

ga2k04 概説

The Outline of "ga2k04"

森 隆史

Takashi MORI

Abstract: This paper summarizes the usage of Distributed Genetic Algorithms Program ga2k04. And it was examined that the difference in performance between ga2k and ga2k04. As a result, both of them show equal performance.

1 はじめに

本研究室では、これまで汎用最適化ツールとして分散遺伝的アルゴリズムプログラム ga2k¹⁾ が開発されてきた。ga2k は、分散遺伝的アルゴリズムであるため、適用範囲が広く、利用面に限って言えば、非常に扱いやすい。しかしながら、プログラムの拡張性、安全性は低く、設計者に対し新しい要求があった場合、それが小さな要求であっても、プログラム全体を大きく変更しなければならない可能性がある。

ga2k04 では、ga2k の拡張性の低さや実行時におけるデータの安全性を考慮し設計されており、設計者にとっても利用者にとっても、扱いやすくなったと考えられる。

2 ga2k との違い

ga2k は、オブジェクト指向型言語である C++ 言語で記述されている。世界を示すクラス Globe、島 (サブ母集団) を示す Island、個体を示す Gene という 3 つのクラスから構成されており、様々な遺伝的操作もその中で実装されている。これに対し、ga2k04 では、世界を示すクラス World、島を示す Island、個体を示す Individual の 3 つと遺伝的操作を行うクラスを合わせて 43 個のクラスから構成されている。遺伝的操作については、それぞれの手法で別々のクラスが用意されており、無駄なデータを持たない。

プログラムの規模は、ga2k が約 2800 行、ga2k04 が約 6700 行と 2.5 倍程度の記述量となっているが、複数のファイルに分割されているため、拡張を行う際、変更を加える部分を容易に見つけることが可能である。

また、ga2k では、あらゆるメンバ変数が public で宣言されていたが、ga2k04 では public 変数は基本的に使用しておらず、private、あるいは protected で宣言されている。これにより、途中でパラメータが変わってしまうなどの危険を防いでいる。

3 ga2k04 の使用方法

実行時のオプション設定以外は、ga2k と同じである。

3.1 コンパイル

ここでは、make を利用したコンパイルの方法を述べる。プログラムをコンパイルするためには、まず、configure スクリプトを実行する。configure スクリプトを実行することにより、Makefile が作成される。

```
$ ./configure
```

コンパイラを変更する場合は、CXX=使用するコンパイラを、コンパイルオプションを変更する場合は、CXXFLAGS=コンパイルオプションをコマンドラインで指定し、以下のように configure スクリプトを実行する。なお、= の前後に空白を入れてはいけない。

```
$ CXXFLAGS="-O2 -funroll-loops" ./configure
```

Makefile を作成したら、以下の要領で make を実行する。

```
$ make
```

実行環境

- gcc version 3.3.1 (cygming special)
Windows XP Professional Edition
- gcc version 2.96
Red Hat Linux 7.3(kernel 2.4.19)
- gcc version 2.95.4
Debian GNU/Linux(kernel 2.4.27)

3.2 実行時のオプション

ga2k04 の基本パラメータを Table 1 に示す。

ただし、これらのパラメータはコマンドラインオプションによって実行時に変更可能である。コマンドラインオプションは、以下のように与える。

```
$ ga2k --eliteSize 5
```

"--eliteSize" の部分が "-e" であっても問題ない。このようにいくつかのオプションについては、2 種類が用意されている。全てのコマンドラインオプションを本紙の最後 Table 2 に示す。いくつか特殊なオプションについて説明しておく。

個体数、島数、染色体長、次元数

個体数と島数、また染色体長と次元数は、それぞれ密接な関係がある。各島の個体数の和が総個体数と同じでなくてはならないし、各設計変数ごとの染色体長の和

Table 1 ga2k04 の基本パラメータ

パラメータ	値
総個体数	400
サブ母集団数	40
個体数/サブ母集団	10
エリート個体数	1
移住個体数/サブ母集団	5
移住間隔	5
染色体長	100
交叉手法	2 点交叉
交叉率	1.0
突然変異率	0.01
選択手法	トーナメント選択
最大世代数	1000
最大評価計算回数	1.0e+6

が全体の染色体長と同じでなくてはならない。そこで、ga2k04 では、密接な関係にあるこれらのパラメータをそれぞれ 1 つのオプションで設定できるように変更した。また、各島の個体数をばらばらにしたい、あるいは各設計変数で制度を変えたいという要求も少なくない。そこで、各島、各設計変数で異なる個体数、染色体長を設定できる機能を付加した。実際の使用方法は以下のようである。

```
$ ga2k --Population 10x20+30x40
```

この場合 10 個体の島が 20 島、30 個体の島が 40 島ということになる。染色体長に関しても同様に各設計変数ごとの染色体長とそれを適用する変数の数を指定すればよい。

また、各島ごとのエリート個体数、交叉率、交叉点数、突然変異率、選択手法、コーディングの種類を、各設計変数ごとに最大値、最小値、定義域の種類を変更することが可能である。この設定方法について説明する。

```
$ ga2k --max 1024 -b 10x10 --max 512 -b 10x10
```

このように設定すると 10 ビットの設計変数が 10 個と 10 ビットの設計変数が 10 個、すなわち 10 ビットの設計変数が 20 個で染色体長 200 で 20 次元となる。ただし、前半 10 変数（変数 0～変数 19）は最大値 512、後半 10 変数（島 10～島 19）は最大値 1024 となる。全ての染色体長の設定をする前に記述されている“--max 1024”は最大値を設定しなかった場合の最大値となる。ただし、この値を設定しなくてもデフォルトに設定されているため、実行上は問題ない。

移住個体数、移住率

本来、移住する個体数を決定するのは移住率というパラメータであり²⁾、例えば島内の個体数が 10 で移住率が 0.5 ならば、5 個体が移住するという事になっている。しかしながら、ga2k04 では、各島の個体数をばらばらに設定できるため、移住率で移住個体数を決めてしまうと島内の個体数が奇数になるなどの不都合が起こる可能性がある。そこで、基本的には移住個体数を直接指定するという形をとっている。こうすることで、各島の個体数が異なっている、各島の個体数は変化せず、個体数が奇数になるということもない。なお、あまり推奨

しないが移住率を指定した場合は、移住率を元に移住個体数を決める方法がとられる。

定義域の種類

ga2k では、定義域は自動的に [最小値, 最大値) となり、最大値は範囲から除外されてしまっていた。例えば、最大値が 5.12、最小値が -5.12 で、1 設計変数が 10 ビットで表現される場合、定義域は [-5.12, 5.12) となり、10 ビットで表せるのは、-5.12, -5.11, ..., 5.10, 5.11 の 1024 個となっていた。ga2k04 では、最大値ではなく、最小値を除外したり、両端の値を取れるようにするために、中央値を除外することができる。

4 ga2k04 の解探索性能

ga2k と ga2k04 の解探索性能の比較を行う。基本的なアルゴリズムは同じものであるため、同等の性能を示すことが予想される。

パラメータは Table 1 に示したものをを用いた。対象問題は 10 次元の Rastrigin 関数、Rosenbrock 関数とした。

結果を Fig. 1 に示す。図の縦軸は、関数評価値を横軸は評価計算回数を示しており、最小化問題であるため、縦軸の値が小さいほど良好な性能を示しているといえる。

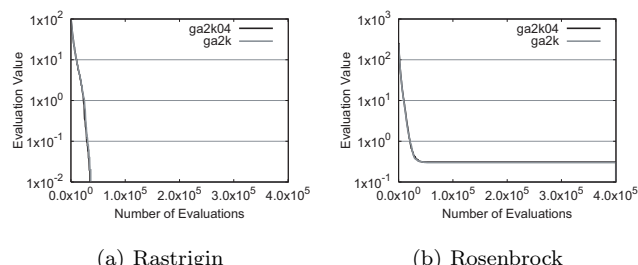


Fig. 1 解探索履歴

Fig. 1 より、ga2k04 は、ga2k と同等の解探索性能を示している。

5 まとめ

ga2k と ga2k04 の違い、ga2k04 の使用方法、解探索性能について簡単に説明を行った。ga2k04 は、拡張性や安全性の面で ga2k を上回っており、また同じ解探索性能を示していることから、非常に扱いやすいプログラムであるといえる。

参考文献

- 1) 廣安知之, 吉田純一, 佐野正樹, 福永隆宏, 片浦哲平, 澤田淳二. 分散遺伝的アルゴリズム ga2k (ver.1.4.1) 仕様書. 同志社大学工学部知的システムデザイン研究室. http://mikilab.doshisha.ac.jp/dia/research/pdga/archive/20030402_ver1.4.1/ga2k.doc.pdf
- 2) Reiko Tanese. Distributed Genetic Algorithms. Proceedings of the Third International Conference on Genetic Algorithms, pp.434-439, 1989.

Table 2 ga2k04 のコマンドラインオプション

--Population	-P	母集団サイズ	‘x’ と ‘+’ を用い、 サブ母集団サイズ、島数も同時に設定する
--eliteSize	-e	エリート個体数	1 島あたりのエリートの数
--initializer	-a	初期母集団生成方法	初期個体の発生法を指定 0：ランダム, 1：ファイルから読み込み
--crossoverRate	-c	交叉率	交叉率を 0 から 1 の実数で設定
--numCrossover	-x	交叉点数	N 点交叉の N を整数で設定
--mutationRate	-m	突然変異率	突然変異率を 0 から 1 の実数で設定
--selection	-s	選択手法	選択手法を指定 0：トーナメント選択（デフォルト） 1：ルーレット選択 2：ランキングルーレット選択 3：ルーレットトーナメント選択 4：ランキングルーレットトーナメント選択
--tournamentSize	-t	トーナメントサイズ	トーナメントサイズを整数で設定 (トーナメント選択の場合のみ有効)
--migrantSize	-M	移住個体数	各島の移住個体数を整数で設定する
--migrationRate		移住率	移住率を 0 から 1 の実数で設定 (各島の個体数が変動する可能性がある)
--migrationInterval	-I	移住間隔	移住間隔を整数で設定
--generation	-g	最大世代数	最大世代数を 0 から 1 の実数で設定
--numEvaluation	-N	最大評価計算回数	最大評価計算回数を整数で設定
--trial	-z	試行回数	試行回数を整数で設定
--thleshold	-T	最適値	最適値を実数で設定 (最大化問題であるため、 設定した値以上になると最適解発見とみなす)
--stop	-E		最適解発見で探索を終了する (引数は無い)
--output	-o	解探索履歴の出力	各世代の評価計算回数, 評価値を出力する (引数は無い)
--seed	-r	乱数の種	乱数の種を整数で設定
--chromosomeLength	-b	染色体長	‘x’ と ‘+’ を用い、 設計変数ごとのビット長, 次元数も同時に設定する
--min		定義域の最小	定義域の最小値を次数で設定
--max		定義域の最大	定義域の最大値を次数で設定
--domainType		定義域の種類	定義域の中で省く点を指定 0：最大値, 1：中央値, 2：最小値
--coder	-C	コーディングの種類	コーディングの方法を指定する 0：グレイコード, 1：バイナリコード
--function	-f	評価関数	評価関数を指定 0：Rastrigin 関数, 1：Schwefel 関数, 2：Griewank 関数, 3：Rosenbrock 関数, 4：Ridge 関数
--parallelType		並列処理の方法	並列処理の方法を指定 0：いくつかの島の処理を各島に割り当てる 1：マスタースレーブモデル