

ga2k の島数とエリート数の調査と自作 ga コードのテストプログラミング 狩野 浩一

1 前回からの課題

以下に今回行った課題を示す .

- ga2k のパラメータ検討
- C++による SGA の作成
- テストプログラミングの学習および作成

2 ga2k のパラメータ検討

ga2k のエリート数と島数のパラメータが解探索に与える影響を調査した . エリート数の母体数に占める割合を 0.25, 0.5, 0.75, 1 のそれぞれのパターンで固定して , 島数を 1, 10, 20, 50, 100 に変化させた . 今回用いたパラメータを Table 1 に示す . エリート数の各島数における解探索の様子を Fig. 1 に示す .

エリート数の割合が大きく , 島数が 10 の場合に最適な結果が得られた . エリート数の割合が増加するにつれて解探索の精度が向上した原因は , 対称問題に 10 次元の Rastrigin 関数を用いて行ったため , 局所解に陥ることが少なく , 解探索が進んだことによると考えられる .

Table 1 パラメータ

総個体数	400	染色体長	100
設計変数	10	交差率	1.0
移住間隔	5	移住率	0.5
突然変異率	0.005	試行回数	300

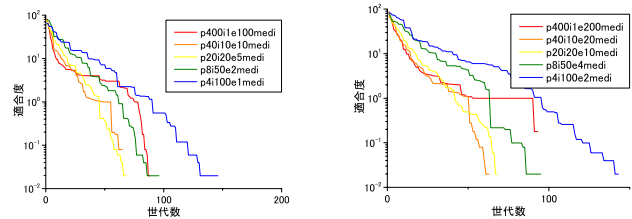
3 自作 SGA コードのテストプログラミング

C++のテストフレームワークである CppUnit を用いて , C++で作成中の SGA コードに対し , テストプログラミングを行った . これまでにテストを行ったメソッドを以下に示す .

3.1 Gene クラス

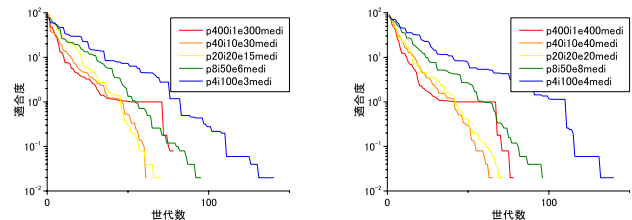
個体に関するデータや関数を扱うクラス .

- Initialize 関数 : 染色体の初期化を行う . 染色体のビット列が 0 と 1 で構成されているか , 適合度に正しく初期値が入力されているかをテストにより確認した .



(a) Rastrigin

(b) Ridge



(c) Schwefel

(d) Griewank

Fig. 1 エリート数の割合による解探索の違い

3.2 Island クラス

島内の全個体に関するデータや関数を扱うクラス .

- Initialize 関数 : 島内の染色体の初期化を行う . 島内の各個体のビット列が 0 と 1 で構成されているか , 適合度に正しく初期値が入力されているかを確認した .
- Fitness 関数 : 引数として渡された個体の適合度を計算する . あらかじめ適合度の分かっている引数を渡し , その値と比較して確認した .
- Mutation 関数 : 突然変異を行う . 突然変異率を 1.0 とし , ビット列が反転していることを確認した .

4 翌月への課題

- 自作 SGA に対するテストの完了
- C++による SGA の作成及び DGA への拡張
- MPI を用いた DGA の並列化
- Java を用いたインタフェースの作成