

MicroAVS 超入門

Tutorial for Micro AVS

赤塚 浩太

Kouta AKATSUKA

Abstract: This paper introduces Micro AVS which is a visualization tool. It's exceedingly effective in presentations and tracing searches. I summarize its features and the way of making files. And I report my study utilizing Micro AVS. In this study, all GA individuals are plotted the three-dimensional space with hamming distance, fitness value and topology.

1 はじめに

本レポートでは、計算・実験により出力される数値データの可視化（ビジュアライゼーション）ツールである、MicroAVS について紹介する。これまで、GA などの研究において個体の分布などの大量の数値データを把握するには、excel のグラフ機能にたよるか、java などを用いて各自で可視化する必要があった。しかし、MicroAVS を用いることで、java などを用いるより少ない労力で、excel より効果的なグラフを作成することができる。そこで、本レポートでは、MicroAVS の初歩的な使用方法を紹介したいと思う。

2 MicroAVS の特徴

MicroAVS の特徴は次の点である。

- 豊富な可視化機能** 可視化機能では、3次元棒グラフ、鳥瞰図をはじめとする様々な可視化を行うことができる。
- 離散データの格子構造データへの変換** 離散データから、自動的にデータの補間を行い、格子構造データへ変換することが可能である。また、格子構造データに変換することで、離散データでは利用できない多くの可視化手法を利用できる。
- 豊富なデータタイプ** 離散データ、画像データ、有限要素法解析データ、地形データなど様々なデータタイプに対応している。
- アニメーション機能** 同じデータを世代やステップ毎に取ったデータを連続で読みこみ、アニメーション指せることができる。また、速度ベクトルのデータに基づいてパーティクルを飛ばすシミュレーションを行うことができる。
- 豊富な出力形式** 図であれば、bmp、gif や jpeg など一般的な画像形式に加え、ps(eps) 形式での保存が可能。また、3次元データを VRML や Java アプレットへ出力できる他、アニメーションデータを AVI や MPEG に変換することも可能である。

このように MicroAVS は、豊富な出力形式 (特に EPS) や離散データの補間機能、アニメーション機能など数多くの非常に有用な機能を有する。

3 データの作成方法

3.1 はじめに

MicroAVS を使用するには、まずデータを作成する必要がある。そこでここでは、MicroAVS にいくつか用意されているデータ作成方法の中で、もっとも作りやすく汎用性があると思われる方法について説明する。

3.2 データファイルの構造

データファイルは、プログラムなどで出力した数値の羅列されたデータファイル (*.dat) と、各数値の意味を MicroAVS に認識させるためのフィールドファイル (*.fld) の 2 つからなる。また、フィールドファイルは、データ形式などを指定するヘッダ部と、データの値を読みこむファイルを指定する Variable 部、データの座標を読み込むファイルを指定する Coord 部からなる。

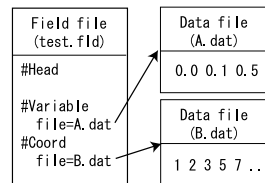


Fig. 1 ファイル分割

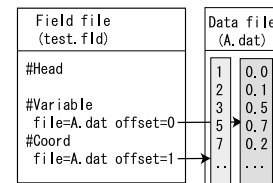


Fig. 2 複数列ファイル

Variable 部で読みこむデータファイルと、Coord 部で読みこむデータファイルを別々に用意する方法 (Fig.1) と同じファイルに 2 列作り、列によって Variable か Coord かを変え offset で何列目を読みこむかを指定する方法 (Fig.2) がある。

3.3 ヘッダ部

ヘッダ部には、データの構造などを書く。ヘッダファイルの例を次に示す。

```
# AVS field file
ndim = 2
dim1 = 10
dim2 = 10
nspace = 3
veclen = 1
data = float
field = uniform
label = pressure
```

- ・ **# AVS...** ファイルが AVS の *.fld ファイルであることを示す。
- ・ **ndim** データが何次元配列か (計算空間の次元数) を示す。可視化空間の次元とは関係ない。離散データの場合 1 を指定する。
- ・ **dim1,2,...** 各次元方向へのデータのサイズを指定する。
- ・ **nspace** データが存在する物理空間の次元数を指定する。可視化空間の次元である。
- ・ **veclen** データ成分の数を示す。各座標地点におけるデータの個数。
- ・ **data** データ成分のデータ型を、short,byte,integer,float,double の中から指定する。
- ・ **field** 座標値に関する情報を指定する。uniform の場合等間隔となり、座標の指定は必要無くなる。rectilinear の場合直行不等間隔となり、各軸方向毎の代表となる座標を指定する。irregular の場合、全てのデータ成分に対する座標を指定する。
- ・ **label** 各データ成分にラベルをつける。複数の場合はスペースで区切る。

3.4 Variable 部, Coord 部

Variable 部, Coord 部ともに、次の書式となる。

```
variable(coord) n file=*.dat filetype=ascii skip=2 offset=3 stride=4
```

- ・ **variable/coord** その行がデータ成分の読みこみか座標値の読みこみかを指定する。
- ・ **n** variable, coord 毎に、何番目のデータかを指定する。
- ・ **file** 読みこみの対象ファイルを指定する。
- ・ **filetype** 通常は ascii を指定する。
- ・ **skip** 対象ファイルの先頭を何行読み飛ばすかを指定する。
- ・ **offset** 対象ファイルの何列目を読みこむかを指定する。
- ・ **stride** 次のデータまで、何項目データを読み飛ばすかを指定する。通常は列数分指定する。

3.5 2次元離散スカラーデータの例

各データが2次元の座標と各点でのデータ値をもって、データ自体は離散である場合の例を示す。

```
# AVS field file
ndim = 1
dim1 = 10
nspace = 2
veclen = 1
data = float
field = irregular
label = fitness
variable 1 file=a.dat filetype=ascii
skip=1 offset=2 stride=3
coord 1 file=a.dat filetype=ascii
skip=1 offset=0 stride=3
coord 2 file=a.dat filetype=ascii
skip=1 offset=1 stride=3
```

```
a.dat
X Y Dat
0.0 0.0 0.1
2.0 0.0 0.2
5.0 0.0 0.3
3.0 1.0 0.4
1.0 2.0 0.5
3.0 2.0 0.6
5.0 2.0 0.7
0.0 4.0 0.8
0.0 5.0 0.9
3.0 5.0 1.0
```

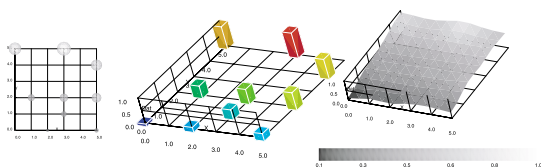


Fig. 3 出力例

なお, Fig. 3 は出力例であり, 左からマーカー表示, 3D 棒グラフ表示, 構造データへ変換後鳥瞰図表示したものである。

3.6 複数ステップデータの作成

世代やステップ毎にデータを用意しアニメーション等を行う, 複数ステップデータの作成においても, これまでに述べたデータ構造など基本的な事は変わらない。もっ

とも簡単な複数ステップデータの記述方法は, Variable 部, coord 部をステップ毎に用意し, 各ステップの始めと終わりに time と EOT を追加する。time はステップの始まりを意味し, またステップ毎のラベルを指定するものである。

```
# AVS field file
ndim = 1
dim1 = 10
nspace = 2
veclen = 1
data = float
field = irregular
label = fitness

time file=step1.dat filetype=ascii skip=1
variable 1 file=step1.dat filetype=ascii
skip=2 offset=2 stride=3
coord 1 file=step1.dat filetype=ascii
skip=2 offset=0 stride=3
coord 2 file=step1.dat filetype=ascii
skip=2 offset=1 stride=3
EOT

time file=step2.dat filetype=ascii skip=1
variable 1 file=step2.dat filetype=ascii
skip=2 offset=2 stride=3
coord 1 file=step2.dat filetype=ascii
skip=2 offset=0 stride=3
coord 2 file=step2.dat filetype=ascii
skip=2 offset=1 stride=3
EOT
```

```
step1.dat
X Y Dat
step=1
0.0 0.0 0.1
2.0 0.0 0.2
5.0 0.0 0.3
3.0 1.0 0.4
1.0 2.0 0.5
3.0 2.0 0.6
5.0 2.0 0.7
0.0 4.0 0.8
0.0 5.0 0.9
3.0 5.0 1.0
```

```
step2.dat
X Y Dat
step=2
1.0 0.0 0.2
3.0 1.0 0.3
4.0 2.0 0.5
2.0 2.0 0.8
3.0 2.0 0.9
0.0 3.0 1.0
1.0 3.0 1.5
2.0 2.0 1.7
9.0 1.0 1.9
0.0 0.0 2.0
```

ただし, この方法ではステップ数が多くなると非常に手間がかかる。そこで, Fig. 4 のように複数ステップの dat ファイルを1つにまとめ, DO と ENDDO, NSTEP を用いて複数ステップの variable 部, coord 部をまとめることもできる。

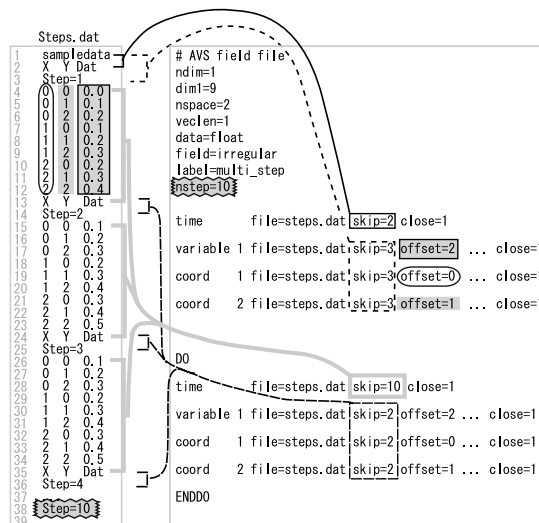
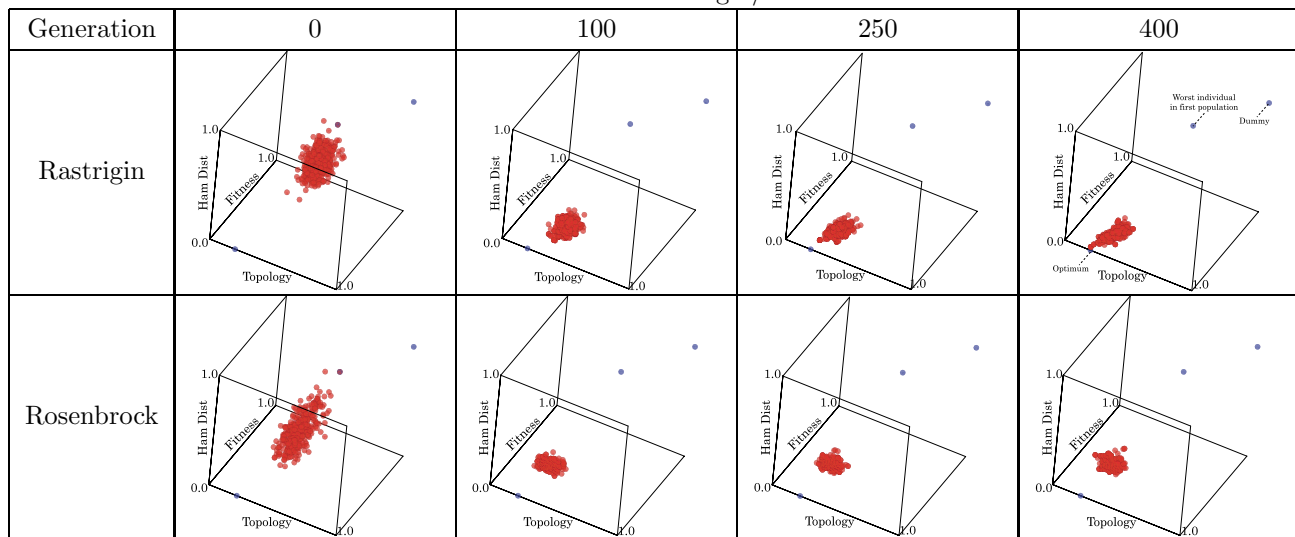


Fig. 4 複数ステップを1ファイルに記述した例

左側が dat ファイル, 右側が fld ファイルである。fld ファイル中ヘッダ部は nsteps という総ステップがいくつになるかを指定する項目が増えた以外は変化はない。データ部では最初のステップを呼び出す部分と, 2 ステップ目以降を呼び出す部分を分ける必要がある。最初のステップの部分はこれまでとほぼ同じである。time 部分の skip には 1 行目と 2 行目を読み飛ばすため 2 を指定し, variable, coord 部分の skip には 1~3 行目を読み飛ば

Table 1 Result of Rastrigin/Rosenbrock



すために3を指定する．なお，複数ステップを同一ファイルから呼び出す場合はclose=1を各部分に書き，ファイルを閉じないようにする必要がある．続いて，DO～ENDDOで囲まれた部分(2～10ステップ)では，各ステップの間で読み飛ばす必要のある行数をskipに指定する．time部分とvariable部分などは別々に扱われるため，それぞれお互いを読み飛ばす必要がある．すなわち，time部分では，4行目から次にtimeで読む必要がある14行目までの10行をskipする必要がある．また，variable，coord部では前ステップの最後に呼んだ行から，現ステップの最初に読む行までの間の「X Y Dat」「STEP=」の2行を読み飛ばす必要があるため，skip=2を指定する．

4 研究紹介

4.1 はじめに

これまで，GA探索中の個体情報を用いて探索の様子を把握するには，対象問題の設計変数を2次元に固定し，コード化前の状態での設計変数値と関数値を元に，個体が関数上にどのように分布しているかを把握するものが大半であった．しかし，GAでは探索の際にコード化を行うため，設計変数値と関数値では個体の散らばりを正確に把握することは困難であり，ハミング距離などを用いて多様性を測る研究も多く行われてきた．そこで，本研究では個体の分布をハミング距離，位相，関数値の3軸を用いて把握する事を試みる．

4.2 具体的な方法

現在は研究段階であり，個体が真の解にどのように到達するのかを把握する事が目的のため，3軸は真の解を原点とし関数値は初期世代における最悪個体の値，ハミ

ング距離，位相はビット長を1.0として可視化を行った．

GA探索中の任意の世代において，全個体の関数値，真の解からのハミング距離，位相を求め，3軸の空間にプロットする．GAにはSGA400個体を用い，対象問題はRastrigin関数とRosenbrock関数を用いた．交叉には交叉率1.0の1点交叉，突然変異には突然変異率1/Lのgenesis型，染色体長は10Dim x 10Bitを用いた．

4.3 実験結果

実験結果をTable 1に示す．Rastrigin関数では世代が進む毎に個体集団が最適解に近づいていくのに対し，Rosenbrockでは100世代以降あまり変化が見られないことがわかる．

4.4 考察

実験結果より，Rastrigin関数とRosenbrock関数では個体集団の動きに大きな差があることがわかった．しかし，現在の方法では個体集団と真の解を比較しているため，GA探索中に関数の性質を把握することはできない．今後はGA探索中に関数の性質を把握できる手法を考案する必要がある．

5 おわりに

本レポートでは，MicroAVSに関してその特徴とデータの作成方法について述べた後，実際にMicroAVSを用いた研究を紹介した．このように，MicroAVSは数値データの可視化ツールとして非常に有用であり，EXCELやORIGIN，MAPLE等に加えて利用する価値が十分にある．

参考文献

- 1) MicroAVS オンラインマニュアル