

自作 SGA プログラムのテストと変更
中山 靖一

1 研究活動内容

- 自作 SGA のプログラム変更
- 自作 SGA のテスト
- MOGA に関する調査

2 自作 SGA のプログラム変更

前回の月例発表会までに作成した自作 SGA のプログラムを, STL(Standard Template Library) を用いてプログラムを変更した. 変更したプログラムと ga2k との解探索性能比較のために, 対象問題として Rastrigin, Schwefel, Griewank, Rosenbrock, Ridge 関数を用いて数値実験を行った. 数値実験の際に, 設定したパラメータは Table 1 である.

Table 1 パラメータ

個体数	400
遺伝子長	100
設計変数	10
交叉方法	2点交叉
交叉率	0.6
突然変異率	0.01
エリート個体数	1
最大世代数	1000
選択方法	トーナメント選択
トーナメントサイズ	4
試行回数	300

自作 SGA, ga2k で Rastrigin 関数, Ridge 関数の数値実験の結果を中央値で表したものを Fig. 1 に示す. Fig. 1 より, 自作 SGA は, Rastrigin 関数, Ridge 関数について ga2k とほぼ同等の解探索の性能を持っていることが確認できた. 紙面の都合により, Schwefel, Griewank, Rosenbrock 関数の実験結果は割愛するが, ga2k とほぼ同等の解探索の性能を持っていることを確認した.

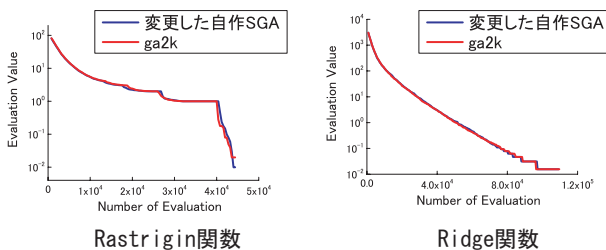


Fig. 1 数値実験結果

3 自作 SGA のテスト

CPPUNIT を用いて, 作成した自作 SGA のプログラムの主要なモジュール(個体の初期化, トーナメント選択, 交叉, 突然変異, エリート保存, 個体のソート)に対してテストを行った. その結果, テストしたモジュールが正常に動作していることを確認した.

4 MOGA に関する調査

MOGA は, 1993 年に Fonseca によって提案された手法である. MOGA の主な特徴を以下に示す.

- パレートのアプローチ
- ニッチング
- ランキング法を用いた適合度の割り当て

MOGA は, 解の優越関係に基づいて定められるランクを用いて選択演算を行うパレートのアプローチの代表的な手法である. Fonseca の用いたランキング法は, 個体 X_i が n_i 個の個体に優越されているとき, X_i のランク $r(X_i)$ を式 (1) のように定める.

$$r(X_i) = 1 + n_i \quad (1)$$

ニッチングは, 非劣解集合の多様性保持のために行う. 式 (2) に示すシェアリング関数を用いてニッチングを行う. α_{share} は, ある一定の近傍のことである. $d_{i,j}$ は, 個体 i と j の目的関数空間での距離である.

$$Sh(d_{i,j}) = \begin{cases} 1 - \frac{d_{i,j}}{\alpha_{share}}, & (if d_{i,j} \leq \alpha_{share};) \\ 0, & otherwise. \end{cases} \quad (2)$$

ニッチングカウントは式 (3) によって求める. μ_{r_i} は, ランク r_i における個体の数を表している.

$$nC_i = \sum_{j=1} \mu_{r_i} Sh(d_i, j) \quad (3)$$

5 今後の課題

今後の課題は, MOGA の特徴であるランキング法を用いた適合度の割り当てについて調査を行い, MOGA を実装することである. 今回のテストプログラミングでは, 先にプログラムを作成していたので, テストファーストでプログラムを作成できなかった. 今後は, テストファーストでプログラムの作成する.