

テストプログラミングによる GA の作成

藤本 万里子

1 課題の達成状況

今月は、プログラミング実習を中心に行った。本章では、それらの詳細について述べる。

1.1 テストプログラミング

実装する GA の各モジュールに対してテストコードを作成し、モジュールの実装後テストを行った。その結果、各モジュールはテストを通過し、正常に動作することを確認できた。なお、テストを行う際には、C 言語のテスティングフレームワークである Check¹を利用した。

1.2 分散 GA の実装

単一母集団 GA (SPGA) と逐次分散 GA の実装を行い、その後、性能を確認するため ga2k との比較を行った。適用した数学的テスト関数は、Rastrigin 関数と Griewank 関数である。なお、特に文中に記述がない場合、Table 1 のパラメータを用いた。

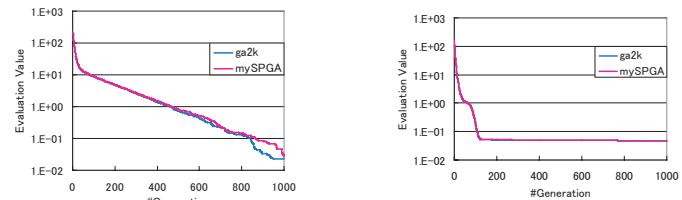
Table 1 Parameters

パラメータ	値
個体数	400
エリート個体数	5
次元数	20
染色体長	200
選択手法	トーナメント選択
トーナメントサイズ	4
交叉方法	2 点交叉
交叉率	1.0
突然変異率	0.005
最大世代数	1000
試行回数	300

SPGA の比較結果

自作 SPGA と ga2k の解探索履歴を Fig. 1 に示す。Fig. 1(a) が Rastrigin 関数、Fig. 1(b) が Griewank 関数に適用した結果である。グラフの横軸は世代数、縦軸は関数評価値であり、300 回試行の平均値を示している。なお、本実験は最小化問題であるため、グラフが下降するほど最適解に近づいていることになる。

Fig. 1(b) では、自作 SPGA と ga2k の解探索履歴はほぼ一致した。Fig. 1(a) では、探索の終盤で解探索履



(a) Rastrigin 関数

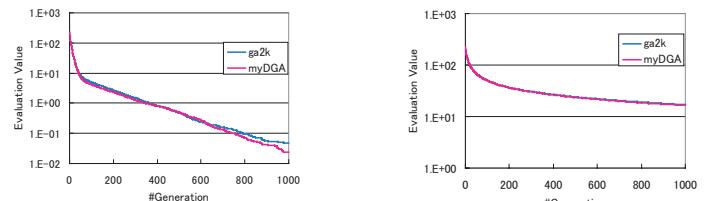
(b) Griewank 関数

Fig. 1 自作 SPGA と ga2k の解探索履歴

歴に差が見られるが、これは、収束が遅い試行が数試行確認されたので、その影響を受けたことが原因であると考えられる。以上より、実装した SPGA が正常に動作していることが確認された。

分散 GA の比較結果

SPGA を拡張し、逐次分散 GA の作成を行った。自作分散 GA と ga2k の解探索履歴を Fig. 2 に示す。分割母集団数 10、分割母集団内の個体数 40 とした結果を Fig. 2(a)、分割母集団数 40、分割母集団内の個体数 10、交叉率 0.6、突然変異率 0.05 とした結果を Fig. 2(b) 示す。示す。なお、グラフの見方は Fig. 1 と同様である。



(a) 解探索履歴 1

(b) 解探索履歴 2

Fig. 2 自作分散 GA と ga2k の解探索履歴

Fig. 2 より、自作分散 GA と ga2k の解探索履歴はほぼ一致しており、実装した分散 GA が正常に動作することが確認された。

2 今後の課題

Java による GUI の作成

解探索履歴等を示す GUI を、Java で作成する。

¹<http://check.sourceforge.net/>