アルゴリズム論(第11回)

http://coconut.sys.eng.shizuoka.ac.jp/ds/07/

安藤和敏 (静岡大学工学部)

2006.06.25

2.1. 木と道

2.1.1. 最短路問題 (ベルマン-フォード法)

図 2.1 (教科書の図 2.8 と同じ) のように, 負の長さを持つ枝があるネットワークに対しては, ダイクストラ法は必ずしも最短路を見出さない. 本日の講義では, 長さが負であるような枝があるネットワークにおいても, 最短路を見出すアルゴリズム (ベルマン-フォード法) について紹介する. 負の長さの枝の存在は非現実的に思えるかも知れないが, もっと

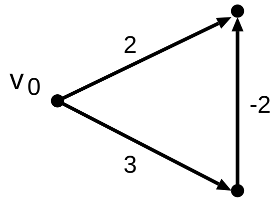

図 2.1: 負の長さの枝があるネットワーク

複雑な問題を解くためにそのようなネットワークを扱う必要がある.

図 2.2 のように, ネットワークに負の長さの有向閉路が存在する場合, 最短路問題は解を持たない. では, 負の長さの有向閉路の存在は許すけれども, 問題を最短路として初等

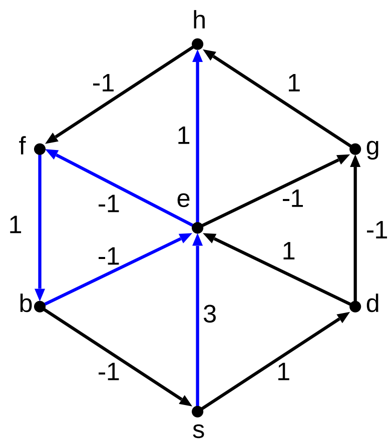

図 2.2: 負の長さの有向閉路が存在するネットワーク

的な有向道 (⇒ p. 13) の中で最短なものを見付けることとしたらどうだろう. しかし, それば解くのが非常に困難な問題になる.

ここでは, 問題を

「負の長さの閉路を見付けるか, または, 始点からそれ以外の全ての点への最短路を見付ける」

と設定しよう．この問題を解くアルゴリズムとして, ベルマン-フォード法 (アルゴリズム 1) が知られている.

アルゴリズム 1 ベルマン-フォード法 (始点を v_0 とする)

入力: ネットワーク $\mathcal{N} = (G = (V, A), l)$.

出力: もしあれば負の長さの有向閉路を, そうでなければ v_0 からその他の各点 v への最短路, 及び, 最短路長.

1: $p(v_0) \leftarrow 0,\ p(u) \leftarrow +\infty\ (u \in V \setminus \{v_0\}),\ k \leftarrow 1.$

2: 各枝 $(v, w) \in A$ に対して,

 (*) $p(w) > p(v) + l(v, w)$ ならば

  $p(w) \leftarrow p(v) + l(v, w),\ q(w) \leftarrow v.$

3: (i) Step 2 で p の更新 (*) が全くされなければ停止する.

 (ii) p が更新されたとき,

  (a) $k < n(= |V|)$ ならば $k \leftarrow k + 1$ として Step 2 へ戻り,

  (b) $k = n$ ならば停止する (このとき負の長さの有向閉路が存在する).

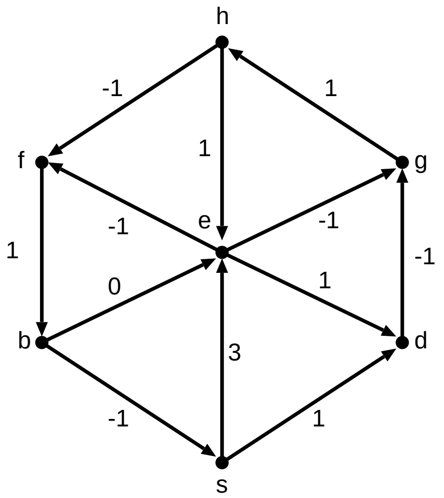

図 2.3: ネットワーク $\mathcal{N} = (G = (V, A), l)$

ベルマン-フォード法を, 始点を $v_0 = s$ として図 2.3 のグラフに対して実行した結果は表 2.1 のようになる．さらに, 実行結果を図で表現すると, 図 2.4 のようになる.

Step 2 における枝の選択の順番は任意であるが, 次のような順序で枝を調べてみよう.

s から出る枝, b から出る枝, d から出る枝, e から出る枝, f から出る枝, g から出る枝, h から出る枝.

この順序で枝を調べた場合, ベルマン-フォード法の実行の経過は表 2.1 のようになる.

Step 3 (i) でベルマン-フォード法が終了したとき, 全ての枝 $(v, w) \in A$ に対して $p(v) + l(v, w) \geq p(w)$ が成り立つ．即ち,

$$l_p(v, w) = l(v, w) + p(v) - p(w) \geq 0. \tag{2.1}$$

表 2.1: ベルマン-フォード法の動き (各自で記入せよ).

		s	b	d	e	f	g	h
	p	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
	q	—	—	—	—	—	—	—
$k=1$	p							
	q							
$k=2$	p							
	q							
$k=3$	p							
	q							
$k=4$	p							
	q							
$k=5$	p							
	q							
$k=6$	p							
	q							
$k=7$	p							
	q							

さらに, $(q(u), u)$ $(u \in V \setminus \{v_0\})$ は v_0 を根とする有向木を成し, この有向木上の枝 (v, w) に対しては,

$$l_p(v, w) = l(v, w) + p(v) - p(w) = 0$$

が成り立つ. (w のポテンシャルが最後に変更された時点での v のポテンシャルを $p'(v)$ とすると, $p(w) = p'(v) + l(v, w)$ が成り立っていたはずである. その後のアルゴリズムの実行によっても, ポテンシャルは増加しないので, $p'(v) \geq p(v)$ が成り立つ. もし $p'(v) > p(v)$ であるならば, $p(w) > p(v) + l(v, w)$ であるから, (2.1) と矛盾する. ゆえに, $p(w) = p'(v) + l(v, w) = p(v) + l(v, w)$ である.) 補題 2.2(⇒ p. 43) から, v_0 からこの有向木上の道が最短路である.

補題 2.4: ネットワーク $\mathcal{N} = (G = (V, A), l)$ に対して, 適当なポテンシャル $p\colon V \to \mathbb{R}$ が存在して, $l_p(a) \geq 0$ $(a \in A)$ とできるための必要十分条件は, $\mathcal{N}$ に負の長さの閉路が存在しないことである.

(証明) もし, すべての枝 $a \in A$ に対して $l_p(a) \geq 0$ となるようなポテンシャル $p\colon V \to \mathbb{R}$ が存在するのならば, 任意の有向閉路

$$C = (u = v_0, a_1, v_1, a_2, v_2, a_3, \cdots, a_k, v_k = v)$$

に対して, l_p に関する C の長さ $l_p(C)$ は非負である. さらに, 補題 2.1 によれば

$$l_p(C) = l(C) + p(u) - p(v) = l(P)$$

であるから, l に関する C の長さ $l(C)$ も非負である. つまり, 長さが負であるような閉路は存在しない.

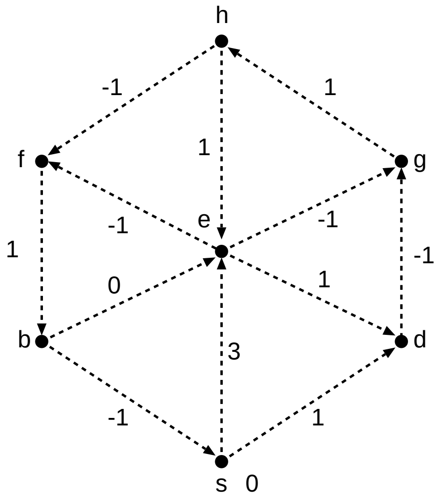

図 2.4: p と q の図的表現 (各自で記入せよ).

逆に, 負の長さの閉路が存在しないのであれば, ベルマン-フォード法はStep 3(i) で停止する. そのとき, すべての枝 $(v, w) \in A$ に対して

$$p(w) \leq p(v) + l(v, w)$$

である. すなわち, $l_p(v, w) \geq 0$ がすべての枝 $a \in A$ に対して成り立つ. □