

sGA プログラミング
岩橋 崇史

1 目標課題

- sGA プログラムの作成
- 自作 sGA と ga2k の性能比較
- MPI を用いた DGA の構築

2 研究の進捗状況

2.1 作成した sGA を OneMax 問題に適用

DGA の構築を行う前に、基本的な GA の概念や遺伝的操作の理解を深めるために、C 言語で sGA の作成を行った。

プログラミングにおいて各遺伝的操作が正常に動作しているかを確認するために、OneMax 問題に適用した。

2.2 プログラムの概要

作成したプログラムの概要を Table 1 に示す。

Table 1 プログラムの概要

染色体	{ 0,1 } のビット列
選択	トーナメント選択
交叉	一点交叉
対象問題	OneMax 問題

2.3 遺伝的操作の詳細

自作したプログラムにおけるトーナメント選択を Fig. 1 に、一点交叉を Fig. 2 に示す。

トーナメント選択は母集団の中から個体数をランダムに選び出して、その中で一番適合度の高いものを選択する手法である。

一点交叉は 2 個体を選び、ビット列上の交叉点を境に、両者の情報の入れ替えを行う。

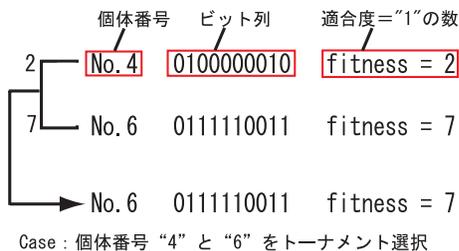


Fig. 1 トーナメント選択

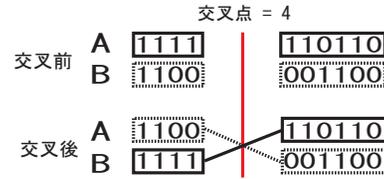


Fig. 2 一点交叉

2.4 プログラムの実行結果

Table 2 で示すパラメータを固定し、交叉率 0.4 と 0.8 で比較を行った。Fig. 3 は交叉率 0.4 と 0.8 での解探索の推移を示している。

Table 2 パラメータ

個体数	20
遺伝長	100
突然変異率	0.01
エリート戦略	なし
世代数	1000

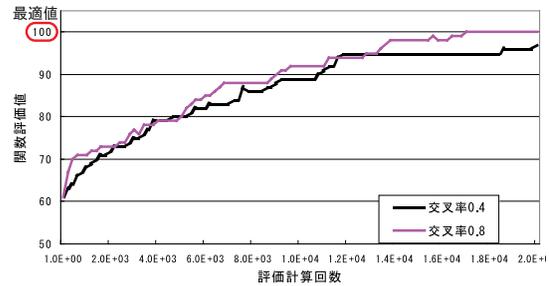


Fig. 3 実行結果

Fig. 3 より、交叉率 0.4 では評価計算回数 1.E+04 以降は解の停滞が著しく、評価計算回数 2.E+04 でも最適解に達していない。しかし、交叉率 0.8 では交叉率 0.4 よりも解探索が良好で、解の停滞が少なく、評価計算回数 1.7E+04 で最適解に到達していることがわかる。

よって、OneMax 問題を対象としたときには、交叉率が高い方が、より解探索精度が高いことがわかった。

3 今後の課題

作成した sGA を連続最適化問題のテスト関数である Rastrigin 関数に適用する。そして、自作のプログラムを検証するために、ga2k における関数評価値の推移との比較を行う。

そして、MPI を用いた DGA の構築を目指す。