数理解析研究所講究録 1241 巻 2001 年 157-165

A Ranged Laminar Family in Graphs and Its Application

Hiroshi Nagamochi Yuusuke Abe Toshimasa Ishii

Department of Information and Computer Sciences,
Toyohashi University of Technology,
Toyohashi, Aichi, 441-8580, Japan.
{naga,yusuke,ishii}@algo.ics.tut.ac.jp

Abstract Let $G=(V,E)$ be a simple undirected graph with a set V of vertices and a set E of edges weighted by nonnegative reals. For a given real $k>0$, it is known that there exists a laminar family $\mathcal{X}_k$ of subsets of vertices such that each subset corresponds to a cut with size less than k and destroying all cuts in $\mathcal{X}$ by adding a new vertex s and some weighted edges between s and V destroys all other cuts with size less than k. We prove that such laminar families $\mathcal{X}_k$ for all positive reals $k>0$ can be obtained as a compact representation which we call a ranged laminar family. The time complexity for computing the ranged laminar family is $O(|V||E|+|V|^2\log|V|)$. As an application of this, we show that given ranged laminar family the source location problem for all demands $k>0$ can be solved simultaneously in $O(|V|^2)$ time.

1 Introduction

The connectivity in graphs is one of the basic concepts, and has wide applications in practice. such as the design and analysis of reliable networks. For example, the problem of augmenting a given graph to be a higher connected graph by adding a smallest number of new edges is applied to design of reliable networks. The problem is called the connectivity augmentation problem, and has been studied extensively (see [3, 5] for a recent survey). Given a graph G and an integer $k>0$ called a target, the problem of making G k-edge connected by adding a smallest set of new edges is called the edge-connectivity augmentation problem. Watanabe and Nakamura [12] first proved that the problem is polynomially solvable, and Frank [2] gave a general framework to handle the edge-connectivity augmentation problem by using the edge-splitting technique. Based on the Frank's algorithm and a nice property in the minimum cut algorithm [9], Nagamochi et al. [7] proved that the real weighted version of the edge-connectivity augmentation problem (where edges of G are weighted by nonnegative reals and the objective is to minimize the sum of weights of edge to be added to make G k-edge-connected) can be solved for all real targets $k>0$ in the sense that optimal solutions for all k are represented by a compact representation.

On the other hand, Nagamochi and Ibaraki [6] modified the minimum cut algorithm so as to reduce the time complexity of the Frank's algorithm. As a byproduct, they [8] obtained the following result: For a given real $k>0$, there exists a laminar family $\mathcal{X}_k$ of subsets of vertices such that each subset corresponds to a cut with size less than k and destroying all cuts in $\mathcal{X}$ by adding a new vertex s and some weighted edges between s and V destroys all other cuts with size less than k. Recently, by using this result, it was shown [1] that the source location problem with uniform demand $k>0$ can be solved in linear time (see section 4 for the definition of the problem).

In this paper, we prove that such laminar families $\mathcal{X}_k$ for all positive reals $k>0$ can be obtained as a compact representation which we call a ranged laminar family. The time complexity for computing the ranged laminar family is $O(|V||E| + |V|^2 \log |V|)$. As an application of this, we show that from a given ranged laminar family, the source location problem for all demands $k > 0$ can be solved simultaneously in $O(|V|^2)$ time.

2 Preliminaries

Let $G = (V, E)$ be an edge-weighted undirected graph with a set V of vertices, a set E of edges. We denote $|V|$ by n and $|E|$ by m, and we may write the vertex set and the edge set of a graph G as $V(G)$ and $E[G]$, respectively. We denote edge weights by a function $c_G : E \rightarrow R^+$, where R^+ denotes the set of nonnegative reals, and we may write the weight $c_G(e)$ of edge $e = (u, v)$ as $c_G(u, v)$ A singleton set x may be simply written as x. For two disjoint subsets $X, Y \in V$, 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 we 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 be written as $d_G(X)$. A cut separates $x, y \in V(G)$ is called an *(x, y)-cut*. An (x, y)-cut with minimum size called a *minimum (x, y)-cut*, and its size is defined by $\lambda_G(x, y)$. A total ordering $v_1, v_2, \ldots, v_n$ of all vertices in $V(G)$ *maximum adjacency ordering* (*MAO*, for short) if it satisfies

$$d_G(\{v_1, v_2, \ldots, v_i\}, v_{i+1}) = \max_{u \in \{v_{i+1}, \ldots, v_n\}} d_G(\{v_1, v_2, \ldots, v_i\}, u) \quad (1 \leq i \leq n-1).$$

Lemma 2.1 [6, 9, 11] *For an MAO $v_1, v_2, \ldots, v_n$, $\lambda_G(v_{n-1}, v_n) = d_G(\{v_n\}, V(G) - \{v_n\})$ holds for the last two vertices v_{n-1} and v_n. An MAO in a graph G with n vertices and m edges can be found in $O(m + n \log n)$ time.* □

Notice that we can choose an arbitrary vertex as the first vertex v_1 in an MAO and that if $d_G(u) \geq k$ for all $u \in V - v_1$ then there exists a pair of vertices s and t with $\lambda_G(s, t) \geq k$.

A family $\mathcal{X} = \{X_1, X_2, \ldots, X_p\}$ of sets of vertices is called a *laminar family* if it satisfies $X_i \cap X_j = \emptyset, X_i \subset X_j$ or $X_j \subset X_i$ for each $X_i, X_j \in \mathcal{X}$. A member X in a laminar family $\mathcal{X}$ is called *maximal* (resp., *minimal*) in $\mathcal{X}$ if there is no member $Y \in \mathcal{X}$ with $Y \supset X$ (resp., $Y \subset X$). For a function $t : V \rightarrow R^+$, we may write $\sum_{v \in X} t(v)$ as $t(X)$.

Definition 2.1 *For a graph $G = (V, E)$ and a target $k \geq 0$, a k-laminar family of G is defined to be a laminar family of subsets of V such that*

(1) $d_G(X) < k$ *for all $X \in \mathcal{X}$,*

(2) *For any $|V|$-dimensional vector t such that $d_G(X) + t(X) \geq k$, $X \in \mathcal{X}$, it holds*

$$d_G(Y) + t(Y) \geq k, \quad Y \in 2^V - \{\emptyset, V\}.$$

For example, Fig. 1 shows a 7-laminar family $\mathcal{X}_7$ in the graph, In section 3.1, we review how to compute a k-laminar family for a given G and $k \geq 0$.

For two reals $a, b \in R^+$ with $a < b$, the interval $[a, b]$ is called a *range* and its size $\pi([a, b])$ is defined as $b - a$. For a range $r = [a, b]$, a and b are denoted by $L(r)$ and $U(r)$, respectively. Let $R = \{[a_1, b_1], [a_2, b_2], \ldots, [a_q, b_q]\}$ be a set of ranges. The size of R denoted by $\pi(R)$, is defined as the sum $\sum(b_i - a_i)$ of all range sizes in R. For a given $h \in R^+$, the *upper h-truncation* (resp., *under*

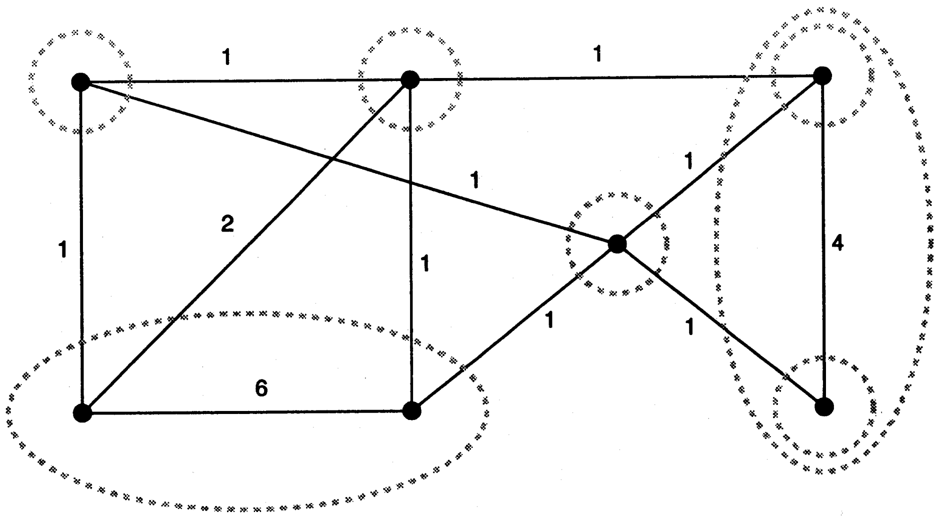

Figure 1: A k-laminar family for $k = 7$.

h-*truncation*) of a set R of ranges is defined by $R|^h = \{[a_i, \min\{h, b_i\}] | a_i < h, i = 1, 2, \ldots, q\}$ (resp., $R|_h = \{[\max\{a_i, h\}, b_i] | b_i > h, i = 1, 2, \ldots, q\}$). A ranged laminar family is a family $\mathcal{X}$ of cuts X with a range $r(X)$. For a real $k \geq 0$ and a ranged laminar family $\mathcal{X}$, we denote by $\mathcal{X}/k$ the laminar family $\{X \in \mathcal{X} \mid L(r(X)) < k < U(r(X))\}$.

Definition 2.2 *A ranged laminar family $\mathcal{X}$ is called valid if $\mathcal{X}/k$ is a k-laminar family for each target $k \geq 0$.*

Fig. 2 shows a ranged laminar family $\mathcal{X}$ of the graph G in Fig. 1. Notice that $\mathcal{X}/7$ is equal to $\mathcal{X}_7$ in Fig. 1.

In this paper, we design an $O(nm + n^2 \log n)$ time algorithm for computing a valid ranged laminar family.

3 Computing a Ranged Laminar Family

In this section, we give an algorithm for computing a ranged laminar family after describing the algorithm [8] for computing a k-laminar family for a given real $k \geq 0$.

3.1 Computing a k-laminar family

We describe the algorithm in [8] for computing a k-laminar family $\mathcal{X}_k$. Given a graph $G = (V, E)$, we first add to G a new vertex s to obtain a graph with the designated vertex s. We then add a new edge between s and each vertex $u \in V(G)$ with $d_G(u) < k$, where weight of the edge (s, u) is given by $k - d_G(u)$ so that $d_H(u) = k$ holds in the resulting graph H. Then, we initialize a laminar family $\mathcal{X}_k$ of subsets of $V(G)$ by $\{\{u\} \mid d_G(u) < k\}$.

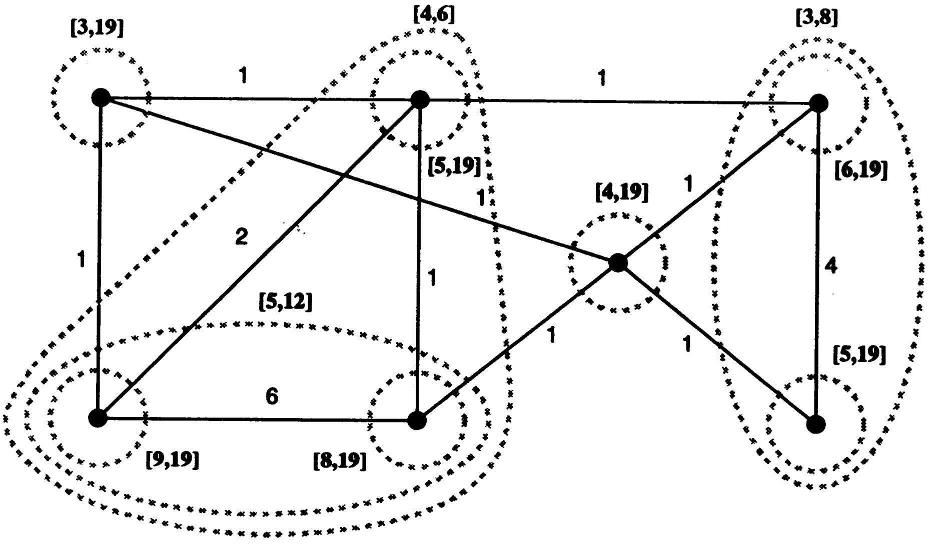

Figure 2: A given graph G and a ranged laminar family $\mathcal{X}$ for G.

Then we repeat the following procedure for contracting two vertices into a single vertex until the graph has three vertices (including s). We maintain the condition that

$$d_H(u) \geq k \text{ holds for all } V(H) - s \tag{1}$$

in the current graph H (this holds for the initial graph H). By (1) and Lemma 2.1, there exits a pair of vertices $v, w \in V(H) - s$ such that $\lambda_H(v, w) \geq k$ (such v, w are given by the last two vertices in an MAO starting from s as the first vertex). Since no cut with size less than k separates these v and w, we contract u and v into a single vertex x^*. After this, we check whether (1) still holds or not, i.e., whether $d_H(x^*) \geq k$ holds or not. If $d_H(x^*) \geq k$, then we repeat the same procedure. Otherwise, we repeat the same procedure after updating H and $\mathcal{X}_k$ as follows. We increase weight of edge (s, x^*) by $k - d_H(x^*)$, and we add to $\mathcal{X}_k$ the set X^* of all vertices in $V(G)$ that have been contracted into x^*. Thus,

$$\text{for each cut } X \in \mathcal{X}_k,\ d_H(X) = k \text{ holds in the graph } H \text{ immediately after } X \text{ is added to } \mathcal{X}_k \tag{2}$$

Clearly, the final $\mathcal{X}_k$ is a laminar family of subsets of $V(G)$. The algorithm is described as follows.

Algorithm k-LAMINAR-FAMILY
Input : An edge-weighted graph $G = (V, E, c_G)$ and a real $k \in R^+$.
Output : A k-laminar family $\mathcal{X}_k$.
Let $U = \{u_1, u_2, \ldots, u_p\}$ be the set of vertices $u_i \in V$ such that $d_G(u_i) < k$.
$V' = V \cup \{s\}$; $E' = E \cup \{(s, u_1), (s, u_2), \ldots, (s, u_p)\}$;
for each $u_i \in U$ **do**
  $c_H(s, u_i) := k - d_G(u_i)$
end;
Let $H = (V', E', c_H)$ be the obtained graph.
$\mathcal{X}_k := \{\{u_1\}, \{u_2\}, \ldots, \{u_p\}\}$;
while $V(H) > 4$ **do**
  Find two vertices $v, w \in V(H) - s$ such that $\lambda_H(v, w) \geq k$;

10 Contract v and w into a single vertex x^*, and let H be the resulting graph;
11 **if** $d_H(x^*) < k$ **then**
12 Let H denote the graph obtained from H by setting $c_H(s, x^*) := c_H(s, x^*) + k - d_H(x^*)$ (after creating edge (s, x^*) with $c_H(s, x^*) := 0$ if $(s, x^*) \notin E[H]$);
13 Let X^* denote the set of vertices in $V(G)$ that have been contracted into x^* so far, and set $\mathcal{X}_k := \mathcal{X}_k \cup \{X^*\}$;
14 **end**
15 **end**;
16 Output $\mathcal{X}_k$.

Theorem 3.1 [8] *For a graph G and a target $k \geq 0$, Algorithm k-LAMINAR-FAMILY correctly computes a k-laminar family in $O(nm + n^2 \log n)$ time.* □

3.2 Algorithm for computing a ranged laminar family

The basic idea for finding a ranged laminar family is to use the same approach for computing the edge connectivity augmentation function [7]. We try to perform Algorithm k-LAMINAR-FAMILY for all targets $k \geq 0$. In order to execute this in a finite space, we maintain the computation process for all k using a compact representation. For this, we represent graphs H during execution of k-LAMINAR-FAMILY for all $k \geq 0$ by a ranged graph, which is defined as follows. Let $H = (V \cup \{s\}, E \cup E_H(s)$ have a designated vertex s, a set E of edges not incident to s and a set $E_G(s)$ of edges incident to s, where each edge $e \in E$ has a nonnegative weight $c_H(e)$, but each vertex $v \in V$ has a set $R(v)$ of ranges (instead of a nonnegative weight). Such a graph H is called a *ranged graph*, which represents infinitely many weighted graphs in the following sense.

Definition 3.1 *Given an arbitrary target $k \geq 0$, we let H and k correspond to an edge-weighted graph $H|^k = (V \cup \{s\}, E \cup E_H(s))$ such that*

$$c_{H|^k}(e) = c_H(e) \text{ for } e \in E \text{ and } c_{H|^k}(e) = \pi((R(v)|^k)) \text{ for } e = (s, v) \in E_H(s)$$

(see Section 2 for the definition of $\pi(\cdot)$ and $R|^k$).

With the notion of ranged graphs, let us perform Algorithm k-LAMINAR-FAMILY for all targets $k \geq 0$. Given an edge-weighted graph $G = (V, E)$, we first add a new vertex s to V, and add one edge between s and each $v \in V$, where $E_H(s)$ denotes the set of all edges between s and V. Now we set a range set $R(v)$ of each $v \in V$ to be $R(v) = \{[d_G(v), \infty]\}$. It is easy to see that for each $k \geq 0$ $H|^k$ is the graph constructed in line 6 of k-LAMINAR-FAMILY. In what follows, we use K as a sufficiently large value in the sense that for any two $k, k' \geq K$ $H|^k$ and $H|^{k'}$ have the essentially same structure (we will see that $K = 2\max_{v \in V} d_G(v)$ suffices). Then let each each $v \in V$ has range

$$R(v) = \{[d_G(v), K]\}.$$

At this point, the initial laminar family $\mathcal{X}_k$ in line 7 of k-LAMINAR-FAMILY may be different for each k. However, all these families $\mathcal{X}_k$ can be compactly represented by a ranged laminar family. That is, we initialize a ranged laminar family $\mathcal{X}$ by $\{\{u\} \mid u \in V(G)\}$ and set the range $r(X)$ of each $X \in \mathcal{X}$ by $[d_G(u), K]$. We easily see that $\mathcal{X}/k$ is equal to $\mathcal{X}_k$ in line 7 of k-LAMINAR-FAMILY.

Now we proceed to the procedure of contracting vertices in k-LAMINAR-FAMILY. Here we have to resolve an important problem such that if a pair of two vertices $v, w \in V(H) - s$ to be contracted are different for distinct targets k and k', we cannot maintain the computation process by a single ranged graph H. Fortunately, it is ensured that there is a common pair of vertices v and w such that $\lambda_{H|^k}(v, w) \geq k$ for all $k \geq 0$.

Lemma 3.1 [7] *Let $H = (V \cup \{s\}, E \cup E_H(s))$ be a ranged graph with $|V(H)| \geq 3$. Assume that for each vertex $u \in V(H) - s$,*

$$\text{the range set } R(u) \text{ contains a range } r \text{ with } L(r) \leq d_H(u, V(H) - \{s, u\}) \text{ and } U(r) = K. \quad (3)$$

Then, for the last two vertices v, w in an MAO starting from s in the weighted graph $H|^K$, such that $\lambda_{H|^k}(v, w) \geq k$ holds for all $0 \leq k \leq K$. □

Lemma 3.1 shows that u, v can be used in common as pair of vertices to be contracted for all k and such a pair can be found in $O(m + n \log n)$ time.

We finally consider how to update the range sets $R(v)$, $v \in V(H)$ and the ranged laminar family $\mathcal{X}$ after contracting two vertices v and w. Suppose that

$$\text{for each } u \in V(H) - s, \text{ all ranges } r \in R(v) \text{ satisfy } U(r) = K \quad (4)$$

(this is true for the initial ranged graph H). For the vertex x^* just contracted from v and w, we set $R(x^*)$ to be the union of $R(v)$ and $R(w)$. In the resulting ranged graph H, if $d_{H|^k}(x^*) \geq k$ for all $k \in [0, K]$, then we can proceed to the next iteration of the procedure. Assume $d_{H|^k}(x^*) < k$ for some $k \in [0, K]$. That is, by (4), $R(x^*)$ contains no range r with $L(r) \leq d_H(x^*, V(H) - \{s, x^*\})$ (in other words, x^* does not satisfies (3)). In this case, we modify some ranges in $R(x^*)$. Let $k^* = d_H(x^*, V(H) - \{s, x^*\})$. For this, we compute k' such that $\pi(R(x^*)|^{k'}) = k' - k^*$.

Note that for a target k with $k \leq k^*$ or $k \geq k'$, the current ranged laminar family $\mathcal{X}$ satisfies the condition (2) with $\mathcal{X}_k = \mathcal{X}/k$. To meet (2) for targets $k \in [k^*, k']$, we divide each range

$$r \in R(x^*) \text{ with } L(r) < k' < U(k) \quad (5)$$

into two ranges $r' = [L(r), k']$ and $r'' = [k', U(r)]$ ($U(r) = K$), and then replace the set of all ranges $r' = [L(r), k']$ by a single range $[k^*, k']$, where we merge the $[k^*, k']$ and some range $[k', K]$ into $[k^*, K]$ to satisfy (4). (Note that this operation can be written as $R(x^*) := (R(x^*) - \{r\})|_{k'} \cup \{[k^*, K]\}$ for a range $r \in R(x^*)$.)

we need to add to $\mathcal{X}$ the set X^* of vertices in $V(G)$ contacted into x^*, setting its range $r(X^*) = [d_H(x^*, V(H) - \{s, x^*\}), k']$. Then for the resulting $\mathcal{X}$ and all $k \in [0, K]$, the condition (2) holds for $\mathcal{X}_k = \mathcal{X}/k$. This implies that for each target $k \in [0, K]$, $H|^k$ and $\mathcal{X}/k$ can be viewed as those H and $\mathcal{X}_k$ computed during the execution of k-LAMINAR-FAMILY. Therefore, the final ranged laminar family $\mathcal{X}$ obtained by the algorithm is valid. Note that it suffices to choose K so that the k' in line 13 is always less than $U(r)$ in (5). Therefore, by setting $K := 1 + 2\max_{v \in V} d_G(v)$, any k' is less than $U(r) = K$. the choice of K The algorithm description is given as follows.

ALGORITHM RANGED-LAMINAR-FAMILY

Input : An edge-weighted undirected graph $G = (V, E, c_G)$

Output : A ranged laminar family $\mathcal{X}$

begin
  $V' := V \cup \{s\}$; $E(s) = \{(s,v) | v \in V\}$;
$\mathcal{X} := \emptyset$; $K := 1 + 2\max_{v \in V} d_G(v)$;
  for each vertex $u \in V$ **do**
    $R(u) := \{[d_G(u), K]\}$; $X := \{u\}$; $r(X) := [d_G(u), K]$; $\mathcal{X} := \mathcal{X} \cup \{X\}$;
  end;
  Let $H = (V', E' = E \cup E(s))$
  be the resulting ranged graph;
  while $|V(H)| \geq 4$ **do**
    Find vertices $v, w \in V(H) - s$ such that $\lambda_{H|^k}(u,v) \geq k$ holds for all $0 \leq k \leq K$;
    Contract v and w into a single vertex x^*;
    $R(x^*) := R(v) \cup R(w)$;
    Let H be the resulting ranged graph;
    $k^* := d_H(x^*, V(H) - \{s, x^*\})$;
    Let $X^* \subset V(G)$ be the set of vertices contracted into x^*;
    /* We assume $R(x^*) = \{[a_1, K], [a_2, K], \ldots, [a_{|X^*|}, K]\}$,
    where $a_1 \leq a_2 \leq \cdots \leq a_{|X^*|}$ */
    if $k^* < a_1$ then
      Find $k' \in R^+$ such that $\pi(R(x^*)|^k) = k' - k^*$;
      $R(x^*) := (R(x^*) - \{[a_1, K]\})|_{k'} \cup \{[k^*, K]\}$;
      $\mathcal{X} := \mathcal{X} \cup \{X^*\}$; $r(X^*) := [k^*, k']$;
    end;
    Denote the ranged graph resulting as H;
  end;
  Output $\mathcal{X}$;
end.

Clearly, $|\mathcal{X}| \leq 2n-2$. Since we need to compute the MAO at most $n-1$ times and other computations are minor, the running time of RANGED-LAMINAR-FAMILY is $O(nm + n^2 \log n)$.

Theorem 3.2 *For a given graph G, Algorithm RANGED-LAMINAR-FAMILY correctly computes a valid ranged laminar family $\mathcal{X}$ in $O(nm + n^2 \log n)$ time.* □

4 The source location problem for all demands

Let $G = (V, E)$ be a simple undirected graph with a cost function $cost : V \to R^+$, a weight function $c_G : E \to R^+$ A general form of the source location problem [10] asks to find a minimum cost subset $S \subseteq V$ for a demand function $d : V \to R^+$ such that for each $v \in V - S$ there are $d(v)$ edge-disjoint (or vertex-disjoint) paths between v and S. This has an application to the problem of finding an optimal location of mirror servers on computer networks [1, 4].

In what follows, we consider the source location problem which asks to find a minimum cost subset $S \subseteq V$ for a uniform demand $k > 0$ such that there are k edge-disjoint paths between each vertex

$v \in V - S$ and S. It is known [1] that this source location problem can be solved in linear time if a k-laminar familiy $\mathcal{X}_k$ has been obtained. In this paper, we shows that the source location problem for all demands k can be solved simultaneously by using a valid ranged laminar family.

4.1 Algorithm for the source location problem for a fixed demand

We solve the source location problem for a fixed demand k by using a k-laminar family $\mathcal{X}_k$. Since $d_G(X) < k$ holds for each cut $X \in \mathcal{X}_k$ by Definition 2.1(1), we must choose at least one source from each minimal subset $X \in \mathcal{X}_k$ (otherwise, removal of $E_G(X)$ would separate some vertex $v \in X$ and S). Conversely, if we select a vertex from each minimal subset $X \in \mathcal{X}_k$ then any other cut Y with $d_G(Y) < k$ includes at least one source Definition 2.1(2). If a cost function $cost : V \to R^+$ is given, then it suffices to choose a vertex with the minimum cost from each minimal subset $X \in \mathcal{X}_k$.

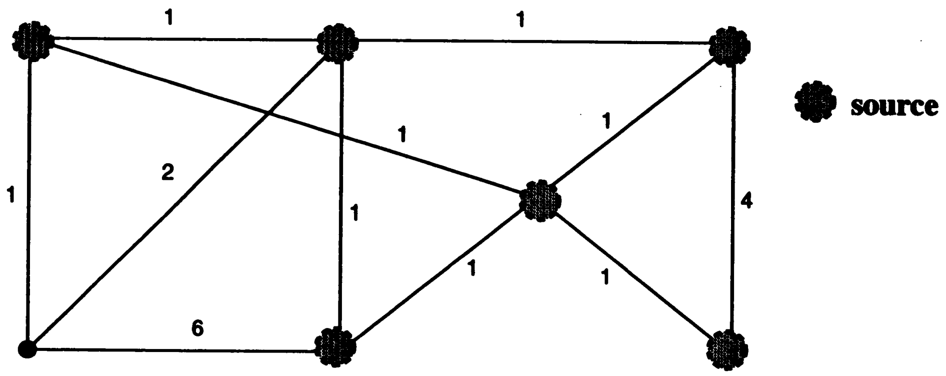

Figure 3: An obtained location of sources for $k = 7$.

Theorem 4.1 *The source location problem for a demand k can be solved in $O(n)$ time by using k-laminar family.* □

For the graph G in Fig. 2 and demand $k = 7$, the S of minimum number of sources is shown in Fig. 3.

4.2 Algorithm for the source location problem for all demands

Suppose that for a given graph G, a valid ranged laminar family $\mathcal{X}$ is obtained. To solve the source location problem for all demands, we first sort boundary values $L(r(X)), U(r(X))$ for all cuts $X \in \mathcal{X}$, and let $Z = \{z_1, z_2, \ldots, z_q | z_1 < z_2 < \cdots < z_q\}$ be the resulting sequence ($q \leq 2n-2$). Though structure of a k-laminar family may be changed at each boundary values z_j ($1 \leq j \leq q$), it does not change in the interval (z_j, z_{j+1}) ($1 \leq j < q$). Therefore, by computing a solution for each interval, we can obtain optimal source sets for all demands k.

Let $\mathcal{X}_{[z_j, z_{j+1}]}$ be the k-laminar family for $k \in (z_j, z_{j+1}]$. Then we choose a vertex with the minimum cost from each minimal subset $X \in \mathcal{X}_{[z_j, z_{j+1}]}$. The set $S_{[z_j, z_{j+1}]}$ of chosen vertices is an optimal source set for demand $k \in (z_j, z_{j+1}]$.

Theorem 4.2 *For a given graph G and a valid ranged laminar family $\mathcal{X}$ of G, the source location problem for all demands can be solved in $O(n^2)$ time.* $\square$

5 Conclusion

In this paper, we gave an $O(nm + n^2 \log n)$ time algorithm for computing a ranged laminar family for a given graph. As an application of this, we showed that the source location problem for all demands k can be solved in $O(n^2)$ time from a given ranged laminar family.

References

[1] K. Arata, S. Iwata, K. Makino, and S. Fujishige: Locating sources to meet flow demands in undirected networks, Lecture Notes in Computer Science, 1851, pp.300-313, 2000.

[2] A. Frank: Augmenting graphs to meet edge-connectivity requirements, SIAM J. Discrete Mathematics, vol.5, pp.25–53, 1992.

[3] A. Frank: Connectivity augmentation problems in network design, Mathematical Programming: State of the Art (eds: J. R. Birge and K. G. Murty), The University of Michigan, Ann Arbor, MI, pp.34–63, 1994.

[4] H. Ito, H. Uehara and M. Yokoyama: A faster and flexible algorithm for a location problem on undirected flow networks, IEICE Trans. Fundamentals, vol.E83-A, no.4, pp.704-712, 2000.

[5] H. Nagamochi: Recent development of graph connectivity augmentation algorithms, Inst. Electron. Inform. Comm. Eng. Trans. Fundamentals, vol.E83-D, pp.372-383, 2000.

[6] H. Nagamochi and T. Ibaraki: Computing edge-connectivity of multigraphs and capacitated graphs, SIAM J. Discrete Mathematics, vol.5, pp.54-66, 1992.

[7] H. Nagamochi and T. Ibaraki: Augmenting edge-connectivity over the entire range in $\tilde{O}(nm)$ time, Journal of Algorithms, vol.30, pp.253-301, 1999.

[8] H. Nagamochi and T. Ibaraki: Polyhedral structure of submodular and posi-modulara systems, Discrete Applied Mathmatics, vol.107, pp.165-189, 2000.

[9] H. Nagamochi, T. Ishii and T. Ibaraki: A simple and constructive proof of a minimum cut algorithm, Inst. Electron. Inform. Comm. Eng. Trans. Fundamentals, vol.E82-A, pp.2231–2236, 1999.

[10] H. Nagamochi, T. Ishii and H. Ito: Minimum cost source location problem with vertex-connectivity requirements in digraphs, Information Processing Letters (to appear).

[11] M. Stoer and F. Wagner: A simple min cut algorithm, JACM, vol.44, pp.585–591, 1997.

[12] T. Watanabe and A. Nakamura: Edge-connectivity augmentation problems, J. Comp. System Sci., vol.35, pp.96–144, 1987.