Parallel Simulated Annealing using Genetic Crossover

Tomoyuki Hiroyasu*, Mitsunori Miki* and Maki Ogura**

(Received April 28, 2000)

This paper proposes Parallel Simulated Annealing using Genetic Crossover. The proposal algorithm consists of several processes operated as simulated annealing (SA). The genetic crossover is used to exchange information between the solutions every fixed interval. This operation reduces the computational cost while SA requires a lot of costs particularly in continuous problems. The proposed algorithm is applied to the continuous test problems. It is found that proposed algorithm could search the global optimum solution effectively, compared to conventional distributed genetic algorithms and sequential SA by the numerical examples. Therefore, using the proposed algorithm for the problems that have a lot of local optimums, such as identifying protein structures, it expects that the proposed algorithm is more effective than the conventional SA algorithms.

Key words : Simulated Annealing, Parallel Computing, Genetic Crossover

キーワード : シミュレーテッドアニーリング，並列処理，遺伝的交叉

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

廣安知之・三木光範・小掠真貴

1. はじめに

タンパク質の構造同定は，最適化問題の 1 つである．タンパク質は生命現象に直接関わる重要なものであるため，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おり，エネルギーを最小とするような構造を最適化手法を用いて求めることで，構造同定が可能である．これまで，タンパク質の構造同定においてはシミュレーテッドアニーリング (Simulated Annealing : SA) が主として使用されてきた [1]．

SA は，過熱炉内の物質を徐々に冷却し低エネルギーの状態を得るという焼きなましの過程をシミュレートした手法である [2]．近傍探索をランダムに行いながら，解が改悪する場合でも次の状態に遷移する可能性を持つことで局所解に陥ることを防いでいる．SA ではこのような状態遷移を繰り返し，与えられた初期状態から最終的には最適状態に行きつくことが期待される [3) 4)]．先に述べたように，数値計算を用いた構造解析では SA を中心に研究が進められている．しかしながら，分子の数が大きな問題では，その組み合わせの数から成功例がほとんどない．そのため，分子数の大きな問題に対しても構造同定が可能な手法の開発が望まれる．

そこで本研究では，タンパク質の構造を決定するエネルギー関数は大域的にも局所的にも極小値を持っているという特徴に着目する．すなわち，局所的な探索

* Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6434, Fax:+81-774-65-6796, E-mail:mmiki@mail.doshisha.ac.jp , tomo@is.doshisha.ac.jp
** Graduate Student, Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6716, Fax:+81-774-65-6716, E-mail:ogu@mikilab.doshisha.ac.jp

が得意な SA に，大局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問題に有効である遺伝的アルゴリズム (Genetic Algorithm : GA) のオペレータを取り入れることで，タンパク質の構造同定に対して有効な手法を提案できると思われる．

本研究では基礎的な研究としてそのような手法である遺伝的交叉を用いた並列 SA を提案し，テスト関数に適用させることにより，その性能を検討している．

2. 並列シミュレーテッド アニーリング

2.1 シミュレーテッド アニーリング

基本的なシミュレーテッドアニーリング (SA) アルゴリズムでは，現状態から次状態を生成する「生成処理」，現状態から次状態に遷移するかどうかを判定する「受理判定」，現在の温度から次の温度を生成する「クーリング」の 3 つの過程が重要となる [3]．生成処理では，現状態が与えられて次状態が生起する確率分布として一様分布や正規分布が用いられる．受理判定には通常以下のような Metropolis 基準が用いられる．

$$A(x,x',T)=\begin{cases}1 & if\ \Delta E\le 0\\ \exp(-\frac{\Delta E}{T}) & otherwise\end{cases}$$

また，クーリングでは通常式 (1) で示される指数型アニーリングが用いられる．

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

本来 SA のクーリングでは，最適解の漸近収束性を保証するためには対数型アニーリング

$$T_{k+1}=\frac{T_k}{\log k} \tag{2}$$

を用いるべきだが，対数型アニーリングでは収束があまりにも遅いため，実装するときには指数型アニーリングを用いて計算速度を向上させる．

2.2 SA の並列化

SA は最適化問題を解く有効な手段であるが，マルコフ連鎖をたどる処理であるため，本来強い逐次性があり並列化は容易ではない．しかし計算の効率化と解品質の向上を図るために，SA を並列化する研究が盛んに行われている．

マルコフ連鎖とは，現時点の状態と出力が定まれば次の状態は一意に定義されるという状態遷移のことである．SA ではこのマルコフ連鎖を次々にたどることで最適解に到達することが数学的に保証されている．そのため SA を並列化する際も，マルコフ連鎖を分断しないことで理論上は最適解が求められる．

並列 SA のアルゴリズムの多くはマルコフ連鎖を分断せずにアニーリングを行うモデルである．その一つである温度並列 SA(TPSA) は，複数のプロセッサに異なる温度を与え，各プロセッサでは一定温度でアニーリングを行い，一定の間隔で隣接する温度プロセッサ間で解を交換するという手法である [5]．TPSA では各プロセッサが一定の温度を保つため，クーリングスケジュールが不要となる．また，時間的に一様であるため任意の時点で探索を終了することができ，解の劣化を防ぐことができる [5]．

並列 SA をシストリックにする方法 [6] もあり，その手法では P 個のプロセッサを用いて実行するとき，長さ N の連鎖を長さ $L=N/P$ のサブ連鎖に分割する．プロセッサ 0 で長さ L のアニーリングをした後，一定温度でアニーリングを続けるプロセッサ 1 と，クーリングを行ってアニーリングを続けるプロセッサ 0 に探索を二分する．二分した探索が競合するところで 2 組の解と温度を選択する．この作業を繰り返し，P 個の異なったプロセッサで実行するという手法である．

SA ではマルコフ連鎖をたどり，無限に冷却を行うことで，理論上は探索が最適解に到達するため，これらのモデルはマルコフ連鎖が分断されないように並列化している．しかし限られた時間の中では理論を満たす探索は不可能に近い．マルコフ連鎖を分断しないことに注意しすぎるため，かえって探索の速度向上を妨げているともいえる．次章で提案する手法はマルコフ連鎖を分断した並列 SA である．

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

3.1 遺伝的交叉を用いた並列 SA のアルゴリズム

本研究で提案するシミュレーテッド アニーリング (SA) は Fig. 1 に示すように，並列に実行している各 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 の探索点の総数 (SA の並列数) を個体数，アニーリングステップ (計算繰り返し回数) を世代数と呼ぶこととする．

このモデルでは，解の伝達時に並列に実行している SA からランダムに親として 2 個体を選択し，設計変数間交叉を行う．設計変数間交叉とは各設計変数の間でのみ交叉を行うことをいう．Fig. 2 に示すように，も

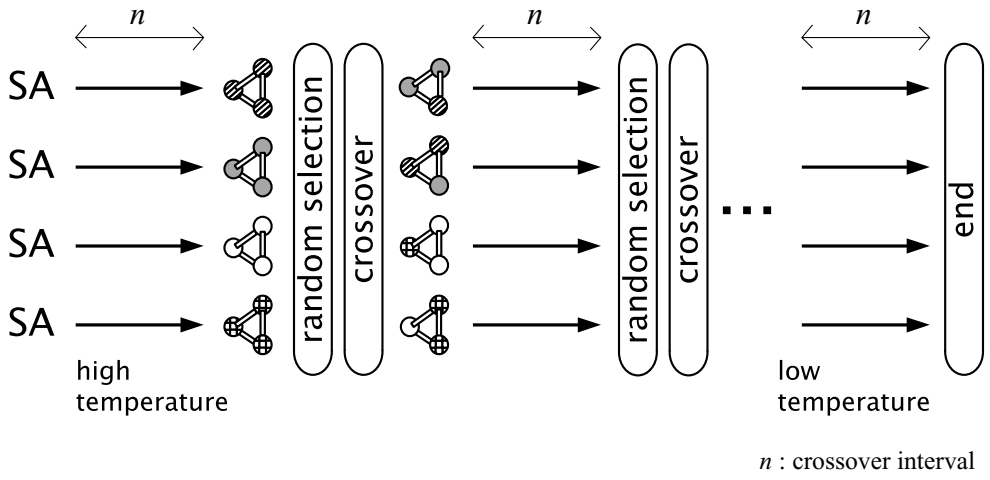


Fig. 1. Simulated annealing using genetic crossover.

との親と生成した子との 4 個体間のうち評価値の高い 2 個体を選択して，この 2 個体から次の探索を続けるという方法である．ある設計変数の最適値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値を他の 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．連続問題をあまり得意としない SA に GA オペレータを適用することで，タンパク質の構造同定の問題に対して有効な手法となることが期待される．

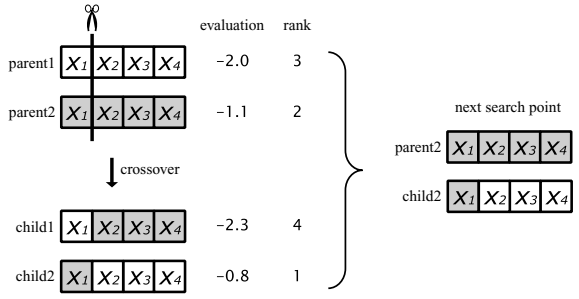


Fig. 2. Crossover.

実装した並列 SA のアルゴリズムでは，生成処理の過程において 3.2 節に示すような近傍範囲を決定する手法を取りいれた．終了条件も 3.3 節のように設定した．また受理判定には Metropolis 基準を用い，温度スケジュールには指数型アニーリング $T_{k+1} = 0.93 \cdot T_k$ を用いた．

3.2 近傍の範囲

SA を連続問題に適用する場合，解摂動に用いる近傍の範囲を決めるパラメータの調節が必要となるが，探索が無駄にならないように近傍の範囲を決定することは難しい．この問題に対して Corana は，受理率が低すぎたり受理率が高すぎたりすることによる無駄な探索が生じることを防ぐため，受理率が 50 %になるように近傍の範囲を適応的に調節するというアルゴリズムの SA を提案している [7]．本研究では遺伝的交叉を用いた並列 SA にこのアルゴリズムを組み込んでいる．

3.3 終了条件

終了条件の実装方法はいくつか考えられる．しかし SA は改善ばかりでなく改悪も認めるアルゴリズムであるため，状態が改悪されたときに終了すると解の品質が悪くなる．終了条件にアニーリング回数を用いるとこの問題が避けられない．また十分な探索がされないうちにアニーリングを終了してしまう可能性もあるため，対象問題ごとにどれぐらいのアニーリング回数で探索が十分行われるのかを知る必要が生じる．

そこで本研究ではアニーリング回数ではなく，解の摂動が許容誤差内でしかおこらなくなったときとした．解の摂動による終了条件を用いることで，解の品質を調節することができ，探索の途中でアニーリングを終了することもない．

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

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 つの並列 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

4.1.1 対象問題

対象問題として，大域的に局所解の多い Rastrigin 関数と，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 関数の 2 つを用いた．それぞれの関数は次式で表される．設計変数の数は 2 とした．

$$f_{Rastrigin} = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i)) \quad (3)$$
$$f_{min} = 0,\ [x_i = 0],\quad n = 2$$
$$f_{Griewank} = 1 + \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}(\cos(\frac{x_i}{\sqrt{i}})) \quad (4)$$
$$f_{min} = 0,\ [x_i = 0],\quad n = 2$$

一般的に Rastrigin 関数は GA での探索が，Griewank 関数は SA での探索が向いているとされている．

なお，本数値実験では，終了条件を「解の 1.0e-4 以上の値が 100 世代変化しないこと」としたため，1.0e-4 以下の解が得られればそれを最適解とした．

4.1.2 パラメータ

各並列 SA を同じ条件下で比較するため，パラメータの値を Table 1 のように固定した．計算はそれぞれの個体数・初期温度について 10 回ずつ行い，平均値を結果とした．

Table 1. Parameter of parallel SAs.

parameter	value
Population size	20 , 200
Initial temperature	5 , 10 , 20 , 30
Cooling rate	0.93
Communication interval	32
Neighborhood adjustment interval	8

4.1.3 結果と考察

Fig. 3 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．Fig. 3 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．遺伝的交叉を用いた並列 SA は初期温度によらずに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．

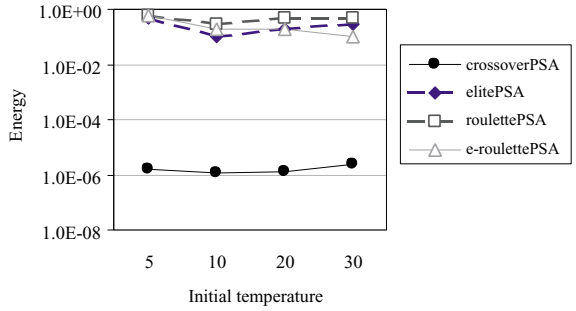


Fig. 3. Results of rastrigin function (population size 20).

各並列 SA の個体数を 10 倍の 200 としたときは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が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．

Fig. 4 はそれぞれの並列 SA の個体数を 20 としたとき，Fig. 5 は 200 としたときの Griewank 関数を解いた結果である．Fig. 4 を見ると遺伝的交叉を用いた並列 SA の解が比較的良いといえるが，どの手法でも 100 %の確率では最適解は求められず有意な差があるとはいえない．しかし 200 個体としたときは結果に大

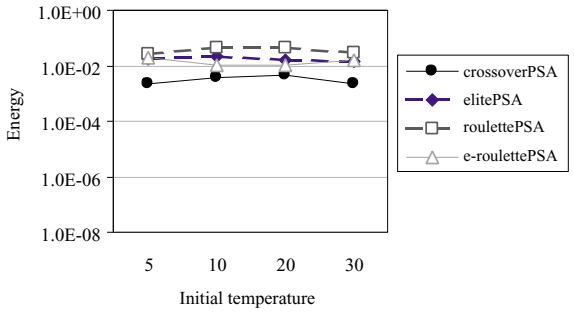


Fig. 4. Results of griewank function (population size 20).

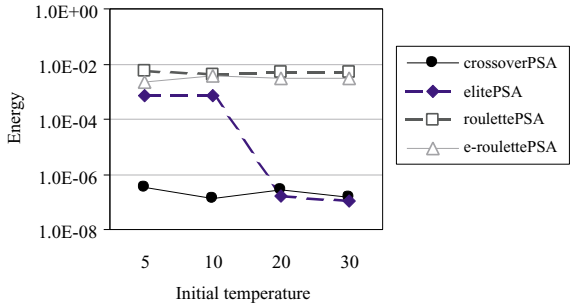


Fig. 5. Results of griewank function (population size 200).

きな差があった．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．遺伝的交叉を用いた並列 SA では初期温度によらず常に最適解が求められた．

Griewank 関数は全体的な景観はなだらかであるが細かく見ると局所解が多数あるため，個体数の少ないときには局所解にはまってしまい，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．

これらの結果から遺伝的交叉を用いた並列 SA の解探索能力が最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，最適解を求められる確率は遺伝的交叉を用いた並列 SA より低い．このためエリート選択を用いた並列 SA は，遺伝的交叉を用いた並列 SA と比較すると解探索能力は低いといえる．

4.2 遺伝的交叉を用いた並列 SA の解探索能力

4.1 節の数値実験によって，遺伝的交叉を用いた並列 SA の解探索能力が優れていることが明らかとなった．ここ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 との比較を行った．また逐次 SA との探索能力の差も示した．

4.2.1 対象問題

Rastrigin 関数，Griewank 関数と，設計変数間に強い依存関係のある Rosenbrock 関数の 3 つの数学的テスト関数を対象問題とした．それぞれの関数は次式で表され，設計変数の数を 10，30 とした．

$$f_{Rastrigin} = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i)) \quad (5)$$
$$f_{min} = 0,\ [x_i = 0],\quad n = 10, 30$$

$$f_{Griewank} = 1 + \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}(\cos(\frac{x_i}{\sqrt{i}})) \quad (6)$$
$$f_{min} = 0,\ [x_i = 0],\quad n = 10, 30$$

$$f_{Rosenbrock} = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (7)$$
$$f_{min} = 0,\ [x_i = 1],\quad n = 10, 30$$

なお，最適解は 4.1.1 節と同様に定義する．

4.2.2 パラメータ

パラメータは Table 2 のように固定し，各個体数，各初期温度ごとに試行を 10 回ずつ行った．

Table 2. Parameter of parallel SA.

parameter	value
Population size	400 , 600
Initial temperature	5 , 10 , 20 , 30
Cooling rate	0.93
Communication interval	32
Neighborhood adjustment interval	8

4.2.3 結果と考察

10 設計変数の Rastrigin 関数を遺伝的交叉を用いた並列 SA で解き，各初期温度において試行を 10 回ずつ行った結果を Fig. 6 に示す．図の横軸は初期温度，左の縦軸はエネルギー，つまり解の値，右の縦軸は最適解が求められた回数を示している．(a) は個体数が 400 のとき，(b) は個体数 600 のときの結果である．'average' は 10 試行の平均値，'best' は 10 試行中での最も良い解の値，'average(correct answers)' は 10 試行中で最適解が求められた場合のみの平均値を示し

ており，棒グラフは 10 試行中最適解が求められた回数を示している．

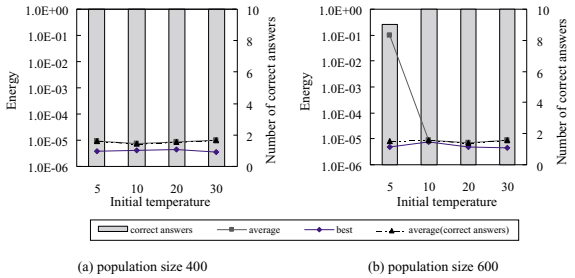


Fig. 6. Results of 10 dimensions rastrigin function.

10 設計変数としても，Fig. 6 に示すように高い確率で最適解を求めることができた．図中の (a) を見ると，既にどの初期温度でも必ず最適解が求められているため，400 個体を用いることで探索は十分行われているといえる．

次に，Rastrigin 関数を 30 設計変数として同様の実験を行った．結果を Fig. 7 に示す．

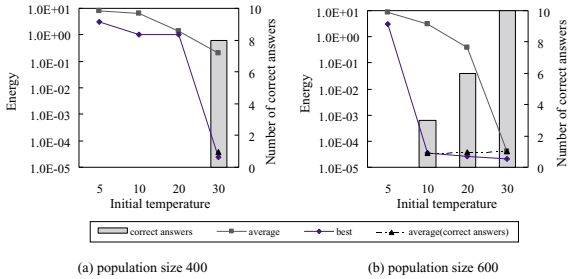


Fig. 7. Results of 30 dimensions rastrigin function.

Fig. 7 より，10 設計変数のときとは異なり最適解が十分に求められていないことがわかる．SA の探索は Rastrigin 関数のような大域的に局所解がある関数を苦手としており，設計変数が増加したことによって不得意な探索領域が広がり，最適解を求められる確率が低下したと思われる．また Fig. 7 の (a) と (b) を比較すると，個体数を多く用いたときの方が最適解を得る確率が高いということがわかる．収束するまでの世代数にはそれほど大きな差がなかったため，個体数が多い分だけ評価計算回数が増加したことがこの理由として考えられる．

10 設計変数の Griewank 関数を対象としたときの結果は，初期温度によらず 9 割程度で最適解が求められた．結果を Fig. 8 に示す．

Griewank 関数を 30 設計変数として実験を行ったときの結果を Fig. 9 に示す．

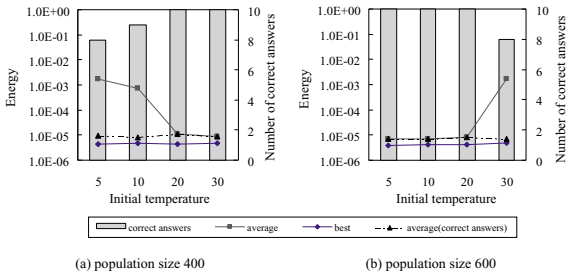


Fig. 8. Results of 10 dimensions griewank function.

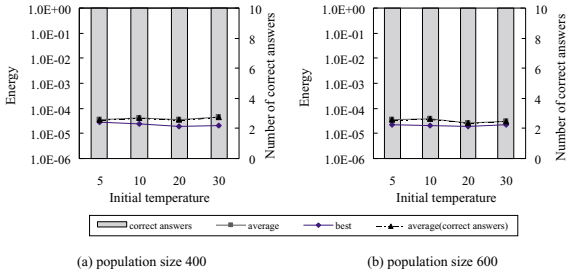


Fig. 9. Results of 30 dimensions griewank function.

Fig. 9 を見ると，10 設計変数のときよりも結果が良好なことがわかる．Griewank 関数は式 (6) で示されるように，第 3 項が各設計変数の積で表されるため，設計変数を増加させた問題の方が最適解を求めやすいと考えられる．

Rosenbrock 関数を 10 設計変数，30 設計変数としたときの結果をそれぞれ Fig. 10，Fig. 11 に示す．

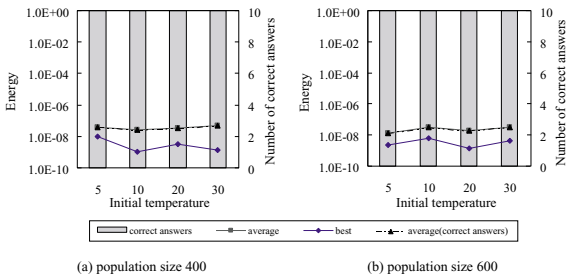
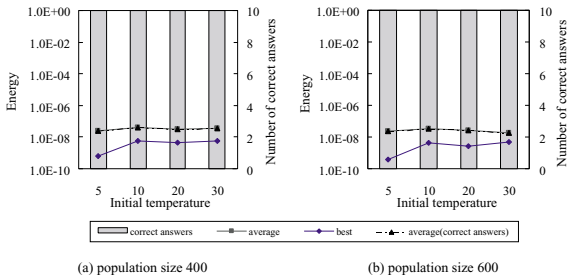


Fig. 10. Results of 10 dimensions rosenbrock function.

Rosenbrock 関数は設計変数間に強い依存関係があるため GA では探索が難しいが，遺伝的交叉を用いた並列 SA では常に最適解が求められることがわかる．また同様の実験で，30 設計変数の Rosenbrock 関数は 10 個体を用いるだけでも最適解が常に求められた．



(a) population size 400　　(b) population size 600

Fig. 11.　Results of 30 dimensions rosenbrock function.

4.2.4　逐次 SA との比較

遺伝的交叉を用いた並列 SA の有効性を検証するため，それぞれの関数を逐次 SA を用いて解き，求められた解を比較した．遺伝的交叉を用いた並列 SA でそれぞれの関数を解いたとき，400 個体を用いると約 8000 世代までに収束したため，ここで用いる逐次 SA は 3200000 世代 (8000 世代*400 個体に相当) の計算を行うもの (表中では SSA-long とする)，8000 世代の計算を独立に 400 回実行するもの (表中では SSA-short とする) の 2 種類として，400 個体を用いたときの並列 SA と性能を比較した．並列 SA と 2 つの逐次 SA の初期温度は 10 に統一した．試行はそれぞれ 10 回ずつ行い，解の値は 10 試行の平均とした．

まず Rastrigin 関数を対象としたときの結果を Table 3 に示す．10 設計変数のときは並列 SA では 10 回とも最適解が求められているが，2 つの逐次 SA では全く最適解が得られていない．また，30 設計変数としたときは並列 SA を用いても最適解を得ることはできなかったが，解の値は 2 つの逐次 SA よりも良好であった．

Table 3.　Comparison of sequential SA and PSA using genetic crossover (rastrigin function).

	PSA	SSA-long	SSA-short
10 dimensions			
Solution	7.64e-6	30.0	25.2
Success rate	1.0	0.0	0.0
30 dimensions			
Solution	6.67	226	216
Success rate	0.0	0.0	0.0

次に Griewank 関数を対象としたときの結果を Table 4 に示す．10 設計変数のときも 30 設計変数のときも逐次 SA では最適解を得ることはできなかったが，遺伝的交叉を用いた並列 SA では高い確率で最適解が求められた．

Table 4.　Comparison of sequential SA and PSA using genetic crossover (griewank function).

	PSA	SSA-long	SSA-short
10 dimensions			
Solution	7.49e-4	3.78	0.273
Success rate	0.9	0.0	0.0
30 dimensions			
Solution	4.13e-5	0.459	0.592
Success rate	1.0	0.0	0.0

Rosenbrock 関数を対象としたときの結果を Table 5 に示す．Griewank 関数のときと同様に，逐次 SA に比べて遺伝的交叉を用いた並列 SA が有効であった．

Table 4，Table 5 からは，逐次 SA ではアニーリングの時間を十分に長くしたり，回数を増加させたりするだけでは最適解が求められないことがわかる．また Table 3 からは，SA での探索が困難な問題に対しても，遺伝的交叉を用いた並列 SA は逐次 SA よりも有効であることが明らかとなった．

Table 5.　Comparison of sequential SA and PSA using genetic crossover (rosenbrock function).

	PSA	SSA-long	SSA-short
10 dimensions			
Solution	2.60e-8	1.92e-3	2.09e-3
Success rate	1.0	0.0	0.1
30 dimensions			
Solution	4.03e-8	1.48e-3	2.59e-3
Success rate	1.0	0.0	0.0

4.2.5　分散 GA との比較

本節ではそれぞれの関数について，分散 GA と遺伝的交叉を用いた並列 SA の解探索能力を比較する．しかし，SA は実数で探索するが GA はビット列で探索をするため，条件を全く等しくして比較することができない．そこで GA のビット数，終了条件，個体数に関してそれぞれを次のように設定した．

並列 SA の終了条件は「解の 1.0e-4 以上の値が 100 世代変化しない」ことであり約 8000 世代までに収束し

たため，分散 GA の終了条件は 8000 世代とし，途中で最適解が求められればそこで終了した．並列 SA は 400 個体で探索を行ったため，分散 GA も 20 個体*20 島の 400 個体での探索とした．また 2 個体*200 島 (duGa)[8] の探索結果とも比較した．分散 GA に用いるビット数は 1 設計変数当たり 30 とした．

対象問題を Rastrigin 関数としたときの結果を Table 6，Griewank 関数としたときの結果を Table 7，Rosenbrock 関数としたときの結果を Table 8 に示す．並列 SA の初期温度は 10 とし，各手法は 10 試行ずつ行った．評価計算回数は 10 試行の平均値である．

Table 6. Comparison of GA and PSA using genetic crossover (rastrigin function).

	PSA	GA(20*20)	duGa
Optimum	1.0e-5*order*	0	0
10 dimensions			
Success rate	1.0	1.0	1.0
Evaluations	3034881	773940	457000
30 dimensions			
Success rate	0.0	0.1	0.2
Evaluations	3117641	3181940	3121600

Table 7. Comparison of GA and PSA using genetic crossover (griewank function).

	PSA	GA(20*20)	duGa
Optimum	1.0e-5*order*	0	0
10 dimensions			
Success rate	0.9	0.0	0.2
Evaluations	3008201	3200400	2676960
30 dimensions			
Success rate	1.0	0.7	0.9
Evaluations	3118041	2922120	1819760

設計変数間に依存関係のある関数を対象とした結果である Table 7 と Table 8 からは，遺伝的交叉を用いた並列 SA は，GA での探索が難しい問題に対しても高い確率で最適解が得られることがわかった．並列 SA の最適解の精度は終了条件を変えることで良くすることができるため，これらの問題に対して高い確率で最適解を得たいときは GA に比べても有効だといえる．しかし Rastrigin 関数を対象問題としたときの Table 6 を見ると，分散 GA の方がより早く最適解を見つけられ，最適解を得る確率も高いことがわかる．GA が得意とする対象問題では，遺伝的交叉を用いた並列 SA の解探索能力は分散 GA に劣っているといえる．

Table 8. Comparison of GA and PSA using genetic crossover (rosenbrock function).

	PSA	GA(20*20)	duGa
Optimum	1.0e-8*order*	0	0
10 dimensions			
Success rate	1.0	0.0	0.0
Evaluations	2750721	3200400	3200400
30 dimensions			
Success rate	1.0	0.0	0.0
Evaluations	2723441	3200400	3200400

5. 結　論

本論文では，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 (SA) 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，遺伝的交叉を用いた並列 SA 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，遺伝的交叉を用いた並列 SA が最も良いふるまいをした．

遺伝的交叉を用いた並列 SA の有効性を検討した数値実験では，逐次 SA や GA では解くことが困難な 10 設計変数以上の Griewank 関数と Rosenbrock 関数を対象問題とした．遺伝的交叉を用いた並列 SA と逐次 SA と比較した結果，提案する並列 SA は収束が速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提案する並列 SA が有効であることが示された．これらから遺伝的交叉を用いた並列 SA は優れた解探索能力があるといえる．

SA はこれまであまり連続問題に対しては適用されていなかったが，このように遺伝的交叉を用いた並列 SA は，様々な連続問題に対して有効であり，GA では解くことのできない問題に対しても良好な解を示し

た．今後はタンパク質の立体構造のエネルギー関数に対して遺伝的交叉を用いた並列 SA を適用することで，現在用いられている手法よりも良いふるまいをし，良好な解が得られることを確認する．

謝　辞

本研究において，岡崎国立共同研究機構の岡本祐幸先生から重要な助言をいただいたことに対し深く感謝する．本研究は平成 10-11 年度文部省科学研究費補助金 (10650104) を受けた．また，平成 9 年度「学術フロンティア推進事業」(電磁環境とインテリジェントエレクトロニクス) に関わる研究費を受けた．

参考文献

1) 木寺詔紀. コンピュータ解析 -最適化をめぐって. 蛋白質 核酸 酵素, Vol. 39, No. 7, 1994.

2) N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Jurnal of Chemical Physics*, Vol. 21, No. 6, pp. 1087–1092, 6 1953.

3) 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.

4) 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.

5) 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, 4 1995.

6) Daniel R. Greening. Parallel simulated annealing techniques. *Physica D*, Vol. 42, pp. 293–306, 1990.

7) M. Marhesi A. Corana, C. Martini, and S. Ridella. Minimizing multimodal functions of continuous variables with the 'simulated annealing' algorithm. *ACM Trans. on Mathematical Sortware*, Vol. 13, No. 3, pp. 262–280, 1978.

8) 廣安知之, 三木光範, 濱崎雅弘, 谷村勇輔. 2 個体分散遺伝的アルゴリズム. 情報処理学会第 60 回全国大会 講演論文集, 2000.

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

page:130-138　(54)-(62)