

テストプログラミングと SGA の実装

平井 聡

1 はじめに

本研究では、「GA によるコ・ジェネレーションシステムの最適化」を対象とする。そのため、遺伝的アルゴリズム (Genetic Algorithm:GA) の基礎知識が必要となる。

先月は GA の基礎知識を取得するため、One Max 問題を対象とし、SGA(Simple GA) のプログラムを実装したが、ga2k と等しい解探索性能が得られなかった。そのため、ga2k と同様の解探索性能を得られるプログラムに修正を行い、Rastrigin や Roenbrock といった対象問題を拡張すること、またテストプログラムを実装することが先月からの課題である。

2 テストプログラム

CPPUNIT を使い、作成したプログラムを自動でテストを行うテストプログラムを作成した。現在テストを行ったモジュールは以下の 4 つである。

- 初期化
- 突然変異
- トーナメント選択
- 交叉

これらのテストプログラムの結果より、実装したプログラムが正常に動作していることが確認できた。しかし、評価とエリート保存はテストできていない。

3 SGA プログラム

実装したプログラムと ga2k の比較を行った。初期パラメータを Table.1 に示す。

Table 1 Parameters

Parameter	Value
Population Size	400
Chromosome Length	100
Maximum generation	500
Crossover rate	1.0
Corssover method	Two point Crossover
Mutation rate	1/Chromosome Length
Selection method	Tournament Selection
Number of elite individuals	5

比較には、100 回試行における解探索推移の中央値を用いた。対象問題は 10 次元の "Rastrigin", "Schwefel", "Rosenbrock", "Griewank", "Ridge" である。解探索比較を Fig. 1 に示す。

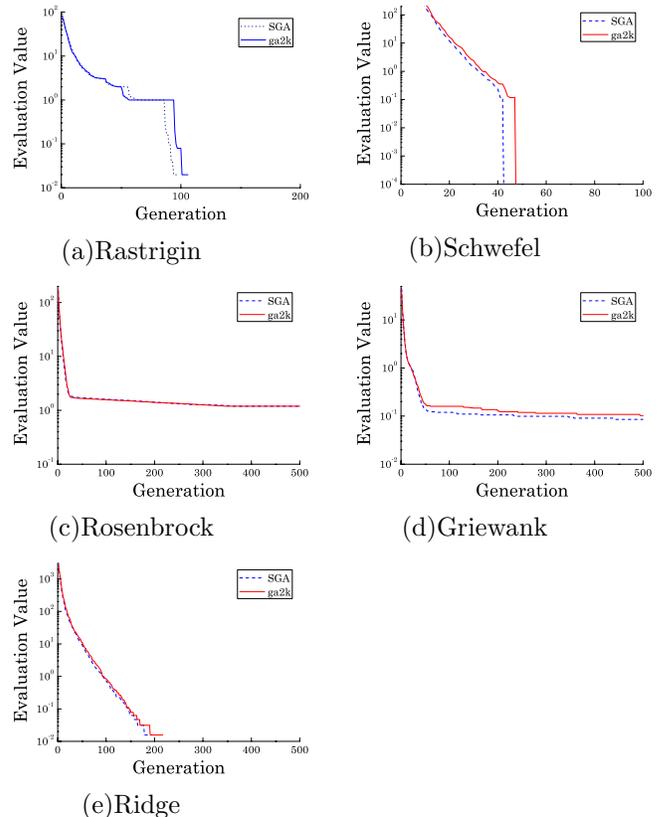


Fig. 1 Comparison Result of ga2k and SGA

Fig. 1 の解探索性能の結果より、実装したプログラムと ga2k はほぼ等しい解探索推移を示すことがわかった。そのため、実装したプログラムと ga2k の性能は、ほぼ等しいことがわかった。

4 今後の課題

今後は、基礎力を上げるために実装したプログラムを分散遺伝的アルゴリズム (Distributed Genetic Algorithm:DGA) に拡張し、その後、MPI を用いて実装する。また、実装を行うプログラムの全モジュールに対して、テストプログラムを作成する。