

Excel によるネットワーク計画法(1)

グラフの表示と接続行列

古林 隆

0. はじめに

時系列データなどを表にして、その上で合計や比率を計算したり、結果を棒グラフなどで表示したりするのに、Excel がよく使われている。また、図形描画を用いて、建物の簡単な平面図や作業の流れ図を描くこともできるし、マクロ（プログラム）を作成すれば、備え付けのツールだけではできない計算を実行することもできる。

このように、Excel は、いろいろな使い方ができるが、ネットワークのデータを表（セル）に書き込んで、それを入力し、ネットワークを描くマクロを作成すると、点の位置や大きさなどが気にいらなければ、データを変えることによって、簡単に絵を描き変えることができるし、計算結果（最短路などの最適解）を示すネットワークを自動的に（マウスをあちこち動かしてお絵かきをしなくても）出力することもできる。

そこで、Excel のマクロで作成した最短路問題や最小木問題などのネットワーク計画法を紹介する。

1. 点と枝の表示

はじめに、二点とそれらの間を結ぶ枝（線）を表示するマクロを示す。

[NP1] 図1に示すように、二点の座標と枝の向きの有無（1：有，0：無）をセルに書き込むと、それらを結ぶ枝を表示する。

メインプログラムを図2に、その中で呼び出すサブルーチンプログラムを図3に示す。

図4に示すように、セルB4，C4に一番目の点のx座標，y座標，セルB5，C5に二番目の点のx座標，y座標，E4に0を書き込んで、「ツール」→「マクロ」→「マクロ」（または「Visual Basic Editor」）→

	A	B	C	D	E	F
1	点と枝の表示					
2						
3						
4		点	x座標	y座標		枝の向き
5		1				
6		2				
7						
8						
9						
10						

図1 NP1のシート

「実行」の順にクリックして、マクロを実行すると、二点を結ぶ無向の枝が表示される。図5は、有向の枝を表示したものである。ここで、セルの数値を変えてマクロを再実行するときは、「図形描画」の中の「オブジェクトの選択」（「矢印」のアイコンをクリック）で、一度表示した図を選択して、それを削除（「はさみ」のアイコンをクリック）する必要がある。

なお、以下では、無向の枝を線分、有向の枝を矢線と言い換えることがある。

◆プログラムの説明

図3に示した三つのサブルーチンの機能、引数の意味は、次のとおりである。

```
Sub np1()
' 点と枝の表示
Dim x1, y1, x2, y2, z, r, cn, cs, ca As Integer
x1 = Cells(4, 2): y1 = Cells(4, 3)
x2 = Cells(5, 2): y2 = Cells(5, 3)
z = Cells(4, 5)
r = 10: cn = 5: cs = 4: ca = 2

If z = 0 Then
Call segment(x1, y1, x2, y2, cs)
Else
Call arrow(x1, y1, x2, y2, r, ca)
End If
Call node(1, x1, y1, r, cn)
Call node(2, x2, y2, r, cn)
End Sub
```

図2 NP1のメイン・プログラム

こばやし たかし 法政大学

〒184-8584 小金井市梶野町3-7-2

```

Sub node(i, x, y, r, c)
    ActiveSheet.Shapes.AddShape(msoShapeOval, x - r, y - r, r * 2, r * 2).Select
    Selection.ShapeRange.Fill.ForeColor.SchemeColor = c
    Selection.ShapeRange.Fill.Solid
    Selection.Characters.Text = Chr$(48 + i)
    With Selection.Font
        .Size = 13
    End With
End Sub

Sub segment(xe, ye, xf, yf, c)
    ActiveSheet.Shapes.AddLine(xe, ye, xf, yf).Select
    Selection.ShapeRange.Line.Weight = 2#
    Selection.ShapeRange.Line.ForeColor.SchemeColor = c
End Sub

Sub arrow(xe, ye, xf, yf, r, c)
    al = Sqr((xe - xf) ^ 2 + (ye - yf) ^ 2)
    p = r / al: q = 1 - p
    xxe = (p * xf + q * xe): yye = (p * yf + q * ye)
    xxf = xe + xf - xxe: yyf = ye + yf - yye
    ActiveSheet.Shapes.AddLine(xxe, yye, xxf, yyf).Select
    Selection.ShapeRange.Line.Weight = 2#
    Selection.ShapeRange.Line.ForeColor.SchemeColor = c
    Selection.ShapeRange.Line.EndArrowheadStyle = msoArrowheadTriangle
    Selection.ShapeRange.Line.EndArrowheadLength = msoArrowheadLengthMedium
    Selection.ShapeRange.Line.EndArrowheadWidth = msoArrowheadWidthMedium
End Sub

```

図3 サブルーチン・プログラム

<サブルーチン node>

機能 点 (円) を表示する。

引数 i, x, y, r, c

- i : 点の番号
- x : x 座標
- y : y 座標
- r : 半径
- c : 色番号

座標 (x, y) を中心にして、半径 r の円を描き、中を色番号 c の色で塗りつぶす。点の番号も表示する。

色番号と色の対応は、表 1 のとおりである。

<サブルーチン segment>

機能 二点を結ぶ線分を表示する。

引数 xe, ye, xf, yf, c

- xe : 線分の始点 (一端) の x 座標
- ye : 始点の y 座標
- xf : 線分の終点 (他端) の x 座標
- yf : 終点の y 座標
- c : 色番号

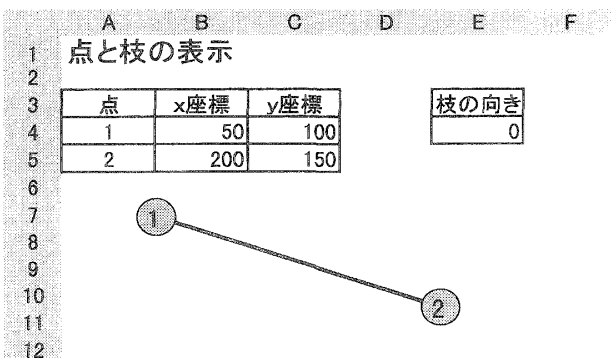


図4 NP1の実行例(1)

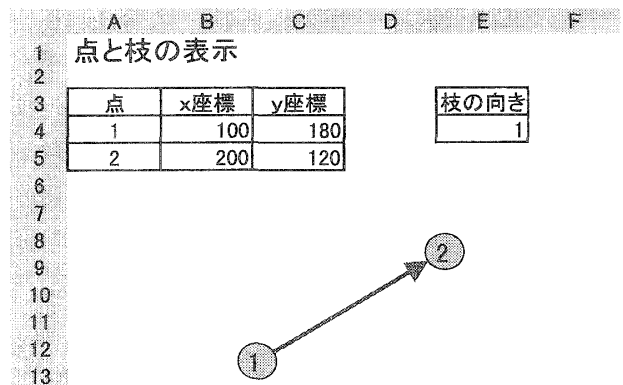


図5 NP1の実行例(2)

表1 色番号と色

色番号	1	2	3	4	5	6	7
色	白	赤	緑	青	黄	紫	空

座標 (xe, ye) と座標 (xf, yf) の間を結ぶ線分を色番号cの色で表示する。

〈サブルーチン arrow〉

機能 二点を結ぶ矢線を表示する。

引数 xe, ye, xf, yf, r, c

xe: 矢線の始点(根元)のx座標

ye: 始点のy座標

xf: 矢線の終点(先)のx座標

yf: 終点のy座標

r: 点の半径

c: 色番号

座標 (xe, ye) から座標 (xf, yf) へ向かう矢線を色番号cの色で表示する。

ここで、矢線が点と重ならないようにするために、長さを短くしてあることに注意されたい。

これらのプログラムは、一度お絵かき(図形描画)をしたときの「マクロの記録」から、不要の部分を削除して作成した。必要なことがどこに書いてあるのかわかりにくい市販の解説書を読むよりはるかに手取り早い。

次に、図2のプログラムでセルに書き込んだデータを読みとる部分を説明する。

セルは、Cells という名前の2次元配列の要素に対応している。1番目の添え字は、行番号、2番目の添え字は、列番号(ABCの順に、1, 2, 3, ……が対応)である。たとえば、セルB4に50を書き込むと、Cells(4, 2)の値は、50になる。

Cellsは、普通の配列と同じように使えるが、セルに数値を書き込むことは、データの入力(read文、scan文などの実行)に当たるから、一度プログラムで使う変数に入れておくほうが、プログラムがわかりやすいし、他の言語に変換するときも楽である。

図1では、セルB4, C4に一番目の点のx座標、y座標を書き込むので、

x1=Cells(4, 2)

y1=Cells(4, 3)

により、変数x1, y1にその点のx座標、y座標が入る(記憶される)。同様に、変数x2, y2に二番目の点のx座標、y座標が入る。

セルE4には、枝の向きの有無にしたがって、1ま

たは0が書き込まれるので、

z=Cells(4, 5)

によって、それを一度変数zに入れて、z=0であれば、サブルーチンsegmentを呼び出し、そうでなければ(z=1であれば)、サブルーチンarrowを呼び出している。なお、点の表示(サブルーチンnodeの呼び出し)が後になっていることに注意されたい。

(矢線の表示だけであれば、点の表示が先でもよい。)

なお、点の色は、黄色(色番号cn=5)、線分の色は青(同cs=4)、矢線の色は赤(同ca=2)になっている。

x座標は、左端が0であり、右が正の向きである。列との関係は、列の幅によって変わるから、適当に決めて、後で変更すればよいが、幅5がだいたい座標間隔100に対応している。y座標は、上端が0であり、下が正方向である。行の高さ10が、だいたい座標間隔100に対応している。図1では、列の幅を7.5、行の高さを15とした。

◆図形描画による図形の変更

一度表示した図形(点や枝)は、「図形描画」で「オブジェクトの選択」をすれば、変更できる。点の色を変えたいときは、「塗りつぶしの色」を、枝の色を変えたいときは、「線の色」をクリックすればよい。また、図形全体を移動したいときは、マウスを図形に近づけて、4方向への小さい矢印が表示されたら、マウスをドラッグすれば(押したまま動かせば)よい。したがって、点の座標を決めるときに、絶対的位置をあまり気にする必要はない。

点毎に位置を変えることも可能であるが、それに接続している枝は、一緒に移動しないから、一部の点の位置を変えたいときは、座標を変えて、もう一度マクロを実行したほうがよい。

2. 点对点接続行列

次に、有向グラフを表示し、その点对点接続行列を出力するマクロを示す。

ここで、点对点接続行列とは、大きさ(次数)が点の数であり、要素が次のように定まっている正方行列G=(g(i,j))である。

点iから点jに向かう枝が存在するとき、

g(i,j)=1, 存在しないとき、g(i,j)=0

とする。

[NP2] 図6に示すように、点の数n, 枝(矢線)

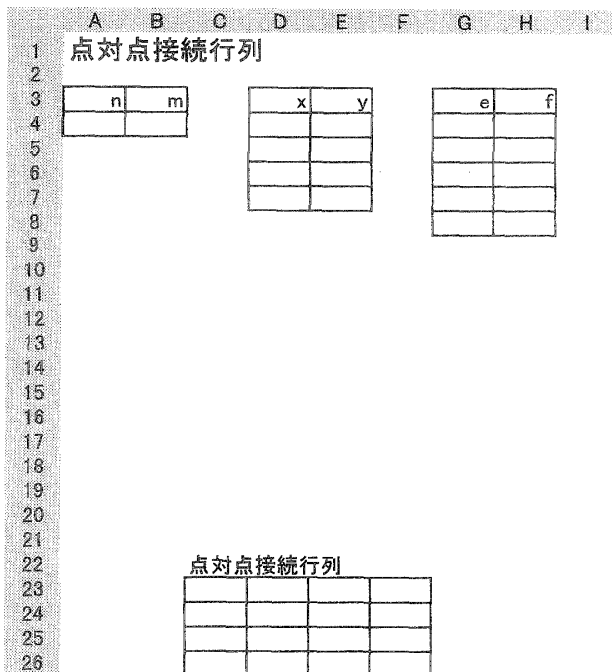


図6 NP2のシート

の数 m , 各点の x 座標, y 座標, 各枝の始点 e , 終点 f をセルに書き込むと, グラフを表示し, その点对点接続行列を指定されたセルに出力する.

プログラムを作成する前に, 表2のような入出力に使用する変数とセル番号の対応表を用意しておくとい.

プログラムを図7に示す(サブルーチンはNP1と同じである). 各点の x 座標, y 座標は, それぞれ配列 x, y に, 各枝の始点, 終点は, それぞれ配列 e, f に記憶されている. 点对点接続行列は, 配列 g に作成されるが, 要素 $g(i, j)$ を出力する(書きこむ)セルの行番号は, $22+i$, 列番号は, $2+j$ であるから,

$$\text{Cells}(22+i, 2+j) = g(i, j)$$

と書けばよい.

実行例を図8に示す.(列の幅は, 例1の2/3になっている.) (次号に続く)

表2 入出力に使う変数とセル番号の対応

変数	意味	セル番号
n	点の数	4, 1
m	枝の数	4, 2
$x(i)$	点 i の x 座標	$3+i, 4$
$y(i)$	点 i の y 座標	$3+i, 5$
$e(k)$	枝 k の始点	$3+k, 7$
$f(k)$	枝 k の終点	$3+k, 8$
$g(i, j)$	接続行列の (i, j) 要素	$22+i, 2+j$

```

Sub np2()
  ' 点对点接続行列
  Dim n, m, i, j, k, r, cn, ca As Integer
  Dim x(10), y(10), e(20), f(20) As Integer
  Dim g(10, 10) As Integer
  n = Cells(4, 1): m = Cells(4, 2)
  For i = 1 To n
    x(i) = Cells(3 + i, 4): y(i) = Cells(3 + i, 5)
  Next i
  For k = 1 To m
    e(k) = Cells(3 + k, 7): f(k) = Cells(3 + k, 8)
  Next k
  r = 10: cn = 5: ca = 4
  ' グラフの表示
  For i = 1 To n
    Call node(i, x(i), y(i), r, cn)
  Next i
  For k = 1 To m
    i = e(k): j = f(k)
    Call arrow(x(i), y(i), x(j), y(j), r, ca)
  Next k
  ' 接続行列の作成
  For i = 1 To n
    For j = 1 To n
      g(i, j) = 0
    Next j
  Next i
  For k = 1 To m
    i = e(k): j = f(k): g(i, j) = 1
  Next k
  ' 接続行列の出力
  For i = 1 To n
    For j = 1 To n
      Cells(22 + i, 2 + j) = g(i, j)
    Next j
  Next i
End Sub

```

図7 NP2のプログラム

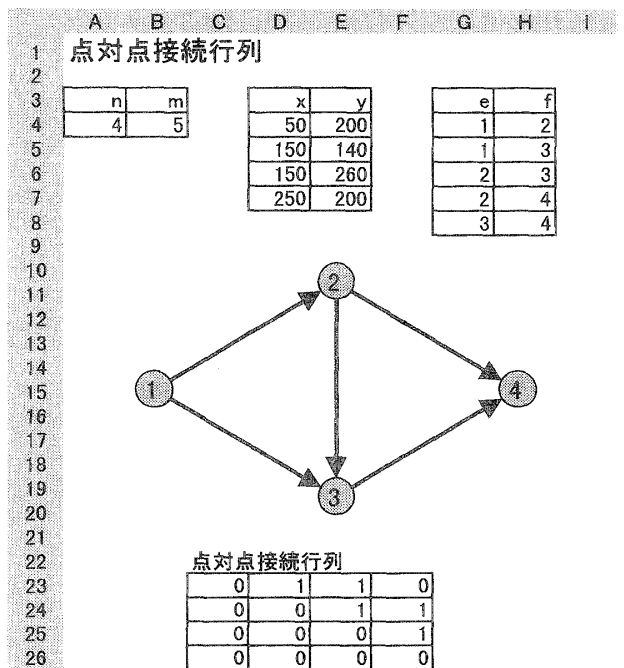


図8 NP2実行例