1995年度日本オペレーションズ・リサーチ学会
秋季研究発表会

# 非巡回的有向グラフ上の s-t パスの列挙

01605630 東京都立大学 *松井 泰子 MATSUI Yasuko
01605000 東京大学 松井 知己 MATSUI Tomomi
02003590 東京工業大学 宇野 毅明 UNO Takeaki

## 1. はじめに

$G = (V, E)$ を頂点集合 $V$ と枝集合 $E$ からなる有向グラフとする。ここでは，$G$ は自己閉路を含まないグラフであるとする。$G$ が有向閉路を含まない時、$G$ を非巡回的グラフと呼ぶ。有向枝 $e = (v_1, v_2)$ において、枝 $e$ が出る頂点 $v_1$ を $e$ の尾と呼び、入る頂点 $v_2$ を $e$ の頭と呼ぶ。$G$ の頂点数を $n$ とし，枝数を $m$ と表す。

本稿では，非巡回的グラフ上の様々な条件下での s-t パスの列挙問題を 3 題定義し、各々に対する解法を提案する。すなわち最初に、(i) 非巡回的グラフ上の s-t パスの列挙解法を、次に (ii) 枝が正の長さを持つグラフ上の s-t 最短パスの列挙解法を述べる。そして、(iii)(i) を用いた、ナップサック問題の最適解の列挙解法を示す。これらの解法の理論的計算量と必要な記憶容量は各々 (i)$O(n+m+\alpha)$, $O(n+m)$, (ii)$O(S+n+m+\alpha), O(n+m)$, (iii)$O(kb+\beta), O(kb)$ である．ただし，$\alpha$ は出力するパスの総数，$S$ は最短路問題の解法に要する計算量，$k$ はナップサック問題において与えられた物の数，$b$ はナップサックの容量，$\beta$ はナップサック問題の最適解の総数を表す．

上記の解法のうち，(i),(iii) は理論的計算量及び記憶容量共に最適であり，(ii) は解 1 個当たりの出力にかかる時間が最適である。

## 2. 非巡回的グラフ上の s-t パスの列挙

非巡回的グラフ上の s-t パス列挙問題は以下のように定義される。

入力：非巡回的グラフ $G = (V, E)$, $s, t \in V$
出力：$G$ 中の全ての s-t パス

初めに、グラフ $G$ に対し前処理を行なう。頂点 $t$ へ到達不可能な頂点をすべて $G$ から削除し、得られたグラフを $G' = (V', E')$ とする。明らかに、$G'$ 上で s-t パスを列挙すれば十分である。$G$ から、ある頂点を削除する時は、削除する頂点に接続する枝もすべて削除するものとする。$G'$ は $G$ から $O(n+m)$ で得る事が出来る。ここで、$s \notin V'$ ならば処理を終了する。次に、$G'$ 中の各頂点において、その頂点を尾とする枝が一意であるかどうかを調べる。もし一意であれば、その枝の頭と尾の頂点を同一視するという操作を、各頂点において、その頂点を尾とする枝が一意である限り繰り返す。その際、自己閉路はグラフから削除する。上記の前処理によって得られたグラフを $G^* = (V^*, E^*)$ とする。$G^*$ のデータは、頂点の隣接リストで持つものとする。

前処理は、解法の理論的計算量の算定を行なう時に、非常に重要である。

**s-t パスの列挙解法 acyclic-enum**

入力：$G^* = (V^*, E^*)$, $s, t \in V^*$
出力：$G$ 中の全ての s-t パス
Step 1: $s = t$ ならば、処理を終了する。
Step 2: all-st-enum$(s, t, ())$ を呼ぶ。

all-st-enum$(s, t, P)$
Step 1: $s = t$ ならば、$P$ を出力し、処理を終了する。
Step 2: $s$ から出ている各枝 $(s, v)$ について、all-st-enum$(v, t, P + (s, v))$ を呼ぶ。

前処理より、$G^*$ 中の $t$ 以外の各頂点を尾とする枝は 2 本以上存在するので、解法が異常終了する事は無い。また、$G$ が非巡回的である事から、解法の有限性は明らか。acyclic-enum で得られる $G^*$ の s-t パスが、$G$ の s-t パスに一対一対応する事は明らか。

以上の acyclic-enum から、非巡回的グラフ上の s-t パスの列挙は $O(n+m+K)$ で実行出来る。ただし、$K$ は、出力されるパスに含まれる辺の本数の総和を表す。ここで、列挙する解の出力に対し、コンパクト出力という概念 [1] を導入すると、理論的計算量は、$O(n+m+\alpha)$ となる。コンパクト出力とは、2 個目以降の解を、直前に出力した解との対称差を出力して、解とみなす手法である。acyclic-enum の変更は、解を 1 個出力した後に、除く辺の集合と加わる辺の集合を別々に待ち、$P$ の代わりに、この 2 つの辺集合の対称差の辺を出力すれば良い。

## 3. 枝が正の長さのグラフ上の $s-t$ 最短パスの列挙



© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

問題の定義は以下の通りである。

入力：有向グラフ $G=(V,E)$, $s,t\in V$, $l:E\rightarrow Z_{++}$
出力：$G$ 上の全ての $s-t$ 最短パス

提案する列挙解法は、コンパクト出力を行なう事により、解1個当たりの出力にかかる理論的計算量は最適である。

提案する解法では、初めに $G$ 中の頂点 $s$ から他の全ての頂点への最短距離を Dijkstra 法を用いて求める。次に、$G$ 中から、$s$-$t$ 最短パスに使われる事の無い辺を除去したグラフ $G'=(V',E')$ を求める。この時、枝の長さは正であることから $G'$ は非巡回的となる。最後に、$G'$ 上での $s$-$t$ パスを acyclic-enum で列挙する。ここで、$G$ の $s$-$t$ 最短パスは、$G'$ 上での $s$-$t$ パスと一対一対応している。

$s$-$t$ 最短パスの列挙解法 direct-enum

入力： $G=(V,E)$, $s,t\in V$, $l:E\rightarrow Z_{++}$
出力： $G$ 中の全ての $s$-$t$ 最短パス
Step 1: $V$ 中の $s$ 以外の全ての頂点に関して、頂点 $v$ への最短距離を求め、$d(v)$ に格納する。
Step 2: $E':=\{(i,j)\in E|l(i,j)=d(j)-d(i)\}$ とし、$G'=(V,E')$ を構築する。
Step 3: $G'=(V,E')$, $s,t\in V$ を入力として、acyclic − enum を呼ぶ。

解法の有限性は、$G'$ が非巡回的グラフである事から明らか。解法の正当性を示す前に定義を行なう。

direct-enum の Step 2 の操作で得られたグラフ $G'=(V,E')$ を最短パスグラフと呼ぶ。最短パスグラフは、次の性質を満たす。

[定理 1] (1)$G'$ 中の任意の $s$-$t$ パスは、元のグラフ $G$ での $s$-$t$ 最短パスである。(2) 元のグラフ $G$ での $s$-$t$ 最短パスは、$G'$ 中の $s$-$t$ パスである。(3)$G$ は非巡回的である。 ∎

定理1より、解法の正当性は明らかである。

理論的計算量と必要記憶容量は、$O(S+n+m+\alpha), O(n+m)$ である。

## 4. ナップサック問題の最適解の列挙解法

ナップサック問題は以下の様に定義される。

$$
\begin{array}{lll}
 & \texttt{maximize} & w^Tx, \\
(\mathrm{KP}) & \texttt{subject to} & a^Tx\leq b, \\
 & & x\in\{0,1\}^k, \\
 & & w\in Z^k_{++}, a\in Z^k_{++}.
\end{array}
$$

問題 **KP** を最短パス問題として解くために、以下の様な補助グラフを構築する。

補助グラフを $G'=(V',E')$ とする。ただし、

$$
\begin{aligned}
V' &= \{t\}\cup\{v_{ij}|\, i=0,\ldots,k,\, j=0,\ldots,b\}, \\
E' &= E_1\cup E_2\cup E_3.
\end{aligned}
$$

ここで、

$$
\begin{aligned}
E_1 &= \{(v_{xy},v_{x'y'})\in V\times V| \\
&\qquad x+1=x', y+ax=y'\leq b\}, \\
E_2 &= \{(v_{xy},v_{x'y'})\in V\times V|\, x+1=x', y=y'\}, \\
E_3 &= \{(v_{kj},t)\in V\times t\,|\, j=0,\ldots,b\},
\end{aligned}
$$

である。次に、$G'$ 中の各枝 $e\in E'$ に対し長さを定義する。関数 $l:E'\rightarrow Z_{++}$ は、各枝に対し、

$$
l(e)=\begin{cases} -w_i+M & e\in E_1, \\ M & e\in E_2, \\ M & e\in E_3, \end{cases}
$$

という長さを与える。ただし、$M$ は十分大きな正の数 (e.g., $1+\max_i w_i$) とする。問題 **KP** の最適解の列挙は、枝が正の長さを持つ非巡回的グラフ $G'$ の s-t 最短パスの列挙と等価である。ゆえに以下のような解法を構築することができる。

ナップサック問題の最適解の列挙解法 knap-enum

Step 1: 入力データから補助グラフ $G'$ を構築し、さらに $G'$ から最短パスグラフ $G^*=(V^*,E^*)$ を構築する。
Step 2: $G^*=(V^*,E^*)$, $s(=v_{00}), t\in V', P(=())$ を入力として、acyclic-enum を実行する。

上記の解法の理論的計算量と必要記憶容量は、$O(kb+\beta), O(kb)$ である.

## 参考文献


[1] S.Kapoor and H.Ramesh "Algorithms for generating all spanning trees of undirected, directed and weighted graphs," in Lecture Notes in Computer Science. (Dehne, F., Sack, J-R. and Santoro, N., eds.), Springer-Verlag (1992) 461-472

[2] R.C.Read and R.Tarjan, "Bounds on Backtrack Algorithms for Listing Cycles,Paths and Spanning Trees," *Networks*, 5 (1975) 237-252.




© 日本オペレーションズ・リサーチ学会. 無断複写・複製・転載を禁ず.