解説

Excelによるネットワーク計画法(2)
最短路と配達路

古林　隆

3. 最短路

枝の長さが与えられているネットワークで，ある一点sから他の一点tへの路の中で，長さが最小であるものを点sから点tへの最短路といい，その長さを最短距離という．ここで，点s, 点tをそれぞれ路の始点，終点という．枝の長さとしては，そこを通過するのにかかる時間やそこに物を運ぶときの費用をとることもあるので，枝の長さは，向きによって異なっていてもよいし，負であってもよいことにする．

最短路・最短距離を求めるアルゴリズムは，次の三つの場合に分けて考えられている．

(a) 枝の長さがすべて非負であるネットワークで，ある一点から他のすべての点への最短路を求める場合

(b) 負の長さの枝が存在するネットワークで，ある一点から他のすべての点への最短路を求める場合

(c) 全最短路（すべての二点間の最短路）を求める場合

最短路を求めることは，それ自体が目的になることもあるが，むしろ他の問題のアルゴリズムの中のサブルーチンとして使われたり，入力データを作るのに使われたりすることが多い．ここでは，巡回セールス問題，配達路問題など多くの問題の入力データを作るのに必要な(c)の場合をとりあげる．

＜注＞　グラフとネットワークは，ほとんど同じ意味で使われるが，今回からは，接続関係以外に，枝に長さや点に配達量などが付与されるので，ネットワークということにする．

[NP 3]　図9に示すように，点の数n，枝の数m，各点のx座標，y座標，各枝の始点e，終点f，順方向（始点から終点への向き）の長さde，逆方向（終点から始点への向き）の長さdfをセルに書き込むと，ネットワークを表示し，すべての2点間の最短距離および次点（最短路で始点の次に通る点）をセルに出力する．さらに，路の始点sと終点tを入力すると，点sから点tへの最短路を矢線で表示する．

ただし，始点と終点が一致している枝（輪ということもある）は，存在しないものとする．

＜注＞　x，yを書き込む行の数は，nであり，e，fなどを書き込む行の数は，mである．n，mが定まってから，足りない行は，挿入（またはコピー）すればよい．

◆アルゴリズム

ここでは，ウォーシャル・フロイド法を用いる．

すべてのi，jに対して，d(i, j)が点iから点jへの最短距離，p(i, j)が次点（点iの次に通る点）になるようにするが，まず，それらの初期値を，次のように定める．

d(i, j)の初期値

(a) 点iと点jの間に枝が存在するとき，点iから点jへの向きの長さ

(b) 点iと点jの間（i≠j）に枝が存在しないとき，十分大きい正数M

(c) i=jのとき，0

p(i, j)の初期値 j

次に，h=1, 2, …, nの順に，すべてのi, jに対して，次の(1)，(2)を実行する．

(1) dh=d(i, h)+d(h, j)とする．

(2) dh<d(i, j)であれば，d(i, j)=dh, p(i, j)=p(i, h)とする．

終了したときのd(i, j)が，点iから点jへの最短距離であり，p(i, j)が次点である．最短路の一部も，また最短路になっているから，次点を順々に見ていけば(i=p(i, j)とすることを繰り返せば)，最短路が求められる．

負の長さの枝が存在する場合，最短距離が存在しな

こばやし　たかし　法政大学
〒184-8584 小金井市梶野町3-7-2

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

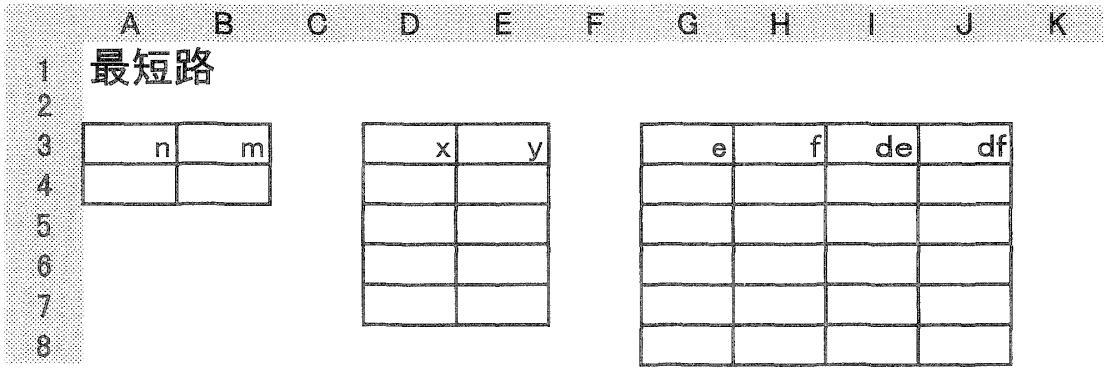


図9　NP3のシート

```
' ネットワークの表示
  r = 10:  cn = 5: cs = 4: ca = 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10　ネットワークの表示のプログラム

```
' 最短距離・次点の計算
  For i = 1 To n
    For j = 1 To n
      d(i, j) = 999: p(i, j) = j
    Next j
    d(i, i) = 0
  Next i
  For k = 1 To m
    i = e(k): j = f(k)
    d(i, j) = de(k): d(j, i) = df(k)
  Next k

  For h = 1 To n
    For i = 1 To n
      For j = 1 To n
        dh = d(i, h) + d(h, j)
        If dh < d(i, j) Then
          d(i, j) = dh: p(i, j) = p(i, h)
        End If
      Next j
    Next i
  Next h

' 最短距離・次点の出力
  u = m + 16
  For i = 1 To n
    Cells(u + i, 1) = i
    Cells(u, 1 + i) = i: Cells(u, n + 1 + i) = i
    For j = 1 To n
      Cells(u + i, 1 + j) = d(i, j)
      Cells(u + i, n + 1 + j) = p(i, j)
    Next j
  Next i
  Stop
```

図11　最短距離・次点の計算と出力のプログラム

いことがある．そのときは，d(i, i) の中に負が現れる．

◆プログラム

ここでも，点と枝を表示するサブルーチンを用いる(前回の図3参照)．

データの入力は，NP2のプログラム（前回の図7）のe(k)，f(k) を読み込む部分の次に，de(k)，df(k) を読み込む部分

de(k)＝Cells(3＋k, 9)　：df(k)＝Cells(3＋k, 10)

を付け加えればよい．

ネットワークの表示の部分を図10に示す．枝の表示に，サブルーチン segment を用いた．点を表示するサブルーチン node より先に実行していることに注意されたい．なお，点の色は，黄色（色番号 cn＝5)，枝の色は，青（色番号 cs＝4）である．変数 ca は，ここでは使用していないが，後で最短路を示す矢線の色番号に用いている（図13参照)．最短距離・次点の計算と出力の部分を図11に示す．ここで，M＝999とした．また，m（枝の数）が大きくなると，最短距離・次点を出力する行を下げる必要があるので，行番号が m によって変わるようにした．

実行例を図12に示す．なお，点番号の背景を塗りつぶすことや表の罫線の一部を太くするのは，プログラムを実行した後シート上で範囲を指定して行った．

最短距離・次点を出力した後，路の始点と終点を与えると，最短路を矢線で表示するようにするには，図13に示すプログラムを付け加えればよい．m＝12のとき，始点は，セルB18に，終点は，セルB19に書き込むことになっている．B18が0（空欄でもよい）であれば，何も行わない．

s＝3，t＝5としたときの実行例を図14に示す．矢線の色は，赤（色番号 ca＝2）である．

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

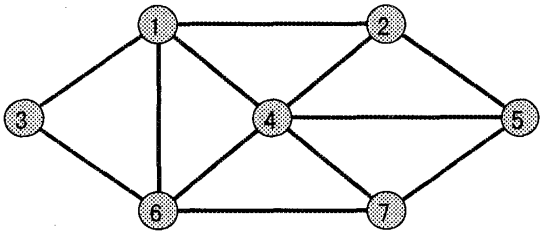
最短路

n	m
7	12

x	y
170	250
290	250
100	300
230	300
360	300
170	350
290	350

e	f	de	df
1	2	60	60
1	3	55	65
1	4	30	40
1	6	80	80
2	4	45	50
2	5	70	65
3	6	50	60
4	5	90	90
4	6	55	55
4	7	70	80
5	7	70	60
6	7	80	75

始点	
終点	
距離	



	最短距離							次　点						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
1	0	60	55	30	120	80	100	1	2	3	4	4	6	4
2	60	0	115	45	70	100	115	1	2	1	4	5	4	4
3	65	125	0	95	185	50	130	1	1	3	1	1	6	6
4	40	50	95	0	90	55	70	1	2	1	4	5	6	7
5	125	65	180	90	0	145	70	2	2	2	4	5	4	7
6	80	105	60	55	140	0	80	1	4	3	4	7	6	7
7	120	125	135	80	60	75	0	4	5	6	4	5	6	7

図12　NP 3 の実行例

```
' 最短路の表示
  u = m + 6
  s = Cells(u, 2): t = Cells(u + 1, 2)
  If s > 0 Then
    Cells(u + 2, 2) = d(s, t)
    i = s
    Do While i <> t
      j = p(i, t)
      Call arrow(x(i), y(i), x(j), y(j), r, ca)
      i = j
    Loop
  End If
```

図13　最短路の表示のプログラム

始点	3
終点	5
距離	185

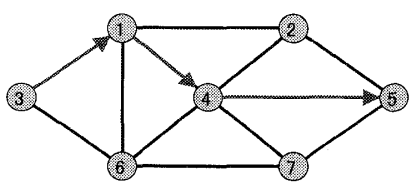


図14　最短路の表示

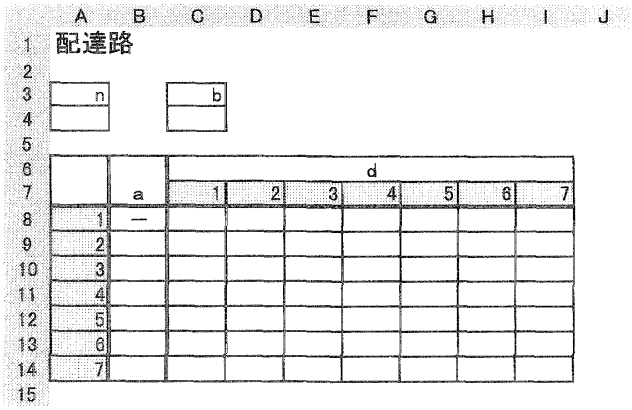
4.　配達路

ネットワークの一点（センターと呼ぶことにする）から他のすべての点へ品物を配達するときに，配達車の走行距離を最小にする配達路を求めることを考える．配達車の容量（積載可能量）が総配達量以上であれば，すべての品物を一度に載せられるので，配達順序だけ求めればよい．走行距離を最小にする配達順序を求める問題は，巡回セールスマン問題と呼ばれている．

ここでは，配達車の容量が総配達量より小さいとする．この場合は，センターから何回かに分けて配達しなければならない．すなわち，配達先をそれらへの配達量の和が配達車の容量を超えないように分けて，それぞれについて，配達順序を決めて，走行距離の和が最小になるようにしなければならない．

[NP 4]　図15に示すように，点の数 n，各点への配達量 a，すべての二点間の最短距離 d，車の容量 b を

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．



配達路

n	b

	a	d						
		1	2	3	4	5	6	7
1	—							
2								
3								
4								
5								
6								
7								

図15　NP4のシート

セルに書き込むと，配達順序，積載量をセルに出力する。ここで，センターは，点1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，最も単純なセービング法を用いることにする。

まず，センターからすべての点へ往復する(n-1)個の配達路を初期解とし，二つの配達路を結合することを繰り返して，総走行距離を減少させていく。

用いる変数の意味は，次のとおりである。ただし，いずれもi=2,3,…,nとする。

q(i)：積載量

センターを出発して最初に配達する点iに対して，その配達路で配達する量の和を示す。その他の点については，0とする。

nex(i)：点iの次に配達する点

nex(i)=1であれば，点iからセンターにもどることを示す。

tail(i)：最後に配達する点

q(i)>0であるiに対して，その配達路（最初に配達する点がiである配達路）で最後に配達する点を示す。

dr(i)：走行距離

q(i)>0であるiに対して，その配達路の走行距離を示す。

s(i,j)(i≠j)：セービング値

点iからセンターにもどって，改めて点jに行くかわりに，点iから直接点jに行くことによって，減少する（節約される）距離を示す。次式で計算できる。

$$s(i,j)=d(i,1)+d(1,j)-d(i,j) \qquad (*)$$

手順は，次のとおりである。

```
' セービング値の計算
  u1 = n + 9
  For i = 2 To n
    For j = 2 To n
      If i <> j Then
        s(i, j) = d(i, 1) + d(1, j) - d(i, j)
        Cells(u1 + i, 2 + j) = s(i, j)
      Else
        Cells(u1 + i, 2 + i) = "   ***"
      End If
    Next j
  Next i

' 初期解
  For i = 2 To n
    q(i) = a(i): nex(i) = 1: tail(i) = i
    dr(i) = d(1, i) + d(i, 1)
  Next i

' 配達路の結合
  Do
    smax = 0
    For i = 2 To n
      If q(i) > 0 Then
        For j = 2 To n
          If i <> j And q(j) > 0 Then
            ti = tail(i)
            If q(i) + q(j) <= b And s(ti, j) > smax Then
              smax = s(ti, j): g = i: h = j
            End If
          End If
        Next j
      End If
    Next i
    If smax > 0 Then
      nex(tail(g)) = h: tail(g) = tail(h)
      dr(g) = dr(g) + dr(h) - smax
      q(g) = q(g) + q(h): q(h) = 0
    End If
  Loop Until smax = 0
```

図16　セービング法のプログラム

(1)　セービング値の計算

上の(*)式でセービング値を計算する。

(2)　初期値の設定

すべてのi(i≠1)に対して，

$q(i)=a(i)$, $nex(i)=1$, $tail(i)=i$,

$dr(i)=d(1,i)+d(i,1)$

とする。

(3)　配達路の結合

次の手順をsmax=0になるまで繰り返す。（後判定）

(3.1)　$i\neq1, j\neq1, i\neq j, q(i)>0, q(j)>0,$

$q(i)+q(j)\leqq b$

をすべて満たす(i,j)の中で，s(tail(i),j)を最大にする対を求め，それを(g,h)，最大値をsmaxとする。ただし，そのような(i,j)が存在しない

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

```
' 配達路の出力
  nr = 0: drt = 0: u2 = u1 + 3 + n
  For i = 2 To n
    If q(i) > 0 Then
      ii = i: nr = nr + 1: k = 2
      Do
        k = k + 1
        Cells(u2 + nr, k) = ii
        ii = nex(ii)
      Loop Until ii = 1
      Cells(u2 + nr, 8) = dr(i)
      Cells(u2 + nr, 9) = q(i)
      drt = drt + dr(i)
    End If
  Next i
  Cells(u2 + nr + 1, 8) = drt
```

図17　配達路の出力のプログラム

ときは，smax＝0 とする．

(3.2)　smax＞0 であれば，配達路を結合する．

$nex(tail(g))=h$, $tail(g)=tail(h)$,

$dr(g)=dr(g)+dr(h)-smax$,

$q(g)=q(g)+q(h)$, $q(h)=0$

とする．

◆プログラム

セービング値を計算して，配達路を求めるまでの部分を図16に，配達路を出力する部分を図17に示す．セービング値を出力する行の番号および配達路を出力する行の番号は，点の数nによって変わるようになっている．

NP3の例で用いたネットワークに配達量などを付与して実行した例を図18に示す．最短距離dは，図12のセルB29からH35までをコピーして，貼り付けた．

求められた配達路は，

R1：1→2→1

R2：1→3→6→4→1

R3：1→7→5→1

の3本であり，総走行距離は，605である．

配達路

n	b
7	1000

	a	d						
		1	2	3	4	5	6	7
1	—	0	60	55	30	120	80	100
2	500	60	0	115	45	70	100	115
3	200	65	125	0	95	185	50	130
4	300	40	50	95	0	90	55	70
5	350	125	65	180	90	0	145	70
6	450	80	105	60	55	140	0	80
7	400	120	125	135	80	60	75	0

セービング値

	2	3	4	5	6	7
2	***	0	45	110	40	45
3	0	***	0	0	95	35
4	50	0	***	70	65	70
5	120	0	65	***	60	155
6	35	75	55	60	***	100
7	55	40	70	180	125	***

配達路

	配達順序	距離	積載量
R1	2	120	500
R2	3 6 4	200	950
R3	7 5	285	750
	距離の総和	605	

図18　NP4の実行例

◆プログラムの改良

上述の方法では，最適解（総走行距離を最小にする配達路）が求められるとは限らない．別の解を求めたければ，(g, h)に選ばれた(i, j)（図18の例では，(3, 6)，(6, 4)，(7, 5)）の一部に対して，$s(i,j)=0$ とおいて，配達路の結合をやり直すようにすればよい．他のセービング値を変える必要はないので，初期解を求める部分にもどれるようにして，指定した(i, j)に対して，$s(i,j)=0$ とおいた後，配達路の結合を実行すればよい．（プログラムを変えたくなければ，無駄が多いが，表の $d(i,j)$ の値を $d(i,1)+d(1,j)$ にすればよい．）

（次号に続く）

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．