最長経路問題の代数的解法についての考察と応用

2014SS034 川瀬諒治
指導教員：小藤俊幸

1　はじめに

ある仕事について最短終了時刻を考える問題を日程計画問題という．この問題を考える際に用いる方法として参考文献 [2] にある方法は，作業の前後関係を表にまとめ，アロー・ダイアグラムと呼ばれる作業進行過程を視覚化したものを作成してクリティカルパスを発見するというものである．本研究ではこの方法で用いられているアロー・ダイアグラムを有向グラフとみなせば，グラフの最長経路を考える問題と読み替えることができることに注目し，参考文献 [1] にある最長経路問題を代数的に解く方法を C 言語でプログラミングし日程計画問題を解決する．C 言語で代数的解法をプログラミングする際，問題となる重み 0 となる辺の扱いや順路をどのように計算するかを考え，得られた結果を考察する．

2　グラフ

グラフとは，いくつかの頂点の集合 V とそれらを結ぶ辺の集合 E によって定義されるものであり，一般に $G(V,E)$ と表される．辺には向きがついている場合とついていない場合がある．ついている場合を有向グラフといい，ついていない場合を無向グラフという．本研究においてはグラフといえば有向グラフを指すものとする．グラフ G のある頂点 $i \in V$ からある頂点 $j \in V$ への辺を $(i,j) \in E$ と表すとする．また，グラフ G の各辺の重みを a_{ij} とする．
図 1 は
V={1,2,3,4},
E={(1,2),(1,3),(2,3),(2,4),(3,4)}
$a_{12}=1$, $a_{13}=4$, $a_{23}=2$, $a_{24}=5$, $a_{34}=7$
と表されるグラフ $G(V,E)$ である．

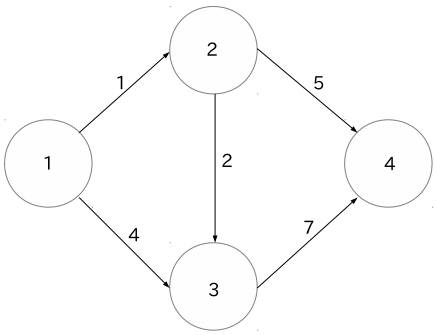

図 1　有向グラフの例

3　最長経路の代数的解法

3.1　最大積

代数的解法を考えるために，まず最大積の考え方を定義する．行列 A を $m \times n$ 行列とし，行列 B を $n \times p$ 行列とする．行列 A，B の (i,j) 成分を A_{ij}，B_{ij} と表現する．行列 A と行列 B に対して最大積をとると，$m \times p$ 行列 C ができるとする．このとき，C_{ij} を以下のように計算したものを最大積とする．

$$C_{ij} = \max(A_{i1}+B_{1j}, A_{i2}+B_{2j}, \cdots, A_{in}+B_{nj}) \quad (1)$$

ただし，$A_{ik} < 0$ または $B_{kj} < 0$ の場合は $A_{ik}+B_{kj} = -1$ とする．そして $*$ は最大積を表す演算子とする．例えば，

$$A = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}, B = \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \end{pmatrix}$$

という行列に対して最大積を計算すると，

$$\begin{pmatrix} 1 & 2 & 3 \end{pmatrix} * \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$= \begin{pmatrix} \max(1+1,-1,3+1) & \max(-1,2+1,-1) \end{pmatrix}$$

$$= \begin{pmatrix} \max(2,-1,4) & \max(-1,3,-1) \end{pmatrix} = \begin{pmatrix} 4 & 3 \end{pmatrix}$$

となる．

3.2　最大和

次に，最大和の考え方を定義する．行列 A，B を $m \times n$ 行列として，その和が $m \times n$ 行列 C となるとする．その行列 C の (i,j) 成分 C_{ij} を

$$C_{ij} = \begin{cases} A_{ij} & (A_{ij} \geq B_{ij} \text{のとき}) \\ B_{ij} & (B_{ij} \geq A_{ij} \text{のとき}) \end{cases} \quad (2)$$

と定義する．
例えば，

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 3 \end{pmatrix}, B = \begin{pmatrix} 2 & -1 \\ 1 & 1 \end{pmatrix}$$

という行列に対して最大和を計算すると，

$$\begin{pmatrix} 1 & 1 \\ 0 & 3 \end{pmatrix} + \begin{pmatrix} 2 & -1 \\ 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \max(1,2) & \max(1,-1) \\ \max(1,0) & \max(3,1) \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$$

となる．

3.3 距離行列

次に，距離行列について定義する. 距離行列とはあるグラフが与えられたとき，グラフの頂点 v と頂点 w の間に辺がある場合その辺の重みを行列の (v,w) 成分に割り当てたものである．一般的に頂点が連結していない場合は 0 を割り当てることが多いが，日程計画問題を考える際に辺の重みが 0 となることがあるので本研究においては連結していない場合，別の数を割り当てなければならない．そこで私は距離行列において頂点が連結していない場合，-1 を割り当てる方法を考えた. そうすることで問題なく代数的解法を用いて日程計画問題を解くことができる.
距離行列 V の $i \in \mathbb{N}$ 個の最大積 V^i の元は,i 個の辺を用いたそれぞれの割り当てられた頂点から頂点への最長距離を表している. ただし，i 個の辺でたどり着くことができない場合は -1 で表される.
つまり，$m \in \mathbb{N}$ 個の頂点を持つグラフに対して

$$\sum_{i=1}^{m-1} V^i = V + V^2 + ... + V^{m-1} \tag{3}$$

の各元は割り当てられた頂点間の最長経路の長さと連結の有無を表している.

4 最長経路問題を解くプログラム

最長経路問題を解くプログラムを実際に C 言語を用いて組んだ. ここでの課題は最長経路の代数的解法は，どの経路を通ったかが計算をしただけでは分からないということである．これをどのように解消したかというと，参考文献 [1] に記載されていた最大積を求める際に添字を残すという方法をもとに，最大積を計算する際 2 重 for 文を用いて計算することを利用し，最短経路を配列に保存する方法を考えた．詳しくは本稿に記載する．プログラムの構成は，最大積を求める関数，最大和を求める関数，配列を用いて順路を計算する関数，$main$ 関数となっている．プログラム全体も本稿に記載する.

5 最長経路の計算の応用

ここで，実際に最長経路問題をプログラムを用いて解いてみる．例題として，参考文献 [2] にある日程計画問題を取り挙げる．インスタントラーメンの作成作業を A:丼準備，B:湯沸かし，C:スープ作り，D:麺ゆで,E:盛り付け，の 5 つの作業に分ける．A,B,C,D,E はそれぞれ作業時間として 3 分，4 分，1 分，3 分，2 分かかるものとする. ここで考えるのはインスタントラーメンを作る作業を最短何分で終了できるかということである. 参考文献 [2] にあるようにこの作業のアロー・ダイアグラムを作成し，それを有向グラフとみなせば，V={1,2,3,4,5}
E={(1,2),(1,3),(2,3)(2,4),(3,4),(4,5)}
$a_{12} = 4$, $a_{13} = 3$, $a_{23} = 0$ $a_{24} = 3$, $a_{34} = 1$, $a_{45} = 2$
というグラフ $G(V,E)$ のクリティカルパスの長さが分かれば最短何分で作業を終了できるかが分かる.

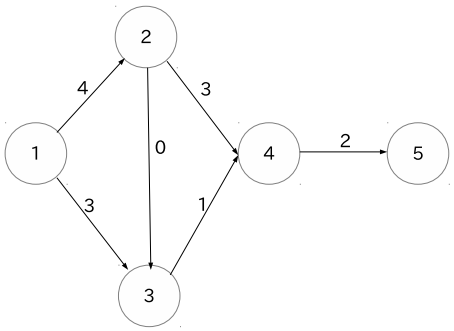

図 2 例題

このとき，このグラフの距離行列は

$$V = \begin{pmatrix} -1 & 4 & 3 & -1 & -1 \\ -1 & -1 & 0 & 3 & -1 \\ -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & 2 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

となる．このグラフの最長経路を計算すると，

$$\sum_{i=1}^{4} V^i = \begin{pmatrix} -1 & 4 & 3 & 7 & 9 \\ -1 & -1 & 0 & 3 & 5 \\ -1 & -1 & -1 & -1 & 3 \\ -1 & -1 & -1 & -1 & 2 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix} \tag{4}$$

となる．これにより (5) の 1 行 5 列の元に注目すると頂点 1 から頂点 5 への最長経路の長さが 9 であることが分かる．そしてこれがこのアロー・ダイアグラムのクリティカルパスであり，例題の作業は最短 9 分で終了できると分かる．さらに，このクリティカルパスは $1 \to 2 \to 4 \to 5$ という経路を通ることが分かる.

6 おわりに

この研究を通して，最長経路を求める際に代数的解法を用いることは，各頂点間の最長距離がどれだけかが行列計算で分かるという点では分かりやすいが，それがどの頂点からどの辺を通りたどる経路なのかは分からないという欠点があることが分かった．その欠点を解消する為にプログラミングにおいては 2 次元配列で計算過程を記録しておく方法を用いた．本研究を通して私は実際にコンピュータに最長経路を計算をさせるならば，C 言語プログラムを組みにくいということや計算量がかなり多くなることから，代数的解法を用いるよりも他のアルゴリズムを用いる方が良いということが分かった．ただ，この代数的解法は行列とグラフ理論とのつながりが明確に現れていると思われるので教育的観点からみれば意味のあることだと思われる.

参考文献

[1] 小野寺力男, グラフ理論の展開と応用, 森北出版,1973.

[2] 松井泰子, 根本俊男, 宇野毅明, 入門オペレーションズ・リサーチ, 東海大学出版会,2008.

[3] 福島雅夫, 新版数理計画入門, 朝倉書店,2011.