

グラフ圧縮による媒介中心性の計算手法

著者	荒巻 和治, キットスワン ナッタポン, 大木 英司
雑誌名	電子情報通信学会論文誌 B
巻	J100-B
号	11
ページ	934-939
発行年	2017-11-01
URL	http://id.nii.ac.jp/1438/00008818/

doi: 10.14923/transcomj.2017JBL4011

研究速報

グラフ圧縮による媒介中心性の計算手法

荒巻 和治[†]キットスワン ナッタポン^{†a)} (正員)大木 英司^{††} (正員:フェロー)

Computation Method for Betweenness Using Graph Compression

Kazuharu ARAMAKI[†], Nonmember,Nattapong KITSUWAN^{†a)}, Member, and Eiji OKI^{††}, Fellow[†] 電気通信大学大学院情報理工学研究所情報・通信工学専攻, 調布市

Department of Communication Engineering and Informatics, Graduate School of Informatics and Engineering, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-shi, 182-8585 Japan

^{††} 京都大学大学院情報学研究所通信情報システム専攻, 京都市

Communication and Computer Engineering, Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto-shi, 606-8501 Japan

a) E-mail: kitsuwan@uec.ac.jp

DOI:10.14923/transcomj.2017JBL4011

あらまし 本論文はグラフの各点の媒介中心性を求める計算手法を提案する。それは次数が1である点をグラフから除き、圧縮されたグラフで計算する。提案手法が、従来の手法の次数が1である点が存在するグラフで生じる冗長な計算を回避し、計算量を削減することを示す。

キーワード 媒介中心性, グラフ解析, グラフ圧縮, 通信ネットワーク

1. ま え が き

現実世界には、インターネット、道路網、電力網、ソーシャルコミュニティなど、多数のネットワークが存在する。それらの構造や特性を把握し、通信をコントロールすることは常に求められている。Freeman [1] が定義した媒介中心性 (betweenness centrality) は、現実のネットワークグラフを解析するとき用いられる基本的な指標の一つとされている。

与えられた静的なグラフから各頂点の媒介中心性の厳密解を求める手法として、Brandes [2] の手法がある。この手法は、グラフ内の各点を始点とする最短経路の集合を求めて、媒介中心性を計算する。これは、一つのグラフに対して、複数回の最短経路の探索をしているため、経路木が同一であった場合に、冗長な計算をしている。これに対して、Wangらは、グラフを関節点で分割し、分割された各部分グラフで最短経路探索を行い、媒介中心性を計算する手法を導入した [3]。Wangの手法は、最短経路の探索領域を小さくすることで、Brandesの手法の冗長な計算部分を削減する。

この削減効果は発見された関節点の位置と数に依存する。グラフ内のある点の次数が1であるとき、その点の接続先の点は関節点であるので、次数が1である点とそれ以外の点にグラフが分割される。次数が1である点の媒介中心性は一意に0であるので、次数が1である点が含まれた部分グラフの経路を探索し、媒介中心性を計算することは冗長である。

一方、媒介中心性の近似解を求める手法 [4] や、辺の挿入が起きる動的なグラフにおいて、媒介中心性の厳密解を求める手法 [5] が開発されている。文献 [4] では、グラフの幾つかの点をサンプリングして得られる最短経路の集合から近似解を求める。文献 [5] では、各点の媒介中心性が既知であるグラフにおいて、ある辺の挿入が起きたとき、その挿入部分を含めた最短経路の集合を得ることで、全ての点の媒介中心性の再計算を行わずに求める。本論文では、静的なグラフにおいて、近似計算を行わずに、高速に媒介中心性の厳密解を求めることに着目する。グラフの解析に、厳密解と近似解のどちらを採用するかは、近似解と厳密解の計算時間と近似解の精度から、用途に応じて判断される。

本論文は、グラフ圧縮を用いた媒介中心性の計算手法を提案する。提案手法は、与えられたグラフから次数が1である点を除くことでグラフを圧縮し、圧縮されたグラフで媒介中心性を求める。提案手法が、次数が1である点が存在するグラフで生じる従来手法 [2] の冗長な計算を回避し、計算量を削減することを計算時間の漸近的上界から示す。計算機実験により、提案手法は従来手法より計算時間を削減することを示す。

Wangの手法 [3] では、関節点でグラフを分割し、計算を行うことで、従来手法の冗長な計算を回避するが、次数が1である点が含まれた部分グラフの経路を探索するという冗長性がある。これに対して、提案手法は、グラフの圧縮により従来手法の冗長な計算を回避する。提案手法は、Wangの手法における次数が1である点が含まれた部分グラフの経路を探索するという冗長な計算を回避する。

2. 準 備

辺に重みのない連結な無向グラフを $G = (V, E)$ で表す。 V はサイズが n の点集合であり、 E はサイズが m の辺集合である。任意の2点 $i, j \in V$ について、 (i, j) は点 i と点 j を結ぶ辺を表す。点 $i \in V$ から幾つかの点と辺を経由して点 $j \in V$ まで到達するとき、経路した辺の本数を $l_i(j)$ で表す。 $d_i(j)$ は考えられる

$l_i(j)$ で最も小さい値を表す. $d_i(j)$ は点 i から点 j までの最短経路の距離である. σ_{ij} は点 i から点 j までの最短経路の本数を表す. $\sigma_{ij}(v)$ は式 (1) で定義する.

$$\sigma_{ij}(v) = \sigma_{iv} \cdot \sigma_{vj} \quad (1)$$

$\sigma_{ij}(v)$ は σ_{ij} のうち点 v を経由する経路の本数である. 媒介中心性 $C_B(v)$ は Freeman [1] によって以下の式で定義されている.

$$C_B(v) = \sum_{i,j \in V \setminus \{v\}; i \neq j} \frac{\sigma_{ij}(v)}{\sigma_{ij}} \quad (2)$$

$C_B(v)$ はグラフ内の各 2 点間の最短経路の本数に対する, 点 v を含むグラフ内の各 2 点間の最短経路の本数の割合の総和である.

3. 従来手法

従来手法である Brandes の手法 [2] について, 概要, 計算時間, 及び, 問題点について説明する.

3.1 従来手法の概要とアルゴリズム

従来手法の概要について説明する. 点 $v \in V$ について, v と点 v でない異なる 2 点 $i, j \in V$ の $\delta_{ij}(v)$ は式 (3) で定義される.

$$\delta_{ij}(v) = \frac{\sigma_{ij}(v)}{\sigma_{ij}} \quad (3)$$

$\delta_{ij}(v)$ は σ_{ij} に対する, $\sigma_{ij}(v)$ の割合である. 式 (2), (3) から $C_B(v)$ は式 (4) で表される.

$$C_B(v) = \sum_{i,j \in V \setminus \{v\}; i \neq j} \delta_{i,j}(v) \quad (4)$$

$C_B(v)$ を求めるには, 点 v と点 v でない異なる各 2 点 $i, j \in V$ に対して, σ_{ij} と $\sigma_{ij}(v)$ を求める必要がある. ある異なる 2 点 $i, j \in V$ について, $P_i(j)$ は点 i と点 j 間の最短経路上で点 i から点 j に到達する一つ手前の点の集合を表す. 従来手法は, 各点 $s \in V$ を始点とする幅優先探索をして点 s を端点とする最短経路の集合を得る. 得られた最短経路から各点 v に対して $\delta_{sv}(P_s(v))$ を求めることで, G 内の点 i でない各 2 点 t, v の $\delta_{st}(v)$ をまとめて計算する. 式 (4) より, $\delta_{st}(v)$ の総和をとり媒介中心性を求める.

従来手法のアルゴリズムの擬似コードを付録の Algorithm 1 に示す.

3.2 従来手法の計算時間

1 回の幅優先探索の走査には $O(m)$ かかる. n 個の点をそれぞれ始点として幅優先探索を走査させるので,

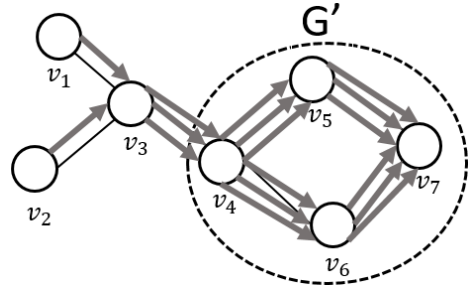


図 1 次数が 1 の点が存在するグラフの例 (最短経路の探索の様子)

Fig. 1 Conventional method with graph one-degree vertices.

従来手法の計算時間は, 与えられた $G = (V, E)$ に対して $O(nm)$ である.

3.3 従来手法の問題点

グラフに次数が 1 である点が存在するとき, 従来手法は冗長な計算をしている. 図 1 で示されるグラフは, 次数が 1 である 2 点 v_1, v_2 が同一の点 v_3 につながり, 点 v_3 がある部分グラフ G' に繋がっている. このグラフ内の各点の媒介中心性を求めるとき, 従来手法はグラフ上の各点を始点とした最短経路を探索する. 点 v_1, v_2, v_3 を始点とする最短経路の集合は部分グラフ G' 内において同一のものを得るので, 3 点 v_1, v_2, v_3 が部分グラフ G' 内の各点に与える媒介中心性の加算は同じになる. よって, 従来手法は次数が 1 である点が存在するグラフで冗長な計算をしている.

4. 提案手法

提案手法は Step 1 と Step 2 から構成される. Step 1 で与えられたグラフ $G = (V, E)$ を圧縮して $G' = (V', E')$ を生成する. Step 2 で G' 内で最短経路を探索し, 媒介中心性を計算する.

4.1 Step 1

与えられたグラフ G から次数が 1 である点 $v \in V$ を除き G' を生成する. このとき除かれる点 v の $C_B(v)$ は一意に 0 である. それは点 v がグラフ内の点 v を除いた任意の 2 点間の最短経路上に存在しないからである. $D(v)$ は点 v の次数を表す. 以下の手順に従って, 与えられたグラフ G からサイズ n の配列 A と次数が 1 である点を除いた $G' = (V', E')$ を得る.

- (1) 各点 $v \in V$ について, $A(v) = 0$ とする.
- (2) 各点 $v \in V$ に対して, $D(v) = 1$ ならば以下の処理を行う.
 - (a) $(v, w) \in E$ である点 w について, $A(w)$ を 1

増加させる。

(b) $C_B(v)$ を 0 とする。

(c) V から点 v 除き, E から (v, w) を除く。

(3) $G = (V, E)$ を $G' = (V', E')$ とする。

$A(v)$ は点 $v \in V$ に繋がっている次数が 1 である点の個数を表す。

4.2 Step 2

Step 2 は計算式の導出と計算手順から構成される。

4.2.1 で, G' と配列 A から媒介中心性を得られる計算式 (12), (18) を導出する。4.2.2 で, Step 1 から得た G' と A から, 導出した計算式 (12), (18) を用いて媒介中心性を計算する。

4.2.1 計算式の導出

G' と配列 A から媒介中心性を得られる計算式 (12), (18) を導出する。異なる 2 点 $i, v \in V$ に対して, $\delta_i(v)$ を式 (5) で定義する。

$$\delta_i(v) = \sum_{j \in V \setminus \{i, v\}} \delta_{ij}(v) \quad (5)$$

$\delta_i(v)$ は各点 $j \in V \setminus \{i, v\}$ の $\delta_{ij}(v)$ の総和である。式 (2), (3), (5) より, 式 (6) を得る。

$$C_B(v) = \sum_{i \in V \setminus \{v\}} \delta_i(v) \quad (6)$$

$S(v)$ はグラフ G で点 $v \in V$ に繋がっている次数が 1 である点の集合を表す。従って $A(v)$ は集合 $S(v)$ のサイズになる。点 $v \in V$ について, $S(v) \subset V$ か $v \notin S(v)$ であることから, 式 (6) より式 (7) が得られる。

$$C_B(v) = \sum_{i \in S(v)} \delta_i(v) + \sum_{i \in \overline{S(v)} \setminus \{v\}} \delta_i(v) \quad (7)$$

式 (7) の右辺の第 1 項について説明する。任意の点 $i \in S(v)$ について, 点 i と点 $j \in V \setminus \{i, v\}$ の最短経路上に必ず点 v が存在する。 V のサイズが n であるから, 式 (8) が成り立つ。

$$\delta_i(v) = n - 2 \quad (8)$$

$S(v)$ のサイズが $A(v)$ であることから, 式 (7) の右辺の第 1 項は式 (9) になる。

$$\sum_{i \in S(v)} \delta_i(v) = A(v)(n - 2) \quad (9)$$

式 (7) の右辺の第 2 項について説明する。ある点

$i \in \overline{S(v)} \setminus \{v\}$ の $A(i)$ が 1 以上であるとする。このとき, 点 $\forall w \in S(i)$ と点 $\forall j \in V \setminus \{i, w\}$ について, 点 w から点 j に到達する最短経路上で点 w から次の点 i になる。 $\sigma_{wj} = \sigma_{ij}$ であるため, 式 (5) から式 (10) が成り立つ。

$$\delta_w(v) = \delta_i(v) \quad (10)$$

任意の異なる 2 点 $a, b \in V' \setminus \{v\}$ について, $S(a)$ と $S(b)$ は独立な集合なので, 各点 $i \in \overline{S(v)} \setminus \{v\}$ は, V' または点 v でない各点 a の $S(a)$ のいずれかの要素となる。したがって, 式 (7) の右辺の第 2 項は式 (10) より式 (11) となる。

$$\begin{aligned} \sum_{i \in \overline{S(v)} \setminus \{v\}} \delta_i(v) &= \sum_{i \in V' \setminus \{v\}} (\delta_i(v) + \sum_{w \in S(i)} \delta_w(v)) \\ &= \sum_{i \in V' \setminus \{v\}} \delta_i(v)(1 + A(i)) \end{aligned} \quad (11)$$

まとめると, 式 (7), (9), (11) から式 (12) が得られる。

$$C_B(v) = A(v)(n - 2) + \sum_{i \in V' \setminus \{v\}} \delta_i(v)(1 + A(i)) \quad (12)$$

式 (12) の右辺の第 2 項の $\delta_i(v)$ について説明する。式 (5) は式 (13) になる。

$$\delta_i(v) = \sum_{j \in S(v)} \delta_{ij}(v) + \sum_{j \in \overline{S(v)} \setminus \{v\}} \delta_{ij}(v) \quad (13)$$

異なる 3 点 $i, v, w \in V$ とある点 $j \in V \setminus \{i, v\}$ について, $\delta_{ij}(v, w)$ は式 (14) で定義される。

$$\delta_{ij}(v, w) = \frac{\sigma_{iv} \cdot \sigma_{wj}}{\sigma_{ij}} \quad (14)$$

$\delta_{ij}(v, w)$ は, 点 i と点 j の最短経路の本数に対する, 点 i と点 j の最短経路で辺 (v, w) を通る経路の本数の割合である。点 j によって $\delta_{ij}(v)$ は式 (14) より式 (15) の場合分けが考えられる。

$$\delta_{ij}(v) = \begin{cases} 1 & (j \in S(v)) \\ \sum_{w \in V' : v \in P_i(w)} \delta_{ij}(v, \{v, w\}) & (j \in \overline{S(v)} \setminus \{v\}) \end{cases} \quad (15)$$

よって, 式 (13), (15) から式 (16) を得る。

$$\delta_i(v) = A(v) + \sum_{j \in S(v)} \sum_{w \in V': v \in P_i(w)} \delta_{ij}(v, \{v, w\}) \quad (16)$$

$\delta_{ij}(v, \{v, w\})$ は以下の場合分けになる.

$$\delta_{ij}(v, \{v, w\}) = \begin{cases} \frac{\sigma_{iv}}{\sigma_{ij}} & (j = w) \\ \frac{\sigma_{iv}}{\sigma_{iw}} \cdot \frac{\sigma_{ij}(w)}{\sigma_{ij}} & (j \neq w) \end{cases} \quad (17)$$

したがって、式 (1), (5), (16), (17) から式 (18) が導出できる.

$$\begin{aligned} \delta_i(v) &= A(v) + \sum_{w \in V': v \in P_i(w)} \sum_{j \in S(v)} \delta_{ij}(v, \{v, w\}) \\ &= A(v) + \sum_{w \in V': v \in P_i(w)} \left(\frac{\sigma_{iv}}{\sigma_{iw}} + \sum_{j \in S(v) \setminus w} \frac{\sigma_{iv}}{\sigma_{iw}} \cdot \frac{\sigma_{ij}(w)}{\sigma_{ij}} \right) \\ &= A(v) + \sum_{w \in V': v \in P_i(w)} \frac{\sigma_{iv}}{\sigma_{iw}} (1 + \delta_i(w)) \end{aligned} \quad (18)$$

第 1 の等号は式 (16) の和の順序を交換している、第 2 の等号は式 (17) を代入している。第 3 の等号は式 (1), (5) の代入し、因数分解で式 (18) を得る。

導出した式 (18), (12) は、 V' と A から媒介中心性を求める式である。従来手法の計算式と異なる点は計算式に V' と A を用いている点である。

4.2.2 計算手順

式 (12) で表現される $C_B(v)$ を計算する手順を説明する。式 (12) は、Step 1 で求めた G' と A 、及び、式 (18) で求めた $\delta_i(v)$ を含む。 G' 内で各点を始点とする最短経路探索を行う。ある点 $s \in V'$ を始点とする最短経路探索を行うとき、各点 $j \in V'$ に対して、 σ_{sj} , $P_s(j)$ を得る。点 s を始点とする最短経路で辺の本数が最も大きい経路の点 s でない端点 t について、 $\delta_s(t)$ は 0 である。点 t から順々に最短経路上の一つ手前の点 v に対して、式 (18) を用いて $\delta_s(v)$ を計算する。ここで得た任意の点 $v \in V' \setminus \{s\}$ に対する $\delta_s(v)$ を式 (12) の右辺の第 1 項に従い $C_B(v)$ に加算する。式 (12) の右辺の第 1 項に従い、始点 s の $C_B(v)$ に $A(s)(n-2)$ を加算する。

アルゴリズムの擬似コードを付録の Algorithm 2 に示す。

4.3 提案手法の計算時間

G 内の次数が 1 である点を c とする。次数が 1 以上の点の数を n' とする。提案した計算手法は、グラフ内の辺を全て調べることで、次数が 1 である点を全て取り除くので、圧縮にかかる時間は $O(m)$ である。圧縮

されたグラフに対して n' 個の点をそれぞれ始点とする幅優先探索を行うので、圧縮後の計算には $O(n'm)$ かかる。 $O(n'm)$ は $O(m)$ に支配的であるので、提案手法は圧縮と計算を合わせて $O(n'm)$ かかる。次数が 1 である点が存在するグラフにおいて $n > n'$ なので、 $O(nm) > O(n'm)$ となる。

5. 計算機実験

従来アルゴリズムと提案したアルゴリズムを計算機で走査させ、二つの計算時間を比較した。評価で用いるネットワークは Barabasi-Albert モデル [6] である。Barabasi-Albert モデル (BA モデル) は、現実世界に存在するネットワークの多くがもつスケールフリー性をモデル化したグラフである。

n 個の点の BA モデルを 10 回生成し、従来手法と提案手法を走査させ、それぞれの計算時間を測定した。計算時間は 10 回の平均をとっている。点数 n は 250 から 10000 まで 250 ずつ増加させた。計算環境として、CPU は Core i5 1.8, GHz, OS は Windows[®]10, メモリは 4.0 GB, 使用言語は Java 1.8.0 を用いた。

実験結果を図 2 に示す。提案手法の計算時間は従来手法に対して平均 65% 短縮している。このとき生成されたグラフは平均して全体の 40% の点の次数が 1 であった。

6. むすび

グラフ圧縮を用いる媒介中心性の計算手法を提案した。提案手法は、与えられたグラフから次数が 1 である点を取り除き、圧縮されたグラフで媒介中心性を求

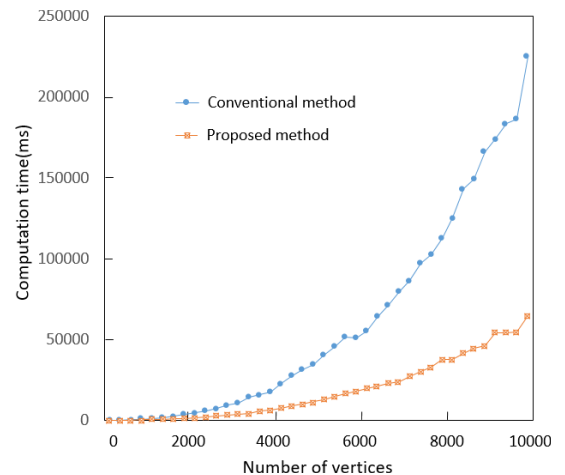


図 2 計算時間の比較
Fig. 2 Comparison of computation times.

める. 提案手法は従来手法で次数が1である点が存在するグラフで生じる冗長な計算を回避し, 計算量を削減することを計算時間の漸近的上界から示した. 計算機実験により, 提案手法が従来手法より計算時間が短くなることを確認した.

文 献

- [1] L.C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol.40, no.1, pp.35-41, 1977.
- [2] U. Brandes, "A faster algorithm for betweenness centrality," *J. Math. Sociology*, vol.25, no.2, pp.163-177, 2001.
- [3] L. Wang, F. Yang, L. Zhuang, H. Cui, F. Lv, and X. Feng, "Articulation points guided blackundancy elimination for betweenness centrality," *Proc. 21st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 2016.
- [4] D.A. Bader, S. Kintali, K. Madduri, and M. Mihail, "Approximating betweenness centrality," *Algorithms and Models for the Web-Graph. WAW 2007. Lecture Notes in Computer Science*, ed. A. Bonato, F.R.K. Chung, pp.124-137, Springer, San Diego, Dec. 2007.
- [5] M. Kas, M. Wachs, K.M. Carley, and L.R. Carley, "Incremental algorithm for updating betweenness centrality in dynamically growing networks," *Proc. IEEE/ACM Int. Conf. on Advances in Social Netw. Analysis and Mining*, Ontario Canada, pp.33-40. Aug. 2013.
- [6] A. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol.286, pp.509-512, Oct. 1999.

付 録

Algorithm 1 Conventional method

```

Require:  $G = V, E$ 
Ensure:  $C_B[v], v \in V$ 
 $C_B[v] \leftarrow 0, v \in V$ 
for  $s \in V$  do
   $S \leftarrow \text{empty stack}$ 
   $P[w] \leftarrow \text{empty list}, w \in V$ 
   $\sigma[t] \leftarrow 0, t \in V$ 
   $\sigma[s] \leftarrow 1$ 
   $d[t] \leftarrow -1, t \in V$ 
   $d[s] \leftarrow 0$ 
   $Q \leftarrow \text{empty queue}$ 
   $\text{enqueue } s \Rightarrow Q$ 
  while  $Q$  not empty do
     $\text{dequeue } v \leftarrow Q$ 
     $\text{push } v \Rightarrow S$ 
    for neighbor  $w$  of  $v$  do
      if  $d[w] < 0$  then
         $\text{enqueue } w \Rightarrow Q$ 
         $d[w] \leftarrow d[v] + 1$ 
      end if
      if  $d[w] = d[v] + 1$  then
         $\sigma[w] \leftarrow \sigma[w] + \sigma[v]$ 
         $\text{append } v \Rightarrow P[w]$ 
      end if
    end for
  end while
   $\delta[v] \leftarrow 0, v \in V$ 
  while  $S$  not empty do
     $\text{pop } w \leftarrow S$ 
    for  $v \in P[w]$  do
       $\delta[v] \leftarrow \delta[v] + \frac{\sigma[v]}{\sigma[w]}(1 + \delta[w])$ 
    end for
    if  $w \neq s$  then
       $C_B[w] \leftarrow C_B[w] + \delta[w]$ 
    end if
  end while
end for

```

Algorithm 2 Proposed method

Require: $G = \{V, E\}$
Ensure: $C_B[v], v \in V$
 $C_B[v] \leftarrow 0, v \in V$
 $A[v] \leftarrow 0, v \in V$
 $V' \leftarrow \emptyset$
for $v \in V$ **do**
 if $D(v) = 1$ **then**
 $A[w] \leftarrow A[w] + 1, (v, w) \in E$
 else
 $V' \leftarrow V' \cup \{v\}$
 end if
end for
for $s \in V'$ **do**
 $S \leftarrow \text{empty stack}$
 $P[w] \leftarrow \text{empty list}, w \in V$
 $\sigma[t] \leftarrow 0, t \in V$
 $\sigma[s] \leftarrow 1$
 $d[t] \leftarrow -1, t \in V$
 $d[s] \leftarrow 0$
 $Q \leftarrow \text{empty queue}$
 enqueue $s \Rightarrow Q$
 while Q not empty **do**
 dequeue $v \leftarrow Q$
 push $v \Rightarrow S$
 for neighbor w of $v, w \in V'$ **do**
 if $d[w] < 0$ **then**
 enqueue $w \Rightarrow Q$
 $d[w] \leftarrow d[v] + 1$
 end if
 if $d[w] = d[v] + 1$ **then**
 $\sigma[w] \leftarrow \sigma[w] + \sigma[v]$
 append $v \Rightarrow P[w]$
 end if
 end for
 end while
 $\delta[v] \leftarrow 0, v \in V$
 while S not empty **do**
 pop $w \leftarrow S$
 $\delta[w] \leftarrow \delta[w] + A[w]$
 for $v \in P[w]$ **do**
 $\delta[v] \leftarrow \delta[v] + \frac{\sigma[v]}{\sigma[w]}(1 + \delta[w])$
 end for
 if $w \neq s$ **then**
 $C_B(w) \leftarrow C_B(w) + \delta[w]$
 end if
 end while
 $C_B(s) \leftarrow C_B(s) + A[s](n - 2)$
end for

(平成 29 年 3 月 30 日受付, 6 月 17 日再受付,
7 月 4 日早期公開)