すべての目標値に対し辺連結度増加問題を$\tilde{O}(mn)$時間で解くアルゴリズム

永持 仁， 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 $G=(V,E,c_G)$を辺に非負の実数値重みを持つ無向グラフとする。Gの各点対間に重みを付加してグラフの辺連結度を指定された目標値kに増加させる問題を考える。このとき、新たに加える重みの総量は最小にするものとする。このGの辺連結度をkに増加させるために付加すべき重みの最小必要量を$\Lambda_G(k)$と記し、$G^*(k)$で最適に辺連結度をkにさせた（1つの）グラフを表すとする。最近、与えられたグラフGに対する関数$\Lambda_G(k)$, $k\in[0,+\infty]$を決定する$O(nm+n^2\log n)$時間のアルゴリズムが提案された。ここで、$n=|V|$, $m=|E|$. 本報告では、関数Λ_Gを計算するために用いられたデータに基づけば、すべての$k\in[0,+\infty]$に対する最適解$G^*(k)$を$O(n\log n)$個の閉路として表現でき、そのような閉路集合を$O(nm+n^2\log n)$時間で計算できることを示す。

Augmenting Edge-Connectivity over the Entire Range in $\tilde{O}(nm)$ Time

Hiroshi NAGAMOCHI 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, and $G^*(k)$ be the resulting graph obtained from G. Recently, it is shown that function Λ_G over the entire range $k\in[0,+\infty]$ can be computed in $O(nm+n^2\log n)$ time, where n and m are the numbers of vertices and edges, respectively. This paper shows that all $G^*(k)$ in the entire range can be obtained from $O(n\log n)$ weighted cycles, and such cycles can be computed in $O(nm+n^2\log n)$ time.

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 $\lambda_G(V) = 7$.

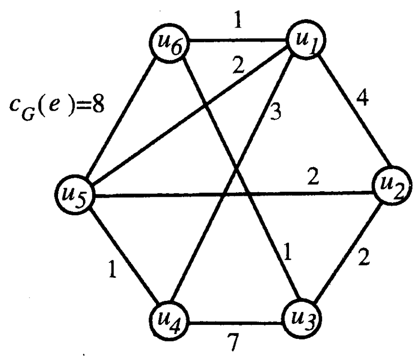

Figure 1: An edge-weighted graph G.

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.

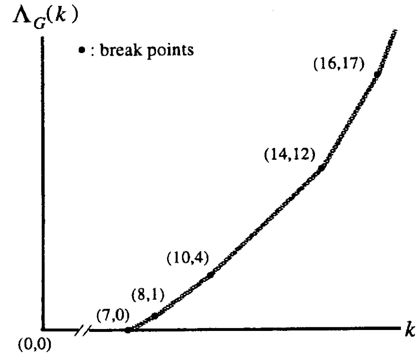

Figure 2: Edge connectivity augmentation function $\Lambda_G(k)$ of G in Fig. 1.

Watanabe and Nakamura [8] 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] 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 [3] improved it to $O(mn^2 \log(n^2/m))$. Recently, Nagamochi and Ibaraki [5] 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 [5] 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.* □

Recently, [6] reported the following result 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.* □

To show the above results, they modified the $O(n(m + n \log n))$ time algorithm in [5] 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 $k \geq 0$ (they do not rely on any parametric search technique of mathematical programming).

This paper presents how to construct graph $G^*(k)$ which has edge-connectivity k and is obtained from G by adding new $\Lambda_G(k)$ weights. We shows that all $G^*(k)$ in the entire range $k \in [\lambda_G(V), +\infty]$ can be compactly represented by G and a set of $O(n \log n)$ cycles $\Pi = (C_1^*, C_2^*, \ldots, C_p^*)$, $(p = O(n \log n))$ on V, each C_i^* of which has a range $[\lambda_{i-1}, \lambda_i]$ $i = 1, 2, \ldots, p$ such that $\lambda_G(V) = \lambda_0 < \lambda_1 < \cdots < \lambda_{p-1} < \lambda_p = +\infty$: $G^*(k)$ for a $k \geq \lambda_G(V)$ is obtained from G by increasing the weights of edges in C_i^* by $(\lambda_i - \lambda_{i-1})/2$ for $i = 1, 2, \ldots, i_k$ and weight of edges in $C_{i_k+1}^*$ uniformly by $(k - \lambda_{i_k})/2$, where i_k is the largest index i such that $\lambda_i < k$. This hierarchical structure of optimal solutions over all k is known so far only for the integer version of the edge-connectivity augmentation problem [7].

Theorem 2 *Given an edge-weighted graph $G = (V, E, c_G)$, there is a set*

$$\Pi = \{(C_i^*, [\lambda_{i-1}, \lambda_i]) \mid i = 1, 2, \ldots, p\}$$

of $p \leq 8n \log n$ weighted cycles that represent all optimal graphs $G^*(k)$ in the entire range $k \in [\lambda_G(V), +\infty]$. Such Π can be computed in $O(mn + n^2 \log n)$ time. □

Our new algorithm runs faster by factor of $O(\log n)$ than the previously fastest algorithm [5] for the augmentation problem for a single fixed k.

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 $u \neq v$ 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 $X - Z \neq \emptyset \neq Z - X$. 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 $\{x, y\}$).

Let us review 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 2 [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 $\mathcal{X}$.* □

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

This section briefly reviews how to compute $\Lambda_G(k)$ over the entire range $k \in [0, +\infty]$. Since it is known in [6] that all break points of Λ_G occur in the range $[0, 2\max_{v \in V} d_G(v)]$, we only consider reals k such that $0 \leq k \leq K$, where

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

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.

3.1 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 $c_{G'} : E' \rightarrow \boldsymbol{R}^+$, 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 $[b^* - \pi(R), b^*]$.

3.2 Totally optimal ranged graph

We now extend the optimality conditions in Lemma 2 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 2, we can easily compute $\Lambda_G(k)$ for any $k \in \boldsymbol{R}^+$ by

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

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 $G'|^k$.

3.3 Algorithm SIMUL-AUGMENT

Given a graph $G = (V, E, c_G)$, the following algorithm SIMUL-AUGMENT [6] constructs 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'.

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
  V' := V ∪ {s}; E'(s) = {(s,v) | v ∈ V};
  K := 1 + 2 max_{v∈V} d_G(v);   X := ∅;
  for each vertex u ∈ V do
     R_{G'}(u) := {[d_G(u), K]};
     X := X ∪ {({u}, [d_G(u), K])}
  end; { for }
  Let G' = (V', E' = E ∪ E'(s), c_G, R_{G'})
  be the obtained ranged graph;
  H := G';
  while |V[H]| ≥ 4 do
     Find vertices v, w ∈ V[H] − s such that
     λ_{H|^k}(v, w) ≥ k, 0 ≤ k ≤ K;
     Contract v and w into a vertex x*;
     R_H(x*) := R_H(v) ∪ R_H(w);
     { Assume R_H(x*) = {[a_1, K], [a_2, K],
        ..., [a_r, K]}, where a_1 ≤ ··· ≤ 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 $\mathcal{X} := \{(X, [a, b]) \in \mathcal{X} \mid X \subset V - X^*\}$
$\cup \{(X^*, [d_G(X^*), k'])\}$
$\cup \{(X, [\max\{a, k'\}, b]) \mid$
$(X, [a, b]) \in \mathcal{X},\ X \subset X^*,\ k' < b\}$;

- -

22 **end**; { if }
23 Denote the ranged graphs resulting from H and G', respectively, as H and G' again
24 **end**; { while }
25 Output G', $\mathcal{X}$ and $R^* = R_H(V[H] - s)$
26 **end**. { SIMUL-AUGMENT }

4 All Optimal Solutions over the Entire Range

From the discussion given so far, an optimally augmented graph $G^*(k)$ with edge-connectivity k can be constructed once a totally optimal ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$ is given. Such G' can for example be obtained by Algorithm SIMUL-AUGMENT with lines 18-20 (line 23 is necessary only to compute a totally critical covering ranged collection $\mathcal{X}$). For the above example example, we have $R_{G'}(u_1) = \{[10, 27]\}$, $R_{G'}(u_2) = \{[8, 27]\}$, $R_{G'}(u_3) = \{[10, 27]\}$, $R_{G'}(u_4) = \{[7, 10], [14, 27]\}$, $R_{G'}(u_5) = \{[7, 10], [16, 27]\}$, $R_{G'}(u_6) = \{[10, 27]\}$.

We first show the next. For a range $r = [a, b]$, a (resp., b) is called the *bottom* (resp., *top*) of r, denoted by $bot(r)$ (resp., $top(r)$).

Lemma 3 *Aligning $A = \cup_{u \in X^*} A_u$ in line 16 can be carried out in $O(|R|)$ time, where R is the set $R_{G'}(V)$ of ranges obtained after line 17.*

Proof: During execution of SIMUL-AUGMENT, we maintain a list $List[R_{G'}(V)]$ of the ranges in $R_{G'}(V)$, where the ranges are arranged in the nondecreasing order of their tops, and each element r in the list has two data, its bottom $bot(r)$ and the vertex u_r with $r \in R_{G'}(u_r)$. To compute $A := \cup_{u \in V} A_u := \cup_{u \in V} R_{G'}(u)|^{k'}$, we first remove ranges $[a_u, b_u] \in R_{G'}(u)$ with $a < k < b$ (if any) fro all $u \in X^*$ from $List[R_{G'}(V)]$ by traversing the list, and add ranges $[a_u, k'], [k', b]$ for those $u \in X^*$ to the resulting list. This can be carried out in $O(|R_{G'}(V)| + |X^*| \log |X^*|)$ time. We then divide the list $List[R_{G'}(V)]$ into two lists $List[R_{G'}(X^*)|^{k'}]$ and $List[R_{G'}(X^*)|_{k'} \cup R_{G'}(V - X^*)]$, where each of these lists is sorted with respect to the tops of ranges. This can be done in $O(|R_{G'}(V)|)$ time. By traversing the list $List[R_{G'}(X^*)|^{k'}]$, we can obtain two sorted lists $List[R_{G'}(u)|^{k'}]$ and $List[R_{G'}(u)|_{k'}]$ in $O(|R_{G'}(V)|)$ time. Based on the list $List[R_{G'}(u)|^{k'}]$, we can align $A = \cup_{u \in X^*} A_u = R_{G'}(u)|^{k'}$ and obtain a sorted list $List[A']$ of the resulting set A' of ranges. in $O(|A|)$ time. The list for the resulting entire set $R := A' \cup R_{G'}(X^*)|_{k'} \cup List[R_{G'}(V - X^*)$ can be updated in $O(|R|)$ time by merging all these sorted lists into a single list. Therefore, aligning $A = \cup_{u \in X^*} A_u$ in line 16 can be done in $O(|R|)$ time. □

If we take this approach, the entire running time of Algorithm SIMUL-AUGMENT (with lines 18-20) becomes $O(mn + n^2 \log n + nr_{max}) = O(mn + n^3)$ time, where r_{max} denote the maximum number of ranges in $R_{G'}(V)$ attained during execution of SIMUL-AUGMENT. Note that $r_{max} \leq n^2$ obviously holds since at most $|X^*|$ new ranges in $R_{G'}(V)$ are created in each iteration of the while-loop.

However, we can modify the alignment operation in lines 18-20 so that $r_{max} = O(n \log n)$ holds, which improves the above running time of SIMUL-AUGMENT as stated in the next theorem.

Theorem 3 *There is a totally optimal ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$ such that $|R_{G'}(V)| \leq (3n - 1) + (2n - 3)\log_2(n - 1)$, and such G' can be obtained in $O(mn + n^2 \log n)$ time.*

Proof: See Appendix for a proof sketch. □

Our next step is to consider how to compute an optimally augmented graphs $G^*(k)$ for the entire range of k. Given a totally optimal ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$ and a fixed $k \geq \lambda_G(V)$, our discussion so far tells that graph $G'_k = (V \cup \{s\}, E \cup E'(s), c_{G'_k})$ in Lemma 2 is computed by setting $G'_k = G'|^k$. Then by Lovász's edge-splitting theorem, graph $G^*(k)$ can be obtained from G'_k by splitting off the edges incident to s. Since an $O(n(m + n \log n) \log n)$ time edge-splitting algorithm is known [5], an optimal solution for each fixed k can be obtained $O(nm + n^2 \log^2 n) + O(n(m + n \log n) \log n) = O(mn \log n + n^2 \log^2 n)$ time. However, this requires to invoke the edge-splitting algorithm for all k, and provides no structural information of optimal solutions in the entire range of k.

In this section, we show that optimal solutions for all k can be represented by a set of $p = O(n \log n)$ cycles $C_1^*, C_2^*, \ldots, C_p^*$ on V, each C_i^* of which has a range $[\lambda_{i-1}, \lambda_i]$ $i = 1, 2, \ldots, p$ such that $\lambda_G(V) = \lambda_0 < \lambda_1 < \cdots < \lambda_{p-1} < \lambda_p = +\infty$; optimally augmented $G^*(k)$ for each k is obtained from G by increasing the weight of the edges in C_i^* by

$(\lambda_i - \lambda_{i-1})/2$ for $i = 1, 2, \ldots, i_k$ and weight of edges in $C^*_{i_k+1}$ by $(k - \lambda_{i_k})/2$, where i_k is the largest index i such that $\lambda_i < k$.

Let $R_{G'}(V)$ be a range set of an totally optimal ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$, where we assume without loss of generality that $R_{G'}(V)$ contains no range $[a, b]$ with $a = b$. Let $\{\lambda_i \mid i = 0, 1, \ldots, p\} = \{top(r),\ bot(r) \mid r \in R_{G'}(V)\}$, where $\lambda_i < \lambda_{i+1}$, $i = 0, 1, \ldots, p-1$. For example, we have $\lambda_0 = 7$, $\lambda_1 = 8$, $\lambda_2 = 10$, $\lambda_3 = 14$, $\lambda_4 = 16$ and $\lambda_5 = 27$ in our running example. By Theorem 3, we can assume

$$p \leq 6n + 4n \log_2 n.$$

Without loss of generality, we also assume that each $R_{G'}(v)$ contains no range $[a, b]$ with $a < \lambda_i < b$ for any λ_i; if necessary, split each $[a, b]$ with $a < \lambda_i < b$ into $[a, \lambda_i]$ and $[\lambda_i, b]$, which may increase the number of ranges in $R_{G'}(V)$, but does not change the above p. Then, we consider a sequence of $p+1$ graphs $G'|^{\lambda_i}$, $i = 0, 1, \ldots, p$. For each $i = 0, 1, \ldots, p-1$, let $G^*_i = (V, E^*_i, c_{G^*_i})$ denote a graph obtained from $G'|^{\lambda_i}$ by splitting off the edges incident to s and removing the designated vertex s. We can assume $\lambda_{G^*_i}(V) \geq k$ by Lovász's edge-splitting theorem. In fact, $\lambda_{G^*_i}(V) = \lambda_i$ holds for all i, since $G'|^{\lambda_i}$ has a (λ_i, s)-critical covering collection.

Then G^*_i, $i = 0, 1, \ldots, p$ satisfy $G^*_i = G^*(k)$ for $k = \lambda_i$. We now show that such G^*_i can be easily obtained from G^*_{i-1} without really applying splitting algorithms, and that all G^*_i can be characterized by cycles C^*_i as noted above. In other words, a totally optimal ranged graph G' contains all the information necessary to construct $G^*(k)$ for the entire range of k.

We first show an important property of G^*_i. Let $V_1, V_2, \ldots, V_h$ be different subsets of V in G^*_i such that each V_j, $j = 1, 2, \ldots, h$ satisfies

$$\lambda_{G^*_i}(u, v) > \lambda_i \text{ for all } u \neq v \in V_j, \text{ or } |V_j| = 1,$$

and is maximal (with respect to $|V_j|$) subject to this property. By $\lambda_{G^*_i}(V) = \lambda_i$, we have $h \geq 2$. Each V_j is called a *λ_i-component* of G^*_i, and a λ_i-component V_j is called a *λ_i-leaf* if $d_{G^*_i}(V_j) = \lambda_i$ (i.e., V_j itself is a minimum cut in G^*_i). From definitions, we easily see that the following properties hold:

(i) The set of λ_i-components is a partition of V.

(ii) For any two λ_i-components V_ℓ and V_j, G^*_i has a minimum cut $X_{\ell,j}$ which separates V_ℓ and V_j.

(iii) Every minimum cut X in G^*_i contains a λ_i-leaf $V_j \subseteq X$, since any minimum cut $X' \subseteq X$ with the minimal cardinality $|X'|$ is a λ_i-leaf of G^*_i.

Lemma 4 *Let V_j be a λ_i-component in G^*_i. Then $d_{G^*_i}(X) = d_{G'|^{\lambda_i}}(X)$ holds for any cut $X \subseteq V_j$.*

Proof: Let ΔE^*_i denotes the set of edges e whose weights have been increased by the edge splitting operation to obtain G^*_i from $G'|^{\lambda_i}$, i.e.,

$$\Delta E^*_i = \{e \in E \mid c_{G^*_i}(e) > c_G(e)\} \cup \{e \in E^*_i - E \mid c_{G^*_i}(e) > 0\}$$

(note that $c_G(e) = c_{G'|^{\lambda_i}}(e)$, $e \in E$). Then it is sufficient to show that ΔE^*_i contains no edge (u, v) such that the both end points u and v belong to some V_j. Since there exists a critical covering collection $\mathcal{X}|^{\lambda_i}$ in $G'|^{\lambda_i}$, the end points u and v of any edge $e = (u, v)$ in ΔE^*_i must belong to different critical cuts $X', X'' \in \mathcal{X}|^{\lambda_i}$, respectively, in order to make G^*_i λ_i-edge-connected. That means that $\lambda_{G^*_i}(u, v) = \lambda_i$ holds for each $e = (u, v) \in \Delta E^*_i$. Therefore, the lemma follows. □

We next introduce another sequence of graphs, which are used to construct G^*_i for all i. Let $\tilde{G}_{i+1}$, $i = 0, 1, \ldots, p-1$ denote the graph obtained from G^*_i by putting back vertex s and edges (s, v) with weights $\pi(R_{G'}(v)|^{\lambda_{i+1}}_{\lambda_i})$ for all $v \in V$; i.e., $\tilde{G}_{i+1} = (V \cup \{s\}, E^*_i \cup E'(s), c_{\tilde{G}_{i+1}})$ with $E'(s) = \{(s, v) \mid v \in V\}$ and

$$c_{\tilde{G}_{i+1}}(e) = \begin{cases} \lambda_{i+1} - \lambda_i & \text{if } e = (s, v) \in E'(s) \text{ and } R_{G'}|^{\lambda_{i+1}}_{\lambda_i}(v) \neq \emptyset \\ 0 & \text{if } e = (s, v) \in E'(s) \text{ and } R_{G'}|^{\lambda_{i+1}}_{\lambda_i}(v) = \emptyset \\ c_{G^*_i}(e) & \text{if } e \in E^*_i. \end{cases}$$

In other words, $\tilde{G}_{i+1}$ is a graph obtained from $G'|^{\lambda_{i+1}}$ by splitting those edges (s, u), $u \in V$ with weight $R_{G'}(u)|^{\lambda_i}$ at s, leaving the remaining weight $R_{G'}(u)|^{\lambda_{i+1}}_{\lambda_i}$ in each edge (s, u). The next lemma claims that the graph $\tilde{G}^*_{i+1}$ defined in the above maintains the optimality of $G'|^{\lambda_{i+1}}$.

Lemma 5 *For each $i = 0, 1, \ldots, p-1$, if $\lambda_{G^*_i}(V) \geq \lambda_i$, then $\lambda_{\tilde{G}_{i+1}}(V) \geq \lambda_{i+1}$.*

Proof: Consider the set $\Gamma_{\tilde{G}_{i+1}}(s) (\neq \emptyset)$ of the neighbors of s (vertices adjacent to s by an edge with a positive weight) in $\tilde{G}_{i+1}$. From definitions of $\tilde{G}_{i+1}$ and G^*_i, for any cut $X \subset V$,

$$d_{\tilde{G}_{i+1}}(X) = d_{G^*_i}(X) + |X \cap \Gamma_{\tilde{G}_{i+1}}(s)| \cdot (\lambda_{i+1} - \lambda_i)$$

holds. Then, if $X \cap \Gamma_{\tilde{G}_{i+1}}(s) \neq \emptyset$, then $d_{\tilde{G}_{i+1}}(X) \geq \lambda_{i+1}$ holds by $\lambda_{G^*_i}(V) \geq \lambda_i$. Therefore, we assume that $\tilde{G}_{i+1}$ has a cut Z such that

$$Z \subseteq V - \Gamma_{\tilde{G}_{i+1}}(s) \text{ and } d_{\tilde{G}_{i+1}}(Z)\ (= d_{G^*_i}(Z)) < \lambda_{i+1}, \quad (3)$$

and derive a contradiction.

We first show that cut Z in (3) is not contained in any λ_i-component V_j. If $Z \subseteq V_j$ for some λ_i-component V_j, then $d_{G^*_i}(Z) = d_{G'|^{\lambda_i}}(Z)$ by Lemma 4, and hence $d_{\tilde{G}_{i+1}}(Z) = d_{G^*_i}(Z) = d_{G'|^{\lambda_i}}(Z) = d_{G'|^{\lambda_{i+1}}}(Z)$ by $Z \cap \Gamma_{\tilde{G}_{i+1}}(s) = \emptyset$. However, $d_{G'|^{\lambda_{i+1}}}(Z) = d_{\tilde{G}_{i+1}}(Z) < \lambda_{i+1}$ would contradict that the totally optimal ranged graph G' satisfies $\lambda_{G'|^{\lambda_{i+1}}}(V) \geq \lambda_{i+1}$. Hence, $Z \not\subseteq V_i$ for any λ_i-component V_j.

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 $\mathcal{X} := \{(X, [a, b]) \in \mathcal{X} \mid X \subset V - X^*\}$
$\cup \{(X^*, [d_G(X^*), k'])\}$
$\cup \{(X, [\max\{a, k'\}, b]) \mid (X, [a, b]) \in \mathcal{X},\ X \subset X^*,\ k' < b\}$;
\- -
22 **end**; { if }
23 Denote the ranged graphs resulting from H and G', respectively, as H and G' again
24 **end**; { while }
25 Output G', $\mathcal{X}$ and $R^* = R_H(V[H] - s)$
26 **end**. { SIMUL-AUGMENT }

4 All Optimal Solutions over the Entire Range

From the discussion given so far, an optimally augmented graph $G^*(k)$ with edge-connectivity k can be constructed once a totally optimal ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$ is given. Such G' can for example be obtained by Algorithm SIMUL-AUGMENT with lines 18-20 (line 23 is necessary only to compute a totally critical covering ranged collection $\mathcal{X}$). For the above example example, we have $R_{G'}(u_1) = \{[10, 27]\}$, $R_{G'}(u_2) = \{[8, 27]\}$, $R_{G'}(u_3) = \{[10, 27]\}$, $R_{G'}(u_4) = \{[7, 10], [14, 27]\}$, $R_{G'}(u_5) = \{[7, 10], [16, 27]\}$, $R_{G'}(u_6) = \{[10, 27]\}$.

We first show the next. For a range $r = [a, b]$, a (resp., b) is called the *bottom* (resp., *top*) of r, denoted by $bot(r)$ (resp., $top(r)$).

Lemma 3 *Aligning $A = \cup_{u \in X^*} A_u$ in line 16 can be carried out in $O(|R|)$ time, where R is the set $R_{G'}(V)$ of ranges obtained after line 17.*

Proof: During execution of SIMUL-AUGMENT, we maintain a list $List[R_{G'}(V)]$ of the ranges in $R_{G'}(V)$, where the ranges are arranged in the nondecreasing order of their tops, and each element r in the list has two data, its bottom $bot(r)$ and the vertex u_r with $r \in R_{G'}(u_r)$. To compute $A := \cup_{u \in V} A_u := \cup_{u \in V} R_{G'}(u)|^{k'}$, we first remove ranges $[a_u, b_u] \in R_{G'}(u)$ with $a < k < b$ (if any) fro all $u \in X^*$ from $List[R_{G'}(V)]$ by traversing the list, and add ranges $[a_u, k'], [k', b]$ for those $u \in X^*$ to the resulting list. This can be carried out in $O(|R_{G'}(V)| + |X^*| \log |X^*|)$ time. We then divide the list $List[R_{G'}(V)]$ into two lists $List[R_{G'}(X^*)|^{k'}]$ and $List[R_{G'}(X^*)|_{k'} \cup R_{G'}(V - X^*)]$, where each of these lists is sorted with respect to the tops of ranges. This can be done in $O(|R_{G'}(V)|)$ time. By traversing the list $List[R_{G'}(X^*)|^{k'}]$, we can obtain two sorted lists $List[R_{G'}(u)|^{k'}]$ and $List[R_{G'}(u)|_{k'}]$ in $O(|R_{G'}(V)|)$ time. Based on the list $List[R_{G'}(u)|^{k'}]$, we can align $A = \cup_{u \in X^*} A_u = R_{G'}(u)|^{k'}$ and obtain a sorted list $List[A']$ of the resulting set A' of ranges. in $O(|A|)$ time. The list for the resulting entire set $R := A' \cup R_{G'}(X^*)|_{k'} \cup List[R_{G'}(V - X^*)$ can be updated in $O(|R|)$ time by merging all these sorted lists into a single list. Therefore, aligning $A = \cup_{u \in X^*} A_u$ in line 16 can be done in $O(|R|)$ time. □

If we take this approach, the entire running time of Algorithm SIMUL-AUGMENT (with lines 18-20) becomes $O(mn + n^2 \log n + nr_{max}) = O(mn + n^3)$ time, where r_{max} denote the maximum number of ranges in $R_{G'}(V)$ attained during execution of SIMUL-AUGMENT. Note that $r_{max} \leq n^2$ obviously holds since at most $|X^*|$ new ranges in $R_{G'}(V)$ are created in each iteration of the while-loop.

However, we can modify the alignment operation in lines 18-20 so that $r_{max} = O(n \log n)$ holds, which improves the above running time of SIMUL-AUGMENT as stated in the next theorem.

Theorem 3 *There is a totally optimal ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$ such that $|R_{G'}(V)| \leq (3n - 1) + (2n - 3) \log_2(n - 1)$, and such G' can be obtained in $O(mn + n^2 \log n)$ time.*

Proof: See Appendix for a proof sketch. □

Our next step is to consider how to compute an optimally augmented graphs $G^*(k)$ for the entire range of k. Given a totally optimal ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$ and a fixed $k \geq \lambda_G(V)$, our discussion so far tells that graph $G'_k = (V \cup \{s\}, E \cup E'(s), c_{G'_k})$ in Lemma 2 is computed by setting $G'_k = G'|^k$. Then by Lovász's edge-splitting theorem, graph $G^*(k)$ can be obtained from G'_k by splitting off the edges incident to s. Since an $O(n(m + n \log n) \log n)$ time edge-splitting algorithm is known [5], an optimal solution for each fixed k can be obtained $O(nm + n^2 \log^2 n) + O(n(m + n \log n) \log n) = O(mn \log n + n^2 \log^2 n)$ time. However, this requires to invoke the edge-splitting algorithm for all k, and provides no structural information of optimal solutions in the entire range of k.

In this section, we show that optimal solutions for all k can be represented by a set of $p = O(n \log n)$ cycles $C_1^*, C_2^*, \ldots, C_p^*$ on V, each C_i^* of which has a range $[\lambda_{i-1}, \lambda_i]$ $i = 1, 2, \ldots, p$ such that $\lambda_G(V) = \lambda_0 < \lambda_1 < \cdots < \lambda_{p-1} < \lambda_p = +\infty$; optimally augmented $G^*(k)$ for each k is obtained from G by increasing the weight of the edges in C_i^* by

$(\lambda_i - \lambda_{i-1})/2$ for $i = 1, 2, \ldots, i_k$ and weight of edges in $C^*_{i_k+1}$ by $(k - \lambda_{i_k})/2$, where i_k is the largest index i such that $\lambda_i < k$.

Let $R_{G'}(V)$ be a range set of an totally optimal ranged graph $G' = (V \cup \{s\}, E \cup E'(s), c_G, R_{G'})$, where we assume without loss of generality that $R_{G'}(V)$ contains no range $[a, b]$ with $a = b$. Let $\{\lambda_i \mid i = 0, 1, \ldots, p\} = \{top(r),\ bot(r) \mid r \in R_{G'}(V)\}$, where $\lambda_i < \lambda_{i+1}$, $i = 0, 1, \ldots, p-1$. For example, we have $\lambda_0 = 7$, $\lambda_1 = 8$, $\lambda_2 = 10$, $\lambda_3 = 14$, $\lambda_4 = 16$ and $\lambda_5 = 27$ in our running example. By Theorem 3, we can assume

$$p \leq 6n + 4n \log_2 n.$$

Without loss of generality, we also assume that each $R_{G'}(v)$ contains no range $[a, b]$ with $a < \lambda_i < b$ for any λ_i; if necessary, split each $[a, b]$ with $a < \lambda_i < b$ into $[a, \lambda_i]$ and $[\lambda_i, b]$, which may increase the number of ranges in $R_{G'}(V)$, but does not change the above p. Then, we consider a sequence of $p+1$ graphs $G'|^{\lambda_i}$, $i = 0, 1, \ldots, p$. For each $i = 0, 1, \ldots, p-1$, let $G^*_i = (V, E^*_i, c_{G^*_i})$ denote a graph obtained from $G'|^{\lambda_i}$ by splitting off the edges incident to s and removing the designated vertex s. We can assume $\lambda_{G^*_i}(V) \geq k$ by Lovász's edge-splitting theorem. In fact, $\lambda_{G^*_i}(V) = \lambda_i$ holds for all i, since $G'|^{\lambda_i}$ has a (λ_i, s)-critical covering collection.

Then G^*_i, $i = 0, 1, \ldots, p$ satisfy $G^*_i = G^*(k)$ for $k = \lambda_i$. We now show that such G^*_i can be easily obtained from G^*_{i-1} without really applying splitting algorithms, and that all G^*_i can be characterized by cycles C^*_i as noted above. In other words, a totally optimal ranged graph G' contains all the information necessary to construct $G^*(k)$ for the entire range of k.

We first show an important property of G^*_i. Let $V_1, V_2, \ldots, V_h$ be different subsets of V in G^*_i such that each V_j, $j = 1, 2, \ldots, h$ satisfies

$$\lambda_{G^*_i}(u, v) > \lambda_i \text{ for all } u \neq v \in V_j, \text{ or } |V_j| = 1,$$

and is maximal (with respect to $|V_j|$) subject to this property. By $\lambda_{G^*_i}(V) = \lambda_i$, we have $h \geq 2$. Each V_j is called a λ_i-*component* of G^*_i, and a λ_i-component V_j is called a λ_i-*leaf* if $d_{G^*_i}(V_j) = \lambda_i$ (i.e., V_j itself is a minimum cut in G^*_i). From definitions, we easily see that the following properties hold:

(i) The set of λ_i-components is a partition of V.

(ii) For any two λ_i-components V_ℓ and V_j, G^*_i has a minimum cut $X_{\ell,j}$ which separates V_ℓ and V_j.

(iii) Every minimum cut X in G^*_i contains a λ_i-leaf $V_j \subseteq X$, since any minimum cut $X' \subseteq X$ with the minimal cardinality $|X'|$ is a λ_i-leaf of G^*_i.

Lemma 4 *Let V_j be a λ_i-component in G^*_i. Then $d_{G^*_i}(X) = d_{G'|^{\lambda_i}}(X)$ holds for any cut $X \subseteq V_j$.*

Proof: Let ΔE^*_i denotes the set of edges e whose weights have been increased by the edge splitting operation to obtain G^*_i from $G'|^{\lambda_i}$, i.e.,

$$\Delta E^*_i = \{e \in E \mid c_{G^*_i}(e) > c_G(e)\} \cup \{e \in E^*_i - E \mid c_{G^*_i}(e) > 0\}$$

(note that $c_G(e) = c_{G'|^{\lambda_i}}(e)$, $e \in E$). Then it is sufficient to show that ΔE^*_i contains no edge (u, v) such that the both end points u and v belong to some V_j. Since there exists a critical covering collection $\mathcal{X}|^{\lambda_i}$ in $G'|^{\lambda_i}$, the end points u and v of any edge $e = (u, v)$ in ΔE^*_i must belong to different critical cuts $X', X'' \in \mathcal{X}|^{\lambda_i}$, respectively, in order to make G^*_i λ_i-edge-connected. That means that $\lambda_{G^*_i}(u, v) = \lambda_i$ holds for each $e = (u, v) \in \Delta E^*_i$. Therefore, the lemma follows. □

We next introduce another sequence of graphs, which are used to construct G^*_i for all i. Let $\tilde{G}_{i+1}$, $i = 0, 1, \ldots, p-1$ denote the graph obtained from G^*_i by putting back vertex s and edges (s, v) with weights $\pi(R_{G'}(v)|^{\lambda_{i+1}}_{\lambda_i})$ for all $v \in V$; i.e., $\tilde{G}_{i+1} = (V \cup \{s\}, E^*_i \cup E'(s), c_{\tilde{G}_{i+1}})$ with $E'(s) = \{(s, v) \mid v \in V\}$ and

$$c_{\tilde{G}_{i+1}}(e) = \begin{cases} \lambda_{i+1} - \lambda_i & \text{if } e = (s, v) \in E'(s) \text{ and } R_{G'}|^{\lambda_{i+1}}_{\lambda_i}(v) \neq \emptyset \\ 0 & \text{if } e = (s, v) \in E'(s) \text{ and } R_{G'}|^{\lambda_{i+1}}_{\lambda_i}(v) = \emptyset \\ c_{G^*_i}(e) & \text{if } e \in E^*_i. \end{cases}$$

In other words, $\tilde{G}_{i+1}$ is a graph obtained from $G'|^{\lambda_{i+1}}$ by splitting those edges (s, u), $u \in V$ with weight $R_{G'}(u)|^{\lambda_i}$ at s, leaving the remaining weight $R_{G'}(u)|^{\lambda_{i+1}}_{\lambda_i}$ in each edge (s, u). The next lemma claims that the graph $\tilde{G}^*_{i+1}$ defined in the above maintains the optimality of $G'|^{\lambda_{i+1}}$.

Lemma 5 *For each $i = 0, 1, \ldots, p-1$, if $\lambda_{G^*_i}(V) \geq \lambda_i$, then $\lambda_{\tilde{G}_{i+1}}(V) \geq \lambda_{i+1}$.*

Proof: Consider the set $\Gamma_{\tilde{G}_{i+1}}(s) (\neq \emptyset)$ of the neighbors of s (vertices adjacent to s by an edge with a positive weight) in $\tilde{G}_{i+1}$. From definitions of $\tilde{G}_{i+1}$ and G^*_i, for any cut $X \subset V$,

$$d_{\tilde{G}_{i+1}}(X) = d_{G^*_i}(X) + |X \cap \Gamma_{\tilde{G}_{i+1}}(s)| \cdot (\lambda_{i+1} - \lambda_i)$$

holds. Then, if $X \cap \Gamma_{\tilde{G}_{i+1}}(s) \neq \emptyset$, then $d_{\tilde{G}_{i+1}}(X) \geq \lambda_{i+1}$ holds by $\lambda_{G^*_i}(V) \geq \lambda_i$. Therefore, we assume that $\tilde{G}_{i+1}$ has a cut Z such that

$$Z \subseteq V - \Gamma_{\tilde{G}_{i+1}}(s) \text{ and } d_{\tilde{G}_{i+1}}(Z)\ (= d_{G^*_i}(Z)) < \lambda_{i+1}, \quad (3)$$

and derive a contradiction.

We first show that cut Z in (3) is not contained in any λ_i-component V_j. If $Z \subseteq V_j$ for some λ_i-component V_j, then $d_{G^*_i}(Z) = d_{G'|^{\lambda_i}}(Z)$ by Lemma 4, and hence $d_{\tilde{G}_{i+1}}(Z) = d_{G^*_i}(Z) = d_{G'|^{\lambda_i}}(Z) = d_{G'|^{\lambda_{i+1}}}(Z)$ by $Z \cap \Gamma_{\tilde{G}_{i+1}}(s) = \emptyset$. However, $d_{G'|^{\lambda_{i+1}}}(Z) = d_{\tilde{G}_{i+1}}(Z) < \lambda_{i+1}$ would contradict that the totally optimal ranged graph G' satisfies $\lambda_{G'|^{\lambda_{i+1}}}(V) \geq \lambda_{i+1}$. Hence, $Z \not\subseteq V_i$ for any λ_i-component V_j.

Now choose a cut Z with the minimum $d_{G_i^*}(Z)$ among all those cuts Z in (3), and assume furthermore that Z has a minimal cardinality $|Z|$ with this property. Then $d_{G_i^*}(Z) < d_{G_i^*}(Z')$ holds for any nonempty and proper subset Z' of Z. Since Z is not contained in any λ_i-component, there are at least two λ_i-components V_j and V_ℓ such that $V_j \cap Z \neq \emptyset \neq V_\ell \cap Z$. By the property (ii) of λ_i-components, G_i^* has a cut $X \subset V$ with $d_{G_i^*}(X) = \lambda_i$ that separates V_j and V_ℓ. We show that two cuts Z and X cross each other. From property (iii), X contains at least one λ_i-leaf V_a, for which $V_a \cap \Gamma_{\tilde{G}_{i+1}}(s) \neq \emptyset$ must hold, because $V_a \cap \Gamma_{\tilde{G}_{i+1}}(s) = \emptyset$ and Lemma 4 would imply $\lambda_i = d_{G_i^*}(V_a) = d_{G'|^{\lambda_i}}(V_a) = d_{G'|^{\lambda_{i+1}}}(V_a) < \lambda_{i+1}$, a contradiction to $\lambda_{G'|^{\lambda_{i+1}}}(V) \geq \lambda_{i+1}$. Hence, $X - Z \supseteq V_a \cap \Gamma_{\tilde{G}_{i+1}}(s) \neq \emptyset$ by $Z \cap \Gamma_{\tilde{G}_{i+1}}(s) = \emptyset$. Similarly for $(V - X) - Z \neq \emptyset$, since $V - X$ is also a minimum cut in G_i^*. Therefore, two cuts Z and X cross each other, and inequality

$$d_{G_i^*}(Z) + d_{G_i^*}(X) \geq d_{G_i^*}(Z - X) + d_{G_i^*}(X - Z)$$

always holds. Note that $d_{G_i^*}(X) = \lambda_i$, $d_{G_i^*}(X - Z) \geq \lambda_i$ (by $\lambda_{G_i^*}(V) \geq \lambda_i$), and $d_{G_i^*}(Z - X) > d_{G_i^*}(Z)$ (from the choice of Z) hold. These, however, contradict the above inequality. Therefore, there is no cut Z satisfying (3) and the lemma is proved. □

Next we consider how to split the edges incident to s in $\tilde{G}_{i+1}$ in order to obtain G_{i+1}^*. Lemma 5 asserts that any cut X with $X \cap \Gamma_{\tilde{G}_{i+1}}(s) = \emptyset$ satisfies $d_{\tilde{G}_{i+1}}(X) \geq \lambda_{i+1}$. For a cut $X \supseteq \Gamma_{\tilde{G}_{i+1}}(s)$, we consider cut $X' = V - X$, which satisfies $X' \cap \Gamma_{\tilde{G}_{i+1}}(s) = \emptyset$ and $d_{\tilde{G}_{i+1}}(X) > d_{\tilde{G}_{i+1}}(X') \geq \lambda_{i+1}$. That is, any cut X not dividing $\Gamma_{\tilde{G}_{i+1}}(s)$ satisfies $d_{\tilde{G}_{i+1}}(X) \geq \lambda_{i+1}$. Then we describe a way to increase the weight of every cut dividing $\Gamma_{\tilde{G}_{i+1}}(s)$ at least by $\lambda_{i+1} - \lambda_i$. This will mean that the resulting graph can be considered as G_{i+1}^* because $\lambda_{G_{i+1}^*}(V) \geq \lambda_{i+1}$ holds by assumption $\lambda_{G_i^*}(V) = \lambda_i$ and the above argument. Now, arrange the vertices in $\Gamma_{\tilde{G}_{i+1}}(s)$ in an arbitrary order, say

$$u_{i,1}, u_{i,2}, \ldots, u_{i,q},$$

delete all edges (s, u), $u \in \Gamma_{\tilde{G}_{i+1}}(s)$, and increase the weights of edges e in E_i^* as follows, where $q + 1 = 1$ and $\delta_{i+1} = (\lambda_{i+1} - \lambda_i)/2$:

$$c_{G_{i+1}^*}(e) = \begin{cases} c_{G_i^*}(e) + \delta_{i+1} & \text{if } e = (u_{i,j}, u_{i,j+1}), \\ & \quad j = 1, \ldots, q \text{ is an} \\ & \quad \text{edge in } G_i^*, \\ \delta_{i+1} & \text{if } e = (u_{i,j}, u_{i,j+1}), \\ & \quad j = 1, \ldots, q \text{ is not} \\ & \quad \text{an edge in } G_i^*, \\ c_{G_i^*}(e) & \text{otherwise.} \end{cases}$$

In other words, G_{i+1}^* is obtained from G_i^* by adding a cycle $C_{i+1}^* = u_{i,1}, u_{i,2}, \ldots, u_{i,q}$ of weight δ_{i+1}. Clearly, any cut X that divides $\Gamma_{\tilde{G}_{i+1}}(s)$ has the desired property, $d_{G_{i+1}^*}(X) \geq d_{G_i^*}(X) + 2\delta_{i+1} \geq \lambda_{i+1}$.

In th above construction also says that, for any intermediate k with $\lambda_i < k < \lambda_{i+1}$, we can obtain an optimally augmented graph $G^*(k)$ by adding a C_{i+1}^* of weight $(k - \lambda_i)/2$.

Now we are ready to describe optimal solutions for the entire range of $k \in [\lambda_G(V), +\infty]$.

For each λ_i, $i = 1, 2, \ldots, p$ (for a technical reason, we redefine $\lambda_p (= K)$ by $\lambda_p := +\infty$), choose a cycle C_i^* that visits all the neighbors of s in $G'|_{\lambda_{i-1}}^{\lambda_i}$. Then set

$$\Pi = \{(C_i^*, [\lambda_{i-1}, \lambda_i]) \mid i = 1, 2, \ldots, p\}$$

of pairs cycles C_i^* and ranges $[\lambda_{i-1}, \lambda_i]$ characterizes all optimal solutions for the entire range of k: Given a k, $G^*(k)$ can be obtained by finding the maximum i_k such that $\lambda_{i_k} < k$, and by increasing the weights of edges along all cycle C_i^* by $(\lambda_i - \lambda_{i-1})/2$ $i = 1, 2, \ldots, i_k$ and along cycle $C_{i_k}^* + 1$ by $(k - \lambda_{i_k})/2$. Clearly, since each cycle C_i^* can be obtained in $O(n)$ time, Π can be constructed in $O(np) = O(n^2 \log n)$ time. Together with Theorem 3, this finally establishes Theorem 2.

For our running example, we have $\Pi = \{(C_1^* = \{u_4, u_5\}, [7, 8]), (C_2^* = \{u_2, u_4, u_5\}, [8, 10]), (C_3^* = \{u_1, u_2, u_3, u_6\}, [10, 14]), (C_4^* = \{u_1, u_2, u_3, u_4, u_6\}, [14, 16]), (C_5^* = \{u_1, u_2, u_3, u_4, u_5, u_6\}, [16, +\infty])\}$.

5 Concluding Remarks

In this paper, we consider the edge-connectivity augmentation problem which asks to add the minimum amount of weights to a given graph $G = (V, E, c_G)$ to make G k-edge-connected, and present an $O(mn + n^2 \log n)$ time algorithm for finding optimal solutions $G^*(k)$ in the entire range $k \in [0, +\infty]$, where $n = |V|$ and $m = |E|$. The argument developed in this paper can be applied to the following slightly more general augmentation problem, which is studied in [2]. It has lower and upper bound constraints for each vertex $v \in V$; we have to add new weights to G so that the resulting graph G' satisfies

$$\underline{vc}(v) \leq \sum_{w \in V - v} (c_{G'}(v, w) - c_G(v, w)) \leq \overline{vc}(v), \quad v \in V,$$

where $\underline{vc}(v) \leq \overline{vc}(v)$, $v \in V$ are given constants (possibly $\underline{vc}(v) = +\infty$ or $\overline{vc}(v) = +\infty$, and $c_{G'}(v, w)$ (resp., $c_G(v, w)$) is considered to be zero if $(v, w) \notin E(G')$ (resp., $(v, w) \notin E$). Let

$$\mu(G) = \min\{d_G(X, V - X) + \sum_{x \in X} \overline{vc}(x) \mid \emptyset \neq X \subset V\}.$$

(Note that $\mu(G)$ can be easily computed as the s-based-connectivity of the graph G" obtained from G by adding a new vertex s and edge (s, v) with weight $\overline{vc}(v)$ for each $v \in V$.) Then clearly, for any $k > \mu(k)$, G cannot be augmented to be k-edge-connected by adding new weights. With a slight modification, our algorithm can find optimal solutions in the range $k \in [0, \mu(G)]$ in the same time complexity.

Note that the problem of finding a complete feasible splitting in a given edge-weighted graph H with a designated vertex a can be reduced to the problem of augmenting the edge-connectivity of graph $H - a$ with vertex constraints $\underline{vc}(v) = 0$ and $\overline{vc}(v) = c_H(s, v)$, $v \in V(H) - a$. Therefore, we can find a complete feasible splitting in a given graph with n vertices and m edges in $O(mn + n^2 \log n)$ time. This is faster by factor of $O(\log n)$ than the previous fastest $O((mn + n^2 \log n) \log n)$ algorithm due to [5]. However, for the integer version of splitting problem where all weights in the resulting graph must be integers, their bound is still currently best.

Appendix:.

We replace lines 18-20 of SIMUL-AUGMENT with the following procedure ALIGN to guarantee that the number of ranges in the resulting G' is $O(n)$. Let us call the resulting algorithm SIMUL-AUGMENT(ALIGN).

Procedure ALIGN

a1 Let $a_{min} = \min\{a \mid [a, k'] \in R_{G'}(X^*)\}$, and let z_{min} be the vertex in X^* such that $[a_{min}, k'] \in R_{G'}(z_{min})$

a2 **for** each vertex $u \in X^* - z_{min}$ **do**

a3   **if** $|R_{G'}(u)|^{k'}| \geq 2$ **then**

a4   Let $[a', b']$ be the range with the maximum top b' among those satisfying $b' \leq a$ for the range $[a, k']$;

a5 $A_u := (R_{G'}(u)|^{k'} - \{[a', b'], [a, k']\}) \cup \{[a', b' + (k' - a)]\}$

a6   **end** { if }

a7 **end** { for }

a8   Sort ranges in $A = \cup_{u \in X^*} A_u$ in the non-decreasing order of their tops, where $[a_{min}, k'] \in R_{G'}(z_{min})|^{k'}$ is be the last range in the ordering;

a9   Align $A = \cup_{u \in X^*} A_u$ 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;

a10 $R_{G'}(u) := A'_u \cup R_{G'}(u)|_{k'}$ for each $u \in X^*$, merging two ranges $[a, k'] \in A'_u$ and $[k', b] \in R_{G'}(u)|_{k'}$ (if any) into a single range $[a, b] \in R_{G'}(u)$; {this merging occurs at least for $u = z_{min}$}

For the correctness of SIMUL-AUGMENT (ALIGN), we only have to show that alignment of $A = \cup_{u \in X^*} A_u$ into $[k^*, k']$ in line a9 of ALIGN is possible, i.e.,

Lemma 6 *$A = \cup_{u \in X^*} A_u$ in line a9 of ALIGN is gapless.*

Proof: We see $R_{G'_{j-1}}(X^*)$ before line 18 is gapless. Since $R_{G'_{j-1}}(X^*)$ before line 18 of SIMUL-AUGMENT is gapless and there is no range $[a, k'] \in R_{G'_{j-1}}(X^*)|^{k'}$ with $a < a_{min}$, where a_{min} is chosen in line a1, we see that, for any k with $\min\{a \mid [a, b] \in R_{G'_{j-1}}(X^*)\} \leq k \leq a_{min}$, there is a range $[a', b'] \in R_{G'_{j-1}}(X^*)|^{k'}$ with $a' \leq k \leq b' < k'$. Hence, set $A' = \{[a', b'] \in R_{G'_{j-1}}(X^*)|^{k'} \mid b' < k'\} \cup \{[a_{min}, k']\}$ of ranges is gapless. Since only ranges in $R_{G'_{j-1}}(X^*)|^{k'} - A'$ are lowered in preparing each A_u in line a5 of ALIGN, the resulting set $A = \cup_{u \in X^*} A_u$ in line a9 of ranges is still gapless. □

By the similar analysis in Lemma 3, we see that aligning $A = \cup_{u \in X^*} A_u$ in line a8 can be carried out in $O(|R|)$ time, where R is the set $R_{G'}(V)$ of ranges obtained after line a9.

By analyzing how many ranges are created during algorithm SIMUL-AUGMENT(ALIGN), we can prove Theorem 3. The detail is omitted due to space limitation.

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

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

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

[6] H. Nagamochi, T. Shiraki and T. Ibaraki, *Computing edge-connectivity augmentation function in $\tilde{O}(nm)$ time*, Tech. Rep. of Information Processing Society of Japan, AL-53-7, 1996.

[7] D. Naor, D. Gusfield and C. Martel, *A fast algorithm for optimally increasing the edge connectivity*, FOCS, 1990, 698–707.

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