられる²¹⁾．

まず第 1 点は、(1),(2),(3) の処理は並列に実行できるが、(4) の処理を並列に実行すると問題が生じる点である．今、2 並列を考え、共通の状態 x から、一方は状態 y を受理、他方は状態 z を受理したとする．共に元の状態 x を改善する方向の受理とする．このとき、推移を並列に行えば、推移は $x \rightarrow y \rightarrow z$ 、または $x \rightarrow z \rightarrow y$ となるが、 y と z の比較をそれぞれ行うというのは非効率であり、また、結果が改善される保証もない．このような推移を不当な推移と呼び、通常の推移を正当な推移と呼ぶ．

第 2 点は、(4) の処理回数と (1),(2),(3) の処理回数との比 (AG 比) が動的に変化する点である．アニーリング初期は温度が高いため AG 比はほぼ 1 に等しく、後半では温度が低くなり受理が少なくなるため、AG 比はほぼ 0 に等しくなる．この AG 比の動的な変化は、実行の前半と後半で適した並列化の方式が異なることを示唆している．

これらの点を考慮して、これまで研究された並列 SA の分類を次節に示す．

2.4 並列 SA の分類

これまで行われた SA の並列化に関する研究は、次に示す二つのアプローチから考案されている^{25, 26)}。

- **Pseudo-parallelization of SA**

異なるプロセッサで逐次 SA を同時に実行する方法や、逐次 SA のデータ領域を異なるプロセッサに割り当てる方法など、各プロセッサが逐次 SA と同じ処理を同時に行う並列 SA が分類される。これまでの研究で考案された、Data Configuration Partition(DCP)、Parallel independent annealing(PIA)、Parallel Markov Chains(PMC)などの手法がここに分類される。

- **Simulated Annealing Parallel Algorithm (SAPA)**

SAPA では、生成・受理判定・状態推移・クーリングから構成される逐次 SA の処理の各部分を、異なるプロセッサが同時に担当する並列 SA が SAPA に分類される。つまり、逐次 SA の処理を複数のプロセッサで分散する方法である。これまでの研究では、Parallel Markov Chains(PMC)、Parallel Markov Trials(PMT)、Adaptive Parallel Simulated Annealing(APSA)、Speculative Trees、Systolic などの手法がここに分類される。なお PMC は、逐次 SA を独立に実行することと、探索途中で解を配布することから、Pseudo-parallelization of SA と SAPA の両アプローチから考案されたモデルと考えられる。

次にこれまでの研究で考案された並列 SA モデル进行分类する。

Data Configuration Partition(DCP)

DCP は SA が用いるデータ領域を分割し、分割されたデータ領域を各プロセッサに割り当て、それぞれが逐次 SA を行う方法である^{24, 27, 28)}。これらの手法では、各プロセッサが持つデータの整合性を保つために多くの通信が必要となる。

Parallel Independent Annealing(PIA)

PIA には、複数のプロセッサを用いて複数の逐次 SA が同時実行する手法が分類される。これらの手法では、逐次 SA で得られた複数の解のうち、最良のものを最終的な解としている^{4, 27)}。しかし、この PIA で解を得るには逐次 SA と同じ計算時間を要することが欠点といえる。

Parallel Markov Chains(PMC)

PMC は複数の逐次 SA が同時にいくつかのプロセッサで実行される PIA に、解の探索途中で周期的な情報交換を加えたモデルとされている。SA では適当な解がある温度で十分に探索すると、解の生成分布が平衡状態になることが示されている。PMC ではこの特徴を用い、各プロセッサが平衡状態に達しマルコフ連鎖が終了した際に解の交換を行い、各プロセッサは交換した解の中から最良なものを次の初期解として解探索を開始する。解の通信によるオーバーヘッドとマルコフ連鎖の長さの釣り合いがとれた場合、PMC が PIA に比べ高い性能が示すことが報告されている^{4, 24, 28)}。このモデルは温度が高い時に有効であるが、低い場合は局所解に収束する可能性が高いことから性能の保証はされていない。

Parallel Markov Trials(PMT)

このモデルは SAPA に分類され、マスターとなるプロセッサとそれ以外のプロセッサが行う処理が異なる並列 SA のモデルである。マスター以外のプロセッサはマルコフ連鎖の継続、つまり解の探索のみを行い、新しい解が受理されるとマスターにその解の情報を送信する。マスターは解の探索に加え、スレーブから送られる解から良好な解を選択することでより良好な解を探索可能である^{4, 24, 29}。このモデルは温度が低い場合に高い解探索能力を示すとされている。

Adaptive Parallel Simulated Annealing(APSA)

このモデルは PMC と PMT を組み合わせることで、解探索能力の向上を達成している。温度が高い場合は PMC を用い、温度が低くなると PMT を用いている^{4, 28}。このモデルは解の探索途中で PMT を用いるため、SAPA に分類される。

Speculative Trees

このモデルでは現在の解をルートとし、そこから生成される解をノードとした決定木を生成する。そして、それぞれのノードにプロセッサを割り当て、並列に SA を実行する事で計算時間の短縮をはかっている²⁴。このモデルでは通信に大きなオーバーヘッドを有することが示されている。

Systolic

このモデルでは、並列に動作する各プロセッサが、解のマルコフ連鎖から決定する独自の温度で解探索を行う^{24, 30}。このモデルに該当する並列 SA は解の通信モデルなど類似点が多く、それらは共通して計算時間の短縮を実現している。

3 連続最適化問題のためのSA

3.1 連続空間での近傍の定義

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が、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多く用いられている。連続設計変数空間における近傍の設定は、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。一方、物理的に意味のある、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ることによって近傍を定義する。そのような近傍の概念を図示すると、Fig. 3.1 のようになる。Fig. 3.1 は、2 変数の問題空間上で現在の解 x から、近傍内の x' への摂動の様子を示している。この場合、解摂動のための近傍の大きさ (近傍レンジ) を考えることが重要となる。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。近傍レンジが大き過ぎる場合は、得られる解の精度は良好なものとはなりにくい。また、近傍レンジが小さ過ぎる場合は、探索の進行が遅すぎるという問題が生じる。近傍に確率分布を用いる手法としては、以下のようなものが提案されている。

- (1) 近傍レンジを固定、もしくは適応的に変化^{16,17)}させて一様分布を用いる。
- (2) FSA(Fast Simulated Annealing) では、双曲線型の高速アニーリングが実現した。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¹⁴⁾ が用いられる。
- (3) 特に高次元の問題に対しては、際立った効果を発揮すると期待される VFA(Very Fast Annealing) 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立分布¹⁵⁾を用いる。

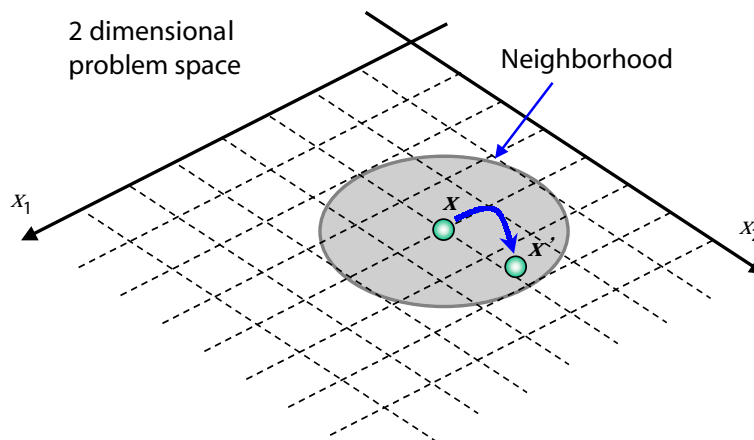


Fig. 3.1 Two dimensional problem space

3.2 連続最適化問題のテスト関数

本節で対象とする，連続最適化問題の標準的なテスト関数について述べる．それらは，式 (3.1) に示す Rastrigin 関数，式 (3.2) に示す Griewank 関数である．ここで， n は設計変数の数を表し，最適解はいずれも 0 である．それぞれの関数の景観と等高線を Fig. 3.2 に示す．等高線は，色が濃いほどエネルギー値が高いことを表し，白い部分が局所最適解である．等高線から分かるように，Rastrigin 関数と Griewank 関数の局所解は格子状に分布している．

$$F_{Rastrigin}(x_i) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \quad (3.1)$$

$$(-5.12 \leq x_i < 5.12)$$

$$F_{Griewank}(x) = 1 + \sum_{i=1}^n \frac{x_i'^2}{4000} - \prod_{i=1}^n \left(\cos\left(\frac{x_i'}{\sqrt{i}}\right) \right) \quad (3.2)$$

$$(x_i = x_i'/100 \mid -5.12 \leq x_i < 5.12)$$

3.3 最適な近傍の設定

連続最適化問題に SA を適用する場合，SA の解探索を効率良く行なう最適な近傍レンジが存在することが予備実験により分かった．ここでは，この最適な近傍を求める実験について述べる．

3.3.1 実験概要

各関数において，最適な解探索を行なう近傍レンジを調べるため，種々の近傍レンジに対して予備実験を行なった．次状態をそれぞれ固定近傍レンジを用いた一様分布によって生成する．

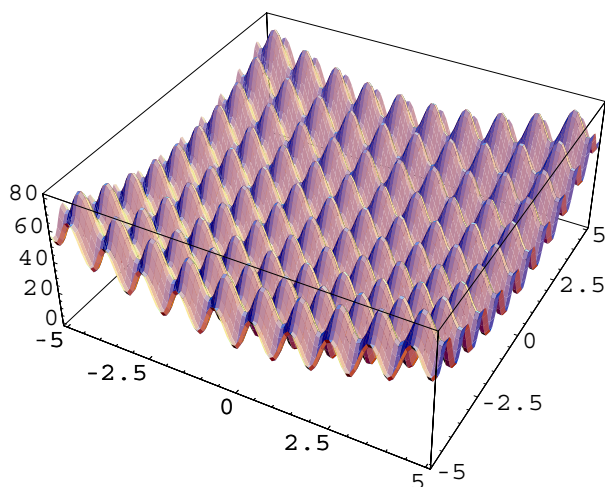
予備実験に用いた近傍は，Rastrigin 関数では最大近傍を設計空間の幅である 5.12，最小近傍をその 1/1000 とし，Griewank 関数では最小近傍を設計空間の 1/10000 とした．最大近傍から最小近傍までを指数的に 100 分割したそれぞれの近傍に対して予備実験を行なう．対象としたのは，2 設計変数と 10 設計変数の各関数である．予備実験に用いたパラメータを Table 3.1 に示す．ただし，クーリングには式 (2.4) の指数型アニーリングを用い，32 ステップでクーリングを行なう．

Table 3.1 Parameters in preliminary experiments

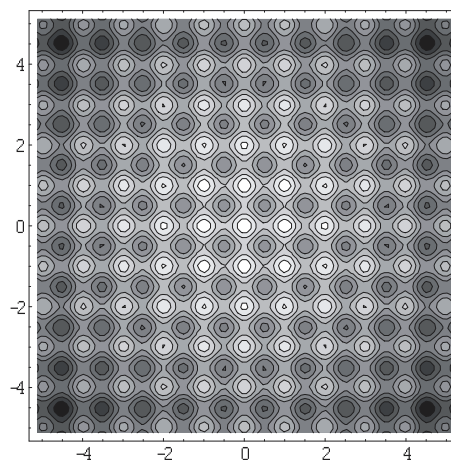
Max temperature	10.0
Minimum temperature	0.01
Number of cooling Steps	32
Cooling cycle	10240
Number of cnnealings	10240 × 32

3.3.2 実験結果

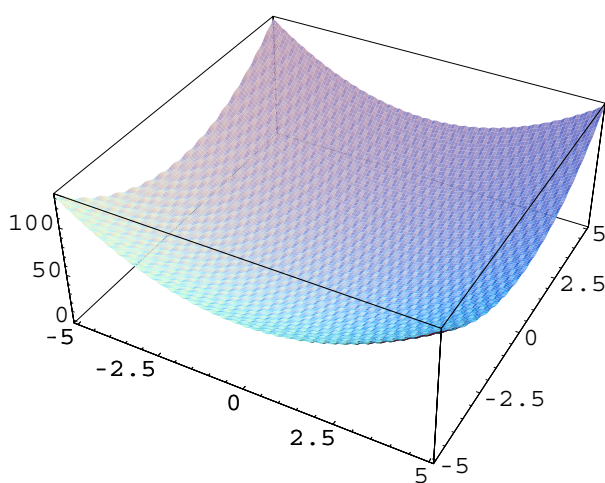
各近傍に対して逐次 SA を 30 試行行い，その中央値の結果を Fig. 3.3 に示す．横軸が近傍レンジ，縦軸がエネルギー値を表し，両軸とも対数表示である．中央値を用いた理由は，複数の局所解が存在し，しかもそれらの関数値に大きな差がある場合には，平均値は最悪値に大きく影響され，正しい評価とはならないからである．



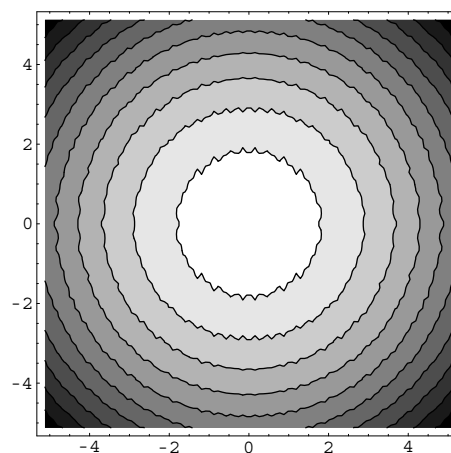
(a) Landscape of Rastrigin



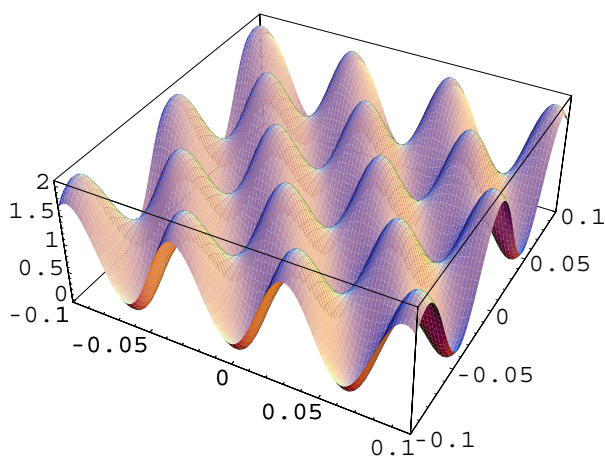
(b) Contour map of Rastrigin



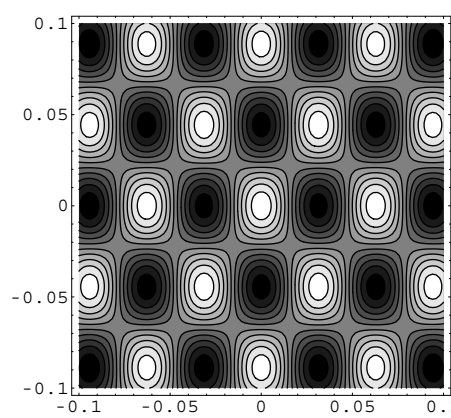
(c) Landscape of Griewank



(d) Contour map of Griewank

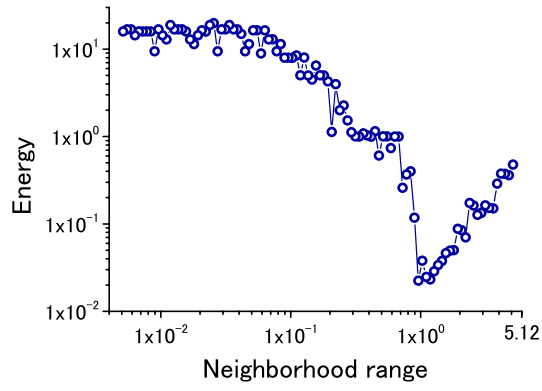


(e) Local landscape of Griewank

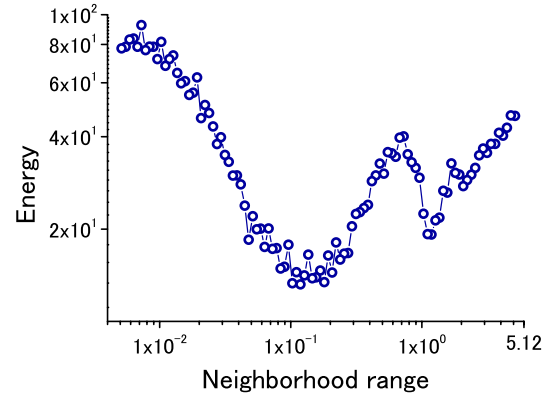


(f) Local contour map of Griewank

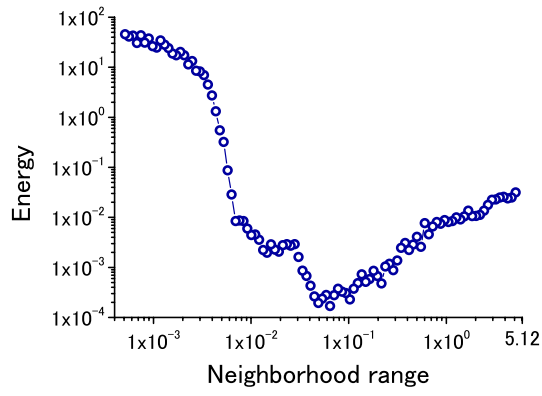
Fig. 3.2 Landscape and contour map (Rastrigin, Griewank)



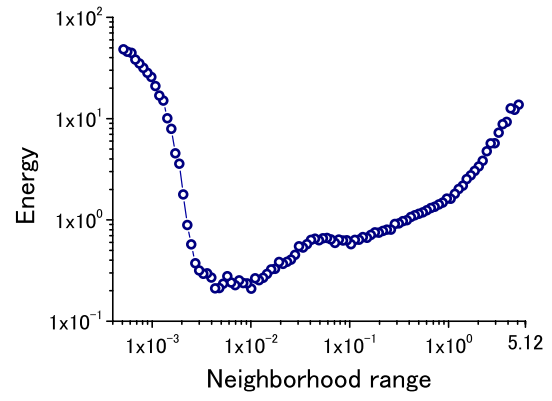
(a) 2 dimensional Rastrigin



(b) 10 dimensional Rastrigin



(c) 2 dimensional Griewank



(d) 10 dimensional Rastrigin

Fig. 3.3 Relation between neighborhood range and energy (Rastrigin, Griewank)

3.3.3 考 察

Fig. 3.3 (a) の結果より, 2 設計変数の Rastrigin 関数では近傍レンジ 1.0 付近が最適な近傍レンジに相当する. この近傍レンジ 1.0 は, Fig. 3.2 (b) の等高線からも分かるように, 隣接する局所最適解どうしの間隔に等しい. すなわち, 最適な近傍レンジとは, 局所解を抜け出すのに最適な近傍であると考えられる. これに対して, 10 設計変数の Rastrigin 関数の最適な近傍レンジは 0.1 程度であり, 2 設計変数の最適な近傍レンジとは異なっている.

一方, 2 設計変数の Griewank 関数では近傍レンジ 0.04 から 0.07 付近が最適な近傍レンジである. この場合も Fig. 3.2 (f) の局所的な等高線から, 局所解を抜け出すのに最適な近傍レンジであることが確認できる. また, 10 設計変数では 0.005 付近が最適な近傍レンジである. このように, 最適な近傍レンジとは対象問題ごとに異なり, その設計変数の数によっても異なることが分かる.

4 近傍並列 SA

4.1 近傍並列 SA の提案

3.3 節の結果から、連続最適化問題において解探索を効率的に行なう近傍レンジがあることが分かった。しかし、最適な近傍は問題によって異なり、その設計変数の数によっても変化する。このため、最適な近傍を設定するには、多くの予備実験が必要である。一方、SA は良好な解を得るためには膨大な計算時間を要するため、その短縮のために並列化が不可欠である。本論文では、これら 2 つの課題に対して、並列 SA に対して種々の近傍を与え、近傍を並列化することによって最適な近傍の調節を自律的に行なう手法を提案する。

4.2 近傍並列 SA のアルゴリズム

SA を並列化し、連続最適化問題において重要となる近傍の設定を自律的に行なう手法を提案する。提案手法は、各プロセスに対してそれぞれ異なる近傍を与え、近傍を並列化して SA の探索を行なう。提案手法は近傍を並列化するため、近傍並列 SA (Neighborhood Parallel Simulated Annealing : NPSA) とする。NPSA の概念図を Fig. 4.1 に示す。また、そのアルゴリズムを Fig. 4.2 に示す。

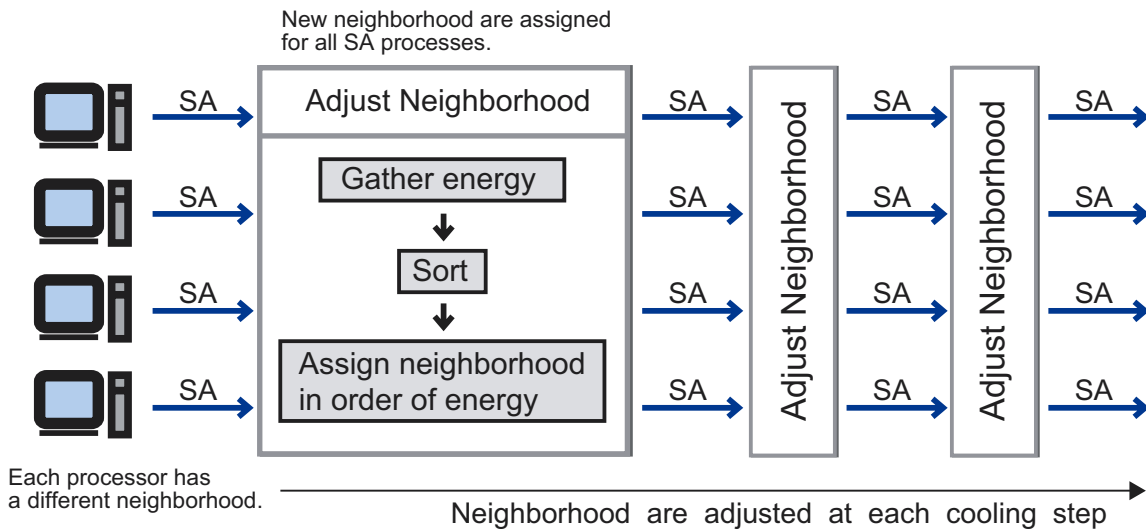


Fig. 4.1 Neighborhood Parallel Simulated Annealing

NPSA は周期的に全プロセスの同期をとり、各プロセスの近傍を自律的に調節するメカニズムを持つ。NPSA の同期時は、各プロセスが持つエネルギー値に対してソートを施し、良好な解探索を行なっているプロセスから順に小さな近傍を割り当てる操作を行なう。こうすることにより、相対的に良好な解探索を行なっているプロセスはその解付近の局所探索をさらに進めることができる。また、局所解に陥っているプロセスは、最終的には相対的に大きな近傍が割り当てられ、大域探索によって局所解から抜け出すことが可能となる。このように、各プロセスは他のプロセスと協調し、探索に応じて適応的に近傍を調節する。ここで、近傍調節のための同期は温度のクーリング時にとることとす

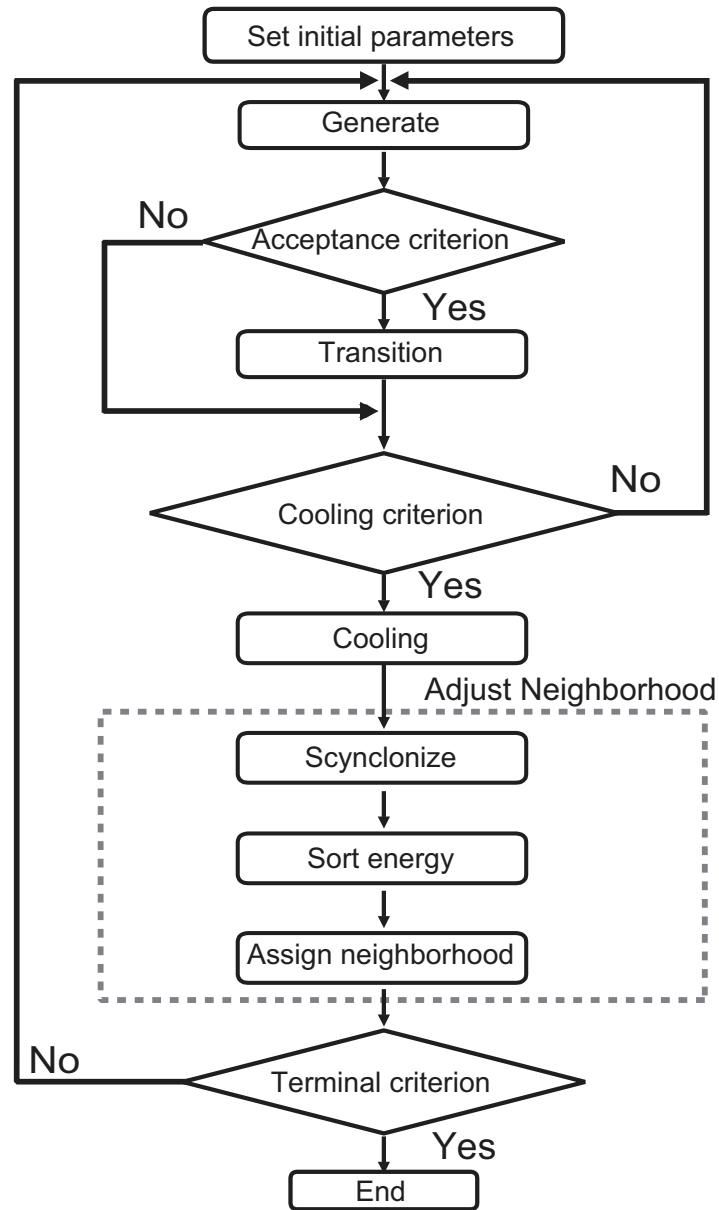


Fig. 4.2 Algorithm of NPSA

る．その理由は，解の収束条件を考えると，各近傍での探索が平衡状態に達してから近傍を調節することが妥当だと考えるからである．

NPSA の同期は，近傍調節を行なうクーリング時にのみ行なうため，その並列化効率は高い．また，この近傍調節以外，NPSA の各プロセスは解探索を独立に行うため，SA の最適解への収束性も保証される．

4.3 近傍並列 SA の探索性能

4.3.1 実験概要

NPSA の近傍調節の効果を 3 種類のアルゴリズムの比較実験によって検証する．その 3 つとは，最適な近傍に設定した逐次 SA (SA) と並列 SA (PSA)，および，今回提案する NPSA である．対象問題は 2 設計変数の Rastrigin 関数と Griewank 関数である．

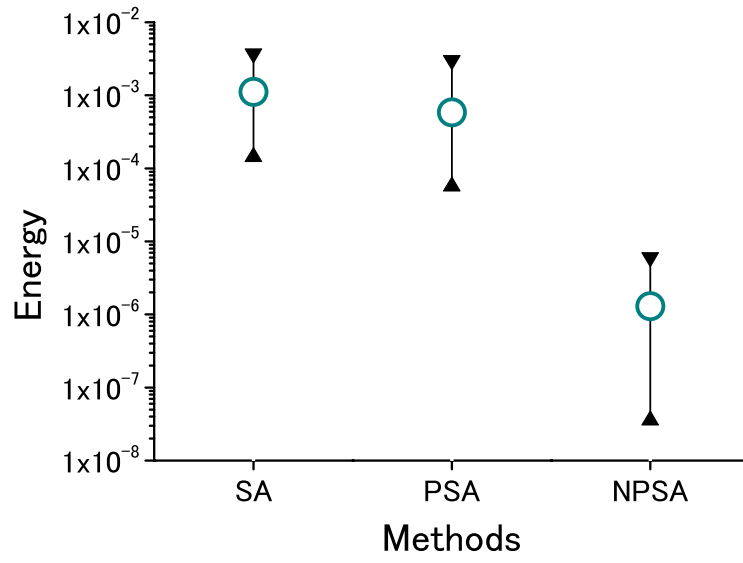
今回の実験で用いたパラメータは，次の通りである．3.3 節の最適な近傍の予備実験より，SA と PSA の近傍は，Rastrigin 関数では 1.0，Griewank 関数では 0.05 とした．一方，NPSA における各プロセスの近傍レンジは最大近傍を設計空間の幅とし，Rastrigin 関数の最小近傍を設計空間の $1/1000$ ，Griewank 関数の最小近傍を設計空間の $1/10000$ とした．各プロセスの近傍は，最大近傍から最小近傍までの近傍間を等比的に割り振ったものである．また，PSA と NPSA の並列プロセス数は 32 とした．このため，並列 SA の各プロセスに割り当てるクーリング周期を逐次 SA のクーリング周期の $1/32$ 倍とし，逐次 SA と並列 SA の総アニーリング数が等しくなるように設定した．予備実験で用いたパラメータを Table 4.1 に示す．

Table 4.1 Parameters in experiments

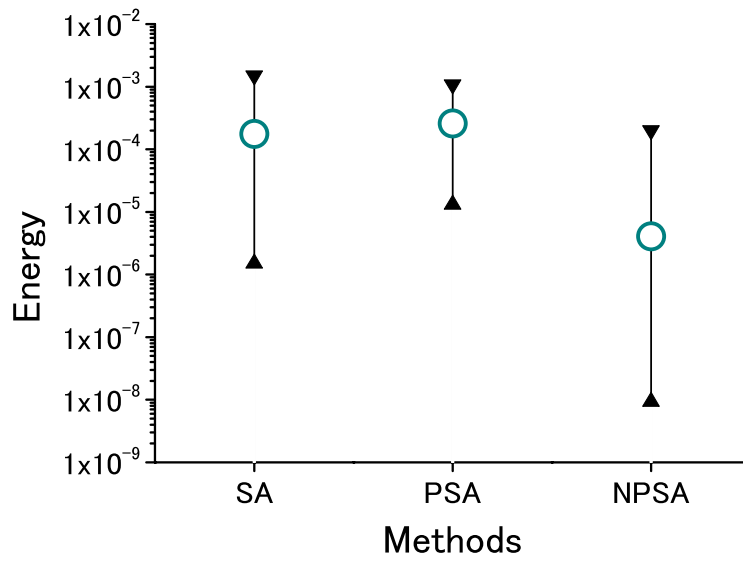
Method	SA	PSA, NPSA
Number of processes	1	32
Max temperature	10.0	10.0
Minimum temperature	0.01	0.01
Number of cooling steps	32	32
Cooling cycle	10240	320
Number of annealings	10240×32	$(320 \times 32) \times 32$

4.3.2 実験結果

Rastrigin 関数，および Griewank 関数に対する 3 つのアルゴリズムの探索性能の結果を，それぞれ Fig. 4.3 (a)，(b) に示す．横軸が比較手法で，縦軸が各アルゴリズムによる 30 試行の最良値，最悪値，中央値を示している．また，PSA と NPSA の各プロセスのうち，最も良好な解を得たプロセスのエネルギーと近傍レンジの履歴をそれぞれ Fig. 4.4，Fig. 4.5 に示す．Fig. 4.4 (a)，Fig. 4.5 (a) が PSA の履歴，Fig. 4.4 (b)，Fig. 4.5 (b) が NPSA の履歴である．横軸が探索のアニーリング数で，左縦軸がエネルギー値，右縦軸が近傍レンジである．また，両縦軸は対数表示である．

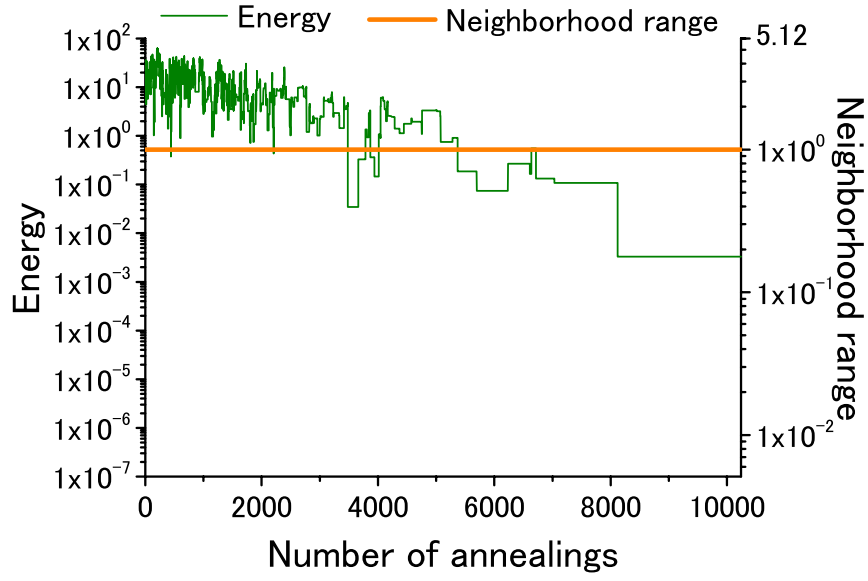


(a) Rastrigin

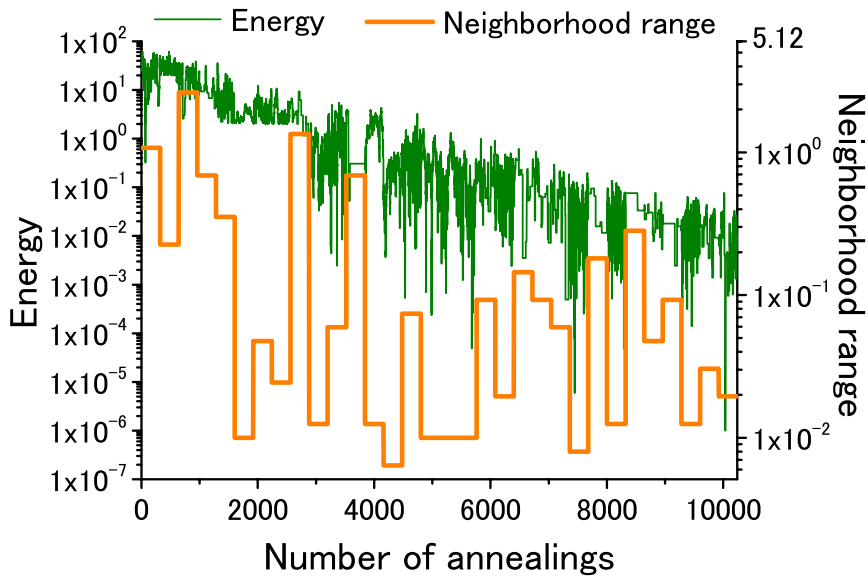


(b) Griewank

Fig. 4.3 Comparison of the qualities of solutions

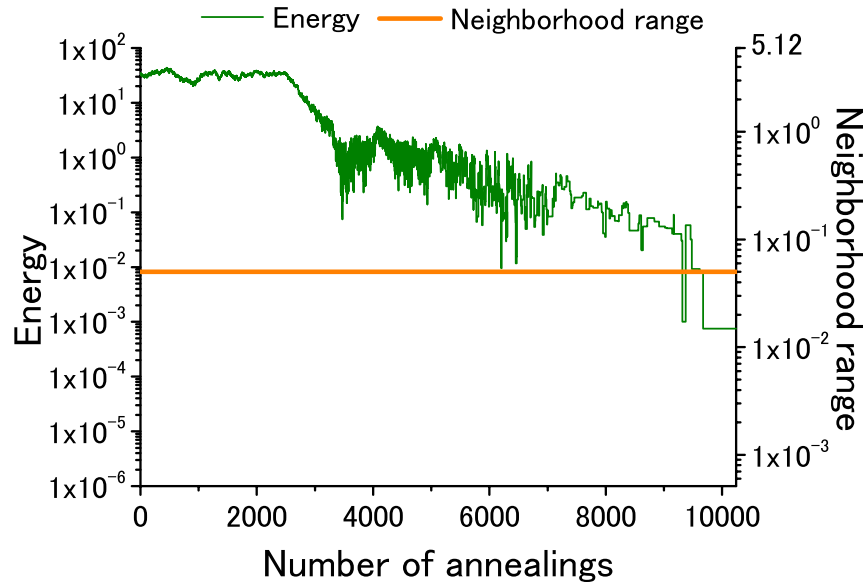


(a) PSA

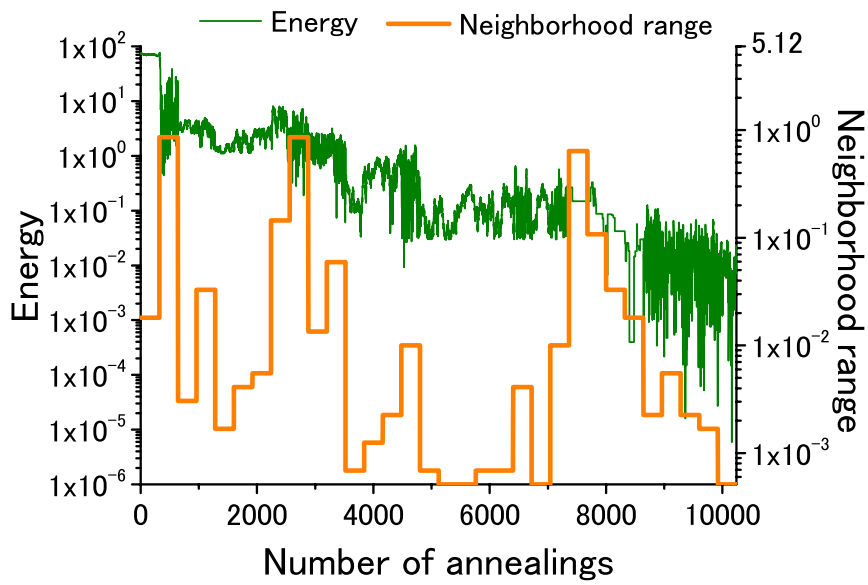


(b) NPSA

Fig. 4.4 History of energy and neighborhood range for Rastrigin



(a) PSA



(b) NPSA

Fig. 4.5 History of energy and neighborhood range for Griewank

4.3.3 考 察

Fig. 4.3 (a), (b) より, 最適な近傍レンジに設定した SA と PSA よりも, NPSA が非常に良好な結果を得ていることが分かる. 一方, Fig. 4.4, Fig. 4.5 に示したエネルギーと近傍レンジの履歴より, PSA では探索の前半では局所最適解を効率的に抜け出すことによって良好な解探索を行なっているが, 探索後半では探索が進んでいない. この理由は, 解探索を通じて近傍レンジを固定しているため, 大域的最適解の領域に到達しても局所的探索が行なわれなためだと考えられる. これに対して NPSA は探索に応じて近傍を適応的に調節しており, 解品質の良好なプロセスが局所探索を行なうことによって効率的な探索を行なっている. このように, NPSA は最適な近傍レンジの設定が不要となるのに加え, 解探索性能の大幅な向上を実現している. さらに, この手法では近傍調節の決定にルールやパラメータを用いないため, 任意の問題に対して容易に適用できる. これらの結果より, 最適に設定した固定近傍を持つ SA や PSA と比べて NPSA の解探索性能が良いことは, 最適な近傍が解の探索過程に応じて変化するためであると考えられる. すなわち, 最適な近傍は問題に固有であるだけでなく, 探索の進行状態によっても変化することが分かる.

5 複雑な多峰性関数への近傍並列 SA の適用

5.1 複雑な多峰性関数

4.3 節において，NPSA は探索の進行状態に応じて適応的に近傍を調節し，最適な近傍に設定した SA と比べて良好な解精度が得られることを示した．本節では，さらに難易度の高い複雑な多峰性関数に NPSA を適用し，有効性を検証する．対象問題は，式 (5.1) に示す Egg Holder 関数，式 (5.2) に示す Rana 関数である．

$$F_{Eggholder}(x) = \sum_{i=1}^{n-1} \left(-x'_i \sin \left(\sqrt{|x'_i - P|} \right) - P \sin \left(\sqrt{|P + x'_i/2|} \right) \right) \quad (5.1)$$

$$P = x'_{i+1} + 47$$

$$(x_i = x'_i/100 | -5.12 \leq x_i < 5.12)$$

$$F_{Rana}(x) = \sum_{i=1}^{n-1} (x'_i \sin(Q) \cos(R) + (x'_i + 1) \cos(Q) \sin(R)) \quad (5.2)$$

$$Q = \sqrt{|x'_{i+1} + 1 - x'_i|}, \quad R = \sqrt{|x'_{i+1} + 1 + x'_i|}$$

$$(x_i = x'_i/100 | -5.12 \leq x_i < 5.12)$$

各関数の景観と等高線を Fig. 5.1 に示す．等高線から分かるように，Egg Holder 関数と Rana 関数の局所最適解の間隔は一樣ではなく，非常に難しい多峰性関数である．

3.3 節と同様に，Egg Holder 関数と Rana 関数の最適な近傍を求める予備実験を行なった．近傍の種類とパラメータに関しては，Rastrigin 関数と同じ条件とした．予備実験の結果を Fig. 5.2 に示す．ただし，Egg Holder 関数と Rana 関数は実験的に求めた準最適解との差をエネルギー値としている．

Fig. 5.2 より，Egg Holder 関数と Rana 関数は最適な近傍レンジの設定が難しい問題といえる．これは，Fig. 5.1 (b) (d) の等高線から分かるように，局所最適解の間隔が一樣でなく，局所解を抜け出すのに最適な近傍レンジが一定ではないからだと考えられる．このような多峰性関数では，近傍に分布を用いるのが一般的である．本研究では，最も単純な分布である正規分布を用いて NPSA の近傍を設計し，探索性能を検証した．5.2 節では，正規分布とその温度スケジュールについて述べる．

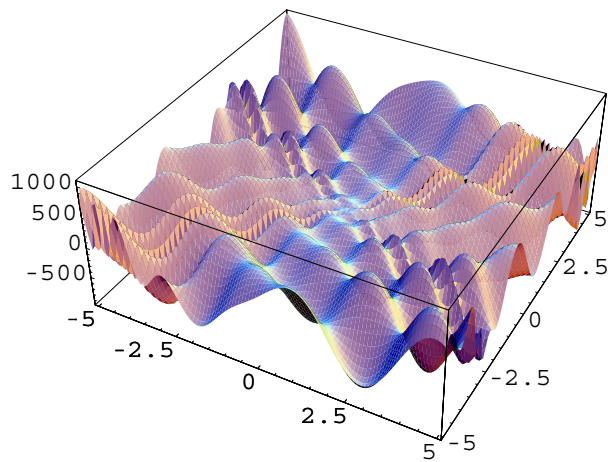
5.2 正規分布と温度スケジュールの設定

5.2.1 正規分布

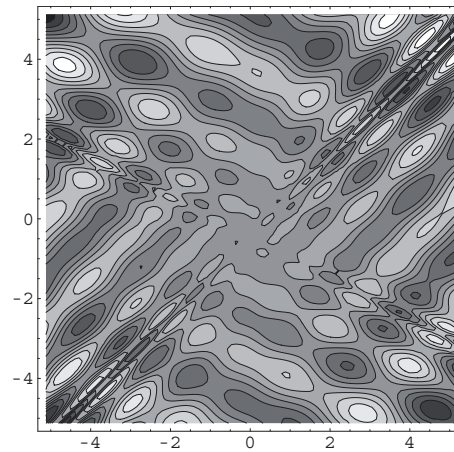
正規分布は，Hinton らによって提案された近傍²²⁾である．エネルギー差を Δx ，温度パラメータを T_k ，対象問題の設計変数の数を D とすると，(5.3) 式で表される．

$$g_k(\Delta x) = \frac{1}{(2\pi T_k)^{D/2}} \exp \left(\frac{-|\Delta x|^2}{2T_k} \right) \quad (5.3)$$

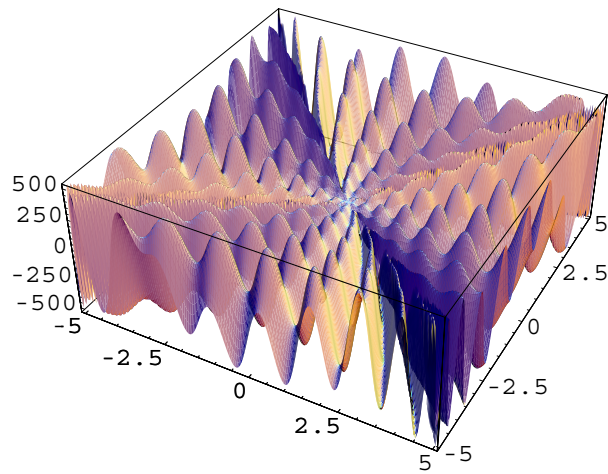
(5.3) 式より，温度の平方根が正規分布の標準偏差となる．このように正規分布近傍は近傍決定に温度を用いており，探索序盤の高温時は近傍の裾野が広い分布を形成して大域探索を行う．また，温度が冷却するにつれて分布の裾野も小さくなり，探索終盤では局所探索を行なう．



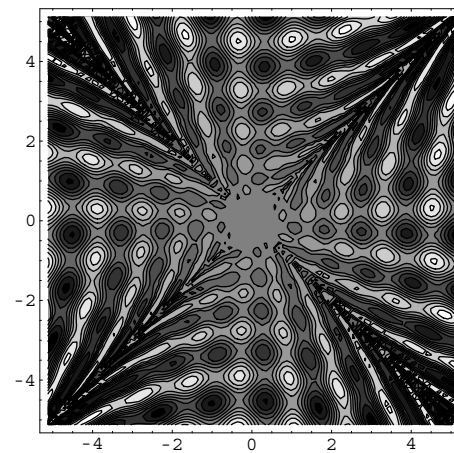
(a) Landscape of Egg Holder



(b) Contour map of Egg Holder

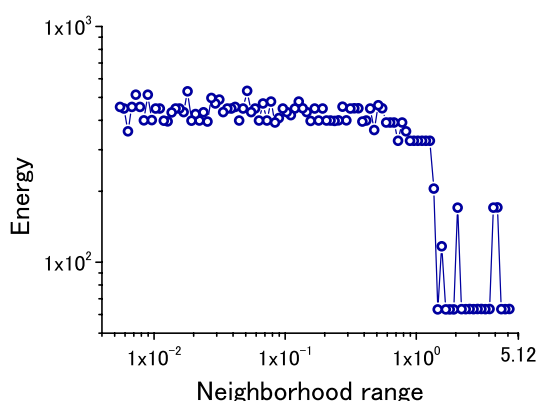


(c) Landscape of Rana

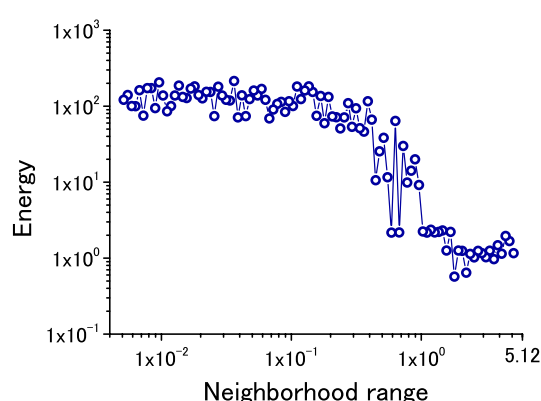


(d) Contour map of Rana

Fig. 5.1 Landscape and contour map (Egg Holder, Rana)



(a) 2 dimensional Egg Holder



(b) 2 dimensional Rana

Fig. 5.2 Relation between neighborhood range and energy (Egg Holder, Rana)

5.2.2 正規分布における温度パラメータの設定

組合せ最適化問題における最高温度は、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ような温度に設定するという考え方が一般的に用いられる．しかしながら、近傍決定に温度スケジュールを利用する場合は温度スケジュールが近傍構造を決定してしまうため、不必要に高温の場合は定義域を超える確率が高くなり、無駄な試行が増えることになる．一方、最低温度に関しても、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、最高温度と同様、最低温度が低すぎると探索領域は非現実的に小さくなる恐れがある．

本研究では、最高温度は定義域をカバーするのに十分な温度に設定する．ここでは、近傍の分布として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．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には、次状態が端に生成される確率は 4.6 % である．また、現在の解が設計空間の端にあるときにもう一方の端に生成される確率は 0.01 % 程度となる．平均値が 0、標準偏差 が 1 である正規分布において、平均値 ± 2 の範囲に含まれる割合を Fig. 5.3 に示す．

一方、最低温度に関しては、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のような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、(5.3) 式で与える近傍が十分小さいということである．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、それとは異なった方法として、最高温度の分布に対して $1/100$ の密度分布の広がりとなるように最低温度を設定した、すなわち、正規分布の標準偏差は設計空間の $1/400$ 程度となり、最低温度は 6.5536×10^{-4} となる．2 設計変数の場合の最高温度時の分布を Fig. 5.4 (a) に、最低温度時の分布を Fig. 5.4 (b) 示す．

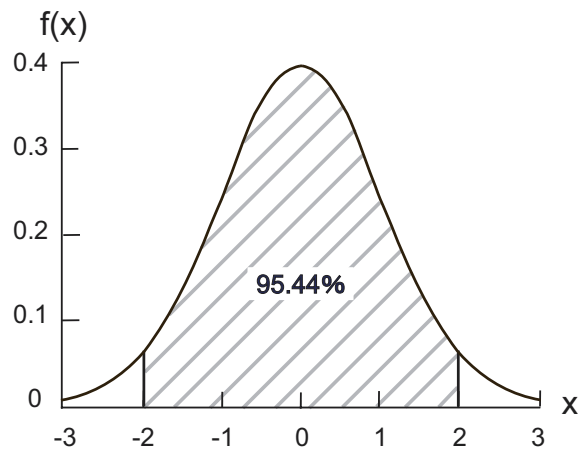
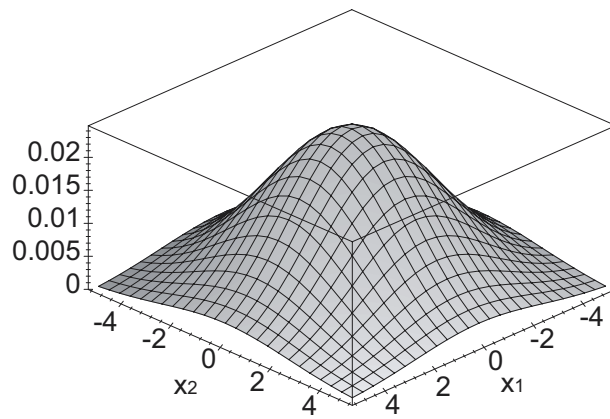
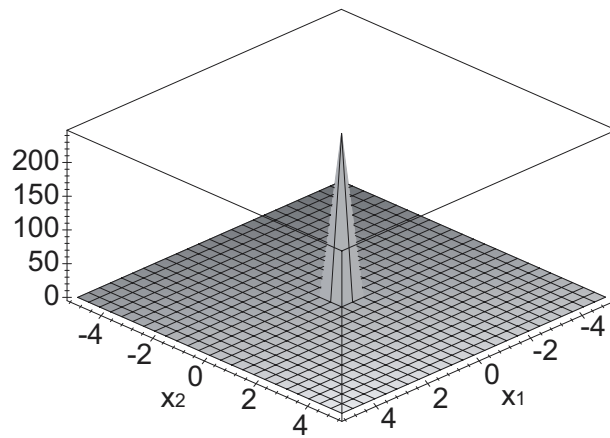


Fig. 5.3 The generation percentage within average ± 2



(a) Normal distribution at max temperature



(b) Normal distribution at minimum temperature

Fig. 5.4 Normal distribution

5.3 数値実験

5.3.1 実験概要

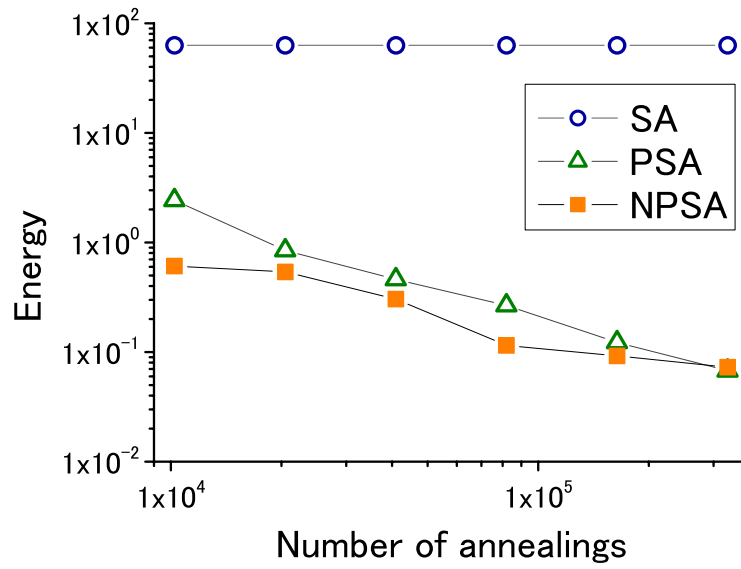
NPSA を複雑な多峰性関数である Egg Holder 関数と Rana 関数に適用し，その解探索性能を検証する．比較手法は，逐次 SA (SA)，並列 SA (PSA)，および NPSA の 3 つのアルゴリズムである．本実験では，探索数を変えた実験を行うことにより，各手法におけるアニーリング数と探索性能の関係を検証する．本実験で用いたアニーリング数は，最小を 10240 とし，それ以降は 2 倍ごとの計 6 パターンの探索数である．すなわち，最大は 327680 となる．本実験で用いたパラメータを Table. 5.1 に示す．

Table 5.1 Parameters in experiments

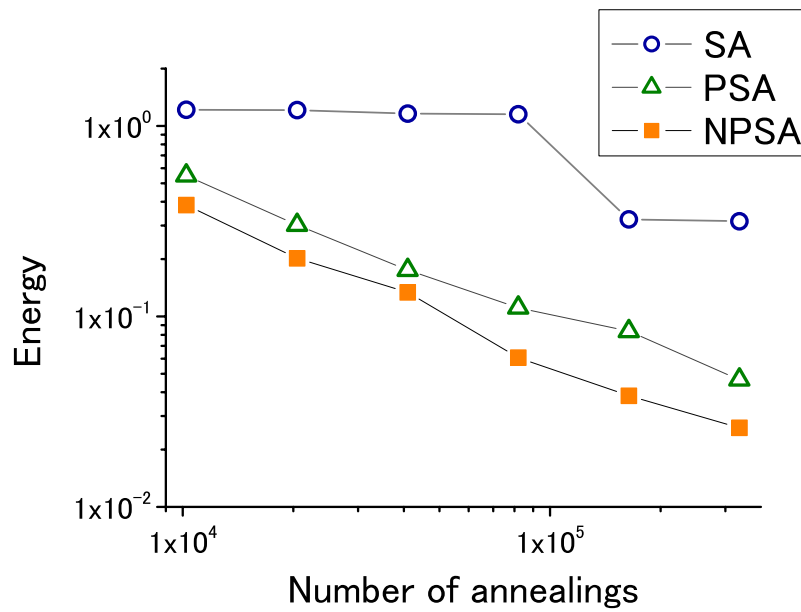
Method	SA	PSA, NPSA
Number of processes	1	32
Number of cooling steps	32	32
Max temperature	6.6636	6.5536
Minimum temperature	6.5536×10^{-4}	6.5536×10^{-4}

5.3.2 実験結果

Egg Holder 関数，および Rana 関数における各手法のアニーリング数と探索性能の関係を Fig. 5.5 (a), (b) に示す．横軸が探索数で，縦軸が各手法の 30 施行におけるエネルギーの中央値の結果である．また，アニーリング数が 327680 の場合の PSA と NPSA の各プロセスのうち，最も良い結果を得たプロセスのエネルギー，および分布に用いた標準偏差の履歴を Fig. 5.6, Fig. 5.7 に示す．Fig. 5.6 (a), Fig. 5.7 (a) が PSA の履歴，Fig. 5.6 (b), Fig. 5.7 (b) が NPSA の履歴である．横軸がアニーリング数であり，1 プロセスは 10240 のアニーリングを行なっている．左縦軸がエネルギー値，右縦軸が分布に用いた標準偏差である．最高温度では標準偏差は 2.56 であり，最低温度では 2.56×10^{-2} となる．

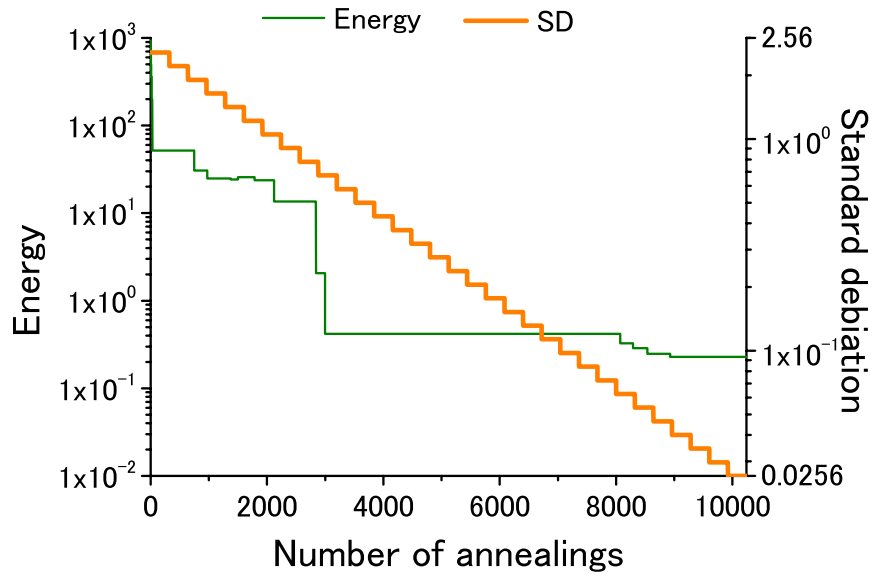


(a) Egg Holder

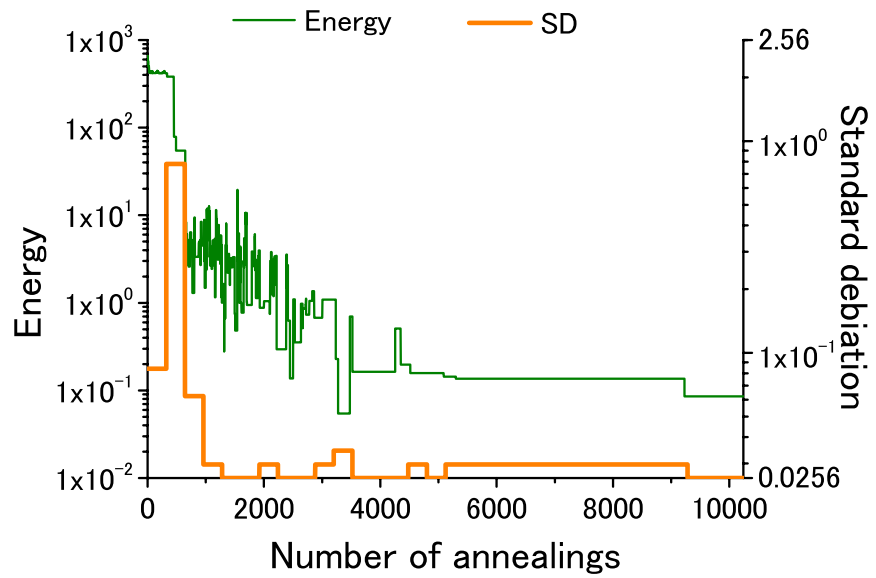


(b) Rana

Fig. 5.5 Performance of methods

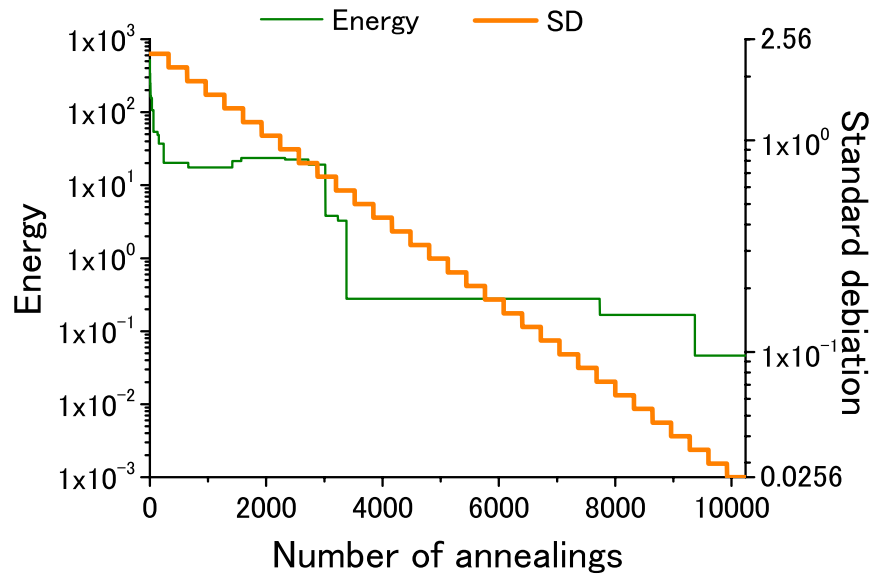


(a) PSA

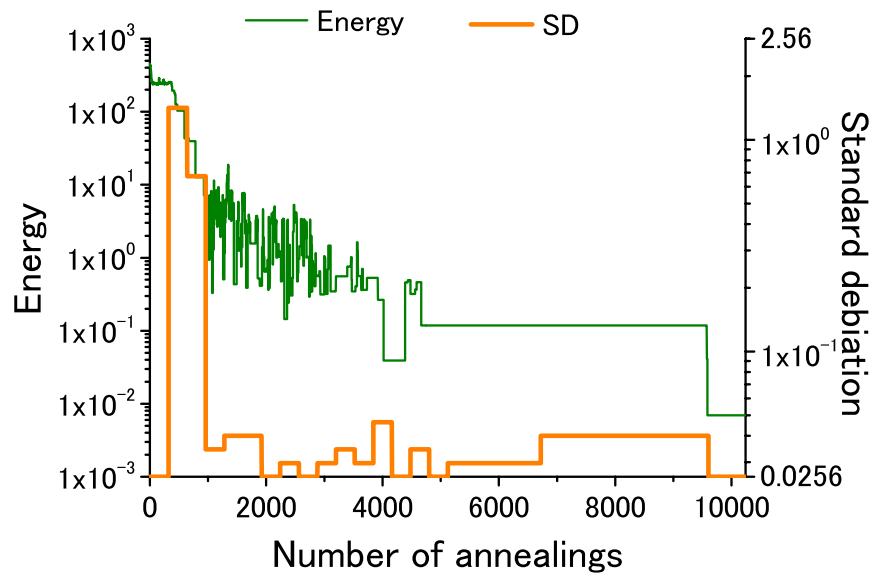


(b) NPSA

Fig. 5.6 History of energy and SD for Egg Holder



(a) PSA



(b) NPSA

Fig. 5.7 History of energy and SD for Rana

5.3.3 考 察

Fig. 5.5 (a), (b) より, 複雑な多峰性関数において逐次 SA の解精度は非常に悪い. これに対し並列化による PSA, さらに近傍を並列化する NPSA は, アニーリング数を増やすことで解精度が向上していることが分かる. 一方, PSA と NPSA を比べると NPSA は特に Rana 関数において解精度の大幅な向上を実現している. また, Egg Holder 関数においても PSA と同等, もしくは高い解探索性能を示した.

PSA と比較して NPSA の異なる点は, PSA が正規分布の標準偏差を温度のクーリングによって小さくするのに対し, NPSA は各プロセスがもつエネルギー値に応じて標準偏差を変化させていることである. 各探索履歴から分かるように, PSA は指数型アニーリングによって階段型の標準偏差のスケジュールとなり, 探索序盤は近傍が大きすぎるために良好な探索が行なわれていない. 一方, NPSA は探索序盤から適応的に近傍を調節しており, 良好な解探索が行なわれている.

以上の結果より, NPSA は探索状況に応じた近傍の調節を行い, 対象問題に依存する近傍の設定が不要である. さらに従来の SA, PSA と比べて解探索性能が向上することから, 提案手法は連続最適化問題における並列化手法として有効であるといえる.

6 結 論

6.1 結 論

本研究では、連続最適化問題に SA を適用する場合に重要となる近傍について検討した。その結果、標準的なテスト関数において、解探索を効率的に行なう近傍レンジが存在することを明らかにした。さらに、この重要となる近傍に着目し、近傍を並列化してその調節を自律的行なう手法として近傍並列 SA (Neighborhood Parallel Simulated Annealing : NPSA) を提案した。提案手法を複雑な多峰性関数に適用し、その有効性を検証した。得られた結論を以下に示す。

- 連続最適化問題の標準的なテスト関数に対して種々の近傍レンジを用いて一様分布で解探索を行なった。その結果、解探索を効率的に行なう最適な近傍レンジが存在することが分かった。
- 最適な近傍とは、局所最適解の分布に依存するため、対象問題やその設計変数の数によって変化する。また、解の探索過程にも応じて変化すると考えられる。
- 連続最適化問題における最適な近傍の設定という問題に対して、近傍を並列化し、探索に応じてその調節を行なう NPSA を提案した。
- 最適な近傍に設定した SA と NPSA を比較した結果、提案手法はこの最適な近傍の設定が不要となるだけでなく、解探索精度の大幅な向上を実現した。
- 逐次 SA では解くことが難しい、複雑な多峰性関数に対して NPSA を適用した。その結果、NPSA は PSA と同等、もしくは高い解探索性能を示した。
- NPSA の近傍とエネルギーの履歴より、NPSA は解の探索に応じて近傍を適応的に調節しており、連続最適化問題における SA の並列化手法として有効であるといえる。

6.2 今後の課題

今後の課題として、以下が挙げられる。

- 今回使用した関数以外の連続最適化問題においても、本提案手法が有効であるかの検証を行なう。
- 正規分布以外の分布に対して NPSA を適用し、検証を行なう。
- 並列化する近傍の種類を固定にするのではなく、最適な近傍への収束を目指す手法の提案。

謝 辞

本研究を遂行するにあたり，多大なる御指導，そしてご協力を頂きました同志社大学 工学部 知識工学科 三木光範教授に心より感謝いたします．

本研究の実施にあたり，ご指導，ご協力を頂きました同志社大学 工学部 知識工学科 廣安知之講師に深く感謝いたします．

本研究の実施にあたり，ご指導，ご協力を頂きました同志社大学大学院 工学研究科 知識工学専攻 知的システムデザイン研究室 小野景子さんに深く感謝いたします．

本研究の実施にあたり，全面的なご指導，ご協力を頂きました同志社大学大学院 工学研究科 知識工学専攻 知的システムデザイン研究室 伏見俊彦さんに深く感謝いたします．

最後に，何事に対しても意欲的に取り組み，常に励まして頂きました知的システムデザイン研究室のみなさんに深く感謝いたします．

参考文献

- 1) Reeves, C.R. 編, 横山, 奈良ら訳. モダンヒューリスティックス. 日刊工業新聞社, 1997.
- 2) Kirkpatrick, S., Gelett Jr. C. D., , Vecchi, M. P. *Optimization by Simulated Annealing*. Science, 1983.
- 3) Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E. *Equation of State Calculation by Fast Computing Machines*. Journ. of Chemical Physics, 1953.
- 4) E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines*. John Wiley & Sons, 1989.
- 5) 金久実. ゲノム情報への招待. 共立出版, 1998.
- 6) E. Aarts and P.J.M. Van Laarhoven. Statistical cooling: A general approach to combinatorial optimization problems. *Philips J. Res*, Vol. 40, pp. 193–226, 1985.
- 7) M. Lundy and A. Mees. Convergence of an annealing algorithm. *Math. Programming*, Vol. 34, pp. 111–124, 1986.
- 8) David T.CONNOLLY. An improved annealing scheme for the qap. *European Journal of Operational Research*, Vol. 46, pp. 93–100, 1990.
- 9) Mark Fielding Harry Cohn. Simulated annealing: Searching for an optimal temperature schedule. *SIAM J. Optim*, Vol. 9, pp. 779–802, 1999.
- 10) Mark Fielding. Simulated annealing with an optimal fixed temperature. *SIAM J.*, Vol. 11, No. 2, pp. 289–307, 2000.
- 11) F.Romeo M.D.Huang and A.Sangiovanni-Vincentelli. An efficient general cooling schedule for simulated annealing. *IEEE*, pp. 225–230, 1986.
- 12) Steave R. White. Concepts of scale in simulated annealing. *Proceeding IEEE Intl. Conf. Comp. Des.(ICCD)*, pp. 646–651, 1984.
- 13) B. Rosen. *Functional Optimization based on Advance Simulated Annealing*. IEEE Workshop on Physics and Computation, 1992.
- 14) Harold Szu and Ralph Hartley. Fast simulated annealing. *Physics Letters A*, Vol. 122, No. 3,4, pp. 157–162, 1987.
- 15) L. Ingber and B. Rosen. Genetic algorithms and very fast simulated reannealing: A comparison. *Mathematical and Computer Modelling*, Vol. 16, No. 11, pp. 87–100, 1992.

- 16) Marchesi M. Martini C. Corana, A. and S. Ridella. *Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing Algorithm*. ACM Trans. on Mathematical Software, 1987.
- 17) 三木光範, 廣安知之, 小野景子. 最適な受理確立を目標とする適応的近傍を持つシミュレーテッドアニーリング. 情報処理学会誌, 2003.
- 18) 瀧 和男木村宏一. 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, , 1990.
- 19) 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.
- 20) Collins, N . E . , Eglese, R . W . and Golden, B . L . *Simulated Annealing-an annotated bibliography*. American J . Math . & Management Sci . , 1988.
- 21) 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, 1994.
- 22) Sejnowski T.J. Hinton, G.E. and D.H. Achley. Boltzmann machines: constraint satisfaction networks that learn. *Technical Report CMU-CS*, pp. 84–119, 1984.
- 23) L. Ingber. Very fast simulated re-annealing. *Mathematical and Computer Modelling*, Vol. 12, pp. 967–973, 1989.
- 24) Daniel R. Greening. Parallel simulated annealing techniques. *Physica D*, Vol. 42, pp. 293–306, 1990.
- 25) Hector Sanvicente S. and Juan Frausto S. A methodology to parallel the temperature cycle in simulated annealing. *Lectures Notes on Computer Science*, pp. 63–74, 2000.
- 26) Hector Sanvicente S. and Juan Frausto S. Mpsa : A methodology to parallelize simulated annealing and its application to the traveling salesman problem. *MICAI2002, LNAI3213*, pp. 89–97, 2002.
- 27) K. Ganeshan K. Krishna and D. Janaki Ram. Distributed simulated annealing algorithms for job shop scheduling. *IEEE Transactions on Systems, man, and Cybernetics*, Vol. 25, No. 7, pp. 1102–1109, 1995.
- 28) R. Luling R. Diekmann and J. Simon. Problem independent distributed simulated annealing and its application. *Proceeding of the 4th IEEE SPDP*, pp. 1–23, 1993.
- 29) James R.A. ALLWRIGHT. A distributed implementation of simulated annealing for the traveling salesman problem. *Parallel Computing*, Vol. 10, .

- 30) P.M.A. Soot J.M. Voogd and R.v.Dantzig. Comparison of vector and parallel implementation of simulated annealing algorithm. *Future Generation Computer Systems, special issue HPCN'94*, Vol. 11, .