

## 設計変数ごとに適応的近傍調節機能を持つ並列シミュレーテッドアニーリング

Parallel Simulated Annealing with Independent Adaptive Neighborhood determined by Evolutionary Computation

宮崎 真

Masashi MIYAZAKI

**Abstract:** Simulated annealing (SA) is an effective general heuristic method for solving many optimization problems. This paper deals with two problems in SA. One is the long computational time of the numerical annealings, and the solution to it is the parallel processing of SA. The other one is the determination of the appropriate neighborhood range in SA, and the solution to it is the introduction of an adaptive mechanism for changing the neighborhood range. The multiple SA processes are performed in multiple processors, and the neighborhood ranges in the SA processes are determined by the evolutionary computation. The proposed method is applied to solve many continuous optimization problems, and it is found that the method is very useful and effective.

## 1 はじめに

Simulated Annealing (SA)<sup>1)</sup> は Metropolis らが 1953 年に発表した焼き鈍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、特に組合せ最適化問題を解く汎用近似解法の 1 つとして用いられている<sup>2, 3)</sup>。SA は組合せ最適化問題に特に有効な手法であるが、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は多く用いられている。しかし、SA におけるパラメータの影響は大きく、特に近傍と温度の調節によって解の精度は大きく異なる<sup>4)</sup>。

連続最適化問題に SA を適用する場合には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できる。一般的に、その近傍幅の大きさにより、エネルギーの変化量も変化し、しかも、その変化量はエネルギー関数に大きく依存する。このため、連続最適化問題に SA を適用する場合、近傍幅の設計は探索性能に大きく影響する。このような問題に対して、SA における解の受理率を用いて近傍幅の調節を行うアルゴリズム<sup>3, 5)</sup>も提案されているが、これらは受理率に依存した調節メカニズムであり、目標とする受理率を予備実験より決定する必要がある。

また、SA には膨大な計算量を要するという欠点がある。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様々な研究がなされている<sup>6)</sup>が、いずれの方法にしても SA のパラメータに関しては経験的にしか与えられていないという問題は残る。

そのため、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注目し、近傍の並列化という新たなアプローチから適応的な近傍幅の調節を行う並列

SA を提案する。そして、代表的な数学関数最小化問題に提案手法を適用し、その有効性を検証する。

近傍幅の調節メカニズムには進化的計算法 (Evolutionary Computation) を用い、並列 SA 上への実装により、近傍幅の調節を自動化できる。また、近傍幅の並列化により並列処理も可能であり、計算量の問題も克服できる。

## 2 最適な近傍幅の設定

連続最適化問題において近傍幅が解精度に与える影響を検証するために、数学的テスト関数を取り上げた。そして、これらの問題に様々な種類の近傍幅を与え、アニーリングを複数回数試行し、それぞれの近傍幅で得た解の精度を比較した。Fig. 1 に Rastrigin 関数に対する実験結果を示す。横軸は各試行での近傍幅、縦軸にそのときに得られたエネルギー値を示す。本実験は最小化問題であるため、エネルギーが低い方が良好である。なお、結果は 30 回試行の中央値である。

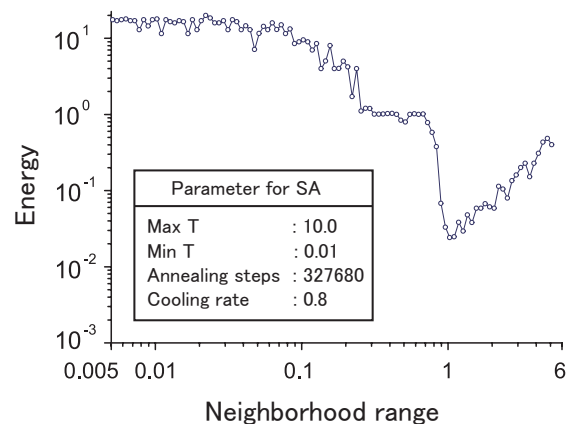


Fig. 1 Rastrigin 関数に対する近傍幅と得られた解精度の関係

Fig. 1 より, Rastrigin 関数では近傍幅が 1.0 付近で最も良好な解が得られた。今回取り上げた他の問題に対して同様の実験を行った結果, 同様の傾向が得られた。しかし, 最適な近傍幅の値は各問題や設計変数の数にも依存することがわかった。

### 3 近傍の並列化

SA を連続最適化問題に適用する場合, 解の精度は近傍幅に大きく依存していること, また, 対象問題の中にはそれぞれ効率的な探索を行うことができる最適な近傍幅が存在することが分かった。しかし, 通常これらの最適な近傍幅は未知であるため, 決定するには膨大な予備実験が必要である。そこで, 探索過程において最適な近傍幅を自律的に決定し, 予備実験を必要としない方法について考える。

最適な近傍幅が未知であるなら, 探索空間に対して複数の近傍幅で同時並列に探索を行うことで, その探索空間上での適切な近傍幅を検出することができ, 良好な探索が可能であると考えられる。これらのことより, SA における近傍の並列化を提案する。近傍の並列化を行うには, SA 自体の並列化が必要となる。並列 SA において, これまでに提案されている手法は以下の通りである。

#### 1. 独立型並列 SA(PSA/FN)<sup>7)</sup>

PSA/FN(Parallel SA/ Fixed Neighborhood) は, 複数のプロセスに異なる初期解を与え, 一定の近傍幅で並列に実行する。最終的に得られた解のうち最も良好なプロセスの解を出力する。

#### 2. 温度並列 SA(TPSA/FN)<sup>8)</sup>

TPSA/FN(Temperature Parallel SA/ Fixed Neighborhood) は, 複数のプロセスに一定の近傍幅と, それぞれ異なる温度を与え, 周期的にプロセス間で解交換を行う並列 SA である。この手法は並列処理との高い親和性だけでなく, 原理的に温度スケジュールが不要である優れた特徴を有している。

#### 3. 適応的近傍を持つ温度並列 SA(TPSA/AN)<sup>9)</sup>

TPSA/AN(Temperature Parallel SA/ Adaptive Neighborhood) は, 適応的近傍を持つ SA と温度並列 SA を組み合わせ, 近傍幅を受理率が 0.5 になるように調節することで近傍幅を自動化し, さらに温度並列 SA により温度スケジュールも自動化している。

これらの手法のうち, 3 の手法は既に適応的近傍を有しているため, 近傍の並列化を行うことができる方法は 1 と 2 である。ここでは近傍の並列化の第一段階として, 1 の方法を基に, 近傍の並列化を行う。

手法 2 の方法に並列化を組み合わせると, 温度の並列化とあわせて, 二次元的な並列化となり, 必要なプロセス数は非常に大きくなる。

## 4 進化的手法を用いた適応的近傍調節機能を持つ並列 SA

### 4.1 基本的概念

近傍調節メカニズムの新しいアプローチとして近傍の並列化を提案する。次に並列化した近傍幅の調節メカニズムを考える。SA における探索は一般的に探索初期では大域的探索を行う必要があるが, 探索終盤においては解の精度を高めるためにある程度小さな近傍幅が必要となる。また, 解の探索状況に応じて必要な近傍幅は異なるため, 初期に設定を行った近傍幅を調節するメカニズムが必要となる。

このようなことを考えた場合, 環境に応じて適応的に変化を生み出せるアルゴリズムに進化的計算手法 (Evolutionary Computation: EC) がある。EC は有利な特徴をもつ個体の情報を伝えることができる自然淘汰の理論にもとづいている。この EC を SA における近傍調節メカニズムに組み込むことで上記の問題点を解決できると考え, 並列化した SA プロセスの近傍幅調節に進化的計算手法を用いた手法を提案する。本論文ではこの手法を進化的計算手法を用いた適応的近傍調節メカニズムを持つ並列シミュレーテッドアニーリング (Parallel Simulated Annealing with Adaptive Neighborhood Determined by Evolutionary Computation: PSA/AN(EC)) と呼ぶ。

PSA/AN(EC) の概念図を Fig. 2 に示す。提案手法のアルゴリズムは, 独立に動作する複数の SA を基本としており, それらの SA に探索空間に対して異なる固有の近傍幅を設定し, 独自に SA を実行する。初期設定の近傍幅で一定期間アニーリングを行った後, 全プロセスで同期をとる。この同期時に各プロセスが持つ近傍幅を個体と見立て, EC オペレータを適用する。この EC オペレータの適用により, 良好な探索を行っている近傍幅が選択され, そうでない近傍幅は淘汰される。

EC 適用後, 新たに生成される近傍幅の集合を次周期における各 SA プロセスの近傍幅に設定し, 再び一定期間アニーリングを行う。この操作を終了条件まで繰り返すことにより, 探索状況に応じて適切な近傍幅で各 SA プロセスは探索を行うことが可能となる。この近傍幅に対しての EC の適用方法については後述する。

### 4.2 PSA/AN(EC) のアルゴリズム

PSA/AN(EC) のアルゴリズムを Fig. 3 に示す。

- 生成処理, 受理判定, 推移

生成では, 現在の状態から近傍範囲内で次状態を生

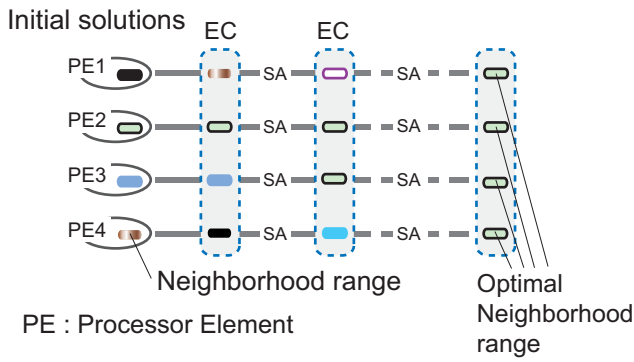


Fig. 2 進化的計算手法を用いた適応的近傍並列 SA

成する処理を行う。受理判定では、生成した次状態へ遷移するかどうかを Metropolis 基準を用いて判定し、受理された場合は遷移、そうでない場合は新たな次状態を生成する。これらの処理は通常の SA と同じである。

- 同期  
一定周期ごとに全プロセスが同期をとり、EC オペレータを適用する。
- 冷却処理  
一定期間アニーリングを行った後、冷却処理を行う。

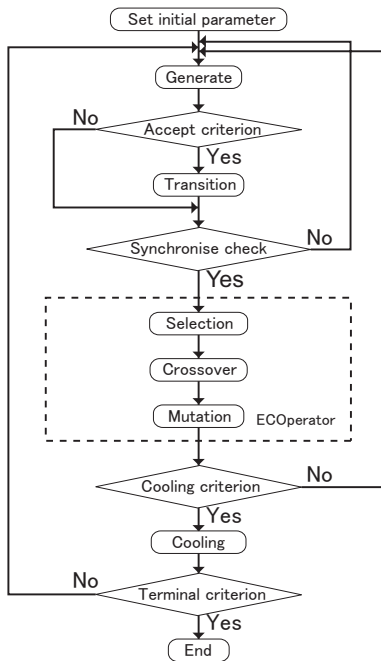


Fig. 3 PSA/AN(EC) のアルゴリズム

#### 4.3 EC による近傍調節

各 SA プロセスは一定期間アニーリングを行った後、全てのプロセスで同期をとり、その際に各 SA プロセス

が持つ近傍幅を個体、エネルギー値を評価値として進化的操作を適用する。この進化的操作は選択、交叉、突然変異より形成される。選択処理では評価値に基づき選択が行われ、これにより、良好な探索を行っている近傍幅が選択される。このとき評価値は各 SA プロセスが持つ最良のエネルギー値 (*Energy*) をもとに以下の式 (1) で算出される。

$$Fitness = \frac{1}{Energy} \quad (1)$$

なお、本研究では選択方法にはトーナメントサイズ 2 のトーナメント選択法を用いた。個体の遺伝子は 10 ビットの 0, 1 で表現し、交叉法は 1 点交叉を用いた。また、突然変異では 1 ビットを確率的に反転させることを行う。ここで、ビット表現の遺伝子から実数値である近傍幅へのデコードはグレイコードを用いた。また、近傍幅はビットからデコードされた値を指数的に変化させることにより決定した。

#### 4.4 PSA/AN(EC) の性能

PSA/AN(EC) の性能を検証するために、Rastrigin 関数、Griewank 関数、Rosenbrock 関数を対象に実験を行った。Rastrigin 関数は局所解が格子状に存在する関数であり、設計変数が 2 の場合、100 個の局所解を持つ。Griewank 関数は設計変数間に依存関係を有する多峰性関数である。大域的には単峰性関数のような性質をもつが、局所的には多数の局所最適化が存在し、最適解を発見するのは困難である。Rosenbrock 関数は設計変数間に依存関係のある単峰性関数である。

比較手法は SA を独立に同時並列に実行する独立型並列 PSA/FN、複数のプロセスに異なる一定温度を設定し、温度調節を自動化した温度並列 SA (TPSA/FN)、および Corana らが提案した適応的近傍を持つ SA を温度並列 SA に応用した手法 (TPSA/AN) である。近傍調節機能を持たない PSA/FN と TPSA/FN は、近傍幅は各問題ごとに、予備実験より求めた最適な近傍幅を用いた。

なお、PSA/AN(EC) の近傍幅の数は SA プロセスの数と同じであり、全てのアルゴリズムで並列数を 32 とした。最高温度は改悪受理確率が 50% となるように予備実験から求めた。

この結果を Fig. 4 に示す。この結果は 30 回試行の中央値を用いている。図より、全ての問題において PSA/FN、TPSA/FN よりも近傍調節機能を有する TPSA/AN と提案手法である PSA/AN(EC) が良好である。中でも PSA/AN(EC) が全ての手法の中で最も性能の高い手法であるといえる。

#### 4.5 EC による近傍調節メカニズムの効果

PSA/AN(EC) の解探索性能が高いことが分かった。続いて、EC による近傍調節メカニズムについて検証を行う。

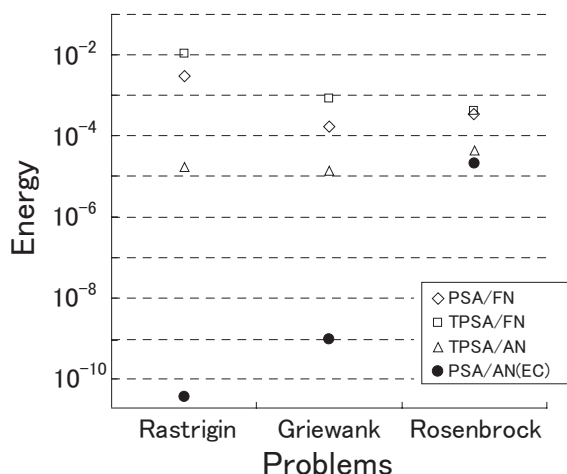


Fig. 4 PSA/AN(EC) の解探索性能

Fig. 5 に Rastrigin 関数を対象とした際の PSA/AN(EC) の 32 プロセスの近傍幅履歴の典型的なものを示す．図中の縦軸は近傍幅，横軸はアニーリングステップ数を示している．Fig. 5 より，Rastrigin 関数には最適な近傍幅が存在したが，PSA/AN(EC) では探索初期では大域的な探索を行う近傍幅に収束している．そして，最適な固定近傍幅を経由して最適化が進むと，さらに精度を向上するために適応的に近傍幅が変化していることがわかる．

次に，Rastrigin 関数において，PSA/AN(EC) の全プロセスの中で最も良好な解を得たプロセスの探索履歴と近傍幅履歴を Fig. 6 に，局所解に陥ったプロセスの履歴を Fig. 7 に示す．

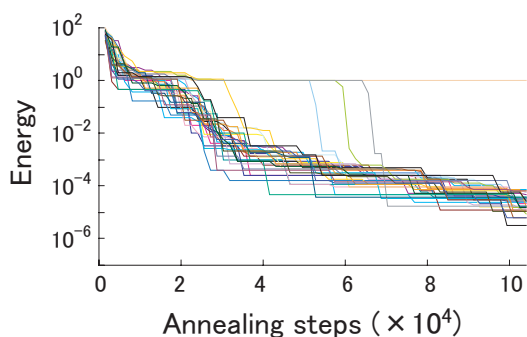


Fig. 5 Rastrigin 関数における PSA/AN(EC) の解探索履歴

Fig. 6 より，最も良好な値を得たプロセスは局所解に陥ることなく良好な探索が行えていることが分かる．このとき，近傍幅は EC により調節されているが，探索終盤においては局所探索を行うことができる小さい近傍幅を得ていることが確認できる．一方，Fig. 7 より局所解に陥ったプロセスは， $6 \times 10^4$  ステップ付近までは局所

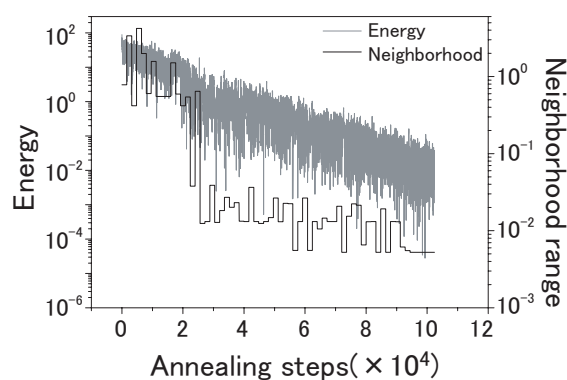


Fig. 6 PSA/AN(EC) における良好な解を得たプロセスのエネルギーと近傍幅の履歴

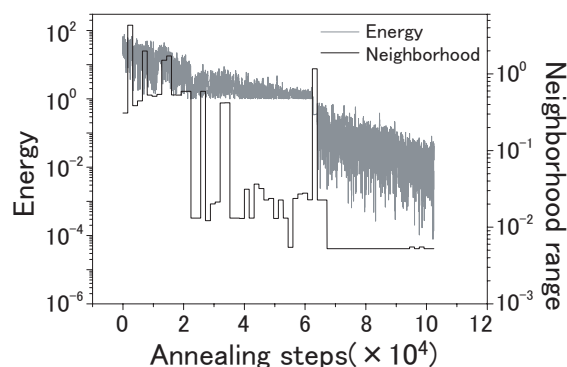


Fig. 7 PSA/AN(EC) における局所解に陥ったプロセスのエネルギーと近傍幅の履歴

解に陥っており，最適化を進めることができていない．しかし，EC により適応的に近傍幅が調整され，局所解を抜け出すことができることがわかる．これより，EC を用いた近傍調節メカニズムは探索状況に応じて適応的に近傍幅を調節できるメカニズムであり，有効に機能しているといえる．

## 5 設計変数ごとに適応的近傍調節機能を持つ 並列 SA

### 5.1 基本的概念

連続最適化問題では，目的関数のランドスケープに応じて適切に近傍幅を調節することで，高い探索能力を得ることができる．このようなことを考えた場合，複雑なエネルギー景観をもつ問題に対しては，設計変数の持つスケールも様々であることが予想される．つまり，探索空間を形成する各設計変数のランドスケープを考慮することが，探索空間全体のランドスケープを考慮することに繋がる．

また，探索状況により，各設計変数における適切な近傍幅はそれぞれ異なることも考えられる．よって，各設

計変数ごとに近傍幅を適応的に調節することで、より効率的な探索ができると考えられる。

しかし、前節で述べた PSA/AN(EC) では全設計変数で共通の近傍幅を用いている。そこで、PSA/AN(EC) に設計変数ごとに独立した近傍幅を与えるアルゴリズムを組み込むことを提案する。このアルゴリズムを PSA/AN(EC)2 とする。

### 5.2 PSA/AN(EC)2 のアルゴリズム

PSA/AN(EC)2 では、Fig. 8 に示すように、各設計変数の近傍幅を個体を構成する要素としている。また、各設計変数の近傍幅をビット表現ではなく、実数値表現を用いた。

交叉は一点交叉で、個体の要素に対して行うため、個体における近傍幅の組合せが異なる個体が生成される。突然変異では、個体を構成する要素を指数的に変化させた。

### 5.3 数値実験

PSA/AN(EC)2 の性能を検証するために、PSA/AN(EC) と比較を行った。対象問題は Rastrigin 関数、Griewank 関数および Rastrigin 関数の  $i$  番目の設計変数を  $1/i$  倍とした Original 関数とし、設計変数の数を 3 とした。

実験結果を Fig. 9 に示す。結果は 30 回試行の中央値をプロットしている。この図より、一般的に PSA/AN(EC)2 が良好な結果を得たことがわかる。

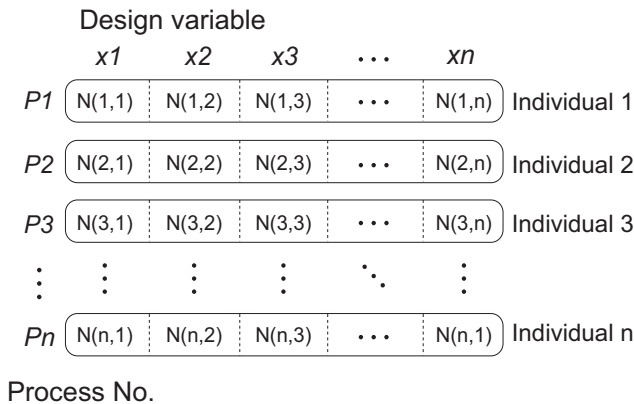


Fig. 8 PSA/AN(EC) における個体

続いて、設計変数の数を 10 とした場合の結果を Fig. 10 に示す。この図より、高次元になると SA および PSA/AN(EC) では十分な精度が得られないが、PSA/ANE(EC)2 ではより良い精度が得られるといえる。よって、提案手法は高次元においていっそう有効な手法であると考えられる。

Fig. 11 に Rastrigin 関数の 10 個の変数のうち、 $x_3$  と  $x_9$  の設計変数における近傍幅の例を示す。この図より、

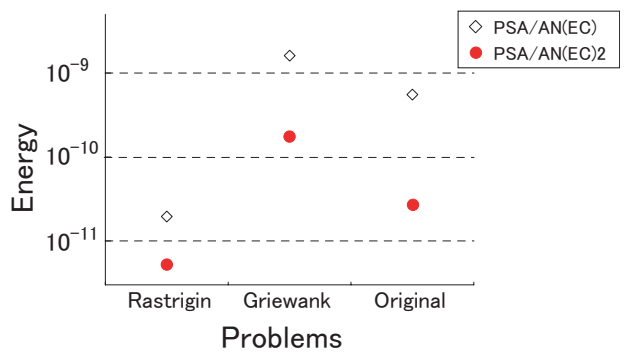


Fig. 9 設計変数の数を 3 とした場合の性能比較

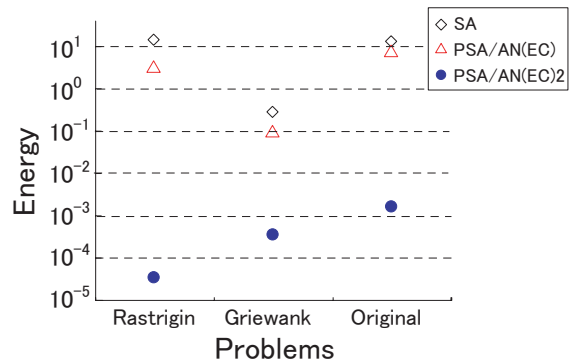


Fig. 10 設計変数の数を 10 とした場合の性能比較

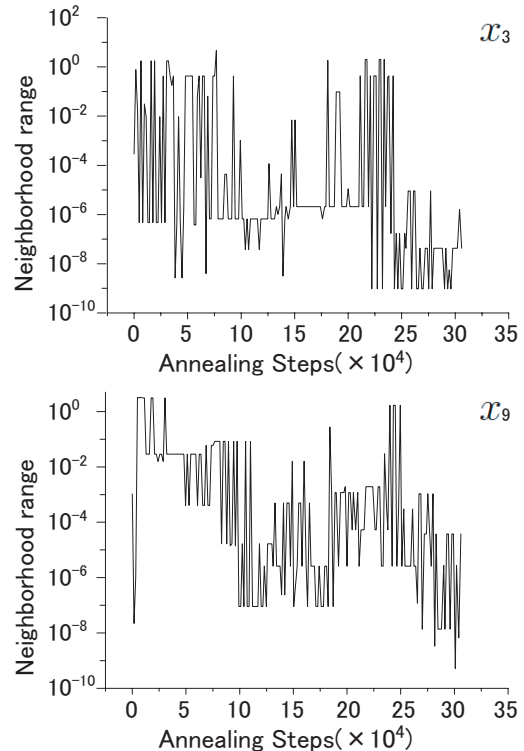


Fig. 11 Rastrigin 関数における近傍履歴



ECによって設計変数ごとに近傍幅が適切に調節されていることが確認できる。このことから、探索状態により、設計変数ごとに適切な近傍幅を調節することは有効であるといえる。

#### 5.4 交叉率の検討

提案手法におけるECオペレータのうち、交叉率が解へおよぼす影響を調べた。Fig. 12にその結果を示す。なお、結果は30回試行の最大値、中央値、最小値をプロットしている。

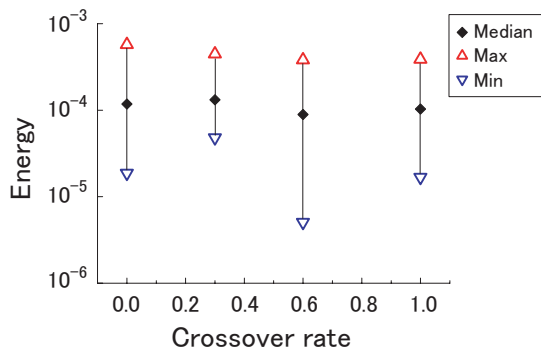


Fig. 12 交叉率の変化による解への影響

図より、交叉率の変化では解に影響が見られず、このため今回の実験で用いた一点交叉法の効果はないと考えられる。この原因は、ある探索点においての最適な近傍幅というものは、他の探索点においての最適な近傍幅であるとは限らないためと考えられる。

## 6 まとめ

本研究では、連続最適化問題へSAを適用する場合に重要となる近傍幅に着目し、設計変数ごとに独立した適応的近傍調節機能を持つ並列SAを提案した。提案手法では、設計変数ごとに独立した近傍幅を持っており、複雑なランドスケープを持つ関数に対して有効であることが分かった。また、次元数が大きい問題に対しても有効であることが分かった。これらのことは、進化的手法(EC)により、各設計変数における近傍幅が探索状況に応じて最適な値に変化したことによる。

今後の課題としては、PSA/AN(EC)2において、他の交叉手法を適用し検討することである。また、提案手法では設計変数間に依存関係のない問題に対し特に有効であるが、今後、設計変数間に依存関係のある問題に対しても検討することが必要である。

## 参考文献

1) Kirkpatrick, S., Gelett Jr. C. D., Vecchi, M. P. Optimization by simulated annealing. *Science*, Vol. 220, No. 4598, pp. 671-680, 1983.

2) Ingber, L. Genetic algorithms and very fast simulated reannealing. A Comparison, *Mathematical and Computer Modeling*, Vol. 16, No. 11, pp. 87-100, 1992.

3) Corana, A., Marchesi, M., Martini, C., 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.

4) 白石洋一 訳 Sadiq M. Sait. 組合せ最適化アルゴリズムの最新手法 - 基礎から工学応用まで -. 丸善株式会社, 2002.

5) 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. *情報処理学会論文誌*, Vol. 44, No. 1, pp. 1-6, 2003.

6) Holmqvist, K., Migdalas, A., Pardalos, P. M. Parallelized heuristics for combinatorial search, in *parallel computing in optimization*, 1997.

7) 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.

8) 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.

9) 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. *情報処理学会論文誌*, Vol. 42, No. 4, pp. 745-753, 2001.