解説

Excelによるネットワーク計画法(3)
最小木と点の彩色

古林　隆

5. 最小木

今までは特に断っていないが，すべての点が枝で連結しているネットワークが与えられているとする．このとき，点の数をnとすると，(n−1)本の枝をうまく選ぶと，それらの枝だけで，すべての点を連結することができる．このようなn個の点と(n−1)個の枝からなる部分ネットワークを木（n個の点を張る木）という．木に含まれる枝だけでは，閉路はできないが，含まれない任意の枝を一本加えると閉路ができる．さらに，枝に長さが付与されているとき，木に含まれる枝の長さの和をその木の長さといい，長さが最小である木を最小木または最短木という．最小木を求めるのは，すべての点を結ぶ道路を建設したり，ケーブルを設置したりするときの費用を最小にしたいときなどに必要になる．

[NP 5]　図19に示すように，点の数n，枝の数m，各点のx座標，y座標，各枝の両端の点e，f，長さdをセルに書き込むと，ネットワークを表示し，最小木を求めて表示する．

＜注＞　x，yを書き込む行やe，f，dを書き込む行が足りなければ，必要なだけ挿入（またはコピー）すればよい．

◆アルゴリズム

ここでは，クラスカル法を用いる．枝を短い方から順に一本ずつ見て，すでに選ばれた枝と閉路ができなければ，その枝を選ぶことを繰り返し，選ばれた枝の本数がn−1になれば，終了する．このように，並んでいる順に，条件を満たせばどんどん選んでいく（食べていく）方法を貪欲法（グリーディ法）という．

初めに，枝を短い方から順に並べ変える．Excelでは，ツールで簡単に並べ替えができるから，入力するときには，あまり気にする必要はない．

短い方からk番目の枝の両端の点をe(k)，f(k)，長さをd(k)とする．（枝の向きは考慮しないから，e(k)，f(k)の順序は結果に影響しない．）すでに選ばれた枝に付け加えると閉路ができるかどうかを判定するため

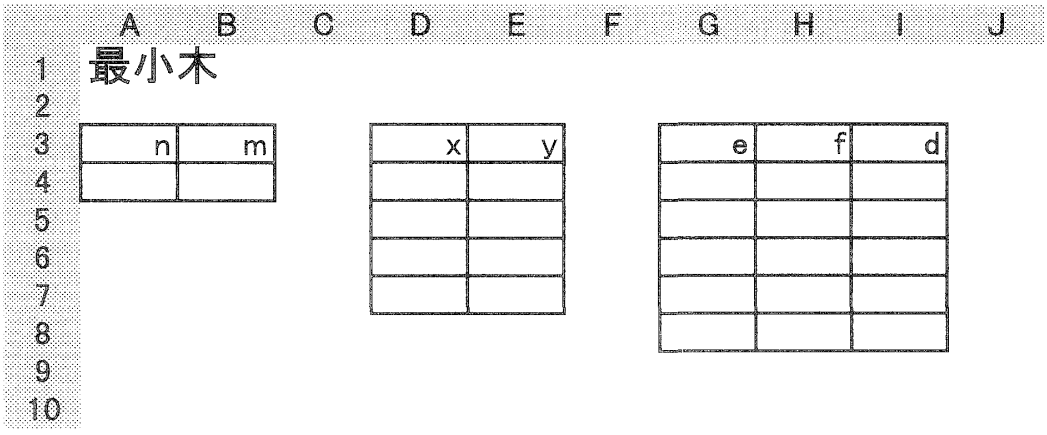

図19　NP 5のシート

こばやし　たかし　法政大学
〒184-8584 小金井市梶野町3-7-2

に，選ばれた枝で連結している点は同じグループになるように，点をグループ分けする．はじめは，枝が一本も選ばれていないから，一個の点だけからなるn個のグループが存在する．枝が一本選ばれる毎にグループの数は1ずつ減っていく．いま見ている枝を選んでよいかどうかは，その両端の点のグループを比較すればよい．同じであれば，両端の点はすでに連結していて，この枝を加えると閉路ができるので，選べないが，異なっていれば，選べる．選ぶと，両端のグループをどちらかに統一すればよい．手順は，次の通りである．

(1) group(i)=i (i=1, 2, …, n), nt=0, dt=0, k=0

(2) nt=n−1になるまで以下を繰り返す（後判定）．

(2.1) k=k+1, i=e(k), j=(k)

(2.2) group(i)≠group(j)であれば，以下の(2.2.1)から(2.2.4)を行う．

(2.2.1) nt=nt+1

(2.2.2) i, j, d(k)をセルに出力し，枝の色を変える．

(2.2.3) dt=dt+d(k)

(2.2.4) group(h)=group(j)であるすべてのhに対して，

group(h)=group(i)とする．

group(i)は，点iが属しているグループの番号，ntは，選んだ枝の数，dtは，それらの長さの和を示している．グループを記憶するのには，前回のNP4の配列q，tail，nexのように，グループの先頭と末尾および各点に対して同じグループに属する次の点を記憶する三つの配列を用意した方がよいが，ここでは，簡単に，各点のグループ番号を記憶する配列だけですませた．

```
Dim n As Integer, m As Integer, i As Integer, j As Integer, k As Integer
Dim x(10) As Integer, y(10) As Integer
Dim e(20) As Integer, f(20) As Integer, d(20) As Integer
Dim r As Integer, cn As Integer, cs As Integer, ct As Integer
Dim group(10) As Integer, nt As Integer, dt As Integer
Dim u As Integer, h As Integer, gj As Integer

' データの入力
n = Cells(4, 1): m = Cells(4, 2)
For i = 1 To n
  x(i) = Cells(i + 3, 4): y(i) = Cells(i + 3, 5)
Next i
For k = 1 To m
  e(k) = Cells(k + 3, 7): f(k) = Cells(k + 3, 8)
  d(k) = Cells(k + 3, 9)
Next k

' ネットワークの表示
r = 10:  cn = 5: cs = 4: ct = 2
For k = 1 To m
   i = e(k): j = f(k)
   Call segment(x(i), y(i), x(j), y(j), cs)
Next k
For i = 1 To n
   Call node(i, x(i), y(i), r, cn)
Next i
```

図20　データの入力とネットワークの表示

◆プログラム

データの入力およびネットワークの表示の部分を図20に示す．（サブルーチンについては，第1回の図3

```
' 最小木の作成と表示
  For i = 1 To n: group(i) = i: Next
  nt = 0: dt = 0:  k = 0: u = m + 17
  Do
    k = k + 1
    i = e(k): j = f(k)
    If group(i) <> group(j) Then
      nt = nt + 1
      Cells(u + nt, 4) = i
      Cells(u + nt, 5) = j
      Cells(u + nt, 6) = d(k)
      Call segment(x(i), y(i), x(j), y(j), ct)
      dt = dt + d(k)
      gj = group(j)
      For h = 1 To n
        If group(h) = gj Then group(h) = group(i)
      Next h
    End If
  Loop Until nt = n - 1

  Cells(u + n, 6) = dt
  For i = 1 To n
     Call node(i, x(i), y(i), r, cn)
  Next i
```

図21　最小木の作成と表示

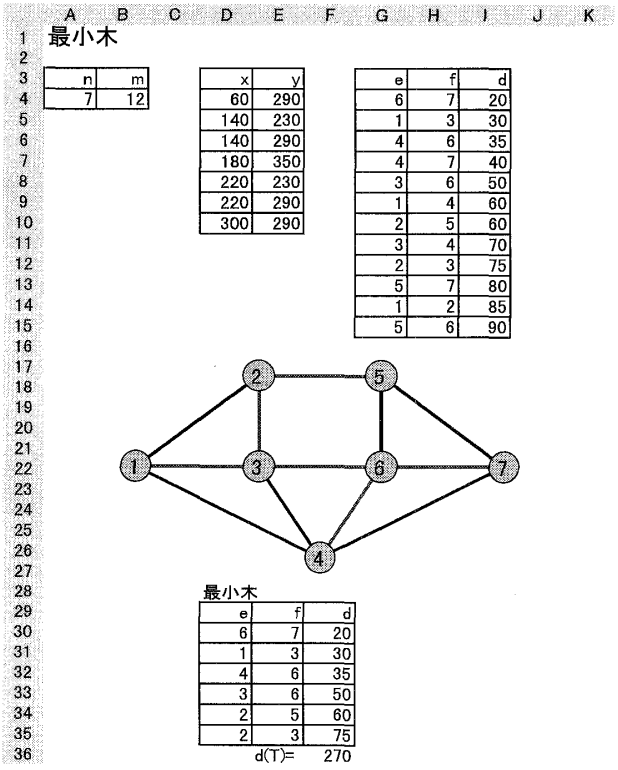

図22　NP5の実行例

を参照されたい。）点および枝の色は，前回のNP 3と同じで，それぞれ，黄（色番号 cn＝5），青（色番号 cs＝4）である．変数 ct は，ここでは使用していないが，後で最小木を示す枝の色番号に用いている．最小木を求めて，表示する部分を図21に示す．枝が一本選ばれ毎に，その両端の点と長さをセルに出力するとともに，枝の色を赤（色番号 ct＝2）に変えている．

実行例を図22に示す．ディスプレイでは，最小木に含まれる枝 (6, 7)，(1, 3)，(4, 6)，(3, 6)，(2, 5)，(2, 3) は，赤で表示されている．

6. 点の彩色

ここでは，隣接している二点（一本の枝の両端の点）の色が異なるように，点を彩色する（点に色を割り当てる）ことを考える．最小の数の色で点を彩色するのは，同じグループに入れてはいけないもの（品物，地域など）の対がいくつか与えられている集合の分割に必要になる．

[NP 6] 図23に示すように，点の数 n，枝の数 m，各点の x 座標，y 座標，各枝の両端の点 e，f をセルに書き込むと，ネットワークを表示し，点の彩色を行う．ただし，e<f とする．

◈アルゴリズム

いろいろなアルゴリズムが提案されているが，ここでは，一つ色を選んでは，その色を割り当てられる点を番号順に見つけて次々割り当てていくことを繰り返すことにする．

点 i に割り当てる色の番号を color(i) に記憶するが，初期値（割り当てられていないとき）は，0 とする．ただし，いま選んでいる色が隣接している点に割り当てられているときは，その色を割り当てることができないので，−1 とする．

隣接点をとりだすために，第1回の NP 2 で求めた接続行列 g(i, j) を用いる．データ入力のときに，e<f とするので，i>j であれば，g(i, j)＝0 であることに注意されたい．

手順は，次のとおりである．

(1) color(i)＝0 (i＝1, 2, …, n)，np＝0，cn＝0 とする．

(2) np＝n になるまで以下を繰り返す（後判定）．

(2.1) cn＝cn＋1 とする．

(2.2) i＝1, 2, …, n の順に，color(i)＝0 であれば，以下を実行する．

(2.2.1) color(i)＝cn，np＝np＋1 とする．

(2.2.2) j＝i＋1, …, n の順に，g(i, j)＝1 で color(j)＝0 であれば，color(j)＝−1 とする．

(2.3) i＝1, 2, …, n の順に，color(i)<0 であれば，color(i)＝0 とする．

なお，この方法では，最適解が得られる（色の数が最小になる）とは限らない．

◈プログラム

データをセルから読みとって，ネットワークを表示し，接続行列を求める部分は，NP 2 と同じであるから省略する．点の色番号の決定とそれらの色で点を表示する部分を図24に示す．ここで，色番号は，1から順につけているが，色番号1は，白であるため，表示するときには，それに1を加えた番号を用いている．

(color(i) が1である点は，赤，2である点は，緑で表示される．)

実行例を図25に示す．

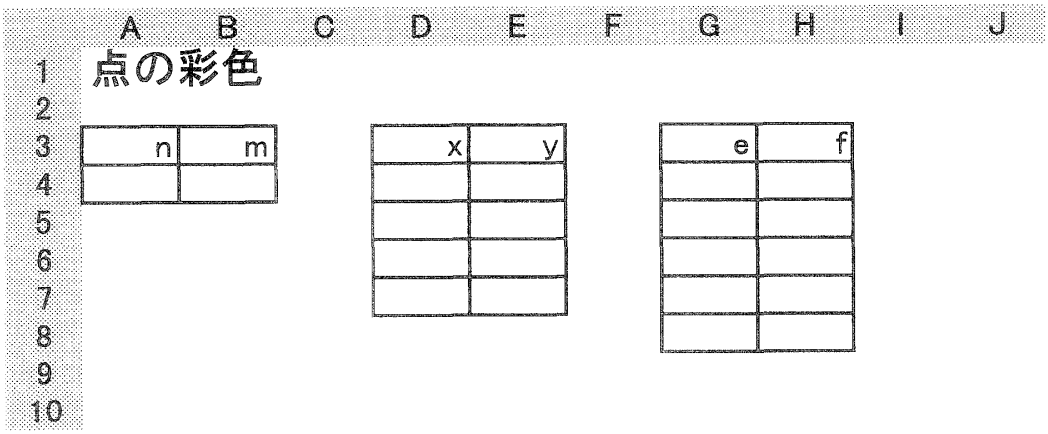

図23 NP 6 のシート

```
' 点の色番号の決定
  For i = 1 To n: color(i) = 0: Next i
  np = 0: cn =0
  Do
    cn = cn + 1
    For i = 1 To n
      If color(i) = 0 Then
        color(i) = cn: np = np + 1
        For j = i + 1 To n
          If (g(i, j) = 1 And color(j) = 0) Then color(j) = -1
        Next j
      End If
    Next i

    For i = 1 To n
      If color(i) < 0 Then color(i) = 0
    Next i
  Loop Until np = n

' 点の彩色
  u = m + 18
  For i = 1 To n
     cn = color(i)
     Cells(u, 2 + i) = cn
     Call node(i, x(i), y(i), r, cn + 1)
  Next i
```

図24　点の色番号の決定

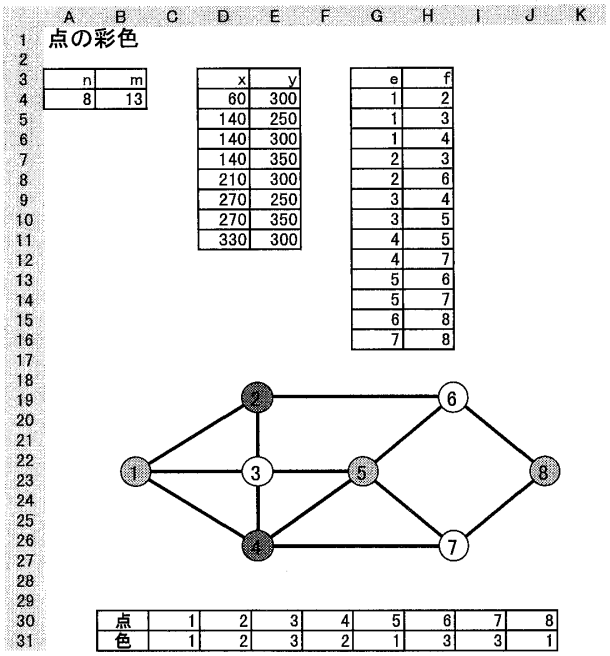

図25　NP 6 の実行例

7.　入力シートと出力シート

これまでに示したプログラムでは，入力（データ）と出力（ネットワーク，計算結果）の関係がわかりやすいように，入力用のシートと出力用のシートを同じ(ActiveSheet) にしている．そのために，枝の数 m によって出力する位置（行番号）をずらせる煩わしさが生じたが，これは，入力用のシートと出力用のシートを別にすることによって除かれる．

シートを指定するときは，Cells の前に，シート名を付ける．たとえば，

n＝Sheet1. Cells(4, 1)

Sheet2. Cells(u, 2＋i)＝cn

とすればよい．サブルーチンで点や枝を表示するシートが，ActiveSheet になっているので，プログラムでは，入力のときだけシートを指定しておいて，出力用のシート（たとえば，Sheet2）を開いて，そこでマクロを実行すれば，ネットワークと計算結果だけが，そのシートに表示される．

8.　おわりに

この解説では，Excel のマクロを使うことに主眼をおいたので，用いたアルゴリズムには，無駄が多いものもある．計算時間をもっと短縮したいと思われるかたは，アルゴリズムの解説書（Excel の解説書ではない）を参照して，プログラムを改良されたい．

Excel では，いろいろなことが計算できて，その結果を出力（表示）することができるが，自分がしたいことに対して，どのようなプログラムを書けばよいかを教えてくれる文献がすぐに見つかるとは限らない．第 1 回にも書いたが，自分の目的に合う文献を探すのに時間をかけるよりは，マクロの記録をとって，それを改良していく方がよい．

皆さんの新たな発見を期待しています．

■訂正

第 1 回（4 月号）の図 2，図 7 のプログラムの中の変数の型宣言文に誤りがありましたので，訂正します．

図 2 の dim 文で，次のように，すべての変数の後に As Integer を付けてください．

Dim x1 As Integer, y1 As Integer,（途中省略）cs As Integer, ca As Integer

図 7 の上の二つの dim 文についても同様に，すべての変数または配列の後に As Integer を付けてください．As Integer などの型の指定は，直前の一つの変数（配列を含む）だけに有効です．省略すると，実数でも文字列でも格納できるバリアント型（16 バイト）になりますので，すべての変数を整数型にするときは，上に記したように，すべての変数に As Integer を付ける必要があります．

本解説で紹介したマクロでは，数値を実数型で記憶しても，不都合は起こらないようですが，バリアント型の変数は，メモリーを多くとる（整数型の 8 倍）だけでなく，処理時間も大きくなりますから，面倒でもすべて整数型の宣言をしましょう．