

適応的近傍を持つシミュレーテッドアニーリング  
小野 景子

1 はじめに

シミュレーテッドアニーリング<sup>1)</sup> (以下 SA と略す) は、組み合わせ最適化問題に有効な汎用アルゴリズムである。一方、連続最適化問題に SA を適用する場合、解振動に用いる近傍は目的関数ごとに設定しなければならない。これに対して、Corana の手法は近傍の幅を受理率を 0.5 になるように調節し、近傍設計を自動化したものである。しかし、近傍調節に用いられる受理率の変化による解の精度の違いは明らかになっていない。

本研究では、最も良い近傍設計はどのようなものかを調べ、問題に適応する振動近傍を持つシミュレーテッドアニーリングを提案する。また、代表的な数学関数最小化問題 (Rastrigin 関数, Griewank 関数) に本手法を適用し、その有効性を検証する。

2 SA/Advanced Adaptive Neighborhood(SA/AAN)

Corana の手法の改良では小さな受理率を実現することが出来なかったが、小さな受理率を実現することの出来る新しいアルゴリズムを提案する。このアルゴリズムは、式 (1) に示す階段関数を用いて受理率から近傍幅を決定する。この時、近傍幅を増加させる拡大率  $H_0$  を、式 (2) のように再帰的に定義し、受理率が下がりにくい時には、拡大率が十分な大きな値になるようにした。

ただし、アニーリング初期には温度が高いため、近傍幅が設計領域全域まで拡大しても、指定された小さな受理率を実現することが出来ない。このため、アニーリング初期は受理率が 0.5 になるように近傍を調節し、その後、固定近傍でアニーリングを行い、受理率が指定された値まで減少した後、提案したアルゴリズムを用いる。

$$\begin{cases} m' = m \times g(p) \\ g(p) = H_0(p'), & \text{if } p > p_1 \\ g(p) = 0.5, & \text{if } p < p_2 \\ g(p) = 1.0, & \text{otherwise} \end{cases} \quad (1)$$

$$\begin{cases} H_0 = H_0 \times H_1, \\ \text{(初期設定: } H_0 = 2.0) \\ H_1 = 2.0, & \text{if } p' > p_1 \\ H_1 = 0.5, & \text{if } p' < p_2 \\ H_1 = 1.0, & \text{otherwise} \end{cases} \quad (2)$$

ここで  $p$  は、近傍の範囲を変更する間隔  $N$  の間に解振動が受理された回数  $n$  から、 $p = n/N$  と計算される。また、ここで  $p'$  は、近傍幅のパラメータ ( $H_0$ ) を変更する

間隔  $L$  の間に解振動が受理された回数  $l$  から、 $p' = l/L$  と計算される。

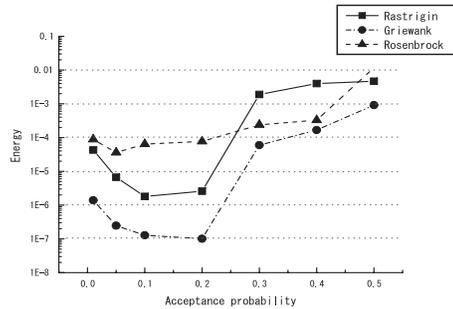


Fig. 1 エネルギーの結果

Rastrigin 関数について一定の受理率を保った場合の最小エネルギーを Fig. 1 に示す。

Fig. 1 より、受理率を 0.5 に保つ従来の方は、良好な精度を与えず、最適な受理率は 0.1 であることが分かる。

Griewank 関数に関数に関しても、従来の方より受理率を 0.1 に保つ方が良質な解を得られることが分かった。

3 TPSA/Advanced Adaptive Neighborhood(TPSA/AAN)

SA/AAN では温度スケジュールを経験的に決めていた。そのため温度スケジュールを決定するためには予備実験が必要不可欠であった。これを解消するために温度スケジュールの自動化が必要であると考えられる。温度並列 SA(Temperature Parallel SA) はこれが可能であるため、SA/AAN に TPSA を適用することを考える。その時、SA には高温時は改悪の受理をやすく、低温時にはしにくいという特徴があるために高温部は一定に保つ受理率を大きく、低温部は小さくする必要があると考えられる。現在、その受理率の決定を適応的に決めることはできないかと研究中である。

参考文献

- 1) 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.
- 2) 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 (1987).