

負コスト枝を含む有向グラフにおける最短単純路問題*

01700900 防衛大学校情報工学科 山田 武夫† YAMADA Takeo

01107880 防衛大学校情報工学科 片岡 靖詞 KATAOKA Seiji

1 はじめに

最短路問題はネットワーク計画法の中でも基本的な問題のひとつであり、ダイクストラ法 [1] を始めとする効率的な解法が開発されているが、枝コストに負のものが含まれる場合には、問題は格段に難しくなる。実際に、NP-困難な問題としてよく知られている最長単純路問題 [2] は、すべての枝コストの符号を反転して考えれば最短単純路問題に他ならない。

問題は次のように設定される。V, Eをそれぞれ節点、枝の集合とする有向グラフ $G = (V, E)$ で、各枝 $e \in E$ にコスト $c_e \in Z$ が付与されているものを考える。Gには始点 s と終点 t が指定されていて、問題は $s-t$ 間のコスト最小の単純パスを求めることである。

本稿では、上の問題に対して動的計画法と分枝限定法による解法をそれぞれ提示する。

2 動的計画法

$d(S, i)$ ($S \subseteq V, s \in S, i \notin S$) を S の点を全て通り、 i に至る $s-i$ 単純路のコストの最小値とする。このとき、

$$d(S, j) = \min_{i \in V_j^+ \cap S} \{d(S \setminus \{i\}, i) + c_{ij}\} \quad (1)$$

が成り立つ。ここに、 V_j^+ は節点 j の先行節点全体の集合である。 $d(\{s\}, s) = 0$ より上を順次計算し、最後に

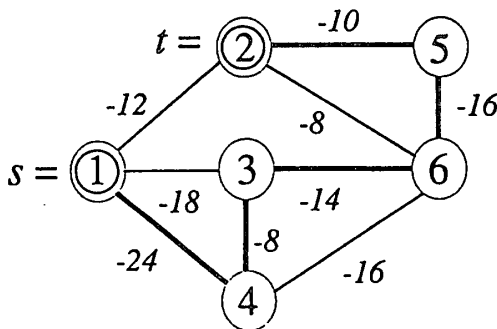


図1: 例題 (その1)

$$d(t) = \min_{S, i \notin S} \{d(S, i) + c_{it}\} \quad (2)$$

とすると、これが求める最適値を与え、同時に最短単純経路も求められる。図1の例題についての計算過程を図2に示す。図中には状態 (S, i) とそこでの値 $d(S, i)$ を、それらの間の関係とともに表示している。図2の太線に沿って、状態 $(\{s\}, s)$ より状態 (t) へ至る経路が最適解を表わしており、これは図1に太線で示したコスト-82の最短路に対応している。

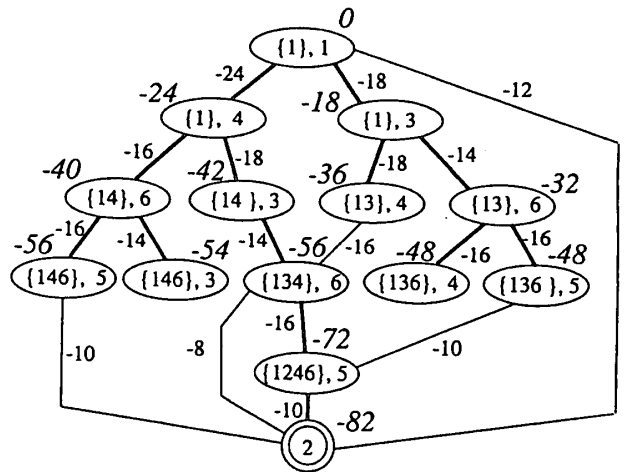


図2: 動的計画法

3 分枝限定法

3.1 部分問題

$F (\subseteq E)$ を s から出る単純パスとし、 $R (\subseteq E \setminus F)$ を F と素な枝集合とする。このとき、 F の枝をすべて含み、 R の枝を全く含まないような $s-t$ パスを (F, R) -許容なパスと呼ぶ。 (F, R) -許容な最短単純パスを求める問題を部分問題 $P(F, R)$ とすると、元の問題は、 $P(\emptyset, \emptyset)$ と表記できる。

3.2 分枝ルール

部分問題 $P(F, R)$ は、 (F, R) -許容なパス $\pi = (F, e_1, e_2, \dots, e_k)$ が得られれば、それをもとに以下の

*成蹊大学, 1999.9.20-21

†E-mail: yamada@cs.nda.ac.jp

ような子問題群に分割することが出来る。すなわち、 $i = 1, \dots, k$ に対して $F_i = F \cup \{e_1, e_2, \dots, e_{i-1}\}$, $R_i = R \cup \{e_i\}$ として k 個の部分問題 $P(F_i, R_i)$ を考えると、各子問題の実行可能領域は互いに素で、その和集合は $P(F, R)$ のそれに一致するので、各 $P(F_i, R_i)$ ($i = 1, \dots, k$) が最適に解ければ、 $P(F, R)$ も解けたことになる。しかしながら、一般に $P(F_i, R_i)$ を最適に解くことは難しいので、以下のように上下界値の情報を用いながら、分枝限定法に組込むことを考える。

3.3 上界値の求め方

上界値 $z^u(F, R)$ および実行可能なパス $\pi^u(F, R)$ は、固定部分 F の終点を始点としたダイクストラ法を用いて求めることができる。負コストの枝が存在するので、得られたパスは最短路とは限らないが、 (F, R) -許容なパスが存在する限りそのような単純パスが一つ得られ、これが上界値 $z^u(F, R)$ を与える。

そこで、得られたパスが現在の暫定値 z^* よりよければ、 $z^* := z^u(F, R)$, $\pi^* := \pi^u(F, R)$ に更新する。

3.4 下界値の求め方

下界値 $z^l(F, R)$ は、 (F, R) -許容パスが、高々 $|V| - |F|$ 本の枝で構成されていることを利用し、動的計画法により求める。すなわち、 $d_k(i)$ を固定部分 F の終点 f から節点 i への k ステップ以内での最短（必ずしも単純でない）経路長とすると、最適性の原理から次が成立する。

$$d_k(j) = \min_{i \in V_j^+} \{d_{k-1}(i) + c_{ij}\}, \forall j \in V, k = 1, 2, \dots \quad (3)$$

初期値は $d_0(i) = \infty$, ($i \neq f$), $d_0(f) = c(F)$ で、 $c(F)$ は F のコストを意味する。このとき、 $d_{|V|-|F|}(t)$ が一つの下界値 $z^l(F, R)$ を与え、この計算と同時にパス $\pi^l(F, R)$ も得られる。

この下界値は、最短路問題において単純パスという条件を緩和したものであるため、 $\pi^l(F, R)$ が単純パスであればこれが最適解となるが、一般にはある節点 $r \in V$ で自分自身に交差する。このような場合、節点 r に着目して下界値をさらに強化することができる。すなわち、

- (i) G から r を除いたグラフに上の方法を適用して得られる値。
- (ii) 同様に、点 f から r までの k ステップ以内のパスについての下界値と、点 r から t までの $|V| - |F| - k$ ステップ以内のパスの下界値の和を考え、 $k = 1, 2, \dots, |V| - |F| - 1$ での最小値。

を求め、これらのうち小さい方を下界値とする。これは、 (F, R) -許容パスを r を通るものとそうでないものに分けて評価するもので、 $z^l(F, R)$ より良い下界値を与える。

4 数値例

以上を ANSI-C 言語で記述し、HP-9000 BL132 上でいくつかの計算を行った。図 3 に示す最長路問題を動的計画法で解いたところ、807 個の状態が生成され、図中に示すパスを得た。また、分枝限定法では 291 個の子問題を生成して、同じ解が得られた。CPU 時間は前者で 0.07 秒、後者で 0.06 秒であった。同じグラフで各枝を双方向枝とした場合には、動的計画法は 227,838 状態、分枝限定法は 808,904 子問題を生成して最適解が得られたが、CPU 時間は前者で 362.8 秒、後者で 97.2 秒であった。

5 むすび

今後さらに計算実験を行ってアルゴリズムの評価と改善を図りたい。

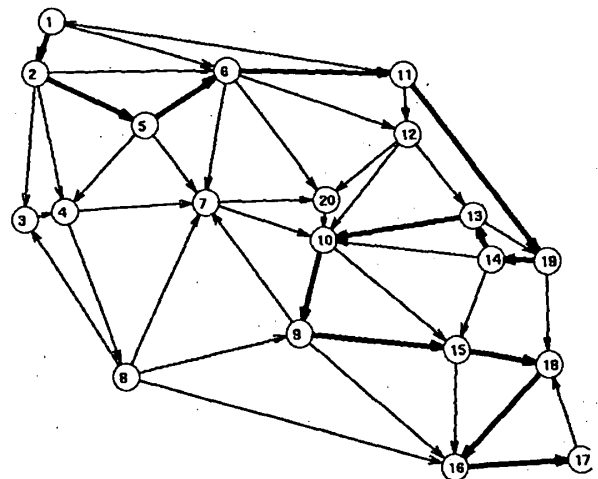


図 3: 例題 (その 2)

参考文献

- [1] R.K. Ahuja, et al.: *Network Flows*, Prentice-Hall, 1993.
- [2] M.R. Garey and D.S. Johnson, *Computers and Intractability*, Freeman, 1979.