

グラフとネットワーク (第11回)

<http://coconut.sys.eng.shizuoka.ac.jp/gn/08/>

安藤和敏 (静岡大学工学部)

2008.12.11

2.1. 木と道

2.2. ベルマン-フォード法

負の長さの枝があるネットワーク上の最短路問題は、ダイクストラ法によって解くことはできない。

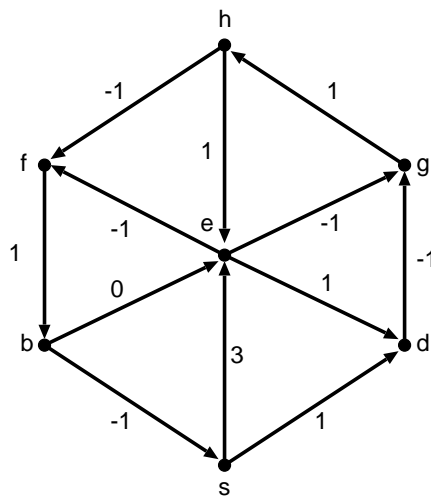


図 2.1: ネットワーク $\mathcal{N} = (G = (V, A), l)$

負の長さの枝があるネットワーク上の最短路問題は、ベルマン-フォード法 (アルゴリズム 1) によって解くことができる。(もちろん、全ての枝の長さが非負 (0 以上) であるようなネットワーク上の最短路問題も、ベルマン-フォード法によって解くことができる.)

ベルマン-フォード法を、始点を $v_0 = s$ として図 2.1 のグラフに対して実行した結果は表 2.2 のようになる。本当は Step 2 における枝の選択の順番は任意であるが、ここでは Step 2 で枝を調べる順序は

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

としよう。

アルゴリズムが終了した時点での p と q を図で表現すると、図 2.2 のようになる。

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).
-

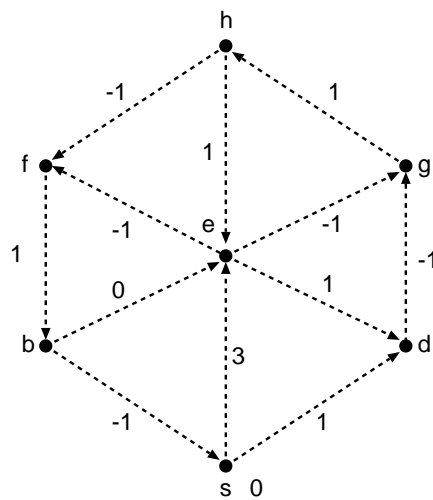


図 2.2: アルゴリズムの終了時の p と q の図的表現 (各自で記入せよ).

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,

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

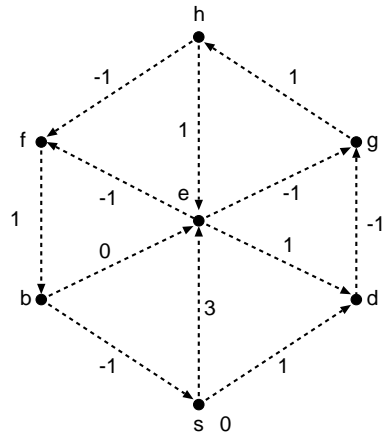
さらに, $(q(u), u) (u \in V \setminus \{v_0\})$ は v_0 を根とする有向木を成し, この有向木上の枝 (v, w) に対しては,

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

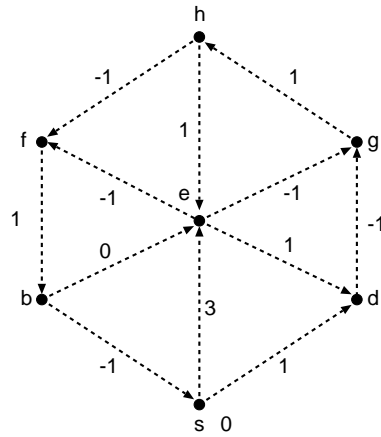
補題 2.2(\Rightarrow p. 43) から, v_0 からこの有向木上の道が最短路である.

アルゴリズムが終了時の有向木を見ると, 枝を調べる順番を別なものとしていたら, たった 2 回の繰り返しでアルゴリズムを終了することができたということが分かる. (どうい
う順番か分かるか?) しかし, この順番はアルゴリズムが終ってみないと分からない.

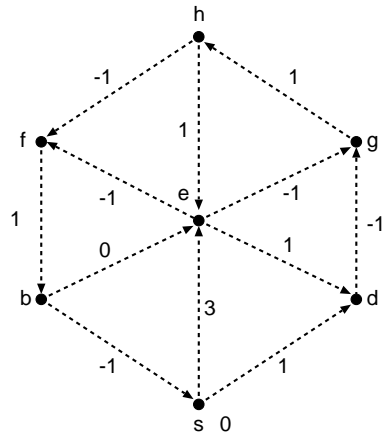
表 2.1: ベルマン-フォード法の動き (各自で記入せよ).



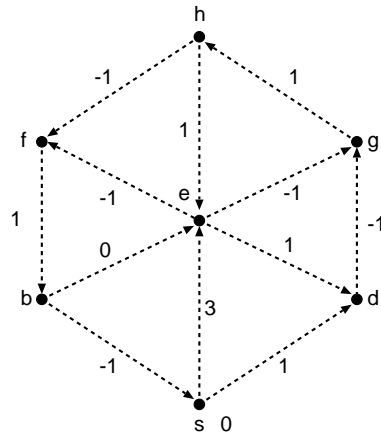
1回目の Step 2 終了時



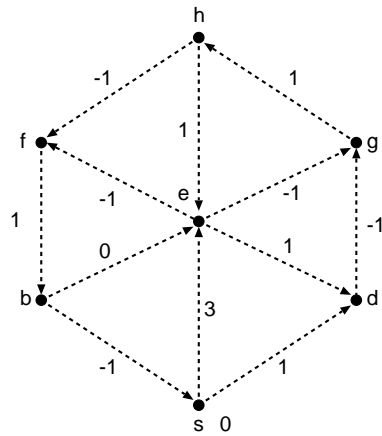
2回目の Step 2 終了時



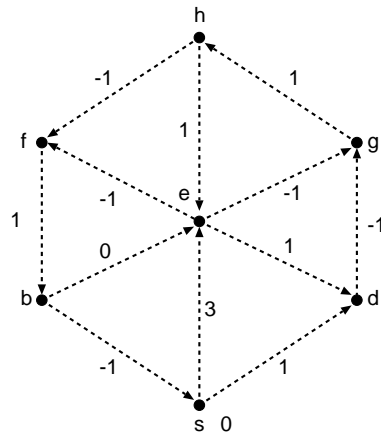
3回目の Step 2 終了時



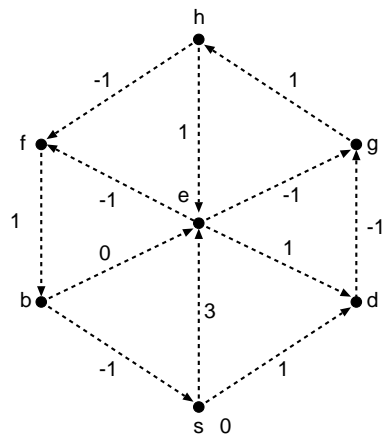
4回目の Step 2 終了時



5回目の Step 2 終了時



6回目の Step 2 終了時



7回目の Step 2 終了時