情報工学実験 II レポート（探索アルゴリズム 1）

曜日＆グループ番号: 金曜日＆グループ 9

実施日:2017 年 1 月 20 日 (金)、提出日:2017 年 2 月 0 日

グループメンバ

（補足：レベル毎に 全員が協力して実施 した上で、レベル毎にレポートをまとめる担当者を決め、全体を一つのレポートとして整理すること。分担方法も自由である。）

- 155706J 久場翔悟: 担当 Level2.1
- 155711E 平木宏空: 担当 Level1.1，1.2，1.3
- 155716F 石塚海斗: 担当 Level2.2
- 155730B 清水隆博: 担当 Level2,Level2.3

提出したレポート一式について

レポート一式は “shell:/net/home/teacher/tnal/2016-search1-mon/group0/” にアップロードした。提出したファイルのディレクトリ構成は以下の通りである。

```
./src/       # 作成したプログラム一式
./report/    # レポート関係ファイル．図ファイルを含む．
./steepestsearch2-1/ #Level2.1 で作成したプログラム一式
```

1　Leve l1: 最適化とは

1.1　Level 1.1: コンピュータと人間の違いを述べよ

1.1.1　課題説明

コンピュータが人間より得意とするモノ、その反対に人間より不得手のモノ、両者について 2 つ以上の視点（立場や観点など）を示し、考察する。

1.1.2　考察

- 視点 1: hoge
  コンピュータならば＊＊が可能であり云々
- 視点 2: fuga
  人間は＊＊しなくてはならないため云々

1.2　Level 1.2: 住宅価格を推定するモデルについて

1.2.1　課題説明

Housing Data Set[2] における RM（平均部屋数）から MEDV（平均価格）を推定するためのモデルについて検討した。

1.2.2　モデルへの入力

1.2.3　モデルにおける処理内容

1.2.4　モデルの出力

（補足：PDF 図を挿入する例）

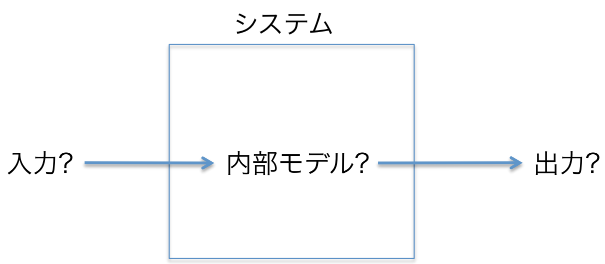

図 1　入出力と内部モデルのイメージ図

1.3　Level 1.3: モデルの良さを評価する方法について

1.3.1　課題説明

Level 1.2 で検討したモデルの適切さを評価する指標について検討した。

1.3.2　評価に用いる情報源

1.3.3　評価手順

1.3.4　評価に基づいた適切さを計る方法

2　Level 2: 最急降下法による最適化

2.1　課題説明

3 種類の連続関数 $y = x^2$、$z = x^2 + y^2$、$y = -x \times \cos(x)$ について、最急降下法の適用を通して探索挙動を観察した。以下ではまず共通部分である最急降下法の探索手続きについて、フローチャートを用いて解説する。その後、3 種類の関数毎にプログラムの変更箇所所、観察意図観察方法、観察結果、考察について説明する。

2.2　Level 2 共通部分

（補足：Level2.1, 2.2, 2.3 には共通する部分が多いため、共通部分は独立して報告すると良いでしょう）

2.2.1　探索の手続き（共通部分）

2.2.2　フローチャート（共通部分）

（手続きとフローチャートはまとめて一つの節にしても構いません）

2.3　Level2.1: $y = x^2$ について

2.3.1　プログラムソース（変更部分）

2.3.2　観察意図と観察方法

刻み値をより小さくすると探索点がより細かく移動するため最適性は良くなるだろう。しかし，小さくしすぎると探索回数が増えてしまい，効率性は悪くなるだろう。逆に刻み値を大きくすると探索回数は減って効率性はよくなるだろう。しかし，その分最適性は悪くなるだろう。

上記のように予想してこれを検証するために，seed 値を 1 に固定し alpha の値を変えることにより探索を行い，結果を観測する。

2.3.3　実行結果

2.3.4　考察

2.4　Level2.2: $z = x^2 + y^2$ について

2.4.1　プログラムソース（変更部分）

以下の図 2 に変更部分のみを示す。

図 2　Level2.2 変更点

```
//main 関数直後
f( argc != 3 ){
(略)
}else{
(略)
alpha = atof(argv[2]);
(略)
}

//f 関数内
//  z = x;
  z = x*x + y*y;

//pd_x 関数内
//  z_dx = 1;
  z_dx = 2*x;

//pd_y 関数内
//  z_dy = 0;
  z_dy = 2*y;
```

2.4.2　観察意図と観察方法

seed 値を固定して alpha 値を変動させせることで, 探索点の刻み幅による探索の最適性及び効率性を検討する。alpha 値を変更することで探索幅も変動することから, 以下の 2 つのような予想が立てられる。1, alpha 値が大きければ探索点の刻み幅も大きくなり, 効率性を向上できるが, 最適性が低下する。2, alpha 値が小さければ探索点の刻み幅が小さくなるため, 最適性を向上できるが, 効率性が低下する。

効率性の観察方法は,seed 値を固定し alpha 値を変動させた際の step 数の推移を各 seed 値 (範囲 1000-10000 1000 刻み) ごとに表して行う。効率性の検討はグラフより, 各 seed 数で最小 step 数の alpha 値を読み取り, 最も優れた alpha 値を多数決で決定し, それをこのプログラムの最大効率であると決定, 改善点を考察する。

また,srand 関数にコマンドライン引数の seed 値が渡されているため, rand 値は同じ seed 値 を入力している限り, 一定である。よって, 同一 seed 値 において, 乱数を考慮しての複数実行, 実行結果の平均値取得等はしない。

最適性の観察方法は,seed 値を固定し alpha 値を変動させた際の終了時座標, これと傾きが 0 になっている座標 (ここでは 0,0) との差の推移を各 seed 値 (範囲 1000-10000 1000 刻み) ごとにグラフで表す。最適性の検討はグラフからその差が最小の alpha の値を各 seed から読み, 最も優れた alpha 値を多数決で決定する。その alpha 値をこのプログラムの最大最適値 であると決定し, 改善点を考察する。

2.4.3　実行結果

1 効率性について

効率性の観察のための手法として, 各 seed 値 (1000-10000 の 1000 刻み 10 種), 各 alpha 値 (0.001 と 0.1-1.0(0.1 刻み) の計 11 種) ごとに最終 step 数を plot していき, その推移傾向を観察することで行った。

以下の図 3 がその結果である。

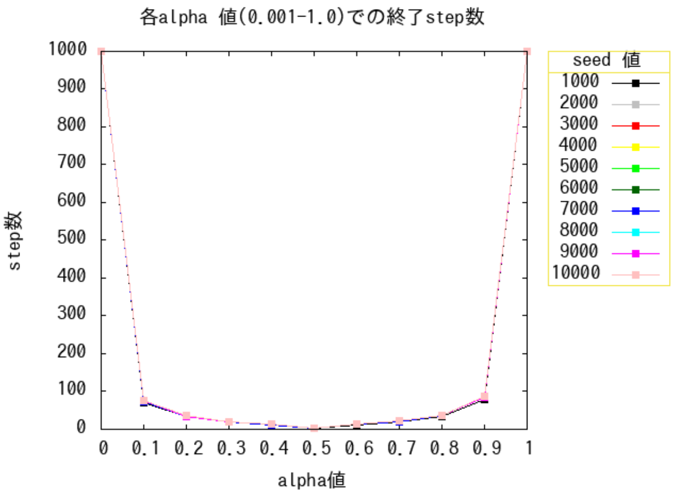

図 3　各 alpha 値 (0.001-1.0) での終了 step 数

alpha 値が 0.001, 1.0 の時に step 数が 1000 となり,alpha 値 0.1-0.9 までの点の差が見えづらい。よって,alpha 値が 0.1-0.9 の範囲の最終 step 数のグラフも図 4 に用意した。

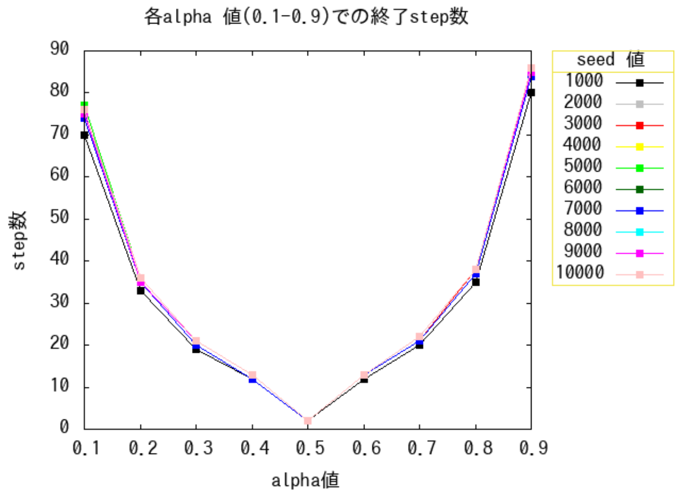

図 4　各 alpha 値 (0.1-0.9) での終了 step 数

2 最適性について

最適性の観察のための手法として, 各 seed 値 (1000-10000 の 1000 刻み 10 種), 各 alpha 値 (0.001 と 0.1-1.0(0.1 刻み) の計 11 種) ごとに終了座標の誤差 (x,y の合計) を plot していき, その推移傾向を観察することで行った。
以下の図 5 がその結果である。

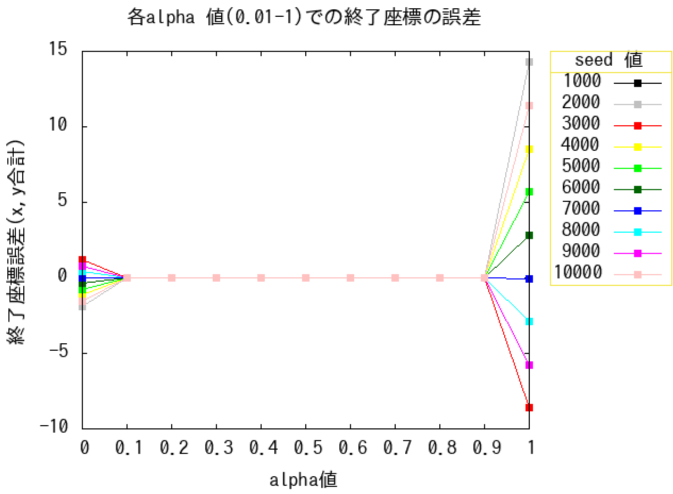

図 5　各 alpha 値 (0.001-1.0) での終了座標誤差 (x,y 合計)

alpha 値が 0.001, 1.0 の時に 最大合計誤差 10 を超え,alpha 値 0.1-0.9 までの点の差が見えづらい。よって,alpha 値が 0.1-0.9 の範囲の合計誤差のグラフも図 4 に用意した。

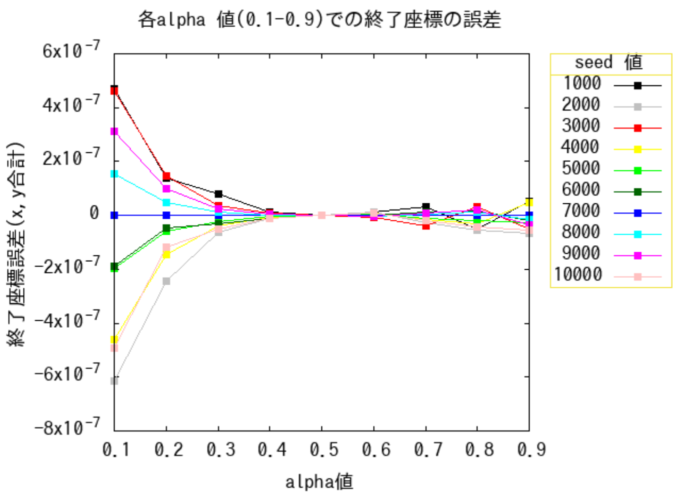

図 6　各 alpha 値 (0.1-0.9) での終了座標誤差 (x,y 合計)

2.4.4　考察

まず, 効率性についての考察を行う。

図 3 より, 最終 step 数が 1000 回以内に収まるのが,$0.001 < alpha < 1.0$ の範囲であることがわかる。このことから,alpha 値が小さすぎると探索できる範囲が狭まり, 最適解まで届かなかったと推察する。 図 4 からは, 総 step 数が 1 番低い点の alpha 値が 0.5 であり,0.5 から離れるごとに step 数が増加する傾向にあることがわかる。alpha 値が 0.5 の時に総 step 数が最小となった理由を, プログラム内の探索点移動に用いられている式より考察する。

$x = x - alpha * pd_x(x, y);$

この式の pd_x(x,y) は x についての偏微分であるため, 置き換えると

$x = x - alpha * 2x;$

となる。この式に alpha = 0.5 を代入すると x には '0' が入ることとなるため, 移動後の x 座標がちょうど傾きが 0 になる点 (最終的に移動したい最適解) になる。y についても同様にして alpha = 0.5 の時に 1 回移動後の座標は'0' となる。

このことから, alpha $= 0.5$ という値は 今回探索した $x^2 + y^2$ の式においてあらゆる座標から最適解の座標を求めることができる値であることがわかる。

次に, 最適性について
効率性の考察にて図 3 より, 最終 step 数が 1000 回となるのが $alpha = 0.001, alpha = 1.0$ で, 言い換えると $alpha = 0.001, alpha = 1.0$ のとき最適解との誤差が大きい。
このことが, 終了座標誤差を表す図 5 からも読み取れる。図 6 からは効率性と同じく alpha=0.5 に最適性が最も高いことがわかる。さらに alpha=0.5 の点へのグラフの収束の度合いにも alpha が 0.5 より小さい時と大きい時で差がある。これは,alpha が 0.5 より小さい場合の最適解からの誤差は探索点が単純に最適解に届かなかったものであるが,alpha が 0.5 より大きい場合では最適解付近まで探索点が到達したが探索点の移動幅の大きさが影響し最適解には至らなかったというもの, という差によるものと推察する。

2.5 Level2.3: $y = x * \cos(x)$ について

Level2.3 では $y = x * \cos(x)$ を最急降下法で探索する．最急降下法では微分した値をアルゴリズムで必要とする為まずはこのモデル式の微分を導出する．

$$y = x * \cos(x) \tag{1}$$
$$y' = \cos(x) - x * \sin(x) \tag{2}$$

2.5.1 プログラムソース（変更部分）

上記 (2) 式で導出した微分値を利用して steepest-decent.c を以下の様に変更した．
まず学習係数 alpha を引数として変更できる用,argv のエラーメッセージを選択する if 文を変更した．第二引数として alpha を受取り,char 型から double 型への変換を行っている．

2.5.2 観察意図と観察方法

今回は学習係数が探索に影響をあたえる物が何かを決定づけるため,seed ではなく初期値位置を固定した状態で探索を行った．探索にはシェルスクリプトを用いた．

2.5.3 実行結果

2.5.4 考察

（補足：参考文献は thebibliography 環境を使って列挙し、本文中で適切な箇所で引用するようにしましょう。例えば下記文献は、アブストラクトや Level 4 で引用しています）

参考文献

[1] 情報工学実験 2: 探索アルゴリズムその 1（當間）
`http://www.eva.ie.u-ryukyu.ac.jp/~tnal/2016/info2/search1/`

[2] Housing Data Set
`http://archive.ics.uci.edu/ml/datasets/Housing`