

近傍並列シミュレーテッドアニーリング*

Neighborhood Parallel Simulated Annealing

安藤 景子¹, 三木 光範², 廣安 知之², 及川雅隆³

Keiko ANDO, Mitsunori MIKI, Tomoyuki HIROYASU, Masataka OIKAWA

¹ 同志社大学大学院 (〒610-0394 京田辺市多田羅都谷 1-3)² 同志社大学工学部 (〒610-0394 京田辺市多田羅都谷 1-3)³ 奈良先端科学技術大学院大学 (〒630-0192 奈良県生駒市高山町 8916-5)

When applying SA to continuous optimization problems, the appropriate adjustment of the neighborhood ranges becomes necessary to obtain the good performance. In this paper, we propose a Neighborhood Parallel Simulated Annealing (NPSA) for continuous optimization problems, which provides global search using the periodic exchange of different neighborhood ranges with parallel computers. The proposed approach, NPSA, automatically determines the appropriate neighborhood range, and shows a good performance in solving typical test problems.

Key Words: Parallel Simulated Annealing, Continuous Optimization Problem, Neighborhood Range

1. はじめに

近年, 社会を構成するシステムは大規模, 複雑化が進み, それに伴いシステムの最適化が重要となっている. 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. 前者の問題は, 主に目的関数の勾配情報をもとに連続的な探索を行い最適解を求めることが多い. 一方, 後者の問題では汎用的な確率的アルゴリズムにより準最適解を求めることが多く, 発見的手法(ヒューリスティック法)である遺伝的アルゴリズム(GA)やシミュレーテッドアニーリング(SA)などが用いられる⁽¹⁾.

SAはMetropolisらが1953年に発表した焼きなましと呼ばれる加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に端を発し, 最適化問題, 特に組み合わせ最適化問題に有効なアルゴリズムである. SAでは, 最小化すべき目的関数はエネルギーと呼ばれ, 任意の非線形性を持った目的関数をほとんど処理できるという大きな利点がある^(2, 3, 4).

SAはGA等と同じく非線形性を持つ種々の目的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. 1つは, 解探索の振る舞いを決めるパラメータ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ということである⁽⁵⁾.

SAで重要となるパラメータは, 近傍と温度である. 組み合わせ最適化問題の場合では, 隣接する2つの要素を入れ替えるなどして近傍を生成する. このため, 近傍の構造を適当に決めると, 温度スケジュールの調節が最も重要となる.

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関係し, 自由に決めることが可能である. 一般的に, 近傍が小さい場合にはエネルギーの変化が小さく, 近傍が大きい場合にはエネルギーの変化が大きい. またその変化は目的関数のランドスケープに大きく依存する. そのため, 連続最適化問題にSAを適用する場合, 近傍の調節が非常に重要になるが, それを一意に定義することは難しい.

これに対し, 目的関数のランドスケープに応じた近傍を適応的に調節する研究が行われている. Coranaは解振動に用いる近傍を受理率が0.5となるように, ランドスケープに応じた近傍調節を自動化した⁽⁶⁾. また, 著者らは任意の受理率を与えることのできる新しい近傍設計を考え, 最適な受理確率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(SA/AAN)を提案した^(7, 8). これらの手法は, 近傍の自動調節を行うため, 問題ごとのチューニングは必要ないが, 近傍

* 原稿受付 2006年6月16日, 改訂年月日 2006年8月29日, 発行年月日 2006年9月8日, ©2006年日本計算工学会. Manuscript received, June 16, 2006; final revision, August 29, 2006; published, September 8, 2006. Copyright ©2006 by the Japan Society for Computational Engineering and Science.

の自動調節のためのパラメータが必要となる。

そこで本研究では、近傍の調節を自動化し、かつ近傍調節のためのパラメータを必要としない手法、近傍並列 SA (Neighborhood Parallel Simulated Annealing: NPSA) を提案する。NPSA では並列処理を行い、各プロセスに異なる近傍を与える。その近傍をプロセス間で交換することにより、近傍の自動調節を行う。

本論文では、まず解探索が良好に行える近傍について検証を行う。次に、提案手法である近傍並列 SA (NPSA) について述べる。最後に、代表的な数学関数最小化問題に本手法を適用し、解探索能力と近傍履歴について検証する。

2. 最適な近傍

SA では、次状態は現在の状態に摂動を加えることで生成される。摂動により生成される次状態の範囲のことを近傍と呼び、連続最適化問題の場合、近傍はユークリッド空間の距離として与えられる。また本論文では、その距離の最大値を近傍幅と呼ぶ。

SA では、一般に現在の解を中心とし、解摂動のための移動距離に関する確率分布を与えることにより近傍を定義する方法が一般的である⁽⁹⁾。この場合、解摂動のための近傍幅を考慮することが重要になり、近傍幅を一定とし探索を行うアルゴリズム⁽¹⁰⁾、温度を用いて近傍幅を決定するアルゴリズム⁽¹¹⁾、目的関数のランドスケープに従い適応的に近傍幅を決定するアルゴリズム^(6, 7, 8) などがある。

近傍幅が大きすぎる場合は得られる解の精度が悪く、小さすぎる場合は探索の進行が遅くなるため、近傍幅の定義が難しい。そこで、最適な近傍幅を求めため、種々の近傍幅に対して数値実験を行った。後に説明する連続最適化問題における標準的なテスト関数である Rastrigin 関数および Griewank 関数に対して、探索開始時から終了時まで一定の近傍幅 (固定近傍幅) とし、近傍幅は最大で設計空間の幅、最小の近傍幅としてその 10^{-3} とし、この間を指数的に 100 分割し、100 種類の近傍について実験を行った。また数値実験では、近傍構造に用いる確率分布として一様分布を用いた。結果を図 1 および図 2 に示す。これらの図の横軸が近傍、縦軸がエネルギー値を表し、2, 3 次元の結果を示している、これらの結果には 30 試行の中央値を用いる。

図 1 および図 2 より、Rastrigin 関数では近傍が 1、Griewank 関数では 2 次元の場合 6、3 次元の場合 8 である場合、最も効率の良い探索が行えており、問題毎に最適な近傍幅が存在すると言える。

これらの結果から、対象問題ごと次元ごとに最適な近傍幅を特定することができれば、その近傍幅で集中的に探索を行うことで良好な結果を得ることができると考えられる。しかし、最適な近傍幅を求めためには、多くの予備実験が必要となる。また、最適な近傍幅は問題ごとに異なるため、最適な近傍幅を一意に定義することは難しい。

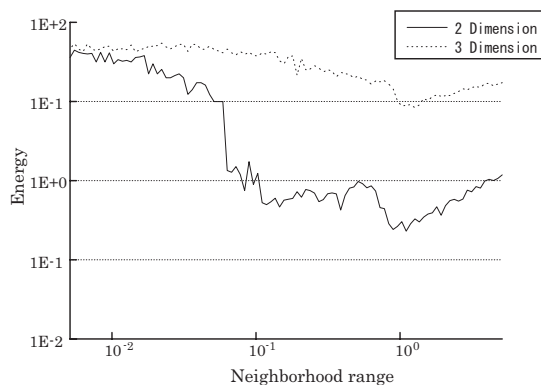


Fig.1 Appropriate Neighborhood range (Rastrigin)

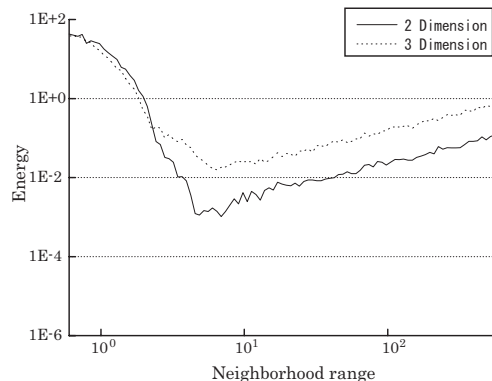


Fig.2 Appropriate Neighborhood range (Griewank)

これに対し、温度を用いて近傍幅を決定するアルゴリズムは、温度の高低により近傍幅を決定するものであり、それらは正規分布の近傍構造を持ち、標準偏差に温度を用いる Boltzman アニールリング、コーシー分布の高速アニールリングなどがある。これらのアルゴリズムでは、温度が高い場合は近傍幅が大きく、温度が低い場合は近傍幅が小さくなるため、高温では大域探索、低温では局所探索を行える。しかしこの方法では、近傍幅と温度が密接に関係しているため、適切な温度スケジュールの設定が難しく、また目的関数のランドスケープを考慮することが出来ないという問題がある。

一方、近傍幅を問題に応じて適応的に決める方法は前述した様にいくつか存在する。これらの方法では、適応的な近傍幅調節に用いるパラメータを与えれば、問題と探索状況に応じて適切な近傍を自動的に求めることができる。

SA では現状態から次状態を生成する際、現状態から近傍の範囲内に次状態を生成し、そのエネルギー値を求める。次状態のエネルギー値が現状態のエネルギー値より低い場合は現状態は次状態に推移し、現状態のエネルギー値より高かった場合は Metropolis 基準を用い確率的に次状態に推移する。

Corana は、現状態から次状態に推移する確率 (受率率) が 0.5 となるように近傍を適応的に変化させ探索を

行う手法を提案した。しかし、受理率を0.5とする妥当性が明らかとなっていなかったため、著者らは任意の受理率になるような近傍調節が可能となる手法、最適な受理確率を目標とする適応的近傍をもつシミュレートドアニーリング (SA/AAN) を提案した。

Coranaの手法, SA/AANは問題ごとに最適近傍を求める必要がない優れた手法であるが, ある周期ごとに周期内での受理率を基に近傍幅を拡大, 縮小し近傍を適応的に調節するため, 基準となる受理率 (Corana:0.5, SA/AAN:任意の値) と近傍幅を拡大, 縮小するパラメータが必要となる。

近傍の定義方法は前述した様に種々存在するが, 適応的な手法を用いた場合でも, 近傍調節のための最低限のパラメータが必要になることから分かるように, 最適な近傍を対象問題に依存することなく自動的に求めることは, 非常に難しいと言える。

3. 近傍並列 SA (NPSA)

3.1 近傍並列 SA (NPSA) のコンセプト

本論文では, 近傍の適応的調節を行い, かつ調節のためのパラメータ (基準となる受理率, 近傍の拡大, 縮小のためのパラメータ) が不要となる新しい近傍調節メカニズムを持つ手法 (NPSA) を提案する。NPSAでは, 異なった近傍を持つ複数のSAを同時並列的に動作させ, 一定のアニーリング周期ごとにプロセス間で通信を行う。全プロセスのエネルギー値から各プロセスのエネルギー値の相対評価を行い, その評価に従いエネルギー値の高いプロセスには大きな近傍を与え大域探索を, エネルギー値の小さいプロセスには小さな近傍を与え局所探索が行えるように近傍を交換する。このように操作を繰り返すことで, 探索に応じた適切な近傍が与えられると考えられる。

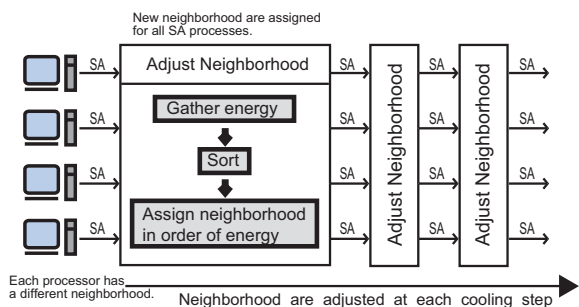


Fig.3 Algorithm of NPSA

3.2 近傍並列 SA (NPSA) のアルゴリズム

NPSAは並列プロセスに対してそれぞれ異なる近傍幅を与え, 各プロセスはその近傍幅で探索を行う。また, 周期的に全プロセスの同期をとりプロセス間で情報交換を行う。探索状況に応じた近傍幅調節を行うため, 探索状況が安定した後に近傍幅調節を行う必要がある。そのため, 近傍幅調節はクーリング周期と同時に進行。同期時には, 各プロセスが持つエネルギー値を1つのプロセスに集め, エネルギー値をソートし, エ

ネルギー値が低い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幅が設定されるよう近傍幅を割り当てる。

こうすることにより, エネルギー値の小さい良好な解探索を行なっているプロセスは小さい近傍幅が割り当てられるため, 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, エネルギー値が悪い局所解に陥っているプロセスには大きな近傍幅が割り当てられるため, 大域探索を行うことになり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幅を調節する。NPSAにおけるプロセス間での近傍幅交換の概念図を図3に, NPSAのアルゴリズムを図4に示す。

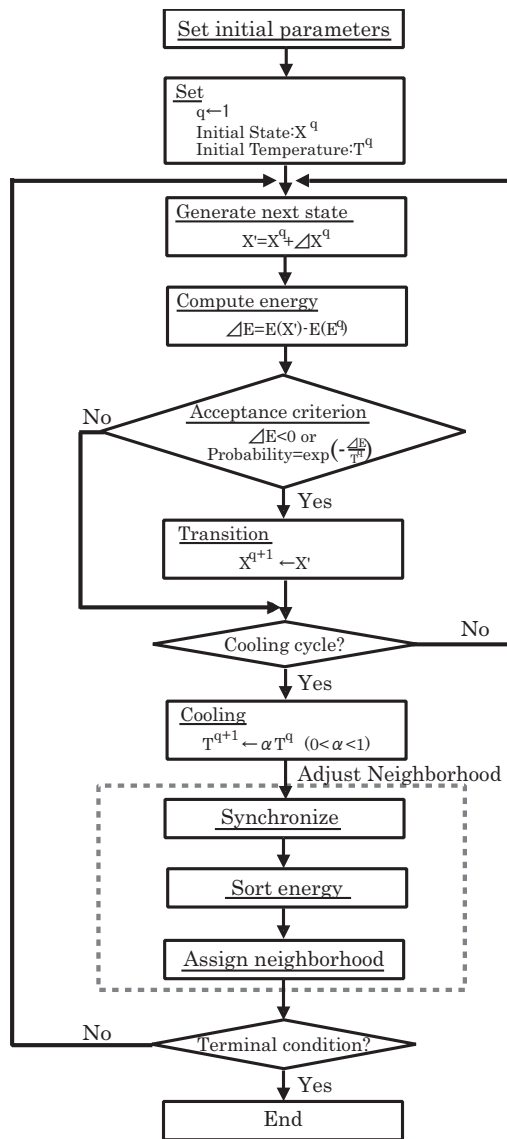


Fig.4 NPSAのアルゴリズム

図中の T は温度を X は解集合を示し, X^q は現在の解を X' は次状態の候補解を示す。また E はエネルギー値 (評価値) を, ΔE は現在の解と次状態候補の解とのエネルギー差を示し, α は冷却率を示す。

初期設定 (Set initial parameters):

初期設定ではNPSAで用いる近傍幅の種類を設定するため、最大近傍幅、最小近傍幅およびプロセス数を設定する。NPSAで用いる近傍幅は、最大近傍幅を設計空間の幅、最小近傍幅をその 10^{-5} とし、その中間の近傍幅は等比的に割り振りプロセス数生成する。

本手法では最小近傍幅を設計空間の 10^{-5} とするが、要求される解精度に合わせて設定することが可能である。Rastrigin関数およびGriewank関数に対する最小近傍幅と精度の関係を図5に示す。図5の横軸が最小近傍幅、縦軸がエネルギー値を示す。この結果より最小近傍幅が大きすぎると精度に影響するが、ある程度小さくすれば良いということが分かった。探索空間の 10^{-5} 程度に設定するのが妥当と考えられるため、本実験では探索空間の 10^{-5} とした。

NPSAではプロセス数は自由に選択することができるため、用いることのできるプロセスの数に合わせて設定することが可能である。ただし、プロセス数は隣り合うプロセス同士の近傍幅の比率と関係する。探索領域の連続性の観点から、比率が0.5以上必要であると考えられるが、プロセス数が多い場合には隣り合うプロセスとの近傍幅の比率は高くなり、プロセス数が少ない場合には比率は低くなる。比率が小さすぎる場合は近傍幅の変化が大きすぎ、大きすぎる場合は変化が小さすぎ無駄にプロセス数が増加する。そこで、問題の特徴の異なる2つの関数を用いてプロセス数と性能について調べる。Rastrigin関数は大きな局所解を無数に持つ関数であり、Griewank関数は大域的には単峰であり局所的には局所解が無数に存在するという特徴を持つ。また、プロセス数16の場合は隣り合うプロセスとの近傍幅の比率が0.5であり、32の場合は比率が0.7程度となる。図6に結果を示す。図6の横軸がプロセス数、縦軸がエネルギー値を示す。この結果より、最適なプロセス数(比率)は問題の特徴により異なるのではなく、比率が0.5である16プロセス以上は必要であると言える。また、プロセス数を増やしすぎても大きく精度の影響しないことより32プロセス程度で十分であると考えられる。

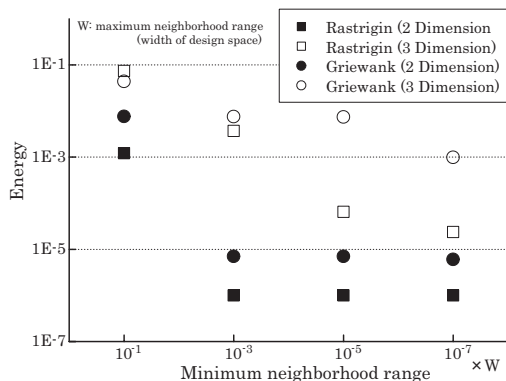


Fig.5 Minimum Neighborhood range

解の生成、受理判定、状態遷移、クーリング (Gener-

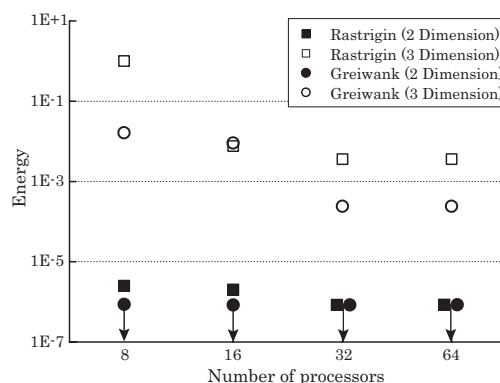


Fig.6 Number of Processors using NPSA

ate next state, Acceptance criterion, Transition, Cooling) :

解の生成 (Generate next state) では、現状態から近傍の範囲内に次状態を生成する処理を行う。受理判定 (Acceptance criterion) では、現状態から次状態へ推移するかを Metropolis 基準に基づいて判断を行う。受理された場合は、次状態への遷移 (Transition) を行う。これらの処理をクーリング (温度の冷却) 期間まで繰り返し、クーリング (Cooling) を行う。これらの処理は通常の SA と同じである。

同期 (Synchronize) :

適応的に近傍を調節するため、プロセス間で同期 (Synchronize) をとり、各プロセスがもつエネルギー値を1つのプロセスに集める。

エネルギー値のソート (Sort energy) :

1つのプロセスに集められたエネルギー値を降順(または昇順)にソートする。

近傍幅の再定義 (Assign neighborhood) :

降順(または昇順)にソートされたエネルギー順に、初期設定で定めた近傍幅の種類から大きな(または小さな)近傍幅が割り当てられるよう、近傍幅とエネルギー値を対応付ける。エネルギー値に対応する近傍幅を元のプロセスに反映させるため、エネルギー値を集めたプロセスから各プロセスに近傍幅の通信を行い、各プロセスの近傍幅の再定義 (Assign neighborhood) を行う。

4. 提案手法の有効性の検証

提案手法における SA の性能および有効性を検証するために、提案手法である NPSA、並列 SA (Parallel SA:PSA)、Corana の手法 (Corana)、SA/AAN、一般的な逐次 SA (SA)、ランダムに探索空間を全探索する手法 (Random) を異なる特徴を持つ3つの標準テスト関数に適用し解探索能力、解探索の履歴および探索に伴う近傍幅履歴について比較を行う。また、提案手法は次元ごとの探索を行うメカニズムを組み込んでいないため、対象とする次元数は2, 3および5次元とする。

4.1 対象問題 対象問題として式(1)に示す Ras-

trigin 関数⁽⁹⁾，式(2)に示す Griewank 関数⁽¹²⁾ および式(3)に示す EggHolder 関数⁽¹³⁾の3つの標準テスト関数を用いる．

Rastrigin 関数，Griewank 関数の最適解は原点に存在し，その時の関数値は0である．Rastrigin 関数は局所解が格子状に存在する多峰性の関数であり，2次元の場合，100個の局所解を持つ設計変数間に依存関係のない多峰性の関数である．Griewank 関数は設計変数間に中程度の依存関係を持ち，大域的には単峰的な関数のようであるが，局所的に無数の局所解を持つ関数である．一方，EggHolder 関数は多峰性で極めて複雑なランドスケープ持っている関数であり，最適解，その時の関数値は設計空間の次元に依存し，その値は既知でないが，ここでは最小値を求める未知の問題として考え，提案手法の有効性の検証に用いる．

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

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

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

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

$$f_E(\vec{x}) = \sum_{i=1}^{n-1} \left(-x'_i \sin \left(\sqrt{|x'_i - P|} \right) - P \sin \left(\sqrt{|P + x'_i/2|} \right) \right) \quad (3)$$

$P = x'_{i+1} + 47$,
定義域 : $-512 < x_i \leq 512$

4.2 温度パラメータ 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，2次元のテスト関数において温度の調節を行った．その決定方法について述べる．

近傍幅の範囲内に次状態を生成される分布として，式(4)のような正規分布を用いる場合は，最高温度の決定方法として，各変数の標準偏差を基に最高温度を決める方法がある．この方法では，標準偏差を変数の設計空間の1/4と設定する．これにより最高温度で設計空間内を十分に探索可能と考えられる⁽⁸⁾．

しかし，今回は分布として一様分布を用いているため，標準偏差を用いた決定方法を用いることができない．そこで近傍をプロセスに割り当てる最大の近傍とし，最高温度におけるアニーリング時の解探索領域と問題空間の定義域との差が最小になるように，予備実験を行い最高温度を決める．最高温度が高すぎる場合は，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，計算効率を下がる．ところが，

最高温度が低すぎる場合は，定義域に探索されない領域が生じるためである．

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．そこで，予備実験によって，求める精度と同じオーダーの解を得られる温度を最低温度とする．

$$g_k(\Delta x) = \frac{1}{(2\pi T_k)^{D/2}} \exp \left(\frac{-|\Delta x|^2}{2T_k} \right) \quad (4)$$

4.3 各パラメータ設定

問題に適応する近傍幅を持つSAの性能を評価するために，式(1)，(2)および(3)に示す3つのテスト関数について表1および2に用いたパラメータを示す．クーリング回数は32とし，PSAとNPSAの並列プロセス数も同じく32とした．一方，NPSAにおける各プロセスの近傍幅は最大近傍幅を設計空間の幅，最小近傍幅をその 10^{-5} とし，その間の近傍幅は指数的に割り振った．また，同期をとってNPSAの近傍調節を行なう周期はPSAのクーリング周期に合わせた．また，並列手法と逐次手法の探索回数は等しくした．

並列プロセス数とクーリング回数とを等しくする理由は特になく，近傍幅が大きく変化しないという条件のもとで任意の数に設定できる．その他のパラメータに関しては，表1および2に示す．

なお，乱数はrand48を用いた．この乱数は，Martin Birgmeierらによって作成された48ビットの線形擬似乱数生成関数で発生させる．この乱数の詳細は文献⁽¹⁴⁾を参照されたい．

Table 1 Parameters (NPSA, PSA)

Function	Rastrigin	Griewank	EggHolder
最高温度	10	20	10
最低温度	0.01	0.001	0.01
クーリング周期	320	960	960
クーリング率	0.8	0.726	0.8
プロセス数	32		
最小近傍幅	設計空間の幅 $\times 10^{-5}$		

Table 2 Parameters(Corana,SA/AAN,SA,Random)

Function	Rastrigin	Griewank	EggHolder
最高温度	10	20	10
最低温度	0.01	0.001	0.01
クーリング周期	320*32	960*32	960*32
クーリング率	0.8	0.726	0.8
プロセス数	1		
最小近傍幅	設計空間の幅 $\times 10^{-5}$		

5. 実験結果および考察

5.1 解探索能力 解探索能力に関する実験結果を図7, 8, 9および表3に示す．Rastrigin 関数に適用した場合に得られた結果を図7に，Griewank 関数に適用した場合の結果を図8に，EggHolder 関数に適用した場合の結果を図9に示す．表3にRastrigin 関数とGriewank 関数の最適解取得率を示す．最適解取得率は

NPSA と NPSA の次に良い性能を示した SA/AAN の結果を示す。図 7, 8, 9 の結果には問題の難しさを示すために、ランダムに探索空間を全探索する手法 (Random) の結果も示している。これにより、これらの問題はランダムに探索空間を全探索するしらみつがし的な探索では求める精度が得られないということが分かる。

図の横軸は次元数、縦軸はエネルギー値 (評価値) を示し、エネルギー値は値が小さい程、良好な結果を示す。これらの結果は、30 回試行の中央値を用いている。中央値を用いた理由は、複数の局所解が存在し、それらの関数値に大きな差がある場合には平均値でなく中央値で比較する方が位置母数の推定量として頑健であるからである。

一般的な SA の手法と SA の改良手法である Corana の手法 (Corana) および SA/AAN と NPSA を比較すると図 7, 8 および 9 より、Rastrigin 関数, Griewank 関数, EggHolder 関数ともに NPSA は良好な結果を示していることが分かる。並列手法である PSA と NPSA を比較すると、NPSA は PSA より明らかに良好な結果を示していることが分かる。Corana の手法と SA を比較すると、SA の改良手法である Corana の手法が SA より劣っている結果があるが、SA は最適な近傍幅を予備実験により求めているためである。

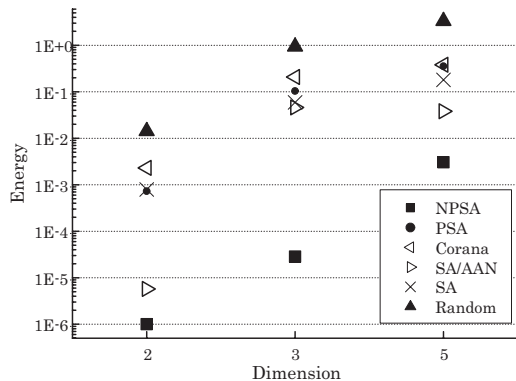


Fig.7 Performance of the methods(Rastrigin)

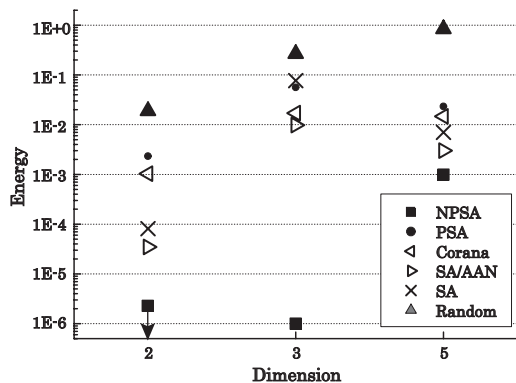


Fig.8 Performance of the methods(Griewank)

Table 3 Success ratio on NPSA and SA/AAN

Dimension	Rastrigin		Griewank	
	NPSA	SA/AAN	NPSA	SA/AAN
2	0.33	0	1	0.2
3	0.17	0	0.53	0
5	0	0	0	0

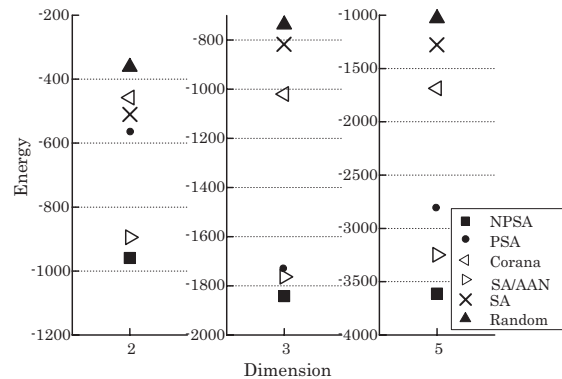


Fig.9 Performance of the methods(EggHolder)

5.2 解探索の履歴および近傍履歴 NPSA と最も標準的で性能の高い PSA の近傍履歴とエネルギー履歴を比較し、NPSA の有効性を考察する。対象問題には 2 次元の Rastrigin 関数を用いた。図 10 に NPSA のエネルギー履歴、図 11 に PSA のエネルギー履歴を示す。図 11 の横軸がアニーリングステップ数 (評価回数)、縦軸がエネルギー値を示し、Current Energy は探索点のエネルギー値を、Best Energy は Current Energy の履歴から求めた最良のエネルギー値を示す。

図 10 および 11 より NPSA が PSA よりよい解が得られていることが分かる。また、NPSA は PSA より早く局所解から抜け出しており、これにより最適解領域での探索が PSA より十分に出来ているといえる。

その理由は図 12 に示す近傍履歴から明らかである。図 12 は 32 プロセス中最終的に最も良いエネルギー値を得たプロセスの履歴を示し、横軸にアニーリングステップ数、縦軸に近傍を示している。PSA は近傍の交換を行わない手法であり、予備実験より求めた最適な近傍幅である 1 を近傍幅として用いる。PSA は予備実験により求めた最適な近傍で探索を行っているが、最適解領域に NPSA と比べ速く収束していない。また、NPSA と比べエネルギー値が悪くなる方向に推移しやすくなっていることから、最適解領域に探索点が入ってもすぐに抜け出してしていることが分かる。そのため、探索終盤になり温度が下がっても、最適解領域での局所探索が十分出来ていないと考えられる。

一方、NPSA は近傍交換時に各プロセスの探索点の解をソートし、エネルギー値が大きい解には大きい近傍を与え、エネルギー値が小さい解には小さい近傍を与えるという、近傍を適応的に変化させるメカニズムを持つため、最適解領域に入っている解では局所探索

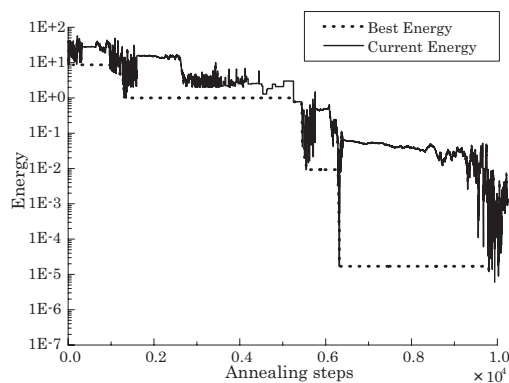


Fig.10 History of energy(NPSA)

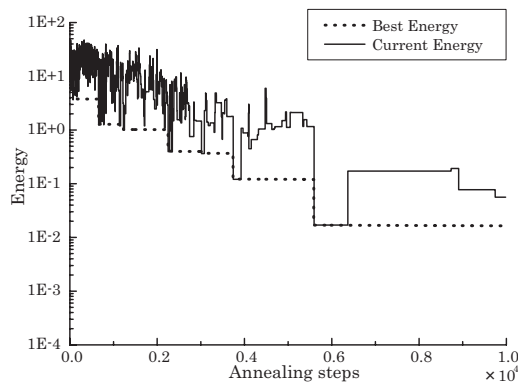


Fig.11 History of energy(PSA)

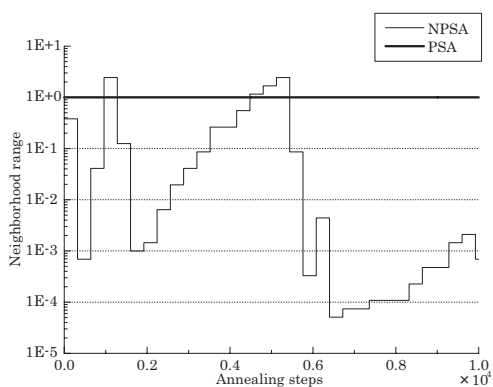


Fig.12 History of neighborhood range

を、局所解領域に入っている解では大域的探索が行える。図12の結果から、探索前半では局所解領域で探索を行っているため、局所解から脱け出すために、徐々に近傍が大きくなっていることが分かる。局所解から脱け出し最適領域に入った探索中盤では、局所探索のために近傍幅が徐々に小さくなり、探索終盤では小さな近傍幅で局所探索を十分行っていることが分かる。

6. まとめ

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場合、近傍の調整が必要不可欠となる。本研究

では、従来の適応的近傍メカニズムで必要とされる多くの近傍調節のためのパラメータを減らし、かつ、並列処理を用いることで処理時間の短縮が期待できる新しい適応的近傍並列アルゴリズム (NPSA: Neighborhood Parallel Simulated Annealing) を提案した。そして実験結果より提案手法がシミュレーテッドアニーリングの拡張アルゴリズムとして有効であることを確認した。

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) 喜多一. シミュレーテッドアニーリング. 日本ファジィ学会誌, (1997).
- (3) Aarts, E. and Korst, J.: Simulated Annealing and Boltzmann Machines, John Wiley & Sons, (1989).
- (4) 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).
- (5) Ingber, L.: Simulated Annealing: Practice versus theory, Journal of Mathl. Comput. and Modelling. Vol.18, No.11, pp29-57 (1993).
- (6) 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).
- (7) 三木光範, 廣安知之, 小野景子: 最適な受理確率を目標とするを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 Vol.44, No.1, pp1-6 (2003).
- (8) 三木光範, 廣安知之, 笠井誠之, 小野景子: 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, 情報処理学会誌 Vol.42, No.4, pp745-753 (2003)
- (9) Bruce E. Rosen, 中野 良平: シミュレーテッドアニーリング, 人工知能学会誌 Vol.9, NO.3, pp.365-371 (1994).
- (10) Szu, H. and Hartley, R.: Fast Simulated Annealing, Physics Letters A, Vol.122, No.3,4, pp.157-162 (1987).
- (11) Rosen, B.: Functional Optimization based on Advance Simulated Annealing, IEEE Workshop on Physics and Computation, PhysComp 92 (Dallas, Texas), pp. 289-293 (1992).
- (12) Whitley, D., Mathias, K., Rana, S. and Dzubera, J.: Evaluating Evolutionary Algorithms, Artificial Intelligence, Vol. 85, pp. 245-276 (1996).
- (13) <http://www.ft.utb.cz/people/zelinka/soma/func.html>
- (14) <http://www.ics.uci.edu/eppstein/projects/pairs>