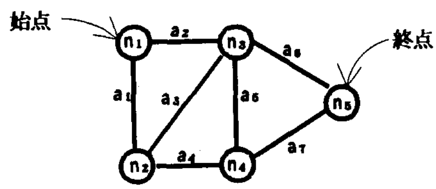
Ⅳ－378　無向グラフを対象とした経路探索手法に関する理論的研究

立命館大学　正員　春名　攻
立命館大学大学院　学生員　○山田　幸一郎
立命館大学大学院　学生員　滑川　達

１．はじめに

本研究では、グラフ理論で取り扱われている接続行列に着目し、無向グラフを対象として、全ての有効可能性のあるルート（基底ルート）を選定するルート探索方法の理論的開発をおこなった。基底ルートとは、グラフ上のＳ～Ｔ間（Sourse to Sink）をサイクルしないルートのことであり、言い替えれば最短路、流量等の問題で検討される価値を秘めているルートのことである。

以後の説明を捉え易くするために図-１に例を示す。



	a_1	a_2	a_3	a_4	a_5	a_6	a_7	N_j
n_1	1	1						
n_2	1		1	1				
n_3		1	1		1	1		
n_4				1	1		1	
n_5						1	1	
A_i								←カウント行列

図-１　対象グラフとその接続行列

２．無向グラフ上での基底ルートの潜在性の検討

対象とするグラフが無向グラフだとしても、取り扱う任意の基底ルートが始点から終点へ向かう方向性を持つルートとなるために、必然的にそのルートを構成している各々のアークは方向を持っていると考えることができる。本研究では、無向グラフから基底ルートの抽出をおこなうために、有向・無向グラフの性質を把握した結果、以下に述べるように、無向グラフに有向グラフの性質を導入する事により、無向グラフの接続行列のみから方向性を持つルートの抽出を可能とした。

特にグラフの性質、すなわち、

「ある流れがアークa_iからあるノードn_jに入り込むと、そのノードn_jからはアークa_i以外の他のアークa_kに流れる。」

ことに着目した。これをグラフ理論で扱っている有向グラフの接続行列（Ｉ）で表わすと、

「アークa_iからノードn_jに入り込む時は、$I(n_j, a_i) = -1$であり、ノードn_jから他のアークa_kに流れる時には、$I(n_j, a_k) = +1$である。」

となる。これを無向グラフの接続行列で示すと、

「アークa_iからノードn_jに入り込む時は、$I(n_j, a_i) = 1$であり、ノードn_jから他のアークa_kに流れる時には、$I(n_j, a_k) = 1$である。」

となる。しかし本研究では、ある流れがアークa_iからノードn_jに流れ込むとき、無向グラフの接続行列であっても仮に、

「アークa_iからノードn_jに入り込む時は、$I(n_j, a_i) = -1$であり、ノードn_jから他のアークa_kに流れる時には、$I(n_j, a_k) = +1$である。」

と考えることができることに着目した。これより無向グラフ上のルートを有向グラフ上のルートのように扱うことができる（図-２参照）。この性質を活かして無向グラフに有向性を持たせ、ルートを合理的かつ効率的に抽出することが可能となってくるのである。それに、任意の全てのルートは、始点から終点に流れる基底ルート、またはサイクルを発生させる非基底ルート以外には存在しないために、ルートは、始点からこの性質を用いることにより最終的にはサイクルをおこなうかあるいは終点にたどり着くまで流れるのである。

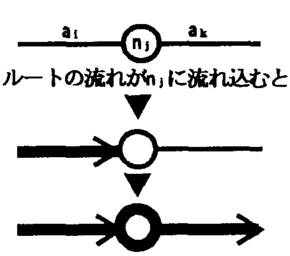


図-２　無向グラフ上にルートが通過する際の図

3．カウント行列設定によるサイクルルート排除

サイクルを「する・しない」を判断させるためには、前述した方向を持ったルートが前進していき、サイクルする時点で何らかのサイクルをおこなうということの判断基準を設け、サイクルルートを排除させる必要があると考えた。本研究では、ルートの通過地点を把握できるように、グラフ上のルートがそれぞれのアーク、ノードに通過する毎にカウントを重ねていくカウント行列を設け、ルートが同地点を2回以上通過するものをサイクルルートとし、これよりサイクルルートの排除を可能とした。

4．基底ルート抽出方法の定式化

研究の結果、全てのルートは、サイクルするもしくは終点にたどり着くまで無向グラフの接続行列（0，1表示）上を方向性を持ちながら、接続行列I上を繰り返し縦→横→縦→横というように$I(n_i, a_j)$は$I(n_i', a_j)$に流れ、そして、$I(n_i', a_j')$へ流れることを繰り返すという性質を有していることを発見した。これより、問題を探索問題に展開でき、前述したサイクルルートの排除と、ルートの流れの性質の利用より、全ての基底ルートを接続行列のみから抽出することを可能とした。

そして、$I^{(1)}(n_i, a_j)$を接続行列の値が1である(n_i, a_j)地点とすると、基底ルート抽出探索の定式は、

始点である$I^{(1)}(s, a_s)$から

$$I^{(1)}(n_i, a_j) \rightarrow I^{(1)}(n_i', a_j)$$
$$\rightarrow I^{(1)}(n_i', a_j')$$

但し、$(n_i, a_j) \neq (n_i', a_j) \neq (n_i', a_j')$を、終点である$I^{(1)}(g, a_g)$に到達する、もしくはサイクルするまで繰り返す。

また、探索していく過程でサイクルルートを排除するために、制約条件を、

$$A_i \leq 1 \quad \cap \quad N_j \leq 1$$

A_i：アーク側のカウント行列
N_j：ノード側のカウント行列
$(i=1, 2, \cdots, l)$　l：アーク数
$(j=1, 2, \cdots, m)$　m：ノード数

とする。

そして、基底ルート$R_k (k=1, 2, ...) = [I^{(1)}(s, a_s), \cdots, I^{(1)}(n_i, a_j), I^{(1)}(n_i', a_j), I^{(1)}(n_i', a_j'), \cdots, I^{(1)}(g, a_g)]$が得られる。ここでs，gは始点、終点のノード、$a_s$，$a_g$はs，gのノードと隣接するアークとする。

サイクルルートは、探索していく過程で

$$A_i = 2 \quad \cup \quad N_j = 2$$

となるまでのルートであり、さらに、

$$A_i = 2 \quad \cap \quad N_j = 2$$

となるルート等の複種類のルートが存在してくる。

この方法を用いて例題のグラフを探索させた結果、図-3（一部省略）の探索をおこなった。

また、計算量を軽減するために、ルートを探索させていく過程で動的計画法を適用することで効率的に基底ルートの抽出をおこなえた。

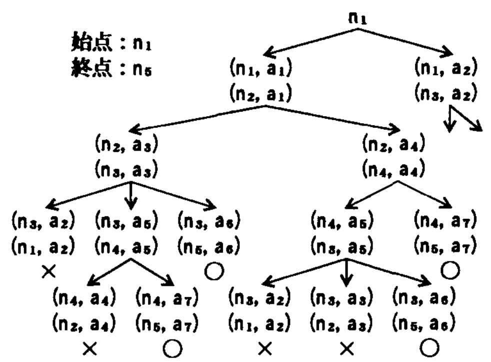


図-3　探索結果のtree図（○:基底ルート，×:非基底ルート）

5．おわりに

本研究では、無向グラフを対象とした全ての基底ルートを、グラフの性質を合理的に導入し、効率的に接続行列のみから抽出することにより可能とした。ここでの抽出方法、抽出されたルートは、様々な分野で取り扱われているネットワーク問題の有効な要因となってくるものと考える。例えば、最も単純なものとして最短経路問題があげられるが、この問題においては、アークに距離、時間等の要素を加え、ルート探索の過程において始点からの距離等の和を考慮することで最短路を効率的に抽出することが可能となる。そして、今後は様々に考えられるより実用的な問題に対しての適用・応用方法を明らかにしていくとともに、より有効な手法を開発していくこととする。