

遺伝的アルゴリズムを用いた 適応的近傍並列シミュレーテッドアニーリング

Parallel Simulated Annealing with Adaptive Neighborhood Determined by Genetic Algorithm

三木 光範*1

Mitsunori Miki

廣安 知之*1

Tomoyuki Hiriyasu

伏見 俊彦*2

Toshihiko Fushimi

*1同志社大学工学部

Dept. of Knowledge Engineering, Doshisha University

*2同志社大学大学院

Graduate Student, Doshisha University

Simulated Annealing is an effective approximate method for solving combinatorial optimization problems. Additionally, this method shows high performance on complicated continuous optimization problems. In case of solving continuous optimization problems with SA, the neighborhood range is very important. We propose Neighborhood Parallel Simulated Annealing model as a new parallel SA model for continuous optimization problems. The proposed method has adaptive Neighborhood determined by Genetic Algorithm, and automatically controls the neighborhood range. The experimental results show better performance than a conventional model.

1. はじめに

シミュレーテッドアニーリング (Simulated Annealing : SA)[1] は、広範囲の組み合わせ最適化問題に有効な汎用近似解法である。しかし、複雑な連続最適化問題においても SA は有効な手法である [3]。SA では確率的に改悪方向への状態遷移を認めることにより局所解からの脱出が可能であり、理論上、最適解に到達することが保証されている [1]。しかし、SA で得られる解は温度や次状態の生成の範囲を決定する近傍と呼ばれる制御パラメータに大きく依存しており、任意の問題に対して、適切な設定を行う必要がある。特に連続最適化問題を扱う場合は近傍が解に与える影響が大きい。

一方、これまでの研究において、探索過程において、近傍幅を適応的に調節するメカニズムを持つアルゴリズムの研究も行われている [2][3]。しかし、適応的に近傍幅を調節するにはルールを適切に決めなくてはならないことや、対象問題に最適な近傍幅の設定を施したもののほうが高い性能を示すことも明らかになっている [4]。しかし、対象問題に最適な近傍幅を設定するためには膨大な予備実験が必要となる。

本研究では、SA を並列化し、遺伝的アルゴリズム (Genetic Algorithm : GA) を用いて自律的に対象問題に最適な近傍幅の設定を行う遺伝的アルゴリズムを用いた並列シミュレーテッドアニーリングを提案する。そして代表的な数学的テスト関数にこの手法を適用し、従来から提案されている SA と比較することでその有効性を検証する。

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られた解の精度を比較した。図 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。

図 1 より、Rastrigin 関数では近傍幅が 1.0 付近で最も良好

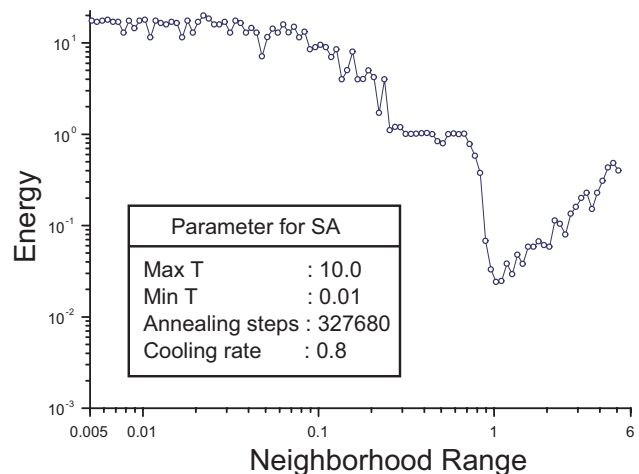


図 1: Rastrigin 関数に対する近傍幅と得られた解精度の関係

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. 遺伝的アルゴリズムを用いた適応的近傍並列シミュレーテッドアニーリング

前節より連続最適化問題において最適な近傍幅が存在することを確認した。しかしこの最適な近傍幅は各問題に依存した値であり、決定するには多くの予備実験を行う必要がある。そこで本研究では遺伝的アルゴリズムを用いて解探索過程で自律的に最適な近傍幅を探索する適応的近傍並列シミュレーテッドアニーリング (Parallel Simulated Annealing with Adaptive : PSA/ANGA) を提案する。図 2 に PSA/ANGA の概念図を示す。

PSA/ANGA では、複数のプロセスが固有の近傍を持ち、独自に SA を実行する。また、各プロセスの近傍幅を GA における個体としてビット配列で表現し、同期時にはそのビット配列に対して GA オペレータを適用する。つまり、SA における近傍幅を GA を用いて最適な近傍幅へ進化させることを期

待する。

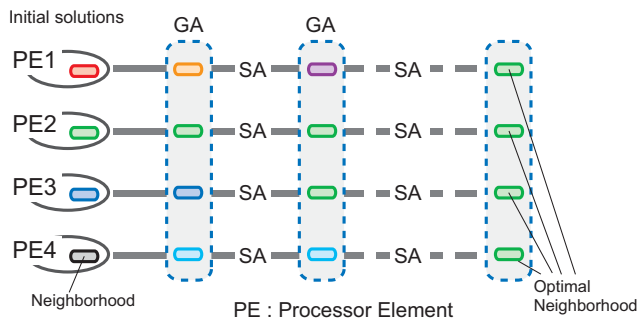


図 2: PSA/ANGA の概念図

- SA オペレータ
解生成, 受理判定, 状態遷移は通常の SA と同一の処理を行う。これらの処理を PSA/ANGA ではそれぞれのプロセスが独自に行う。
- GA オペレータ
一定期間ごとに全プロセスで同期を取り, 各プロセスが計算した評価値から GA オペレータを適用する。

ここでいう評価値とは一定期間での各プロセスの持つ解の品質を示している。GA オペレータは各近傍幅を持つ評価値からの選択, 近傍幅を示すビット配列に対しての交差, 突然変異を用いた。このことで, 評価値の高い近傍が生じ, 各プロセスの近傍幅が最適な近傍幅周辺に収束する。また交差や突然変異を用いることで近傍の多様性を保持している。

4. 数値実験

PSA/ANGA の性能を検証する実験を行う。3つのテスト関数に対して, PSA/ANGA と逐次 SA, Corana の手法 [2] (SA/AN), 最適な受理確率を目標とする適応的近傍を持つ SA [3] (SA/AAN) を適用し, 解精度を比較した。実験結果を図 3 に示す。横軸は各手法, 縦軸はエネルギー値を示す。なお, 結果は 30 試行の中央値を示す。中央値を用いた理由は指数的な差がある場合には平均値で比較すると正しい評価にならないからである。

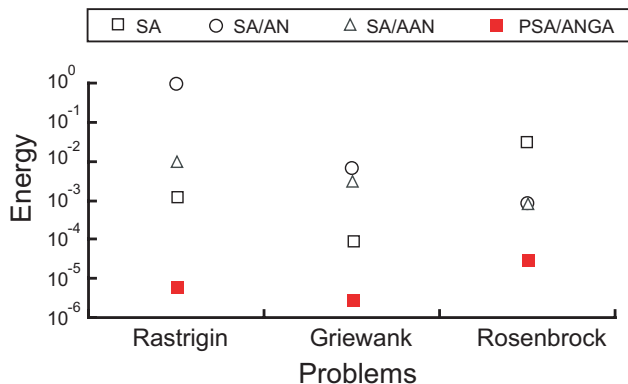


図 3: 各手法の結果

図 3 より, 全ての対象問題において PSA/ANGA が逐次 SA, SA/AN, SA/AAN に比べ, 良好な解を探索していることがわかる。この原因として, GA オペレータを用いた近傍幅の決定により, 各プロセスの近傍幅が最適な値に収束していることが考えられる。図 4 に Rastrigin 関数に PSA/ANGA を適用した際の各プロセスの近傍幅推移を示す。図 4 では, 横軸にアニーリングステップ数, 縦軸に近傍幅を示す。

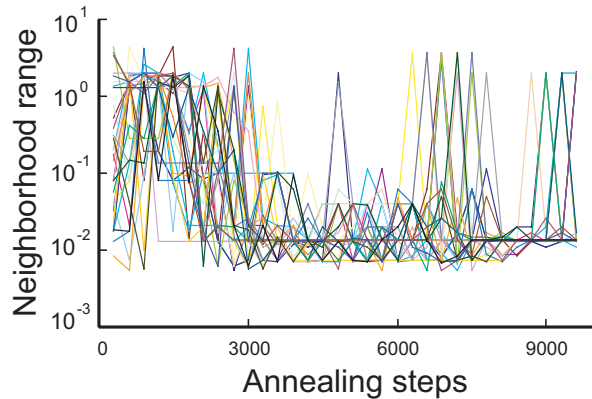


図 4: PSA/ANGA の近傍幅推移

図 4 より, 各プロセスの近傍幅が探索に応じて調節しながら収束していることがわかる。他の対象問題でも図 4 と同様に, 近傍幅が探索過程において自律的に調節しながら最適な値に収束していることがわかった。つまり PSA/ANGA では, 各プロセスが自律的に適切な近傍幅を決定することで優れた解探索能力を示すことがわかった。

5. まとめ

本研究では, 連続最適化問題に SA を適用する際に重要となる近傍幅に注目し, 探索過程において自律的に最適な近傍幅へ進化させるメカニズムを持った手法を提案した。様々なテスト関数を用いて実験を行った結果, 提案手法は従来の逐次 SA, や提案されている Corana の手法, 最適な受理確率を目標とする適応的近傍をもつ SA に比べ, 良好な結果を示すことができた。これは遺伝的アルゴリズムによって近傍幅が対象問題に対して最適な値に変化したことによる。今後はテスト関数以外の問題に PSA/ANGA を適用し, 有効性を検証する。

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クス. 日刊工業新聞社, 1997.
- [2] Corana, A., Marchesi, M., Martini, C., Ridella, S. Minimizing Multimodal Functions.
- [3] 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, 2003.
- [4] 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, 2001