

遺伝的交叉を用いた並列シミュレーテッドアニーリングの検討

廣 安 知 之[†] 三 木 光 範[†]
小 掠 真 貴^{††} 岡 本 祐 幸^{†††}

本研究では、遺伝的交叉を用いた並列シミュレーテッドアニーリング (PSA/GAc) を提案する。PSA/GAc は、複数のプロセス上で並列にシミュレーテッドアニーリングの操作を行い、一定期間のアニーリングを行った後、2 プロセス間で遺伝的アルゴリズムのオペレータである遺伝的交叉により情報交換を行う。その後再び一定期間のアニーリングを行い、遺伝的交叉を行う操作を繰り返す。情報交換に遺伝的交叉を利用することにより、大域的にはいくつかの準最適解が存在し、局所的には無数の準最適解を有するような問題に特に有効であると考えられる。数種のテスト関数に適用した結果、PSA/GAc の優れた性能が明らかとなった。また、実問題への適用例として最適化問題の一つであるタンパク質のエネルギー最小化計算を行い、従来用いられていた手法と比較し、PSA/GAc の有効性を明らかにした。

Examination of Parallel Simulated Annealing using Genetic Crossover

TOMOYUKI HIROYASU,[†] MITSUNORI MIKI,[†] MAKI OGURA^{††}
and YUKO OKAMOTO^{†††}

This paper proposes Parallel Simulated Annealing using Genetic Crossover (PSA/GAc). In this algorithm, there are several processes of Simulated Annealing (SA) working parallel. To exchange information between the solutions, the operation of genetic crossover is performed. Through the continuous test problems, it is found that PSA/GAc can search the solution effectively. The proposed algorithm is also applied to the minimization of protein energy function. Comparing PSA/GAc to the conventional algorithm, it is also found that PSA/GAc is effective algorithm for real world problems.

1. はじめに

シミュレーテッドアニーリング (Simulated Annealing: SA) は最適化問題を解く手法の一つであり、多くの組合せ最適化問題に対して有効な手法である¹⁾。SA の探索は最適解へ収束するという保証を有するが、解を得るまでの計算量が非常に多いという欠点を持つ。特に連続最適化問題を対象とした場合、最適解を得るまでに多くの計算量が必要となり実用的ではない。高速に解を求める手法はいくつか考えられるが、代表的

なものが並列化とハイブリッド化であろう。これまでに、逐次処理である SA を並列化し高速化を図る研究はいくつか行われている¹⁾。SA を並列化する手法は、以下の3つが一般的である²⁾。1つめは、異なるプロセスで独立に SA を行う方法であり、温度を低減するときにすべてのプロセッサから最もよい結果を選択し、各々のプロセッサは新しい温度でこの共通の解から再び探索を始める。2つめの方法は、独立にランダムな近傍解を発生させ、受理判定を行うために独立にプロセッサを使用するものである。1つのプロセッサが受理すべき近傍解を発見した際に、これを他のすべてのプロセッサに伝達し、新しい現在の解の近傍に探索が遷移する。3つめの方法は、共通のメモリで現在の解を記憶し、すべてのプロセッサがそれに基づいて計算を行うものである。各プロセッサは互いに同期することなく逐次 SA を実行し、遷移した結果を共通のメモリに非同期に書き込む。

また、SA と他の最適化手法を組み合わせたハイブリッドアルゴリズムの研究もいくつか行われており、

[†] 同志社大学工学部
Department of Knowledge Engineering and Computer Sciences, Doshisha University

^{††} 同志社大学大学院
Graduate School of Engineering, Doshisha University

^{†††} 岡崎国立共同研究機構 分子科学研究所
Institute for Molecular Science, Okazaki National Research Institutes
総合研究大学院大学
The Graduate University for Advanced Studies

組み合わせる最適化手法として、多点探索を行う遺伝的アルゴリズム (Genetic Algorithm : GA) が挙げられる。SA は局所探索に優れた最適化手法であるが GA は大域探索に優れた手法であるため、これらを組み合わせることで互いの短所を補完することができると思われる。このようなハイブリッドアルゴリズムとして、熱力学的遺伝的アルゴリズム (Thermodynamical Genetic Algorithm : TDGA)³⁾ や Parallel Recombinative Simulated Annealing⁴⁾、遺伝的状態生成処理を取入れた改良型アニーリング法⁵⁾などが提案されており、他にもいくつかの研究が報告されている^{6),7)}。また、GA に局所探索メカニズムを導入したアルゴリズムは特に Memetic Algorithm⁸⁾と呼ばれ、精力的に研究が行われている。しかしこれらのハイブリッドアルゴリズムの多くは、GA のアルゴリズムを中心としたものや、逐次 SA のアルゴリズムを中心としたものであり、並列 SA での探索を基本とし GA のオペレータを取り入れた手法に関する研究は数少ない。

そこで本研究では、連続最適化問題を対象とする場合においても、比較的少ない計算量で良好な解を得ることのできる新たな並列 SA アルゴリズムとして、遺伝的交叉を用いた並列シミュレーテッドアニーリング (Parallel Simulated Annealing using Genetic Crossover : PSA/GAc) を提案する。提案する手法は、並列に複数実行している SA の解の伝達時に、GA のオペレータである遺伝的交叉を用いたハイブリッドアルゴリズムである。局所的な探索が得意な SA に、大域的な探索が得意でかつ部分解の組み合わせで最適解が得られる GA オペレータを取り入れることで、適用可能な対象問題の範囲を拡張することが可能だと考えられる。また、提案手法は並列 SA のアルゴリズムを基本とするため、比較的少ない計算量でも、解の多様性と優れた局所探索能力の維持が可能だと考えられる。

本研究では提案手法をテスト関数に適用させることにより、その性能を評価する。また、最適化問題の一つであるタンパク質のエネルギー最小化計算を行い、その有効性を検討する。

2. SA と GA

2.1 シミュレーテッドアニーリング (SA)

シミュレーテッドアニーリング (Simulated Annealing : SA) の基礎となる考え方は、Metropolis らが 1953 年に発表した焼きなましとよばれる過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する⁹⁾。その後、Kirkpatrick らはこの種のシミュレーションが最適化問題の実行可能解を探索するのに

応用できることを提案した¹⁰⁾。

SA は、高温で熔融状態にある物質を徐々に冷却することにより欠陥の少ない結晶などの低エネルギーの状態を得る物理プロセスを、計算機上で模倣することにより最適化問題を解こうとする汎用近似解法の一つである¹¹⁾。特に組み合わせ最適化問題を解く手法として有効であるが、解の改悪を許すことで局所解からの脱出の可能性があるという特徴から、複数の局所解を持つ連続値最適化問題のための SA アルゴリズムも多く提案されている^{1),10)}。

SA は、理論上は最適解に到達できるという保証を持つが、最適解を得るまでに非常に多くの計算量を要するという欠点も持つ。また、多くのパラメータをチューニングする必要もある。このため、逐次処理である SA を並列化して高速化を図る研究²⁾や、パラメータの自動チューニングの研究¹⁾がなされている。

SA のアルゴリズムでは、現在の状態から次の状態を生成する生成処理 (Generate)、現在の状態から次の状態に遷移するかどうかを判定する受理判定 (Accept Criterion)、現在の温度から次の温度を生成するクーリング (Cooling) の 3 つの過程が重要となる¹⁾。

ある目的関数 $f(x)$ があって、各状態 x に対しエネルギー $f(x)$ が定義されているとする。SA は、与えられた初期状態から探索を始め、状態を遷移させて探索を続けることで最終的にエネルギーが最小となる状態、つまり目的関数の大域的な最小状態を発見することを目的としている。

生成処理では、現在の状態 x から次に遷移すべき状態 x' を返す。状態 x' は確率分布によって状態 x から生成される。状態空間が連続の場合は確率分布として一様分布や正規分布が用いられる。

受理判定では、生成された次の状態 x' のエネルギー $E' = f(x')$ と現在の状態 x のエネルギー $E = f(x)$ との差分 $\Delta E (= E' - E)$ 、および温度パラメータ T が与えられ、次の状態への遷移を受理するか否かを判定する。通常、式 (1) に示す Metropolis 基準¹⁰⁾ が用いられる (ボルツマン因子は 1 とおいた)。

$$P_{\text{Accept}} = \begin{cases} 1 & \text{if } \Delta E \leq 0 \\ \exp\left(-\frac{\Delta E}{T}\right) & \text{otherwise} \end{cases} \quad (1)$$

生成処理、受理判定をある程度繰り返した後クーリングを行う。クーリングでは、アニーリングの第 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。最適解の漸近収束性を保証するためには式 (2) で示される対数型アニーリング以上に急速に冷やしてはならない¹²⁾。

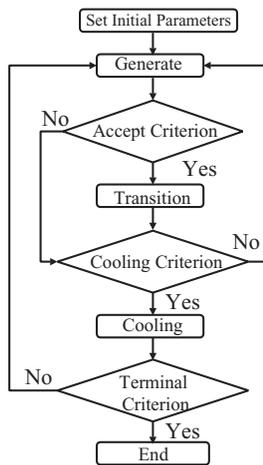


図 1 SA のアルゴリズム

Fig. 1 Algorithm of Simulated Annealing

$$T_{k+1} = \frac{T_k}{\log k} \quad (2)$$

しかし、対数型アニーリングでは収束するまでに膨大な時間を要するため、実用的には式 (3) で示される指数型アニーリングを用いることが多い。

$$T_{k+1} = \gamma T_k \quad (0.8 \leq \gamma < 1) \quad (3)$$

SA の基本的なアルゴリズムを図 1 に示す。

終了条件として最終温度やアニーリングステップ数などが用いられるが、他にも様々な終了条件が考えられている。終了条件に達すれば、その時の状態とエネルギーをそれぞれ最適状態、最適値として出力する。

2.2 遺伝的アルゴリズム (GA)

遺伝的アルゴリズム (Genetic Algorithm : GA) は、1960 年代に Holland らが生物の適応進化を模倣するモデルとして提唱した手法が基礎となっている¹³⁾。1970 年代に入り DeJong による最適化問題を対象とした研究を契機として、1989 年には Goldberg によって GA のアルゴリズムの枠組みが整理された¹⁴⁾。

GA は生物の進化の過程を模倣したアルゴリズムであり、遺伝的な法則を工学的にモデル化したものである。自然界における生物の進化の過程では、いくつかの個体の集合が形成されており、環境への適合度が高い個体が高い確率で選択され生存する。また、交叉によって次世代の個体、つまり、子を生成したり、突然変異によって性質の異なる個体が生成されたりもする。GA のアルゴリズムでは、複数の個体が、目的関数値に基づいて適合度の計算を行い、その後選択 (Selection)、交叉 (Crossover)、突然変異 (Mutation) からなる遺伝的オペレータとよばれる操作を行って、適合

度の高い個体が生存し次世代に増殖する。これらの一連の操作周期は世代と呼ばれる。以下にそれぞれの遺伝的オペレータについて詳しく述べる。

選択方法にはいくつかの種類があるが、代表的な方法としてルーレット選択が挙げられる。ルーレット選択は Holland¹³⁾ によって提案された確率的選択モデルであり、各個体の適合度とその総計を求めて、適合度の総計に対する各個体の適合度の割合を選択確率として個体を選択するという方法である。適合度の高い個体が次世代の個体として選ばれる可能性が大きい、適合度の低い個体でも選ばれる可能性が残されるため、個体群の多様性を維持することができると考えられる。他の選択手法としてエリート保存選択があり、これは個体群の中で最も適合度の高い個体は無条件に次世代に残すという方法である。エリート保存選択は他の選択手法と組み合わせて使用されることが多い。確率に従う選択や交叉、突然変異によって適合度の高い個体が消滅する可能性があるが、エリート保存選択を用いることによってエリート個体 (適合度の最も高い個体) が破壊されず、探索の速度を上げることができると考えられる。

交叉とは、選択された個体間で設計変数の組換えにより新しい個体を生成するというオペレータである。基本的な交叉は個体群の中から任意の 2 個体 (親) をランダムに選び、ランダムな 1 点または多点の交叉点で遺伝子を組替えることにより、新たな 2 個体 (子) を生成する操作である¹⁵⁾。

突然変異は、染色体上のある遺伝子座の値を他の対立遺伝子に置き換えることにより、交叉だけでは生成できない子を生成して、個体群の多様性を維持する働きをする¹⁵⁾。

GA はこれらの遺伝的オペレータを繰り返すことで探索を進めるという手法である。終了条件に達すると、そのときに適合度の最も高い個体を最適解として出力する。

一方、GA の個体をいくつかの集団に分割して探索を行う分散遺伝的アルゴリズム (Distributed Genetic Algorithm) が提案され、従来の GA よりも解探索能力が高いことが明らかとなっている¹⁶⁾。分散 GA では、個体を島と呼ばれる複数の集団に分割し、それぞれの島内で従来の GA の処理を行う。一定期間の探索を行った後、各島から任意に個体を選択し、他の島の個体情報と置き換えることで情報交換を行う。これを移住 (Migration) と呼ぶ。従来の GA の遺伝的オペレータに移住オペレータを加え、これらを繰り返すことで探索を進める。終了条件に達すれば、全個体の中

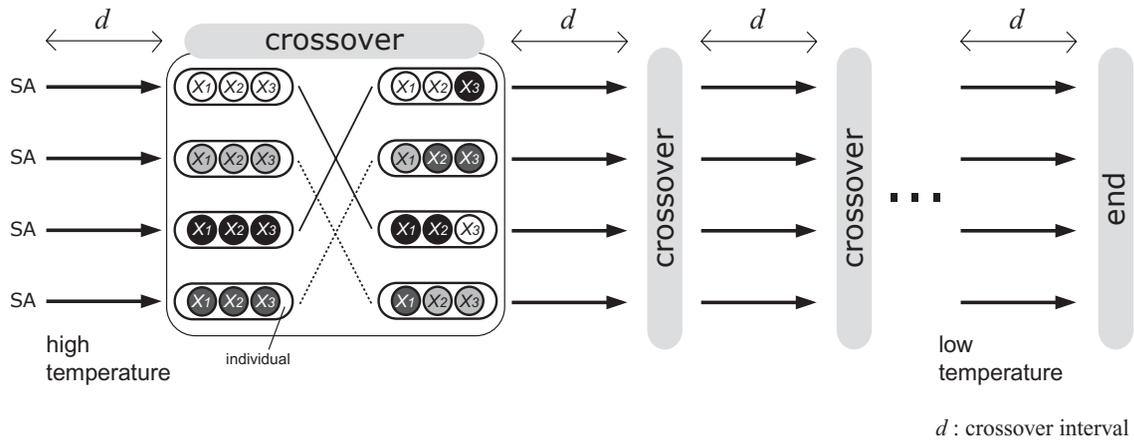


図 2 遺伝的交叉を用いた並列 SA(PSA/GAc) の概念図
 Fig. 2 Model of Parallel Simulated Annealing using Genetic Crossover (PSA/GAc)

で適合度の最も高い個体を最適解として出力する。

2.3 SA と GA のハイブリッドアルゴリズム

前節までに述べた SA と GA は、以下のような異なる特徴を持つ。SA は現在の状態のみを保存しているため、現在の状態の近傍に重点をおいた探索となる。一方 GA は複数の個体を候補解として持つため探索空間は広いが、交叉によって次世代の候補解を生成するために近傍の探索が行えるとは限らない。つまり、SA は局所探索に優れており、GA は大域探索に優れているといえる。

このように、SA と GA は互いに補完する関係にあるため、SA と GA のハイブリッドアルゴリズムがいくつか提案されている。Memetic Algorithm と呼ばれる GA に局所探索を組み込んだ手法⁸⁾、熱力学的遺伝アルゴリズム (TDGA)³⁾ や Parallel Recombinative Simulated Annealing (PRSA)⁴⁾、遺伝的状態生成処理を取入れた改良型アニーリング法⁵⁾ などがある。

TDGA と PRSA は GA での探索を中心とし、次の世代の個体を生成する過程に SA の概念を取り入れたものである。このように、GA を基本としたハイブリッドアルゴリズムの研究は多くなされている。また SA を基本としているアルゴリズムでは、改良型アニーリング法のように逐次 SA によるものが多い。そこで本研究では、並列 SA を基にしたハイブリッドアルゴリズムの提案を行うことを目的とした。

並列 SA を基にしたハイブリッドアルゴリズムは、並列 SA の持つ特徴から特に局所探索を必要とする問題に適していると考えられる。一方 GA を基にしたハイブリッドアルゴリズムは、特に大域探索を必要とする問題に適していると考えられる。このように、並列 SA を基にしたハイブリッドアルゴリズムは、GA を

基にしたハイブリッドアルゴリズムとは解探索において得意とする対象問題の特徴が異なり、また逐次 SA の探索を高速化することができる。次節で提案手法のアルゴリズムを述べる。

3. 遺伝的交叉を用いた 並列シミュレーテッドアニーリング

本研究で提案するアルゴリズムを遺伝的交叉を用いた並列シミュレーテッドアニーリング (Parallel Simulated Annealing using Genetic Crossover : PSA/GAc) と呼び、その概要を図 2 に示す。図 2 に示すように、PSA/GAc では複数のシミュレーテッドアニーリング (SA) を並列に実行する。さらに、一定期間ごとの SA の解の伝達時に、遺伝的アルゴリズム (GA) のオペレータである遺伝的交叉を用いたものである。GA のオペレータを用いた SA であるため、SA の探索点を個体 (Individual)、探索点の総数 (SA の並列数) を個体数 (Population size) と呼ぶこととする。

PSA/GAc での探索手順を以下に示す。また PSA/GAc のアルゴリズムを図 3 に示す。

- step1 初期解を生成し、複数ある探索点が並列に SA の処理を行う。
- step2 アニーリングが一定期間 d に達すると、並列に実行している SA の解からランダムに 2 つずつ解を選びペアを生成する。並列 SA がそれぞれに持つ解を個体と呼ぶこととする。このときすべての個体がペアを組むため、個体数の半数のペアが生成される。
- step3 ペアを組む 2 つの個体を親として、交叉点が 1 つの設計変数間交叉を行い、2 個体の子を生成する。設計変数間交叉とは各設計変数の間でのみ遺伝

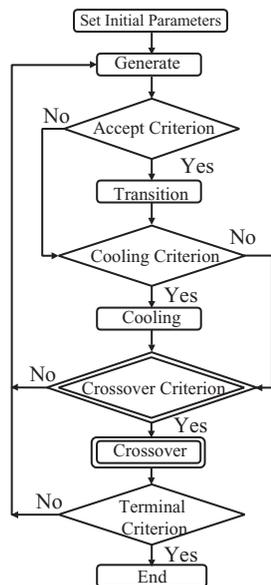


図3 PSA/GAcのアルゴリズム
Fig. 3 Algorithm of PSA/GAc

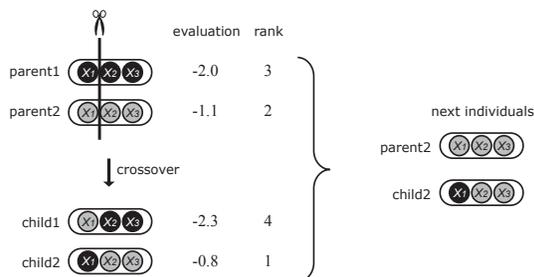


図4 設計変数間交叉と選択
Fig. 4 Crossover and Selection

的交叉を行うことをいう。

step4 もとの親と生成した子との4個体のうち評価値の高い2個体を選択する。

step5 選択された2個体から一定期間 d のアニーリングを行う。

step6 すべてのペアにおいて step3~step5 の処理を行う。

step7 終了条件を満たすまで step2~step6 の処理を繰り返す。

step3, step4 の処理を図4に示す。ある設計変数の最適解が求まっている場合、遺伝的交叉によってその設計変数の最適解を他のSA探索に伝達することができるため、アニーリングの収束を早めることができる。

PSA/GAcはSAでの探索を基にするため、局所的に多くの極小値を持つ問題に有効であるが、GAの遺

伝的交叉オペレータを用いるため、大域的にいくつかの極小値を持つ問題や部分解の組み合わせによって大域的最適解が得られる問題にも有効であると考えられる。

4. 数値実験

4.1 GAオペレータを用いた3種の並列SAとの比較

提案する遺伝的交叉を用いた並列シミュレーテッドアニーリング(PSA/GAc)では、並列に実行している各シミュレーテッドアニーリング(SA)の探索途中の解の伝達に遺伝的アルゴリズム(GA)のオペレータである遺伝的交叉を用いている。本節では、他のGAオペレータではなく遺伝的交叉を用いることの有効性を検証するため、遺伝的交叉以外のGAオペレータを解の伝達方法として用いた他の3種の並列SAと解探索能力を比較した。これらは一定間隔ごとに以下のような方法で解の伝達を行うものである。

- エリート選択を用いた並列SA (elitePSA) : 1つのエリート個体の解を他のすべての個体の新たな探索点とする
- ルーレット選択を用いた並列SA (roulettePSA) : ルーレット選択によりすべての個体の新たな探索点を決定する
- エリート保存を含むルーレット選択を用いた並列SA (e-roulettePSA) : エリート保存を用いたルーレット選択によりすべての個体の新たな探索点を決定する

ルーレット選択を用いた並列SAでは、あるステップにおいて、最大エネルギー値を持つ個体のエネルギー値 E_{max} と各個体のエネルギー値 E_i の差分 ΔE_i をとる。個体数 N のとき、差分の総計 $E_{sum} = \sum_{i=1}^N \Delta E_i$ を求め、 i 番目の個体の期待値が $\frac{\Delta E_i}{E_{sum}}$ となるような選択を行うものとした。エリート保存を含むルーレット選択を用いた並列SAでは、1つのエリート個体を次のステップの探索に保存し、残りの個体を同様のルーレット選択によって決定した。PSA/GAcの解の伝達時に用いる遺伝的交叉は、3節に示したように、設計変数間での一点交叉とした。

対象問題として、式(4)に示す大域的に極小値を多く持つRastrigin関数(f_{Ra})と、式(5)に示す大域的にはなだらかだが局所的に多くの極小値を持つGriewank関数(f_{Gr})の2つを用いた。設計変数の数 n は2とした。各関数の定義域はそれぞれの式に示した通りである。これらの関数は、各設計変数の値が0のときに最適値0をとる。

$$f_{Ra} = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \quad (4)$$

$$x \in [-5.12, 5.12], \quad n = 2$$

$$f_{min} = 0 \text{ at } x_i = 0 \quad (i = 0, 1, \dots, n)$$

$$f_{Gr} = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n (\cos(\frac{x_i}{\sqrt{i}})) \quad (5)$$

$$x \in [-512, 512], \quad n = 2$$

$$f_{min} = 0 \text{ at } x_i = 0 \quad (i = 0, 1, \dots, n)$$

探索に用いる個体数は 20, 200 とし, 初期解は乱数を用いて定義域内に発生した. 生成処理においては, 次の状態を近傍内に確率的に生成し, 確率分布として一様分布を用いた. 近傍の範囲の決定には Corana が提案する SA で用いられたアルゴリズム¹⁷⁾を用いた. Corana のアルゴリズムは, 次の状態を受理する確率が常に 50% となるように近傍の範囲を適応的に調節するものであり, 本実験では 8 ステップごとに近傍の範囲を調節することとした. それぞれの並列 SA は 32 ステップごとに解の伝達を行うこととした. クーリング率 $\gamma = 0.93$ の指数型アニーリングを用い, 次の状態を 20 回受理するごとにクーリングを行った. 初期温度は 5, 10, 20, 30 とした. 終了条件は「解の値の 1.0×10^{-4} 以上の数値が 100 ステップ変化しないこと」とし, 1.0×10^{-4} 以下の値を示す解が得られればそれを最適解とした. また, 1 ステップごとにすべての設計変数について生成処理が行われ, 評価計算を行った後, 1 回の受理判定が課されることとした.

図 5 は, それぞれの並列 SA の個体数を 20 として Rastrigin 関数を解いた結果である. 横軸は初期温度, 縦軸は 1 個体の持つエネルギーつまり解の値であり, 10 試行の平均値を示している. 図 5 からは各並列 SA の結果に大きな違いがあることがわかる. PSA/GAc は初期温度によらず, 常に最適解を求められているが, 他の 3 つの手法ではどのような初期温度でも常に最適解を求めることができなかった.

各並列 SA の個体数を 10 倍の 200 としたときは図 6 に示すように, 解の伝達方法による差はなく, 4 つの手法すべてで常に最適解が求められた. 個体数を増加させても各個体の繰り返し計算回数あまり変わらなかったために全体の計算回数が増え, すべての手法で解が求まったと考えられる.

一方, 個体数を 20 として Griewank 関数を対象としたときは, 図 7 に示すように PSA/GAc の解が比較的良かったが, どの手法でも 100% の確率では最適解が求められず有意な差はなかった. しかし 200 個体を用いた場合には, 図 8 に示すように大きな差が生じた.

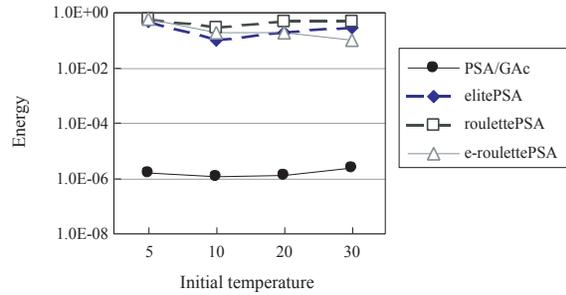


図 5 Rastrigin 関数の結果 (個体数 20)
Fig. 5 Results of rastrigin function (population size 20)

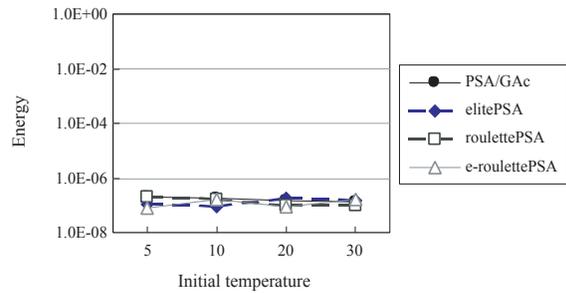


図 6 Rastrigin 関数の結果 (個体数 200)
Fig. 6 Results of rastrigin function (population size 200)

ルーレット選択を用いた並列 SA とエリート保存を含むルーレット選択を用いた並列 SA ではどの初期温度でも常に最適解は求められなかったが, エリート選択を用いた並列 SA では初期温度によっては 100% で最適解が求められることがあった. PSA/GAc では初期温度によらず常に最適解が求められた.

Griewank 関数は大域的にはなだらかであるが局所的には局所解が多数ある景観を有する関数であるため, 個体数の少ないときには局所解にとらわれ, 最適解を求めることができない. 個体数を増やすことで最適解にたどり着く可能性が上がったものと考えられる.

また, それぞれの並列 SA の個体数を 20 として Rastrigin 関数を解いたときの解の履歴を図 9 に, 探索初期の解の履歴を図 10 に示す. 横軸はステップ数, 縦軸は解の値であり, ある 1 試行における最良解の履歴を示している. 図中では PSA/GAc およびエリート選択を用いた並列 SA の結果のみを示しているが, ルーレット選択を用いた並列 SA およびエリート保存を含むルーレット選択を用いた並列 SA の結果はほぼエリート選択を用いた並列 SA と同等であった. 図 9 からは, エリート選択を用いた並列 SA は局所解に捕らわれているのに対し, PSA/GAc は局所解に捕らわれずに最適解へ到達していることがわかる. 本実験では 32 ステップごとに解の伝達が行われている. 図 10

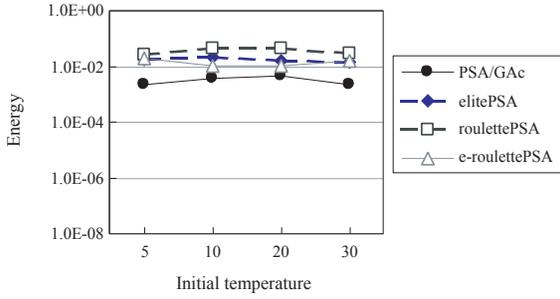


図 7 Griewank 関数の結果 (個体数 20)

Fig. 7 Results of griewank function (population size 20)

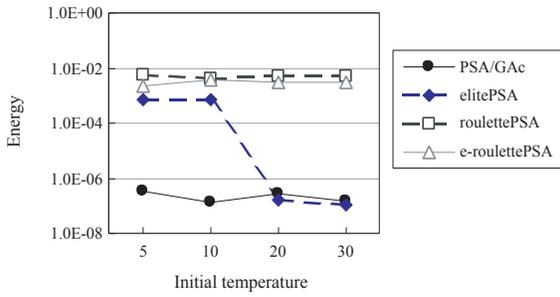


図 8 Griewank 関数の結果 (個体数 200)

Fig. 8 Results of griewank function (population size 200)

では, PSA/GAc において解の伝達, つまり遺伝的交叉が行われるステップのときに大きくエネルギー値が下がる場合があることが示されている. 遺伝的交叉によって一定間隔でエネルギー値が下がり, 局所解に捕らわれない探索が可能になったと考えられる. この結果は, 提案アルゴリズム設計時に期待した通りのものである.

これらの結果から提案手法である PSA/GAc の解探索能力が他の GA の遺伝的操作を使用した場合よりも優れているといえる. またエリート選択を用いた並列 SA も比較的優れた解探索能力を示しているが, 最

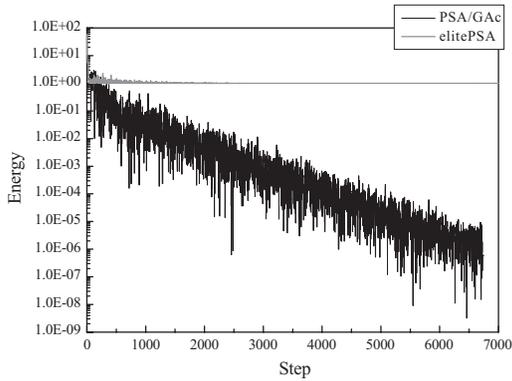


図 9 解の履歴 (Rastrigin 関数)

Fig. 9 History of energy for rastrigin function

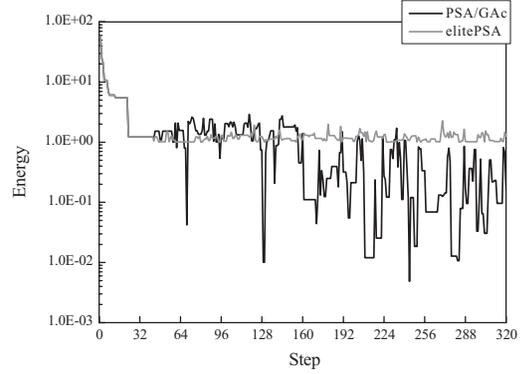


図 10 探索初期における解の履歴 (Rastrigin 関数)

Fig. 10 Early history of energy for rastrigin function

適解を求められる確率は PSA/GAc より低い. このためエリート選択を用いた並列 SA は, PSA/GAc と比較すると解探索能力は低いといえる.

4.2 分散 GA, 逐次 SA との比較

4.1 節の数値実験によって, PSA/GAc の解探索能力が優れていることが明らかとなった. 本節では設計変数を増加させた問題を対象として実験を行い, 分散 GA (DGA) との比較を行った. また逐次 SA (SSA) との探索能力の差も示した. 分散 GA は容易に並列化でき, また逐次処理でも性能が高い^{16),18)}. よって分散 GA との比較を行うことにより, PSA/GAc の探索能力の有効性を検討することができる.

対象問題は, 式 (6) に示す Rastrigin 関数 (f_{Ra}), 式 (7) に示す Griewank 関数 (f_{Gr}) と, 式 (8) に示す設計変数間に強い依存関係のある Rosenbrock 関数 (f_{Ro}) とした. 設計変数の数 n は 10, 30 とした. Rosenbrock 関数は, 各設計変数の値が 1 のときに最適値 0 をとる. 各関数の定義域はそれぞれの式に示した通りである.

$$f_{Ra} = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \quad (6)$$

$$x \in [-5.12, 5.12], \quad n = 10, 30$$

$$f_{min} = 0 \text{ at } x_i = 0 \quad (i = 0, 1, \dots, n)$$

$$f_{Gr} = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \left(\cos\left(\frac{x_i}{\sqrt{i}}\right) \right) \quad (7)$$

$$x \in [-512, 512], \quad n = 10, 30$$

$$f_{min} = 0 \text{ at } x_i = 0 \quad (i = 0, 1, \dots, n)$$

$$f_{Ro} = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (8)$$

$$x \in [-2.048, 2.048], \quad n = 10, 30$$

$$f_{min} = 0 \text{ at } x_i = 1 \quad (i = 0, 1, \dots, n)$$

表 1 PSA/GAc と分散 GA, 逐次 SA を用いて最適解が得られた確率

		PSA/GAc	SSA-long	SSA-short	DGA
Rastrigin	10dimensions	1.00	0.00	0.00	1.00
	30dimensions	0.00	0.00	0.00	0.10
Griewank	10dimensions	0.90	0.00	0.00	0.00
	30dimensions	1.00	0.00	0.00	0.70
Rosenbrock	10dimensions	1.00	0.00	0.10	0.00
	30dimensions	1.00	0.00	0.00	0.00

各手法の評価計算回数は同等にし、終了条件を満たしたときに探索を終了した。PSA/GAc では 400 個体、8000 ステップの探索を行い、32 ステップごとに遺伝的交叉を用いた。遺伝的交叉は、3 節に示したように設計変数間での一点交叉とした。逐次 SA は 3200000 ステップ (8000 ステップ × 400 個体に相当) の探索を行う SSA-long, 8000 ステップの探索を独立に 400 回実行する SSA-short の 2 種類とした。また分散 GA は 20 個体 × 20 島の 400 個体とし、8000 世代の探索を行った。PSA/GAc と 2 種の逐次 SA の初期温度は 10 に統一した。各手法について試行はそれぞれ 10 回ずつ行い、Success rate を表 1 に示した。Success rate とは、試行回数に対して最適解を得られた回数の割合を示している。なお、初期解の発生、次の状態の生成、近傍の範囲、アニーリングスケジュール、終了条件と最適解は 4.1 節と同様に定義する。

まず、PSA/GAc と 2 種の逐次 SA との結果を比較すると、逐次 SA と PSA/GAc の評価計算回数は等しいことから、単に SA のアニーリング時間や回数を増加しただけでは結果が向上しないことがわかる。また分散 GA との結果を比較すると、GA での探索が困難な設計変数間に依存関係のある問題 (f_{Ro}) に関しては、特に PSA/GAc の探索が有効であることがわかる。設計変数間に依存関係のない関数 (f_{Ra}) を対象としたときも、GA に劣らない探索が行われている。また Griewank 関数を対象としたとき、特に GA の探索において、設計変数が増加したときの方が高い探索能力を示している。これは、Griewank 関数は式 (7) に示すように第 3 項が部分解の掛上げであるため、設計変数が増加するほど確率的に最適解を得やすくなることに起因する。30 設計変数の Griewank 関数を対象とした場合にも PSA/GAc の探索が有効であることが示されている。これらの結果から、提案する PSA/GAc の探索能力は非常に高いといえる。

4.3 タンパク質のエネルギー最小化計算

前節までに数種のテスト関数に PSA/GAc を適用

し、その探索能力の高さを明らかにした。本節では、PSA/GAc の実問題への適用例として、タンパク質のエネルギー最小化計算を対象に数値実験を行った。

タンパク質は生命現象に直接関わる重要なものであり、構造を解明することは生命現象の仕組みを説明することにもつながる。タンパク質の立体構造はエネルギーの最小状態に対応しているため、タンパク質のエネルギー最小化計算を行い、最小エネルギー構造を求めることで立体構造予測を行おうとする研究がされている。

これまでタンパク質のエネルギー最小化計算においては SA がよく使用されてきた¹⁹⁾。岡本らは、小規模なタンパク質 (Met-enkephalin) を対象としてエネルギー最小化計算における SA の有効性を確かめている²⁰⁾。

Met-enkephalin は図 11 に示すように、Tyr-Gly-Gly-Phe-Met という 5 個のアミノ酸からなり、ECEPP/2 エネルギー関数^{21)~23)} に基づいた気相中において、 $E \leq -11\text{kcal/mol}$ の領域で最小エネルギー構造をしている²⁴⁾。本実験では、Met-enkephalin の主鎖における 10 個の二面角と、側鎖における 9 個の二面角をそれぞれ設計変数とした。二面角のとり得る値は $[-180^\circ, 180^\circ]$ の範囲で表現した。各設計変数において順に生成・受理判定を行ってから 1 回のクーリングを行うこととし、これらの処理を 1 Monte Carlo sweep (MCsweep) と呼ぶこととする。つまりこのタンパク質は 19 個の設計変数を持っており、1 MCsweep によって 19 回の Metropolis 判定が課されるものとした。生成処理において、次の状態は近傍内に一様分布を用いて確率的に生成した。近傍の範囲 $[min, max]$ は、式 (9) で与えた。

$$max = 180^\circ + \frac{180^\circ(0.3 - 1) \times sweep}{nsweeps} \quad (9)$$

$$min = -max$$

ここで $sweep$ は現在の MCsweep 数を示し、 $nsweeps$

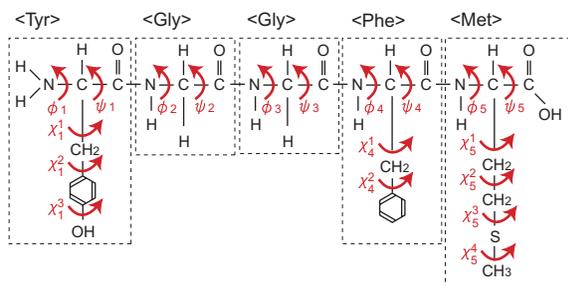


図 11 Met-enkephalin の構造
Fig. 11 Met-enkephalin molecule

表 2 最適構造が得られる確率

Table 2 Success rate for prediction of protein tertiary structure

	Number of MCsweeps	Evaluations	Success rate
PSA/GAc	4992	100005 × 19	0.90
SSA	100000	100000 × 19	0.50

は探索終了までの MCsweep 数を示す。よって近傍の範囲は、探索開始時に $[-180^\circ, 180^\circ]$ 、探索終了時に $[-54^\circ, 54^\circ]$ となる。

本実験では PSA/GAc を用いて Met-enkephalin のエネルギー最小化計算を行い、岡本らの結果²⁵⁾と比較した。岡本らの実験で用いられた逐次 SA(SSA) では、1MCsweep ごとに 19 個の二面角をそれぞれ生成・受理判定するので、100000MCsweeps の評価計算回数は $100000 \times 19 = 1900000$ 回となる。PSA/GAc では解の伝達に遺伝的交叉を用いるときにも評価計算を行う。本実験では 32MCsweeps ごとに遺伝的交叉を用いた。そのため、個体数を 20 としたときに評価計算回数を約 1900000 回とするために、MCsweep 数は 4992 とした。それぞれ試行は 10 回ずつ行い、最適構造が得られた確率を表 2 に示した。Success rate は 4.2 節と同様である。

表 2 から、最適構造を得る確率は、逐次 SA(SSA) を用いる場合よりも PSA/GAc を用いた場合の方が高いことが明らかとなった。本論文で適用しているタンパク質は小規模ではあるが、タンパク質のエネルギー最小化計算においても、PSA/GAc は逐次 SA より探索能力が高いといえる。

5. 結 論

本研究では、遺伝的アルゴリズム (GA) のオペレータである遺伝的交叉を用いた並列シミュレーテッドアニーリング (PSA/GAc) を提案し、その有効性を検討した。

GA の他のオペレータであるエリート選択やルーレット選択などを用いた並列 SA と、PSA/GAc との性能を比較した数値実験では、2 設計変数の Rastrigin 関数と Griewank 関数を対象問題とした。その結果、PSA/GAc が最も良いふるまいをした。

分散 GA や逐次 SA との比較を行い、PSA/GAc の有効性を検討した数値実験では、探索がより困難な 10 設計変数以上の Rastrigin 関数、Griewank 関数、Rosenbrock 関数を対象問題とした。PSA/GAc と逐次 SA とを比較した結果、提案する PSA/GAc は収束が早く解の品質も良いことが示された。また分散 GA

と比較した結果、GA での探索が困難な問題に対しては PSA/GAc が有効であることが示された。これらから PSA/GAc は優れた探索能力を有するといえる。

実問題への適用例として 5 個のアミノ酸からなる Met-enkephalin を対象にエネルギー最小化計算を行った数値実験では、従来用いられていた SA よりも PSA/GAc の探索能力が高いことが明らかとなった。この結果、PSA/GAc はタンパク質のエネルギー最小化計算に有効な手法である可能性が示された。

今後は、現在用いられている手法では解析が困難である大規模なタンパク質のエネルギー最小化計算に PSA/GAc を適用し、良好な解が得られることを確認する。

謝辞 本研究は文部科学省からの補助を受けた同志社大学の学術フロンティア研究プロジェクト「知能情報科学とその応用」における研究の一環として行った。

参 考 文 献

- 1) Bruce E. Rosen, 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, 1994.
- 2) E. H. L. Aarts and J. H. M. Korst. *Simulated annealing and Boltzmann machines*. John Wiley & Sons, 1989.
- 3) 森, 吉田, 喜多, 西川. 遺伝アルゴリズムにおける熱力学的選択ルールの提案. システム制御情報学会, Vol. 9, pp. 82-90, 1996.
- 4) Samir W. Mahfoud and David E. Goldberg. Parallel recombinative simulated annealing: A genetic algorithm. *Parallel Computing*, Vol. 21, pp. 1-28, 1995.
- 5) 小坏成一, 須貝康雄, 平田廣則. 遺伝的状態生成処理を取り入れた改良型アニーリング法によるフロアプラン. 電学論 C, Vol. 112, No. 7, pp. 411-416, 1992.
- 6) Hao Chen and Nicholas S. Flann. Parallel Simulated Annealing and Genetic Algorithms: a Space of Hybrid Methods. *Parallel Problem Solving from Nature*, Vol. 3, pp. 428-438, 1994.
- 7) Liu Yong, Kang Lishan, and D.J. Evans. The annealing evolution algorithm as function optimizer. *Parallel Computing*, Vol. 21, pp. 389-400, 1995.
- 8) Pablo Moscato. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. *Caltech Concurrent Computation Program, Report. 790*, 1989.
- 9) N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Ma-

- chines. *The Journal of Chemical Physics*, Vol.21, No. 6, pp. 1087–1092, 1953.
- 10) S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi. Optimization by Simulated Annealing. *Science*, Vol. 220, No. 4598, pp. 671–680, 1983.
 - 11) 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, Vol. 9, No. 6, 1997.
 - 12) Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 6, pp. 721–741, 1984.
 - 13) J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
 - 14) D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
 - 15) 坂和正敏, 田中雅博. 遺伝的アルゴリズム. ソフトコンピューティングシリーズ 1. 朝倉書店, 1995.
 - 16) Reiko Tanese. Distributed genetic algorithms. *Proc. 3rd International Conference on Genetic Algorithms*, pp. 434–439, 1989.
 - 17) A. Corana, M. Marhesi, C. Martini, and S. Ridella. Minimizing Multimodal Functions of Continuous Variables with the 'Simulated Annealing' Algorithm. *ACM Trans. on Mathematical Software*, Vol. 13, No. 3, pp. 262–280, 1987.
 - 18) Theodore C. Belding. The distributed genetic algorithm revisited. *Proc. 6th International Conference on Genetic Algorithms*, pp. 114–121, 1995.
 - 19) Hikaru Kawai, Takeshi Kikuchi, and Yuko Okamoto. A prediction of tertiary structures of peptide by the Monte Carlo simulated annealing method. *Protein Engineering*, Vol. 3, No. 2, pp. 85–94, 1989.
 - 20) 岡本祐幸. モンテカルロシミュレーションで探るタンパク質の折り畳み機構. 物性研究, Vol. 70, No. 6, pp. 719–742, 1998.
 - 21) F.A. Momany, F.A., R.F. McGuire, A.W. Burgess, and H.A. Scheraga. *J. Phys. Chem.*, Vol. 79, pp. 2361–2381, 1975.
 - 22) G. Nemethy, M.S. Pottle, and H.A. Scheraga. *J. Phys. Chem.*, Vol. 87, pp. 1883–1887, 1983.
 - 23) M.J. Sippl, G. Nemethy, , and H.A. Scheraga. *J. Phys. Chem.*, Vol. 88, pp. 6231–6233, 1984.
 - 24) Yuko Okamoto, Takeshi Kikuchi, and Hikaru Kawai. Prediction of Low-Energy Structures of Met-Enkephalin by Monte Carlo Simulated Annealing. *CHEMISTRY LETTERS*, pp. 1275–1278, 1992.
 - 25) Ulrich H. E. Hansmann and Yuko Okamoto. Numerical Comparisons of Three Recently Proposed Algorithms in the Protein Folding Problem. *JOURNAL OF COMPUTATIONAL CHEMISTRY*, Vol. 18, No. 7, pp. 920– 933, 1997.