アルゴリズム 53−7
(1996. 9. 13)

辺連結度増加関数を$\tilde{O}(mn)$時間で計算するアルゴリズム

永持 仁, 白木 孝, 茨木 俊秀
京都大学 工学研究科 数理工学教室
〒606-01 京都市左京区吉田本町

摘要 辺に非負の重みを持つ無向グラフ $G=(V,E,c_G)$ が与えられたとき, このグラフを k-辺連結にするために付加すべき容量の必要最小量を$\Lambda_G(k)$ と記す. $\Lambda_G: R^+ \to R^+$を辺連結度増加関数と呼ぶ. このとき, Λ_Gは高々$n-1$ 個の折れ点を持つ区分的線形な関数であること, Λ_Gの同定が $O(nm+n^2\log n)$ 時間でできることを示す. ただし, $n=|V|$, $m=|E|$.

Computing Edge-Connectivity Augmentation Function in $\tilde{O}(nm)$ Time

Hiroshi NAGAMOCHI, Takashi SHIRAKI and Toshihide IBARAKI

Department of Applied Mathematics and Physics
Graduate School of Engineering
Kyoto University
Kyoto, Japan 606-01

Abstract For a given undirected graph $G=(V,E,c_G)$ with edges weighted by nonnegative reals $c_G: E \to \boldsymbol{R}^+$, let $\Lambda_G(k)$ stand for the minimum amount of weights to be added to make G k-edge-connected. This paper shows that function $\Lambda_G: R^+ \to R^+$ is a piecewise linear function with at most $n-1$ break points, and that the entire Λ_G can be deterministically computed in $O(nm+n^2\log n)$ time, where n and m are numbers of vertices and edges, respectively.

1 Introduction

Let $G = (V, E, c_G)$ be an edge-weighted undirected graph with a set V of vertices, a set E of edges, and a weight function $c_G : E \rightarrow \boldsymbol{R}^+$, where $\boldsymbol{R}^+$ denotes sets of nonnegative reals. We denote $n = |V|$ and $m = |E|$. An edge with end vertices u and v is denoted by (u, v). A singleton set $\{x\}$ may be simply written as x, and " $\subset$ " implies proper inclusion while " $\subseteq$ " implies " $\subset$ " or " $=$ ". For two disjoint subsets, $X, Y \subset V$, we denote by $E_G(X, Y)$ the set of edges, one of whose end vertices is in X and the other is in Y, and define $d_G(X, Y) = \sum_{e \in E_G(X,Y)} c_G(e)$. A *cut* is defined as a subset X of V with $\emptyset \neq X \neq V$, and the *size* of cut X is defined by $d_G(X, V - X)$, which may also be written as $d_G(X)$. For a subset $X \subseteq V$, define its *inner-connectivity* by $\lambda_G(X) = \min\{d_G(X') \mid \emptyset \neq X' \subset X\}$. In particular, $\lambda_G(V)$ (i.e., the size of a minimum cut in G) is called the *edge-connectivity* of G. G is called *k-edge-connected* if $\lambda_G(V) \geq k$. For example, the graph G in Fig. 1 has $\lambda_G(V) = 7$.

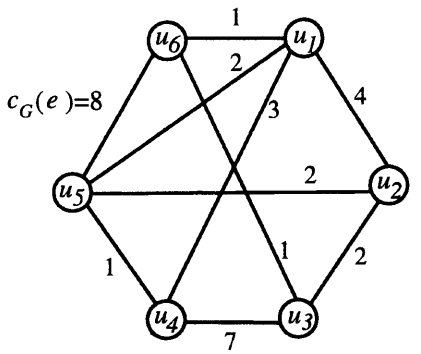

Figure 1: An edge-weighted graph G.

Given a graph $G = (V, E, c_G)$ and a $k \in \boldsymbol{R}^+$, *the edge-connectivity augmentation problem* asks to make G k-edge-connected by adding weights to the edges in G, where the weight of any edge in E can be increased and new edges not in E may be introduced. Let $\Lambda_G(k)$ denote the smallest total amount of weights added to make G k-edge-connected. We call $\Lambda_G(k)$ for $k \geq 0$ *the edge connectivity augmentation function* of G, which is clearly nondecreasing and convex. Since $\Lambda_G(k)$ can be written as the objective function of a linear programming problem with parameter $k \geq 0$, it is piecewise linear. For example, Fig. 2 illustrates function $\Lambda_G(k)$ of the graph G in Fig. 1.

Given a graph $G = (V, E, c_G)$ with an integer-valued weight function $c_G : E \rightarrow \boldsymbol{Z}^+$ and an integer $k \in \boldsymbol{Z}^+$, where $\boldsymbol{Z}^+$ denotes the set of nonnegative integers, the integer version of the edge-connectivity augmentation problem asks to make G k-edge-connected by adding integer weights to the edges in G. Let $\tilde{\Lambda}_G(k)$ denote the smallest total amount of the integer weights added to make G k-edge-connected.

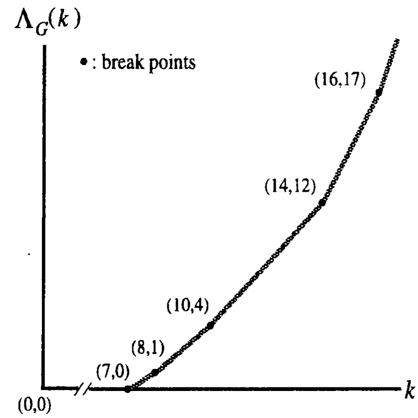

Figure 2: Edge connectivity augmentation function $\Lambda_G(k)$ of G in Fig. 1.

Watanabe and Nakamura [9] first proved that the integer version of the edge-connectivity augmentation problem can be solved in polynomial time for any given integer k. Different from the approach by Watanabe and Nakamura, Cai and Sun [1] first pointed out that the augmentation problem for a given k can be directly solved by applying the Lovász edge-splitting theorem. Based on this, Frank [2] gave an $O(n^5)$ time augmentation algorithm. Afterwards, Gabow [4] improved it to $O(mn^2 \log(n^2/m))$. Recently, Nagamochi and Ibaraki [6] gave an $O(n(m + n \log n) \log n)$ time algorithm. Note that all these algorithms can compute the set of edges to be added to make G k-edge-connected. If only the value $\tilde{\Lambda}_{G(k)}$ is required, the problem becomes slightly easier because [6] also says that $\tilde{\Lambda}_{G(k)}$ for a given k can be computed in $O(n(m + n \log n))$ time.

Clearly, $\tilde{\Lambda}_G(k) \geq \Lambda_G(k)$ holds for all k. However, $\tilde{\Lambda}_G(k)$ is almost the same as $\Lambda_G(k)$ since $\tilde{\Lambda}_G(k)$ can be obtained just by rounding up $\Lambda_G(k)$.

Lemma 1 [1, 2] *Let $G = (V, E, c_G)$ be a graph with an integer-valued weight function $c_G : E \rightarrow \boldsymbol{Z}^+$ and $k \in \boldsymbol{Z}^+$ be an integer with $k \geq \max\{2, \lambda_G(V)\}$. Then $2\Lambda_G(k)$ is an integer, and $\tilde{\Lambda}_G(k) = \lceil \Lambda_G(k) \rceil$ holds.* □

This paper shows the following results of $\Lambda_G(k)$ (hence of $\tilde{\Lambda}_G(k)$).

Theorem 1 *Function Λ_G for the entire range $k \geq 0$ can be deterministically computed in $O(n(m + n \log n))$ time.* □

Theorem 2 *Let (k_i^*, Δ_i^*), $i = 0, 1, 2, \ldots, r$, be all the break points of function $\Lambda_G(k)$, where $\Delta_i^* = \Lambda_G(k_i^*)$ for k_0^* $(= \lambda_G(V)) < k_1^* < k_2^* < \cdots < k_r^*$. Then:*

- (i) *Λ_G has at most $n-1$ break points (i.e., $r \le n-2$).*
- (ii) $\max_{v \in V} d_G(v) \le k_r^* \le \max\left[\max_{v \in V} d_G(v), \max_{X \subset V, |X| \ge 2} \frac{\sum_{v \in X} d_G(v) - d_G(X)}{|X|-1}\right]$ $(\le 2 \max_{v \in V} d_G(v))$.
- (iii) $\Delta_r^* = \frac{1}{2}\sum_{v \in V}\{k_r^* - d_G(v)\}$.
- (iv) *Let $\frac{d\Lambda_G(k)}{dk}$ denote the gradient of function $\Lambda_G(k)$ at $k \notin \{k_i^* \mid i = 0, 1, 2, \ldots, r\}$. Then $\frac{d\Lambda_G(k)}{dk} \in \{0, \frac{2}{2}, \frac{3}{2}, \ldots, \frac{n}{2}\}$, and $\frac{d\Lambda_G(k)}{dk} = \frac{n}{2}$ for $k > k_r^*$.* □

To show the above results, we modify the $O(n(m + n\log n))$ time algorithm in [6] that computes $\Lambda_G(k)$ for a given k, so that the single run of the algorithm simulates its execution for the entire range of $k \ge 0$ (we do not rely on any parametric search technique of mathematical programming). We can show that the resulting algorithm still runs in $O(n(m+n\log n))$ time, and prove the properties in Theorem 2 by tracing its behavior.

2 Preliminaries

For an edge-weighted graph $G = (V, E, c_G)$, its vertex set V and edge set E may also be denoted by $V[G]$ and $E[G]$, respectively, and the weight $c_G(e)$ of edge $e = (u, v)$ by $c_G(u, v)$. Without loss of generality, we assume that G has no multiple edges. For a subset $X \subseteq V$, $G[X]$ denotes the subgraph of G induced by X. For a vertex $v \in V$, a vertex $u \ne v$ adjacent to v by an edge is called a *neighbor* of v in G, and $\Gamma_G(v) = \{w \in V \mid (v, w) \in E\}$ denote the set of all *neighbors* of v in G.

We say that a cut X *separates* two disjoint subsets Y and Y' of V if $Y \subseteq X$ and $Y' \subseteq V - X$ (or $Y \subseteq V - X$ and $Y' \subseteq X$) hold. In particular, a cut X separates x and y if $x \in X$ and $y \in V - X$ (or $y \in X$ and $x \in V - X$). We say that a cut X *divides* a subset $Z \subseteq V$ if $X - Z \ne \emptyset \ne Z - X$. The *local edge-connectivity* $\lambda_G(x, y)$ for two vertices x and y is defined to be the minimum size of a cut that separates x and y (i.e., divides $\{x, y\}$). An ordering $v_1, v_2, \ldots, v_n$ of all vertices in V is called a *maximum adjacency* (MA) *ordering* (also called *legal* in [3]) in G if it satisfies $d_G(\{v_1, v_2, \ldots, v_i\}, v_{i+1}) \ge d_G(\{v_1, v_2, \ldots, v_i\}, v_j)$ $1 \le i < j \le n$.

Lemma 2 [7, 8] *Let $G = (V, E, c_G)$ be an edge-weighted graph.*

(i) *An MA-ordering $v_1, v_2, \ldots, v_n$ of vertices in G can be found in $O(m + n\log n)$ time.*

(ii) *For the last two vertices v_{n-1} and v_n, $\lambda_G(v_{n-1}, v_n) = d_G(v_n)$.* □

Given an MA-ordering $v_1, v_2, \ldots, v_n$ in G, we write $u \le u'$ if $u = v_i$ and $u' = v_{i'}$ for $i \le i'$, and arrange the edges in $E_G(\{v_1, \ldots, v_{i-1}\}, v_i)$ $(i = 2, 3, \ldots, n)$ in the order of

$$e_{i,1} = (u_{i,1}, v_i), \ldots,\ e_{i,r_i} = (u_{i,r_i}, v_i), \qquad (1)$$

where $r_i = |E_G(\{v_1, \ldots, v_{i-1}\}, v_i)|$, so that $u_{i,1} \le u_{i,2} \le \cdots \le u_{i,r_i}$ holds. Now given a real $\delta \ge 0$, we define the weight functions c_{G_δ} and $c_{\overline{G}_\delta}$ as follows:

$$c_{G_\delta}(u_{i,j}, v_i) = \begin{cases} c_G(u_{i,j}, v_i), & \text{if } j \le p_i - 1 \\ \delta - \sum_{1 \le j \le p_i - 1} c_G(u_{i,j}, v_i), & \text{if } j = p_i \\ 0, & \text{if } j > p_i \end{cases}$$

$$c_{\overline{G}_\delta}(e) = c_G(e) - c_{G_\delta}(e) \text{ for all edges } e \in E,$$

where p_i denotes the smallest index such that $\sum_{1 \le j \le p_i} c_G(u_{i,j}, v_i) \ge \delta$ (we interpret $p_i = r_i + 1$ if $\sum_{1 \le j \le r_i} c_G(u_{i,j}, v_i) < \delta$). We call the resulting graphs $G_\delta = (V, E, c_{G_\delta})$ and $\overline{G}_\delta = (V, E, c_{\overline{G}_\delta})$ δ-*skeleton* and δ-*skin* of G (with respect to ordering $v_1, \ldots, v_n$), respectively. The following lemma is an elaboration of the property stated in Lemma 2(ii).

Lemma 3 [7] *For an edge-weighted graph $G = (V, E, c_G)$, let $v_1, v_2, \ldots, v_n$ be an MA-ordering of vertices in G, and δ be a real with $0 \le \delta \le d_G(v_n)$. Then, the δ-skeleton G_δ and the δ-skin $\overline{G}_\delta = (V, E, \bar{c}_\delta)$ of G respectively satisfy*

$$\lambda_{G_\delta}(v_{n-1}, v_n) = \delta\ (= d_{G_\delta}(v_n)),$$

$$\lambda_{\overline{G}_\delta}(v_{n-1}, v_n) = d_G(v_n) - \delta\ (= d_{\overline{G}_\delta}(v_n)). \ \square$$

When we regard G_δ (resp., $\overline{G}_\delta$) as the set of infinite numbers of graph instances corresponding to all δ with $0 \le \delta \le d_G(v_n)$, this lamma claims that a pair of vertices v and w such that $\lambda_{G_\delta}(v, w) = d_{G_\delta}(w)$ (resp., $\lambda_{\overline{G}_\delta}(v, w) = d_{\overline{G}_\delta}(w)$) can be chosen commonly to all δ (i.e., $v = v_{n-1}$ and $w = v_n$) without solving an infinite number of maximum flow problems.

3 Computing $\Lambda_G(k)$ for a Fixed k

In this section, we review the $O(n(m + n\log n))$ time algorithm in [6] for computing $\Lambda_G(k)$ for a given $k \in \boldsymbol{R}^+$. We first give an optimality condition of the edge-connectivity augmentation problem.

Let $s \in V$ be a *designated vertex* in G. A cut X is called *s-proper* if $\emptyset \neq X \subset V - s$. $\lambda_G(V - s)$ (i.e., the size of a minimum s-proper cut) is called the *s-based-connectivity* of G. Obviously $\lambda_G(V) = \min\{\lambda_G(V-s), d_G(s)\}$. A family $\mathcal{X} = \{X_1, X_2, \ldots, X_p\}$ (possibly $p = 0$) of disjoint subsets $X_i \subset V - s$ is called a *collection* in $V - s$. If

$$\sum_{i=1}^{p} d_G(s, X_i) = d_G(s)$$

holds, then $\mathcal{X}$ is called *covering* (i.e., every neighbor of s is contained in some subset $X_i \in \mathcal{X}$). An s-proper cut X is called *(k,s)-critical* in G if it satisfies $d_G(s, X) > 0$, $d_G(X) = k$ and $\lambda_G(X) \geq k$. A collection $\mathcal{X}$ in $V - s$ is called *(k,s)-critical* in G either if $\mathcal{X} = \emptyset$ or if all $X_i \in \mathcal{X}$ are (k,s)-critical.

Lemma 4 [6] *For a graph $G = (V, E, c_G)$ with a designated vertex $s \in V$, and a nonnegative real $k \in \boldsymbol{R}^+$, let $\mathcal{X}$ be a (k,s)-critical covering collection in G. Then $|\mathcal{X}| \neq 1$.* □

Lemma 5 [1] *Let $G = (V, E, c_G)$ be an undirected graph, and k be a nonnegative real. If a new vertex s and a set $E'(s)$ of weighted edges incident to s can be added to G so that the resulting graph $G' = (V \cup \{s\}, E \cup E'(s), c_{G'})$ satisfies the following conditions (i)-(ii), then $\Lambda_G(k) = d_{G'}(s)/2$.*

(i) $\lambda_{G'}(V) \geq k$.

(ii) *G' has a (k,s)-critical covering collection $\mathcal{X}$.* □

We now describe the algorithm of [6] which constructs the G' in Lemma 5. It first adds a new vertex s to G, and continue adding weighted edges between s and some vertices in G until the resulting graph G' satisfies (ii), while maintaining a (k,s)-critical covering collection in the current graph. To compute G', an auxiliary graph H, which is obtained by contracting some vertices in G', is prepared and updated.

A complete description of the entire algorithm is given as follows.

Algorithm AUGMENT

Input: An edge-weighted graph $G = (V, E, c_G)$ and a real $k \geq 0$.

Output: $\Lambda_G(k)$, a graph G' with a set $E'(s)$ of weighted edges satisfying Lemma 5, and a (k,s)-critical covering collection $\mathcal{X}$ of G'.

1 **begin**
2 Let $U = \{u_1, u_2, \ldots, u_p\}$ be the set of vertices $u_i \in V$ such that $d_G(u_i) < k$;
3 $V' = V \cup \{s\}$;
4 $E' := E \cup \{(s, u_1), \ldots, (s, u_p)\}$;
5 **for** each $u_i \in U$ **do**
6 $c_{G'}(s, u_i) := k - d_G(u_i)$
7 **end**; { for }
8 Let $G' = (V', E', c_{G'})$ be the resulting graph (where $c_{G'}(e) = c_G(e)$, $e \in E$);
9 $\mathcal{X} := \{\{u_1\}, \{u_2\}, \ldots, \{u_p\}\}$;
10 $H := G'$;
11 **while** $|V[H]| \geq 4$ **do**
12 Find two vertices $v, w \in V[H] - s$ such that $\lambda_H(v, w) \geq k$;
13 Contract v and w in H into a vertex x^*, and let H be the resulting graph;
14 **if** $d_H(x^*) < k$ **then**
15 Let $X^* \subseteq V - s$ denote the set of vertices that have been contracted into x^* so far;
16 Choose an arbitrary vertex $u \in X^*$, and let $c_{G'}(s, u) := c_{G'}(s, u) + k - d_H(x^*)$;
17 Let G' be the resulting graph;
18 Let H denote the graph obtained from H by increasing $c_H(s, x^*) := c_H(s, x^*) + k - d_H(x^*)$;
19 $\mathcal{X} := \mathcal{X} \cup \{X^*\}$, after discarding all X' with $X' \subset X^*$ from $\mathcal{X}$;
20 **end** { if }
21 **end**; { while }
22 $\Lambda_G(k) := d_H(s)/2$ $(= d_{G'}(s)/2)$;
23 Output $\Lambda_G(k)$, $G' = (V \cup \{s\}, E', c_{G'})$ and $\mathcal{X}$
24 **end**. { AUGMENT }

After each iteration of the while-loop, we easily see from construction that $\mathcal{X}$ is a collection, and that each cut X in $\mathcal{X}$ is (k,s)-critical in the current G'. Hence $\mathcal{X}$ is a (k,s)-critical covering collection in G', and the final $\mathcal{X}$ satisfies (ii) of Lemma 5.

Now we consider (i) of Lemma 5. From construction of H,

$$d_H(u) \geq k \quad \text{for all } u \in V[H] - s \tag{2}$$

holds after each iteration of the while-loop. Then we can choose vertices $v, w \in V[H] - s$ such that

$$\lambda_H(v, w) \geq k \tag{3}$$

(i.e., those in line 12) as follows. We compute an MA-ordering of the vertices in H starting from $v_1 = s$. The last two vertices, say v, w, in this ordering are different from s, and satisfy $\lambda_H(v, w) = c_H(w)$ by Lemma 2(ii), and hence $\lambda_H(v, w) \geq k$ by (2). The time to find such two vertices v and w is $O(|E[H]| + |V[H]| \log |V[H]|) = O(m + n \log n)$.

Since two vertices v, w with $\lambda_H(v, w) \geq k$ are contracted into a single vertex in each iteration of the while-loop, it is not difficult to see that, for each vertex $u \in V[H] - s$,

$$\lambda_{G'}(v, w) \geq k \quad \text{for all } v, w \in Y_u \tag{4}$$

holds after each iteration of the while-loop, where $Y_u \subseteq V - s$ denotes the set of vertices that have been contracted into u so far. From (2) and (4), the final graph G' satisfies $\lambda_{G'}(V) \geq k$, i.e., (i) of Lemma 5. For details, see [6].

4 Computing $\Lambda_G(k)$ for All k

Since all break points of Λ_G occur in the range stated in Theorem 2(ii) (although the proof of this is omitted), we only consider reals k such that $0 \leq k \leq K$, where

$$K = 1 + 2 \max_{v \in V} d_G(v). \qquad (5)$$

Suppose that we first execute algorithm AUGMENT in the previous section for $k = K$. In this case, AUGMENT first adds an edge (s, u_i) of weight $K - d_G(u_i)$ for each vertex $u_i \in V$ in lines 5–7. We now try to modify the rest of AUGMENT so that $\Lambda_G(k)$ for all $k \leq K$ are implicitly computed. For this, we treat the weight $c_G(e)$ of each edge $e \in E_{G'}(s)$ as a set $R(e)$ of ranges, defined in the following, so that $\Lambda_G(k)$ for an arbitrary $k \leq K$ can be effectively retrieved.

4.1 Ranged graph

For two reals $a, b \in \boldsymbol{R}^+$ with $a < b$, the interval $[a, b]$ is called a *range*, and its size $\pi([a, b])$ is defined as $b - a$. Let $R = \{[a_1, b_1], [a_2, b_2], \ldots, [a_t, b_t]\}$ be a set of ranges. The size of R, denoted by $\pi(R)$, is defined as the sum of all range sizes in R:

$$\pi(R) = (b_1 - a_1) + \ldots + (b_t - a_t).$$

For a given $k \in \boldsymbol{R}^+$, we define the following operations on a set R of ranges. For a $\delta \in \boldsymbol{R}^+$, we say that range $[a - \delta, b - \delta]$ is obtained by *lowering* $[a, b]$ by δ. The *upper k-truncation* of a range $[a, b]$ is defined by

$$[a, b]|^k = \begin{cases} [a, \min\{b, k\}] & \text{if } a < k \\ \emptyset & \text{otherwise,} \end{cases}$$

where $\pi(\emptyset)$ is defined to be 0. Based on this, the upper k-truncation of a set R of ranges is defined by

$$R|^k = \{[a_i, b_i]|^k \neq \emptyset \mid [a_i, b_i] \in R\}.$$

Similarly, the *lower k-truncation* of a range $[a, b]$ is defined by

$$[a, b]|_k = \begin{cases} [\max\{a, k\}, b] & \text{if } b > k \\ \emptyset & \text{otherwise,} \end{cases}$$

and the lower k-truncation of a set R of ranges is defined by

$$R|_k = \{[a_i, b_i]|_k \neq \emptyset \mid [a_i, b_i] \in R\}.$$

We may write $(R|^k)_{k'}$ $(k' \leq k)$ as $R|^k_{k'}$.

From a graph $G = (V, E, c_G)$, construct another graph $G' = (V' = V \cup \{s\}, E' = E \cup E'(s), c_G, R_{G'})$ with a designated vertex s such that (a) G' has edges between s and all vertices $v \in V$ (i.e., $E'(s) = \{(s, v) \mid v \in V\}$), (b) c_G is a weight function on E, and (c) $R_{G'}(v)$ is a set of ranges associated with each vertex $v \in V$. We call such a graph as a *ranged graph*. In a ranged graph, we define the weight of edge $e = (s, v) \in E'(s)$ by $\pi(R_{G'}(v))$. Based on this definition, we can extend c_G and $R_{G'}$ into a weight function $c_{G'} : E' \to \boldsymbol{R}^+$, such that $c_{G'}(e) = c_G(e)$ if $e \in E$ and $c_{G'}(e) = \pi(R_G(v))$ if $e = (s, v) \in E'(s)$. Then $d_{G'}(X, Y)$ is similarly defined by using $c_{G'}$.

For notational convenience, $\cup_{x \in X} R_{G'}(x)$ for a subset $X \subseteq V$ may be written as $R_{G'}(X)$. The ranged graph $(V', E', c_G, R_{G'}|^k)$ obtained from a ranged graph G' by upper k-truncating $R_{G'}(v)$ for all $v \in V$ is denoted by $G'|^k$.

Now we say that two range sets R and R' are *equivalent* if $\pi(R|^k) = \pi(R'|^k)$ holds for all $k \in \boldsymbol{R}^+$. A set R of ranges is called *gapless* if $\pi(R|^k) < \pi(R|^{k'})$ holds for any $\min\{a \mid [a, b] \in R\} \leq k < k' \leq \max\{b \mid [a, b] \in R\}$.

Given a gapless set of ranges $R = \{[a_1, b_1], [a_2, b_2], \ldots, [a_t, b_t]\}$, in which $b_1 \leq b_2 \leq \cdots \leq b_t$ is assumed without loss of generality, we modify R into another set of ranges $R' = \{[a_1 - \delta_1, b_1 - \delta_1], [a_2 - \delta_2, b_2 - \delta_2], \ldots, [a_t - \delta_t, b_t - \delta_t]\}$ by lowering each range $[a_i, b_i] \in R$ by $\delta_i \geq 0$, such that R' satisfies $\delta_t = 0$ (i.e., $b_t = b^*$), and $b_i - \delta_i = a_{i+1} - \delta_{i+1}$ for $i = 1, \ldots, t-1$ (i.e., R' is equivalent to a single range $[b^* - \pi(R), b^*]$). We call such R' an *alignment* of R. By definition, an alignment R' is equivalent to a single range $[b^* - \pi(R), b^*]$.

4.2 Totally optimal ranged graph

We now extend the optimality conditions in Lemma 5 to a ranged graph.

Definition 1 *For a given graph $G = (V, E, c_G)$, a ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$, where s is a designated vertex and $E'(s) = \{(s, v) \mid v \in V\}$, is called* totally optimal *if G' satisfies the following conditions* (i) *and* (ii) *for all k with $0 \leq k \leq K$.*

(i) $\lambda_{G'|^k}(V) \geq k$.

(ii) *$G'|^k$ has a (k, s)-critical covering collection $\mathcal{X}^k$.* □

If such a totally optimal ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$ is obtained, then, by Lemma 5, we can easily compute $\Lambda_G(k)$ for any

$k \in \boldsymbol{R}^+$ by

$$\Lambda_G(k) = \frac{d_{G'|^k}(s)}{2} = \frac{\pi(R_{G'}(V)|^k)}{2}. \qquad (6)$$

We describe in the next section how to compute a totally optimal ranged graph G' from a given graph G. To prove total optimality of a ranged graph G', we need to show the existence of a (k, s)-critical covering collection $\mathcal{X}^k$ of $G'|^k$ for all k with $0 \leq k \leq K$. Although (k, s)-critical covering collections $\mathcal{X}^k$ may be different for different k, we are able to show that a set of (k, s)-critical covering collections $\mathcal{X}^k$ for all k can be compactly represented by using the following notion. A pair $(X, [a, b])$ of a cut X and a range $[a, b]$ is called a *ranged cut*, and a set

$$\mathcal{X} = \{(X_i, [a_i, b_i]) \mid i = 1, 2, \ldots, r\}$$

of ranged cuts is called a *ranged collection* if

$$\mathcal{X}|^k = \{X_i \mid (X_i, [a_i, b_i]) \in \mathcal{X},\ a < k \leq b\}$$

is a collection (i.e., X_i's in $\mathcal{X}|^k$ are disjoint) for all real k with $0 \leq k \leq K$. A *ranged collection* $\mathcal{X}$ is called *totally critical covering* (with respect to a ranged graph G') if $\mathcal{X}|^k$ is a (k, s)-critical covering collection in $G'|^k$.

4.3 Main algorithm

Given a graph $G = (V, E, c_G)$, we now construct a totally optimal ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$ and a totally critical covering ranged collection $\mathcal{X}$ of G'. We first run lines 5–7 of algorithm AUGMENT in Section 3 for $k = K$. Then an edge (s, u_i) of weight $K - d_G(u_i)$ is introduced for each vertex $u_i \in V$. However, if $\Lambda_G(k)$ for $k < K$ is to be computed, then this weight must be $\max\{k - d_G(u_i), 0\}$. To take into account this general situation, we introduce a set of ranges $R_{G'}(u_i)$ (initially it consists of a single range $[d_G(u_i), K]$) for each vertex $u_i \in V$, and interpret that the weight of edge (s, u_i) is $\pi(R_{G'}(u_i)|^k)$. At this point, $G'|^k$ satisfies property (2) for any k with $0 \leq k \leq K$, i.e.,

$$d_{G'|^k}(u_i) \geq k \text{ for all vertices } u_i \in V.$$

Also by setting initially the ranged collection $\mathcal{X}$ as $\{(\{u_i\}, [d_G(u_i), K]) \mid u_i \in V\}$, we see that $\mathcal{X}|^k = \{\{u_i\} \mid u_i \in V,\ d_G(u_i) < k\}$ gives a (k, s)-critical covering collection in $G'|^k$ for all k with $0 \leq k \leq K$. That is, this $\mathcal{X}$ is a totally critical covering ranged collection of the current G' (but G' is not totally optimal yet, since (i) of Definition 1 may not be satisfied). The above operations imply that lines 1–9 of AUGMENT are carried out simultaneously for all k with $0 \leq k \leq K$.

Next after setting $H := G'$ in line 10 of AUGMENT, we repeat the while-loop, in which a pair v, w of vertices in H is contracted into a single vertex x^*. H is also a ranged graph such that H is obtained from the current ranged graph G' by contracting some subsets of V into single vertices. Suppose that the following properties (a)–(c) hold after each iteration of the while-loop.

(a) $R_H(u)$ and $R_{G'}(Y_u)$ are equivalent for each $u \in V[H] - s$ and the set of vertices Y_u that have been contracted into u so far.

(b) $d_{H|^k}(u) \geq k$ for all vertices $u \in V[H] - s$.

(c) There is a totally critical covering ranged collection $\mathcal{X}$ of the current G'.

(d) For each vertex $u \in V[H] - s$, $R_H(u)$ is a set of ranges in the from of $\{[a_1, K], [a_2, K], \ldots, [a_{r_u}, K]\}$ (where r_u is the number of vertices contracted into u so far), and contains a range $[a, K]$ such that $a \leq d_H(u, V[H] - \{s, u\})$.

Initially all (a)-(d) are satisfied by the above $H = G'$ and $\mathcal{X}$. We try to contract some two vertices into a single vertex in an iteration of the while-loop until H has only three vertices, while maintaining (a)–(d).

By property (b) and Lemma 2(ii), there is a pair of vertices $v, w \in V[H]$ such that

$$\lambda_{H|^k}(v, w) \geq k. \qquad (7)$$

Furthermore, we ask that such two vertices $v, w \in V[H]$ commonly satisfy (7) for all k with $0 \leq k \leq K$, since otherwise (i.e., if different pairs need be contracted for different k) the single run of AUGMENT cannot simulate the runs for all k. By making use of Lemma 3, we can show that such a pair of vertices always exists. See Appendix for a proof of this fact, which plays a key role in the paper. After finding such vertices v and w, we contract them into a single vertex x^*, as in algorithm AUGMENT, and update as $R_H(x^*) := R_H(v) \cup R_H(w)$.

We then check if (b) still holds in the resulting ranged graph H. If (b) holds, then we proceed to the next iteration of the while-loop, since we can show that the current H, G' and $\mathcal{X}$ satisfy (a)–(d).

On the other hand, if (b) does not hold (i.e., $d_{H|^k}(x^*) < k$ holds for some k with $0 \leq k \leq K$), we have to increase the weight of edge $(s, x^*) \in E[H]$ in $H|^k$ to recover the condition $d_{H|^k}(x^*) \geq k$. This can be achieved only by updating the set $R_H(x^*)$ of ranges. Let X^* denote the set of vertices in G, which have been contracted into x^* in H. By definition,

$$d_{H|^k}(x^*) = k^* + \pi(R_{H|^k}(x^*))$$

holds, where $k^* = d_H(x^*, V[H] - \{s, x^*\})$ ($= d_G(X^*)$, by (a)). Then

$$d_{G'|^k}(X^*) = k^* + \pi(R_{G'|^k}(X^*))$$

holds from (a). We then modify the set of ranges in $R_H(x^*)$ so that $d_{H|^k}(x^*) \geq k$ holds for all k with $0 \leq k \leq K$. In other words, the resulting set of ranges $R_H(x^*)$ must have the property that $R_H(x^*)|_{k^*}^{K}$ is gapless. From (d), we can assume w. l. o. g. $R_H(x^*) = \{[a_1, K], \ldots, [a_r, K]\}$ and $a_1 \leq \cdots \leq a_r$. If $k^* \geq a_1$, then $R_H(x^*)|_{k^*}^{K}$ obviously is gapless and hence $d_{H|^k}(x^*) = k^* + \pi(R_{H|^k}(x^*)) \geq k^* + (k - k^*) = k$ for $0 \leq k \leq K$ (i.e., (b) holds). Therefore, we consider the case of $k^* < a_1$.

To achieve (b) and (c), we determine k' by

$$\pi(R_H(x^*)|^{k'}) = k' - k^*. \qquad (8)$$

Condition (8) means that there is an alingment of $R_H(x^*)|^{k'}$ into $[k^*, k']$. Then we update $R_H(x^*)$ by $R_H(x^*) := \{[k^*, k']\} \cup R_H(x^*)|_{k'}$ and, to maintain (d), we futher merge $[k^*, k']$ and $[a_1, K]|_{k'}$ into a single range $[k^*, K]$.

To maintain (a) (i.e., equivalence between $R_H(x^*)$ and $R_{G'}(X^*)$), some (parts) of the ranges in $R_{G'}(X^*)$ must be lowered. This can be done by the following procedure.

> Define $A_u := R_{G'}(u)|^{k'}$ for all $u \in X^*$, where $k^* + \pi(\cup_{u \in X^*} A_u) = k'$ holds by (a) and (8). Align $\mathcal{A} = \cup_{u \in X^*} A_u$ to obtain $\mathcal{A}' = \cup_{u \in X^*} A'_u$ such that $\mathcal{A}'$ is equivalent to a single range $[k^*, k']$. Finally, let $R_{G'}(u) := A'_u \cup R_{G'}(u)|_{k'}$ for each $u \in X^*$.

Note that, in the above process of updating $R_H(x^*)$ and $R_{G'}(X^*)$, $\pi(R_H(x^*)|^k)$ ($= \pi(R_{G'}(X^*)|^k)$) remains the same for all k with $0 \leq k \leq k^*$ or $k' \leq k \leq K$, but increases for all k with $k^* < k < k'$. Therefore, any cut $X \in \mathcal{X}|^k$ with $X \subseteq X^*$ which was previously (k, s)-critical in G' remains (k, s)-critical for all k with $k' < k \leq K$. Furthermore X^* is now a (k, s)-critical cut for all k with $k^* < k \leq k'$. From these facts, if we update $\mathcal{X}$ by

$$\begin{aligned} \mathcal{X} \;:=\; & \{(X, [a, b]) \in \mathcal{X} \mid X \subset V - X^*\} \\ & \cup\{(X^*, [k^*, k'])\} \\ & \cup\{(X, [\max\{a, k'\}, b]) \mid (X, [a, b]) \in \mathcal{X}, \\ & \quad X \subset X^*,\ k' < b\}, \end{aligned}$$

then the new $\mathcal{X}$ is a totally critical covering ranged collection in G'.

We repeat the above iteration until H has only three vertices. After $n - 3$ iterations of the while-loop, we can prove that the resulting G' satisfies

$$\lambda_{G'|^k}(V) \geq k \text{ for all } k \text{ with } 0 \leq k \leq K$$

from the fact that only those $v, w \in V[H]$ such that $\lambda_{H|^k}(v, w) \geq k$ have been contracted. The final G' is a totally optimal ranges graph of G, since it has a totally critical covering ranged collection.

The entire description of the above algorithm is given as follows. For the purpose of computing Λ_G, we only need to find a set of ranges $R^* = R_H(V[H] - s)$ which is equivalent to a set of ranges $R_{G'}(V)$ in a totally optimal ranged graph G' of G. In the algorithm, however, G' and $R_{G'}$ as well as a totally critical covering ranged collection $\mathcal{X}$ of G' are also explicitly computed (in lines 18-21) to see how $\mathcal{X}$is updated.

Algorithm SIMUL-AUGMENT

Input: A weighted graph $G = (V, E, c_G)$.
Output: A totally optimal ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$ for G, a totally critical covering ranged collection $\mathcal{X}$ of G', and a set of ranges R^* which is equivalent to $R_{G'}(V)$.

begin
1 $V' := V \cup \{s\}$; $E'(s) = \{(s, v) \mid v \in V\}$;
2 $K := 1 + 2\max_{v \in V} d_G(v)$; $\mathcal{X} := \emptyset$;
3 **for** each vertex $u \in V$ **do**
4 $R_{G'}(u) := \{[d_G(u), K]\}$;
5 $\mathcal{X} := \mathcal{X} \cup \{(\{u\}, [d_G(u), K])\}$
end; { for }
6 Let $G' = (V', E' = E \cup E'(s), c_G, R_{G'})$ be the obtained ranged graph;
7 $H := G'$;
8 **while** $|V[H]| \geq 4$ **do**
9 Find vertices $v, w \in V[H] - s$ such that $\lambda_{H|^k}(v, w) \geq k$, $0 \leq k \leq K$;
10 Contract v and w into a vertex x^*;
11 $R_H(x^*) := R_H(v) \cup R_H(w)$;
{ Assume $R_H(x^*) = \{[a_1, K], [a_2, K], \ldots, [a_r, K]\}$, where $a_1 \leq \cdots \leq a_r$ }
12 Let H be the resulting ranged graph;
13 $k^* := d_H(x^*, V[H] - \{s, x^*\})$;
14 **if** $k^* < a_1$ **then**
15 Let $X^* \subset V - s$ be the set of vertices contracted into x^* so far;
16 Find k' such that $\pi(R_H(x^*)|^{k'}) = k' - k^*$;
17 $R_H(x^*) := (R_H(x^*) - \{[a_1, K]\})|_{k'} \cup \{[k^*, K]\}$;

- -

18 $A_u := R_{G'}(u)|^{k'}$ for each $u \in X^*$;
{ $k^* + \pi(\cup_{u \in X^*} A_u) = k'$ holds }
19 Align $\mathcal{A} = \cup_{u \in X^*} A_u$ into $[k^*, k']$, and let $\cup_{u \in X^*} A'_u$ be the resulting set of ranges, where A'_u is obtained from A_u in the alignment;
20 $R_{G'}(u) := A'_u \cup R_{G'}(u)|_{k'}$ for each $u \in X^*$;
21 $\mathcal{X} := \{(X, [a, b]) \in \mathcal{X} \mid X \subset V - X^*\}$
$\cup\{(X^*, [d_G(X^*), k'])\}$
$\cup\{(X, [\max\{a, k'\}, b]) \mid$

```
        (X, [a,b]) ∈ 𝒳, X ⊂ X*, k' < b};
        - - - - - - - - - - - - - - - - - - - - - - - - -
    end; { if }
    Denote the ranged graphs resulting from
    H and G', respectively, as H and G' again
  end; { while }
  Output G', 𝒳 and R* = R_H(V[H] − s)
end. { SIMUL-AUGMENT }
```

The proofs of Theorems 1 and 2 are omitted due to space limitation.

Appendix:

We prove that (d) implies that H has a pair of vertices $v, w \in V[H] - s$ such that $\lambda_{H|^k}(v,w) \geq k$ for all k with $0 \leq k \leq K$.

Lemma 6 *Assume that the set of ranges $R_H(u)$ of each vertex $u \in V[H] - s$ contains a range $[a, K]$ such that $a \leq d_H(u, V[H] - \{s,u\})$. Then:*

(i) *If $V[H] \geq 3$, there is a pair of vertices $v, w \in V[H] - s$ such that $\lambda_{H|^k}(v,w) \geq k$ holds for all k with $0 \leq k \leq K$.*

(ii) *Such a pair of vertices v and w in (i) can be found in in $O(|E[H]|+|V[H]|\log|V[H]|)$ time.*

Proof: (i) Let $k_u^* = d_H(u, V[H] - \{s,u\})$ for $u \in V[H] - s$. It is sufficient to show that the lemma holds in the case where

$$R_H(u) = \{[k_u^*, K]\} \text{ for all } u \in V[H] - s, \qquad (9)$$

since this case has the minimum $\pi(R_H(u)|^k)$ for any k (under the assumption in the lemma statement). Let $s = v_1, v_2, \ldots, v_{n'}$ be an MA-ordering of H starting from s. Then we show that $v = v_{n'-1}$ and $w = v_{n'}$ satisfy the lemma; i.e., $\lambda_{H|^k}(v_{n'-1}, v_{n'}) \geq k$ holds for all k with $0 \leq k \leq K$. From (9), weight $c_{H|^K}(s,u)$ of edge (s,u), $u \in V[H] - s$ is $K - k_u^*$ in graph $H|^K$.

For a given k, we consider the δ-skin $\overline{(H|^K)}_\delta = (V[H], E[H], c_{\overline{(H|^K)}_\delta})$ of graph $H|^K$ (see Section 2 for the definition of δ-skin), where $\delta = K - k$. From the definition of δ-skin, we see that $c_{\overline{(H|^K)}_\delta}(s,u) = \max\{c_{(H|^K)}(s,u) - \delta, 0\}$ holds for $u \in V[H] - s$, because $s = v_1$ has the smallest index among the neighbors of any vertex $u \in V[H] - s$; i.e.,

$$\begin{aligned} c_{\overline{(H|^K)}_\delta}(s,u) &= \max\{K - k_u^* - \delta, 0\} \\ &= \max\{k - k_u^*, 0\}, u \in V[H] - s. \end{aligned}$$

Therefore, $\overline{(H|^K)}_\delta$ satisfies

$$c_{\overline{(H|^K)}_\delta}(e) \leq \pi(R_H(u)|^k) = c_{H|^k}(e)$$

$$\text{for } e = (s,u) \in E_H(s),$$

$$c_{\overline{(H|^K)}_\delta}(e) \leq c_H(e) = c_{H|^k}(e)$$

$$\text{for } e \in E[H] - E_H(s).$$

By Lemma 3,

$$\lambda_{\overline{(H|^K)}_\delta}(v_{n'-1}, v_{n'}) = d_H(v_{n'}) - \delta.$$

From these, we have

$$\begin{aligned} \lambda_{H|^k}(v_{n'-1}, v_{n'}) &\geq \lambda_{\overline{(H|^K)}_\delta}(v_{n'-1}, v_{n'}) \\ &= d_H(v_{n'}) - \delta \\ &= K - \delta \quad \text{(by (9))} \\ &= k, \end{aligned}$$

proving (i).

(ii) We first remove redundant (parts of) ranges from each $R_H(u)$, $u \in V[H] - s$ so that (9) holds. Let H' be the resulting ranged graph. We then obtain an MA-ordering of $H'|^K$ starting from s. The last two vertices $v = v_{n'-1}$ and $w = v_{n'}$ in the ordering satisfy (i) from the above proof. By Lemma 2(i), this can be done in $O(|E[H]| + |V[H]|\log|V[H]|)$ time □

References

[1] G.-R. Cai and Y.-G. Sun, *The minimum augmentation of any graph to k-edge-connected graph*, Networks, 19, 1989, 151-172.

[2] A. Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM J. Disc. Math., 5, 1992, 25-53.

[3] A. Frank, T. Ibaraki and H. Nagamochi, *On sparse subgraphs preserving connectivity properties*, J. Graph Theory, 17, 1993, 275-281.

[4] H.N. Gabow, *Efficient splitting off algorithms for graphs*, STOC, 1994, 696–705.

[5] L. Lovász, Combinatorial Problems and Exercises, North-Holland 1979.

[6] H. Nagamochi and T. Ibaraki, *Deterministic $\tilde{O}(nm)$ time edge-splitting in undirected graphs*, STOC, 1996, 64-73.

[7] H. Nagamochi and T. Ibaraki, *A simple and constructive proof of a minimum cut algorithm*, Tech. Rep. #96001, Kyoto University, Dept. of Applied Mathematics and Physics, 1996.

[8] M. Stoer and F. Wagner, *A simple min cut algorithm*, Lecture Notes in Computer Science 855, Springer-Verlag, 1994, 141–147.

[9] T. Watanabe and A. Nakamura, *Edge-connectivity augmentation problems*, J. Comp. System Sci., 35, 1987, 96-144.