

sGA プログラム作成と学習についての調査
中村 康昭

1 現在の課題

- sGA のプログラム作成
- 学習についての調査

2 研究の進捗状況

2.1 sGA のプログラム作成

今年から GA 班として、現在は sGA のプログラム作成を行っている。現在、Gray コーディングを実装した所であり、自身のプログラムの検証を行っている。以下にテストに用いた対象問題を記す。

2.1.1 OneMax 問題

OneMax 問題では全ビットが 1 となると最適解である。GA の交叉には一点交叉を、選択にはトーナメント選択を用いた。Table 1 に設定したパラメータを示す。

Table 1 パラメータ設定

遺伝子長	100
個体数	10
エリート個体数	1
トーナメントサイズ	2
交叉率	0.6
突然変異率	0.01
最大世代数	1000
試行回数	10

Fig. 1 にプログラムを実行させた結果を示す。

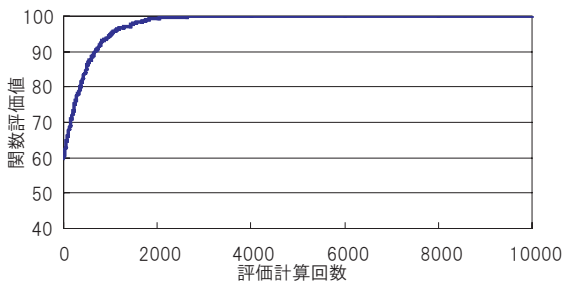


Fig. 1 OneMax 問題における関数評価値の推移

Fig. 1 のグラフにおける関数評価値は全ビット中の 1 の数となっている。遺伝子長が 100 のため、100 に到達

すると最適解に達したということになる。よって、自身のプログラムが正常に動作していることを確認した。

2.1.2 Rastrigin 関数

連続関数最適化問題の一つとして、Rastrigin 関数を用いて検証を行った。Binary コードを用いていたが、最適解を得られなかったため、Gray コードを用いて再度実験を行い、最適解を得ることができた。

パラメータのうち、Table 1 と異なる物を Table 2 に示す。ここでは個体数について 3 つのパラメータで実験を行った。

Table 2 パラメータ設定

設計変数	10
個体数	100, 200, 300
トーナメントサイズ	4

個体数を変化させて比較を行った結果を Fig. 2 に示す。Rastrigin 関数では必ずしも個体が増加すると解探

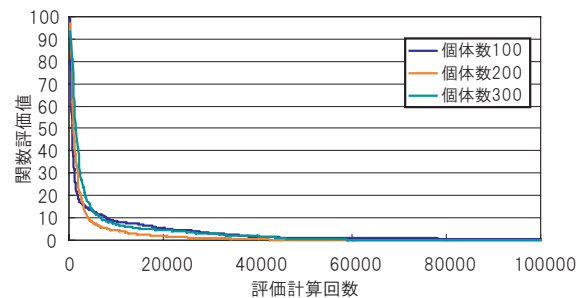


Fig. 2 Rastrigin 関数における関数評価値の推移

索性能が向上するわけではなく、200 個体よりも 300 個体で実験を行った結果の方が最適解を得るのに評価計算を多く必要とする事が分かった。

2.2 学習についての調査

現在、ヘップの学習則 (Hebbian rule) について調査中である。この学習則はニューラルネットワークの分野において、ネットワークの結合の強さを更新している方法の一つである。

3 翌月への課題

引き続き自身のプログラムの検証を進め、学習について様々な手法を調査することが今後の課題となる。