MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Minimizing a Monotone Concave Function with Laminar Covering Constraints

Mariko Sakashita, Kazuhisa Makino,
and Satoru Fujishige

METR 2007–29 April 2007

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.i.u-tokyo.ac.jp/mi/mi-e.htm

Minimizing a Monotone Concave Function with Laminar Covering Constraints*

Mariko Sakashita† Kazuhisa Makino‡ Satoru Fujishige§

April 2007

Abstract

Let V be a finite set with $|V| = n$. A family $\mathcal{F} \subseteq 2^V$ is called *laminar* if for all two sets $X, Y \in \mathcal{F}$, $X \cap Y \neq \emptyset$ implies $X \subseteq Y$ or $X \supseteq Y$. Given a laminar family $\mathcal{F}$, a demand function $d : \mathcal{F} \to \mathbb{R}_+$, and a monotone concave cost function $F : \mathbb{R}_+^V \to \mathbb{R}_+$, we consider the problem of finding a minimum-cost $x \in \mathbb{R}_+^V$ such that $x(X) \geq d(X)$ for all $X \in \mathcal{F}$. Here we do not assume that the cost function F is differentiable or even continuous. We show that the problem can be solved in $\mathrm{O}(n^2 q)$ time if F can be decomposed into monotone concave functions by the partition of V that is induced by the laminar family $\mathcal{F}$, where q is the time required for the computation of $F(x)$ for any $x \in \mathbb{R}_+^V$. We also prove that if F is given by an oracle, then it takes $\Omega(n^2 q)$ time to solve the problem, which implies that our $\mathrm{O}(n^2 q)$ time algorithm is optimal in this case. Furthermore, we propose an $\mathrm{O}(n \log^2 n)$ algorithm if F is the sum of linear cost functions with fixed setup costs. These also make improvements in complexity results for source location and edge-connectivity augmentation problems in undirected networks. Finally, we show that in general our problem requires $\Omega(2^{\frac{n}{2}} q)$ time when F is given implicitly by an oracle, and that it is NP-hard if F is given explicitly in a functional form.

1 Introduction

Let V be a finite set with $|V| = n$. A family $\mathcal{F} \subseteq 2^V$ is called *laminar* if for arbitrary two sets $X, Y \in \mathcal{F}$, $X \cap Y \neq \emptyset$ implies $X \subseteq Y$ or $X \supseteq Y$. Given a

*An extended abstract of this paper appeared in *Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005)*, China, December 2005 [16].

†Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan. sakasita@amp.i.kyoto-u.ac.jp

‡Graduate School of Information Science and Technology, University of Tokyo, Tokyo, 113-8656, Japan. makino@mist.i.u-tokyo.ac.jp

§Research Institute for Mathematical Sciences, Kyoto University, Kyoto, 606-8502, Japan. fujishig@kurims.kyoto-u.ac.jp

laminar family $\mathcal{F}$, a demand function $d : \mathcal{F} \to \mathbb{R}_+$, and a monotone concave function $F : \mathbb{R}_+^V \to \mathbb{R}_+$, the problem to be considered in this paper is given as

$$\begin{array}{lll} \text{Minimize} & F(x) & \\ \text{subject to} & x(X) \geq d(X) & (X \in \mathcal{F}), \\ & x(v) \geq 0 & (v \in V), \end{array} \tag{1}$$

where $\mathbb{R}_+$ denotes the set of all nonnegative reals, and $x(X) = \sum_{v \in X} x(v)$ for any $X \subseteq V$. Here we do not assume that the cost function F is differentiable or even continuous. The present problem has various applications, since laminar families represent hierarchical structures in many organizations. Moreover, the problem can be regarded as a natural generalization of the source location problem and the edge-connectivity augmentation problem in undirected networks, which do not seemingly have laminar structures. We shall show in Section 3 that they can be formulated as (1) by using extreme sets in given networks.

In this paper, we study the following three cases, in which the cost functions F are expressed as

$$F_1(x) = \sum_{X \in \mathcal{F}} f_{\Delta X}(x[\Delta X]) \quad \text{(laminar sum)}, \tag{2}$$

$$F_2(x) = \sum_{v \in V} f_v(x(v)) \quad \text{(separable)}, \tag{3}$$

$$F_3(x) = \sum_{v \in V : x(v) > 0} (a_v x(v) + b_v) \quad \text{(fixed-cost linear)}, \tag{4}$$

where $\Delta X = X - \bigcup\{Y \mid Y \in \mathcal{F},\ Y \subsetneq X\}$, $x[\Delta X]$ denotes the projection of x on ΔX, $f_{\Delta X} : \mathbb{R}_+^{\Delta X} \to \mathbb{R}_+$ and $f_v : \mathbb{R}_+^{\{v\}} \to \mathbb{R}_+$ are monotone concave, and a_v and b_v are nonnegative constants. It is clear that F_2 is a special case of F_1, and F_3 is a special case of F_2 (and hence of F_1).

We consider Problem (1) when the cost function F is given either in a functional form or by an oracle (i.e., we can invoke an oracle for the evaluation of $F(x)$ for any x in $\mathbb{R}_+^V$ and use the function value $F(x)$). In either case, we assume that $F(x)$ can be computed for any $x \in \mathbb{R}_+^V$ in $\mathrm{O}(q)$ time.

We show that if $F = F_1$, the problem can be solved in $\mathrm{O}(n^2 q)$ time, where q is the time required for the computation of $F(x)$ for each $x \in \mathbb{R}_+^V$. We also prove that the problem requires $\Omega(n^2 q)$ time, if $F (= F_2)$ is given by an oracle. This implies that our $\mathrm{O}(n^2 q)$ algorithm is *optimal* if $F (= F_1, F_2)$ is given by an oracle. Moreover, we show that the problem can be solved in $\mathrm{O}(n \log^2 n)$ and $\mathrm{O}(n(\log^2 n + q))$ time if $F\ (= F_3)$ is given in a functional form or by an oracle, respectively, and the problem is intractable in general. Table 1 summarizes the complexity results obtained in this paper.

Table 1: Summary of the results obtained in this paper

	F_1	F_2	F_3	General
Functional Form	$\mathrm{O}(n^2q)$	$\mathrm{O}(n^2q)$	$\mathrm{O}(n\log^2 n)$	NP-hard inapproximable
Oracle	$\Theta(n^2q)$	$\Theta(n^2q)$	$\mathrm{O}(n(\log^2 n+q))$	$\Omega(2^{\frac{n}{2}}q)$

q: the time required for computing $F(x)$ for each $x\in\mathbb{R}_+^V$.

We remark that the results above remain true, even if we add the integrality condition $x\in\mathbb{Z}^V$ to the problem.

Our positive results can be applied to the source location problem and the edge-connectivity augmentation problem (see Section 3 for details). These make improvements in complexity results for the problems. Our results imply that the source location problem can be solved in $\mathrm{O}(nm+n^2(q+\log n))$ time if the cost function F is expressed as F_2, e.g., the sum of fixed setup costs and concave running costs for facilities at $v\in V$, and in $\mathrm{O}(n(m+n\log n))$ time if the cost function F is expressed as F_3, i.e., the cost is the sum of fixed setup costs and linear running costs. We remark that the source location problem has been investigated, only when the cost function depends on the fixed setup cost of the facilities (see (14)). Similarly to the source location problem, our results together with the ones in [12, 14] imply that the augmentation problem can be solved in $\mathrm{O}(nm+n^2(q+\log n))$ time if $F=F_1$, and in $\mathrm{O}(n(m+n\log n))$ time if $F=F_3$. We remark that the augmentation problem has been investigated only when the cost function is linear (see (19)).

The rest of the paper is organized as follows. Section 2 introduces some notation and definitions, and Section 3 presents two applications of our covering problem. Sections 4 and 5 investigate the cases in which F is laminar or separable, and F is linear with fixed costs, respectively. Finally, Section 6 considers the general case.

2 Definitions and Preliminaries

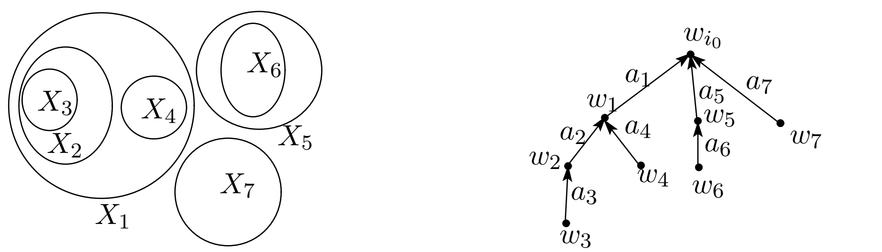
Let V be a finite set with $|V|=n$. A family $\mathcal{F}\subseteq 2^V$ is called *laminar* [6] if for arbitrary two sets $X,Y\in\mathcal{F}$, at least one of the three sets $X\cap Y$, $X-Y$, and $Y-X$ is empty, i.e., $X\cap Y\neq\emptyset$ implies $X\subseteq Y$ or $X\supseteq Y$.

For a laminar family $\mathcal{F}=\{X_i\mid i\in I\}$ define a directed graph $T=(W,A)$

with a vertex set W and an arc set A by

$$
\begin{aligned}
W &= \{w_i \mid i \in I \cup \{i_0\}\} \\
A &= \{a_i = (w_i, w_j) \mid X_i \subsetneq X_j, \mathcal{F} \text{ contains no set } Y \text{ with } X_i \subsetneq Y \subsetneq X_j\} \\
&\quad \cup \{a_i = (w_i, w_{i_0}) \mid X_i \text{ is a maximal set in } \mathcal{F}\},
\end{aligned}
$$

where i_0 is a new index not in I. Since $\mathcal{F}$ is laminar, the graph $T = (W, A)$ is a directed tree toward the root w_{i_0} and is called the *tree representation* (e.g., [6]) of $\mathcal{F}$ (see Fig. 1). For each $X_i \in \mathcal{F}$ let us define the family of the



$\mathcal{F} = \{X_1, X_2, \cdots, X_7\}$ The tree representation $T = (W, A)$ of $\mathcal{F}$

Figure 1:

children, the incremental set, and the depth by

$$
\begin{aligned}
\mathcal{S}(X_i) &= \{X_j \mid a_j = (w_j, w_i) \in A\}, \\
\Delta X_i &= X_i \setminus \bigcup_{X_j \in \mathcal{S}(X_i)} X_j, \\
h(X_i) &= |\{X_j \mid X_j \in \mathcal{F} \text{ with } X_j \supseteq X_i\}|.
\end{aligned}
$$

A function $F : \mathbb{R}^V \to \mathbb{R}$ is called *monotone nondecreasing* (or simply *monotone*) if $F(x) \leq F(y)$ holds for arbitrary two vectors $x, y \in \mathbb{R}^V$ with $x \leq y$, and *concave* if

$$\alpha F(x) + (1 - \alpha)F(y) \leq F(\alpha x + (1 - \alpha)y) \tag{5}$$

holds for arbitrary two vectors $x, y \in \mathbb{R}^V$ and real α with $0 \leq \alpha \leq 1$.

Now, we formulate the problem of minimizing a monotone concave function with laminar covering constraints. Given a laminar family $\mathcal{F} \subseteq 2^V$, a monotone concave function $F : \mathbb{R}_+^V \to \mathbb{R}_+$, and a demand function $d : \mathcal{F} \to \mathbb{R}_+$, we consider the problem given by

$$
\begin{aligned}
(\mathbf{P}) \qquad & \text{Minimize} && F(x) && && (6) \\
& \text{subject to} && x(X) \geq d(X) && (X \in \mathcal{F}), && (7) \\
& && x(v) \geq 0 && (v \in V), && (8)
\end{aligned}
$$

where $x(X) = \sum_{v \in X} x(v)$. We assume without loss of generality that $F(\mathbf{0}) = 0$.

For a function $d : \mathcal{F} \to \mathbb{R}$ we also define the increment $\Delta d : \mathcal{F} \to \mathbb{R}$ by $\Delta d(X) = d(X) - \sum_{Y \in \mathcal{S}(X)} d(Y)$. If $\Delta d(X) \leq 0$, we can remove constraint $x(X) \geq d(X)$ from (7). Hence we assume that every set $X \in \mathcal{F}$ satisfies

$$\Delta d(X) \quad > \quad 0. \tag{9}$$

3 Applications of Our Covering Problem

In this section we introduce two network problems as examples of our problem.

3.1 Source Location Problem in Undirected Flow Networks

Let $\mathcal{N} = (G = (V, E), u)$ be an undirected network with a vertex set V, an edge set E, and a capacity function $u : E \to \mathbb{R}_+$. For convenience, we regard $\mathcal{N}$ as a symmetric directed graph $\hat{\mathcal{N}} = (\hat{G} = (V, \hat{E}), \hat{u})$ defined by $\hat{E} = \{(v, w), (w, v) \mid \{v, w\} \in E\}$ and $\hat{u}(v, w) = \hat{u}(w, v) = u(\{v, w\})$ for any $\{v, w\} \in E$. We also often write $u(v, w)$ instead of $u(\{v, w\})$.

A flow $\varphi : \hat{E} \to \mathbb{R}_+$ is *feasible* with a supply $x : V \to \mathbb{R}_+$ if it satisfies the following conditions:

$$\partial\varphi(v) \stackrel{\text{def}}{=} \sum_{(v,w) \in \hat{E}} \varphi(v, w) - \sum_{(w,v) \in \hat{E}} \varphi(w, v) \leq x(v) \quad (v \in V), \tag{10}$$

$$0 \leq \varphi(e) \leq \hat{u}(e) \qquad (e \in \hat{E}). \tag{11}$$

Here (10) means that the net out-flow value $\partial\varphi(v)$ at $v \in V$ is at most the supply at v.

Given an undirected network $\mathcal{N}$ with a demand $k > 0$ and a cost function $F : \mathbb{R}_+^V \to \mathbb{R}_+$, the source location problem considered in [2] is to find a minimum-cost supply x such that for each $v \in V$ there is a feasible flow φ such that the sum of the net in-flow value and the supply at v is at least k. The problem is rewritten as follows.

$$\begin{array}{lll} \text{Minimize} & F(x) & \\ \text{subject to} & \forall\ v \in V,\ \exists \text{ a feasible flow } \varphi_v \text{ in } \mathcal{N} \text{ with a supply } x\text{:} & \\ & \quad -\partial\varphi_v(v) + x(v) \geq k, & (12) \\ & x(v) \geq 0 \qquad (v \in V). & (13) \end{array}$$

Note that the flow φ_v $(v \in V)$ in (12) may depend on $v \in V$.

The above-mentioned source location problem was investigated in [2, 17] in a special case where the cost function is given as

$$F(x) = \sum_{v \in V : x(v) > 0} b_v. \tag{14}$$

Namely, the cost function depends only on the fixed setup cost of the facilities at vertices $v \in V$ with $x(v) > 0$, and is independent of the positive supply value $x(v)$. Some variants of the problem such as non-uniform demand and directed network cases are also examined in [2, 7, 8, 18].

We show that (12) can be represented by laminar covering constraints as (7).

A *cut* is a nonempty proper subset of V. For a cut X, $\kappa(X)$ denotes the cut capacity of X, i.e.,

$$\kappa(X) \quad = \sum_{\substack{\{v,w\}\in E:\\ v\in X, w\in V-X}} u(v,w),$$

where note that $\kappa(X) = \kappa(V \setminus X)$. By the max-flow min-cut theorem (see, e.g., [1]), (12) is equivalent to

$$\kappa(X) + x(X) \quad \geq \quad k \quad (v \in X \subseteq V). \tag{15}$$

A nonempty $X \subseteq V$ is called an *extreme set* [19] if $\kappa(Y) > \kappa(X)$ holds for each nonempty $Y \subsetneq X$, and let $\mathcal{F}$ be the family of all extreme sets of $\mathcal{N}$. Then, some redundant inequalities in (15) can be removed as follows.

Lemma 1 *Under the nonnegativity condition* (13) *the constraint* (12) (*or* (15)) *are equivalent to*

$$\kappa(X) + x(X) \quad \geq \quad k \qquad (X \in \mathcal{F}). \tag{16}$$

Proof. Let X be any non-extreme set and x be any nonnegative supply satisfying the constraint (16). Then there exists an extreme set $Y \in \mathcal{F}$ such that $Y \subsetneq X$ and we have $\kappa(X) + x(X) \geq \kappa(Y) + x(Y) \geq k$ because of the nonnegativity of x. □

It is known that the family $\mathcal{F}$ of extreme sets is laminar. (For, κ is *posi-modular* [13], i.e., for any $X, Y \subseteq V$, $\kappa(X) + \kappa(Y) \geq \kappa(X \setminus Y) + \kappa(Y \setminus X)$. If there exist $X, Y \in \mathcal{F}$ such that $X \cap Y, X \setminus Y$, and $Y \setminus X$ are all nonempty, then the posi-modularity implies that $\kappa(X) \geq \kappa(X \setminus Y)$ or $\kappa(Y) \geq \kappa(Y \setminus X)$, which contradicts the extremality of X and Y.)

For any $X \in \mathcal{F}$ we denote by $d(X)$ the *deficiency* of X defined by

$$d(X) \quad = \quad \max\{k - \kappa(X), 0\}. \tag{17}$$

Then it follows from the argument given above that (12) can be represented as laminar covering constraints (7) with d given by (17), and hence the source location problem can be formulated as (**P**) in Section 2. We remark that given an undirected network $\mathcal{N} = (G = (V, E), u)$ and a demand $k\,(> 0)$, the family $\mathcal{F}$ of all extreme sets, as well as the deficiency $d : \mathcal{F} \to \mathbb{R}_+$, can be computed in $\mathrm{O}(n(m + n \log n))$ time [12]. Therefore, our results for the laminar covering problem immediately imply the ones for the source location problem considered in [2, 17]. In particular, if the cost function is a separable monotone concave function, i.e., the sum of fixed setup costs and concave running costs for facilities at $v \in V$, the source location problem can be solved in $\mathrm{O}(nm + n^2(q + \log n))$ time. Moreover, we can solve the problem in $\mathrm{O}(n(m + n \log n))$ time if the cost is the sum of fixed setup costs and linear running costs.

3.2 Edge-connectivity Augmentation in Undirected Flow Networks

Let $\mathcal{N} = (G = (V, E), u)$ be an undirected network with a capacity function $u : E \to \mathbb{R}_+$. We call $\mathcal{N}$ *k-edge-connected* if for every two nodes $v, w \in V$ the maximum flow value between v and w is at least k. Given an undirected network $\mathcal{N}$ and a positive real k, the edge-connectivity augmentation problem is to find a smallest set D of new edges with capacity $\mu_D : D \to \mathbb{R}_+$ for which $\mathcal{N}' = (G' = (V, E \cup D), u \oplus \mu_D)$ is k-edge-connected, where $u \oplus \mu_D$ is the direct sum of u and μ_D (see, e.g., [3, 4, 5, 12, 14, 15, 19]). It is known that the problem can be solved in polynomial time. In fact, the node-cost edge-connectivity augmentation problem [5] is polynomially solvable, while the edge-cost one is NP-hard. Here the node-cost is defined by $\sum_{v \in V} c_v(\partial \mu_D(v))$ for a given c_v $(v \in V)$, where $\partial \mu_D(v) = \sum_{e \in D : e \ni v} \mu_D(e)$, and the edge-cost is defined by $\sum_{e \in D} c_e(\mu_D(e))$ for a given c_e $(e \in \binom{V}{2})$. We claim that the constraints of the edge-connectivity augmentation problem can be regarded as the laminar covering constraints, together with nonnegativity constraints.

From the max-flow min-cut theorem, we can see that $x = \partial \mu_D$ must satisfy $\kappa(X) + x(X) \geq k$ for any nonempty $X \subsetneq V$ (see, e.g., [5]). Similarly to the source location problem in the previous section, this implies

$$x(X) \geq d(X) \qquad (X \in \mathcal{F}), \tag{18}$$

where $d(X)$ is given by (17), and $\mathcal{F}$ is the family of all extreme sets in $\mathcal{N}$.

On the other hand, by using splitting technique [9, 10, 11], any x satisfying (18) can create a capacity function $\mu_D : D \to \mathbb{R}_+$ for which $\mathcal{N}'$ is k-edge-connected [9, 10, 11] and moreover, such an x of minimum $x(V)$ can be found in $\mathrm{O}(n(m + n \log n))$ time [14], which proves our claim.

Therefore, the node-cost edge-connectivity augmentation problem can be represented by the laminar covering problem with a linear cost function $F : \mathbb{R}^V_+ \to \mathbb{R}_+$, i.e.,

$$F(x) = \sum_{v \in V} c_v x(v). \tag{19}$$

Our results extend the existing ones for the augmentation problem (see, e.g., [3, 5, 12, 14]). Especially when F is given by (2), the algorithm proposed in this paper together with the ones in [12, 14] solves the augmentation problem in $\mathrm{O}(nm + n^2(q + \log n))$ time. Moreover, if F is given by (4), we can solve the problem in $\mathrm{O}(n(m + n \log n))$ time.

4 The Laminar Cost Case

In this section we consider the problem whose cost function is given by F_1, i.e.,

$$\begin{array}{llll} (\mathbf{P}_1) & \text{Minimize} & \displaystyle\sum_{X \in \mathcal{F}} f_{\Delta X}(x[\Delta X]), & \\ & \text{subject to} & x(X) \geq d(X) \qquad (X \in \mathcal{F}), & (20) \\ & & x(v) \geq 0 \qquad\qquad (v \in V), & \end{array}$$

where $f_{\Delta X}$ is a monotone concave function on ΔX with $f_{\Delta X}(\mathbf{0}) = 0$.

We shall present an $\mathrm{O}(n^2 q)$ time algorithm for the problem and show the $\Omega(n^2 q)$ time bound when the cost function is given by an oracle.

4.1 Structural Properties of Optimal Solutions

This section reveals structural properties of optimal solutions of Problem $(\mathbf{P}_1)$ in (20), which makes it possible for us to devise a polynomial algorithm for Problem $(\mathbf{P}_1)$.

Let $\mathcal{F}$ be a laminar family on V, and $T = (W, A)$ be the tree representation of $\mathcal{F}$. Consider the problem projected on $Y \in \mathcal{F}$ that is given as

$$\begin{array}{llll} (\mathbf{P}_Y) & \text{Minimize} & \displaystyle\sum_{X \in \mathcal{F}: X \subseteq Y} f_{\Delta X}(x[\Delta X]) & \\ & \text{subject to} & x(X) \geq d(X) \qquad (X \in \mathcal{F}, X \subseteq Y), & (21) \\ & & x(v) \geq 0 \qquad\qquad (v \in V). & \end{array}$$

From Assumption (9), the following lemma holds.

Lemma 2 *For any feasible solution $x \in \mathbb{R}^V$ of Problem $(\mathbf{P}_Y)$ in (21), there exists a feasible solution $y \in \mathbb{R}^V$ of (21) such that $y(Y) = d(Y)$ and $y[X] \leq x[X]$ for all $X \subseteq Y$.*

We first show properties of optimal solutions of $(\mathbf{P}_Y)$, from which we derive properties of optimal solutions of $(\mathbf{P}_1)$.

Lemma 3 *For a minimal $Y \in \mathcal{F}$, Problem $(\mathbf{P}_Y)$ has an optimal solution $x = z_v$ for some $v \in Y$ such that*

$$z_v(t) = \begin{cases} d(Y) \ (= \Delta d(Y)) & (t = v) \\ 0 & (t \in V \setminus \{v\}). \end{cases} \tag{22}$$

Proof. Because of Lemma 2 and the monotonicity of the cost function, there is an optimal solution x such that

$$x(Y) = d(Y) \tag{23}$$

and $x(v) = 0$ for $v \in V \setminus Y$. Moreover, any feasible solution x of $(\mathbf{P}_Y)$ satisfying (23) is described as

$$x = \sum_{v \in Y} \frac{x(v)}{d(Y)} z_v,$$

where z_v is defined by (22). Since z_v is feasible, $\frac{x(v)}{d(Y)} \geq 0$ $(v \in Y)$, and $\sum_{v \in Y} \frac{x(v)}{d(Y)} = 1$, it follows from the concavity of $f_{\Delta Y}$ that

$$f_{\Delta Y}(z_{v^*}[\Delta Y]) \leq f_{\Delta Y}(x[\Delta Y])$$

for some $v^* \in Y$. □

We next show properties of optimal solutions for non-minimal $Y \in \mathcal{F}$.

Lemma 4 *Let Y be a non-minimal set in $\mathcal{F}$. Then there exists an optimal solution x of Problem $(\mathbf{P}_Y)$ such that for some $v \in \Delta Y$*

$$\begin{aligned} x(t) &= \begin{cases} \Delta d(Y) & (t = v) \\ 0 & (t \in (V \setminus Y) \cup (\Delta Y \setminus \{v\})), \end{cases} \\ x(X) &= d(X) \quad (X \in \mathcal{S}(Y)), \end{aligned} \tag{24}$$

or for some $X \in \mathcal{S}(Y)$

$$\begin{aligned} x(Z) &= \begin{cases} d(X) + \Delta d(Y) & (Z = X) \\ d(Z) & (Z \neq X,\ Z \in \mathcal{S}(Y)), \end{cases} \\ x(v) &= 0 \quad (v \in (V \setminus Y) \cup \Delta Y). \end{aligned} \tag{25}$$

Proof. By Lemma 2 and the monotonicity of the cost function, there is an optimal solution x such that $x(Y) = d(Y)$ and $x(v) = 0\,(v \in V - Y)$. For such an x, we define δ_v $(v \in \Delta Y)$ and δ_X $(X \in \mathcal{S}(Y))$ by

$$\begin{aligned} \delta_v &= \frac{x(v)}{\Delta d(Y)} \quad (v \in \Delta Y), \\ \delta_X &= \frac{x(X) - d(X)}{\Delta d(Y)} \quad (X \in \mathcal{S}(Y)). \end{aligned}$$

Then we have

$$\sum_{v \in \Delta Y} \delta_v + \sum_{X \in \mathcal{S}(Y)} \delta_X = 1, \text{ and } \delta_v,\ \delta_X \geq 0 \ \ (v \in \Delta Y,\ X \in \mathcal{S}(Y)). \tag{26}$$

We also define z_v $(v \in \Delta Y)$ and y_X $(X \in \mathcal{S}(Y))$ by

$$
z_v(t) \;=\; \begin{cases} \Delta d(Y) & (t = v) \\ 0 & (t \in (V \setminus Y) \cup (\Delta Y \setminus \{v\})) \\ \dfrac{d(X)}{x(X)} x(t) & (t \in X,\ X \in \mathcal{S}(Y)), \end{cases}
$$

$$
y_X(t) \;=\; \begin{cases} \dfrac{d(X) + \Delta d(Y)}{x(X)} x(t) & (t \in X) \\ \dfrac{d(Z)}{x(Z)} x(t) & (t \in Z,\ Z \neq X,\ Z \in \mathcal{S}(Y)) \\ 0 & (t \in (V \setminus Y) \cup \Delta Y). \end{cases}
$$

We can easily see that z_v and y_X are feasible and satisfy the properties shown in Lemma 4, and

$$
x \;=\; \sum_{v \in \Delta Y} \delta_v z_v + \sum_{X \in \mathcal{S}(Y)} \delta_X y_X.
$$

Since δ_v and δ_X satisfy (26) and F is concave, we have

$$
F(x) \geq \sum_{v \in \Delta Y} \delta_v F(z_v) + \sum_{X \in \mathcal{S}(Y)} \delta_X F(y_X).
$$

Hence at least one of z_v $(v \in \Delta Y)$ and y_X $(X \in \mathcal{S}(Y))$ is optimal. $\square$

Let $W^* = \{w_i \mid X_i \in \mathcal{F}\}$. A partition $\mathcal{P} = \{P_1, \cdots, P_k\}$ of W^* is called a *path-partition* of W^* if each $P_j = \{w_{j_0}, w_{j_1}, \cdots, w_{j_{r_j}}\} \in \mathcal{P}$ forms a directed path $w_{j_0} \to w_{j_1} \to \cdots \to w_{j_{r_j}}$ in $T = (W, A)$ with $\Delta X_{j_0} \neq \emptyset$.

We are now ready to describe our structure theorem.

Theorem 5 *Problem* $(\mathbf{P}_1)$ *in* (20) *has an optimal solution* x^* *that can be obtained from a path-partition* $\mathcal{P} = \{P_1, \cdots, P_k\}$ *of* W^* *together with* $v_j \in \Delta X_{j_0}$ $(j = 1, \cdots, k)$ *as follows.*

$$
x^*(t) = \begin{cases} \displaystyle\sum_{w_{j_i} \in P_j} \Delta d(X_i) & (t = v_j,\ j = 1, \cdots, k) \\ 0 & (t \in V \setminus \{v_j \mid j = 1, \cdots, k\}). \end{cases}
$$

Proof. We proceed by induction on the height h of $T = (W, A)$.

Let us first consider the case when $h = 1$. It follows from Lemma 3 that Problem $(\mathbf{P}_Y)$ for any $Y \in \mathcal{F}$ has an optimal solution of form (22). We denote by x_Y such an optimal solution. Then by the separability and monotonicity of F, $x = \sum_{Y \in \mathcal{F}} x_Y$ is an optimal solution of $(\mathbf{P}_1)$ in (20). This x clearly satisfies the condition in the present theorem.

Next, assume that the statement in the theorem is true for $h \leq \ell$ for some integer $\ell \geq 1$. Then consider the case when $h = \ell + 1$. From Lemma 4, Problem $(\mathbf{P}_Y)$ for a maximal set Y in $\mathcal{F}$ has an optimal solution of form (24) or (25). If $x = z_v$ of form (24) is optimal, then we consider Problem $(\mathbf{P}_X)$ for each $X \in \mathcal{S}(Y)$. By the induction hypothesis on h, each such problem has an optimal solution x_X based on some path-partition $\mathcal{P}_X$ of X. Define

$$\begin{aligned} \mathcal{P}_Y &= \bigcup_{X \in \mathcal{S}(Y)} \mathcal{P}_X \cup \{Y\}, \\ x_Y &= \sum_{X \in \mathcal{S}(Y)} x_X + e_v, \end{aligned}$$

where $e_v(t) = \Delta d(Y)$ if $t = v$, and $e_v(t) = 0$ otherwise. We can see that x_Y is an optimal solution of Problem $(\mathbf{P}_Y)$ and is based on path-partition $\mathcal{P}_Y$.

On the other hand, if $x = y_X$ of form (25) is an optimal solution of Problem $(\mathbf{P}_Y)$ for a maximal $Y \in \mathcal{F}$, then we consider Problem $(\mathbf{P}_Z)$ (in (21)) for each $Z \in \mathcal{S}(Y)$ with $Z \neq X$ and also consider the following problem.

$$\begin{array}{lll} \text{Minimize} & \displaystyle\sum_{Z \in \mathcal{F}: Z \subseteq X} f_{\Delta Z}(x[\Delta Z]) & \\ \text{subject to} & x(X) \geq d(X) + \Delta d(Y), & \\ & x(Z) \geq d(Z) \qquad (Z \in \mathcal{F}, Z \subsetneq X), & \\ & x(v) \geq 0 \qquad (v \in V). & \end{array} \tag{27}$$

By induction on h, Problem $(\mathbf{P}_Z)$ (in (21)) for each $Z \in \mathcal{S}(Y)$ with $Z \neq X$ has an optimal solution x_Z based on a path-partition $\mathcal{P}_Z$, and similarly, Problem (27) has an optimal solution x_X based on a path-partition $\mathcal{P}_X$. Hence we can construct an optimal solution $x_Y = \sum_{X \in \mathcal{S}(Y)} x_X$ based on the path-partition

$$\mathcal{P}_Y = \bigcup_{Z \in \mathcal{S}(Y): Z \neq X} \mathcal{P}_Z \,\cup\, (\mathcal{P}_X \setminus \{P\}) \cup \{P \cup \{w_Y\}\} \;,$$

where P denotes the set in $\mathcal{P}_X$ that contains the node corresponding to X, and w_Y denotes the node corresponding to Y.

Finally, by letting $x = \sum_{Y \in \mathcal{S}(X_{i_0})} x_Y$ and $\mathcal{P} = \bigcup_{Y \in \mathcal{S}(X_{i_0})} \mathcal{P}_Y$, we have a desired optimal solution x and its corresponding path-partition $\mathcal{P}$. □

4.2 A Polynomial Algorithm

In this section we present a polynomial algorithm for Problem $(\mathbf{P}_1)$ in (20). The algorithm applies dynamic programming to compute an optimal path-partition of W^*.

For any $Y \in \mathcal{F}$, we denote by w_Y the node in W corresponding to Y, and by $w_{j_0}(= w_Y), w_{j_1}, \cdots, w_{j_{h(Y)-1}}, w_{j_{h(Y)}}(= w_{i_0})$ the directed path from

w_Y to the root w_{i_0}. Our dynamic programming solves the following $h(Y)$ problems for each $Y \in \mathcal{F}$.

$$
\begin{aligned}
(\mathbf{P}(Y,k)) \qquad & \text{Minimize} && \sum_{X\in\mathcal{F}:X\subseteq Y} f_{\Delta X}(x[\Delta X]) && (28)\\
& \text{subject to} && x(Y) \geq d(Y) + \sum_{i=1}^{k} \Delta d(X_{j_i}), && (29)\\
& && x(X) \geq d(X) \qquad (X \in \mathcal{F}, X \subsetneq Y), && (30)\\
& && x(v) \geq 0 \qquad (v \in V), && (31)
\end{aligned}
$$

where $Y \in \mathcal{F}$ and $k = 0, 1, \cdots, h(Y) - 1$. Let $\alpha(Y,k)$ denote the optimal value of Problem $(\mathbf{P}(Y,k))$. By Theorem 5, these problems $(\mathbf{P}(Y,k))$ have optimal solutions based on a path-partition $\mathcal{P}$ of $\{w_i \mid X_i \in \mathcal{F}, X_i \subseteq Y\}$. For $P_j \in \mathcal{P}$ containing $w_Y \in W$ (that corresponds to Y), let v_j be the node in ΔX_{j_0} given in Theorem 5. We store v_j as $\beta(Y,k)$. It follows from Lemmas 3 and 4 that $\alpha(Y,k)$ and $\beta(Y,k)$ can be computed as follows.

For each minimal $Y \in \mathcal{F}$ (which corresponds to a leaf in T) the following z_v^k for some $v \in Y$ gives an optimal solution, due to Lemma 3.

$$
z_v^k(t) = \begin{cases} \sum_{i=0}^{k} \Delta d(X_{j_i}) & (t = v) \\ 0 & (t \in \Delta Y \setminus \{v\}). \end{cases}
$$

Hence we have

$$
(\alpha(Y,k), \beta(Y,k)) = \left(\min_{v\in Y} f_Y(z_v^k), \arg\min_{v\in Y} f_Y(z_v^k) \right) \tag{32}
$$

for $k = 0, \cdots, h(Y)-1$, where $\arg\min_{v\in Y} f_Y(z_v^k)$ denotes a vertex $v^* \in Y$ satisfying $f_Y(z_{v^*}^k) = \min_{v\in Y} f_Y(z_v^k)$.

For a non-minimal $Y \in \mathcal{F}$, Lemma 4 validates the following recursive formulas.

$$
\begin{aligned}
\alpha(Y,k) \;&=\; \min \Bigg\{ \min_{X\in\mathcal{S}(Y)} \Big\{ \alpha(X,k+1) + \sum_{\substack{Z\in\mathcal{S}(Y)\\ Z\neq X}} \alpha(Z,0) \Big\}, \\
&\qquad \min_{v\,\in\,\Delta Y} \Big\{ f_{\Delta Y}(z_v^k) + \sum_{X\in\mathcal{S}(Y)} \alpha(X,0) \Big\} \Bigg\}, && (33)
\end{aligned}
$$

$$
\beta(Y,k) \;=\; \begin{cases} \beta(X,k+1) & \text{if } \alpha(Y,k) = \alpha(X,k+1) + \sum_{\substack{Z\in\mathcal{S}(Y)\\ Z\neq X}} \alpha(Z,0), \\ v & \text{if } \alpha(Y,k) = f_{\Delta Y}(z_v^k) + \sum_{X\in\mathcal{S}(Y)} \alpha(X,0). \end{cases} \tag{34}
$$

By using (32), (33), and (34), our algorithm first computes each α and β from the leaves toward root w_{i_0} of T. Then we obtain an optimal value

$\sum_{X \in \mathcal{S}(X_{i_0})} \alpha(X, 0)$ of Problem $(\mathbf{P}_1)$ in (20). Next, we compute an optimal solution x^* by using β from the root toward the leaves of T.

Our algorithm is formally described as follows.

Algorithm DP

Input: A laminar family $\mathcal{F}$, a demand function $d : \mathcal{F} \to \mathbb{R}_+$, and a cost function F as in (20).

Output: An optimal solution x^* for Problem $(\mathbf{P}_1)$ in (20).

Step 0. $\tilde{W} := W$.

Step 1. (Compute α and β) While $\tilde{W} \neq \{w_{i_0}\}$ do

Choose an arbitrary leaf $w \in \tilde{W}$ of $T[\tilde{W}]$.

/* Let Y be the set in $\mathcal{F}$ corresponding to w. */

(1-I) Compute $\alpha(Y, k)$ and $\beta(Y, k)$ for $k = 0, \cdots, h(Y)-1$ by using either (32) or ((33) and (34)).

(1-II) $\tilde{W} := \tilde{W} \setminus \{w\}$.

Step 2. $\tilde{W} := W \setminus \{w_{i_0}\}$, and $x^*(v) := 0$ for all $v \in V$.

Step 3. (Compute an optimal x^*) While $\tilde{W} \neq \emptyset$ do

Choose an arbitrary node w of $T[\tilde{W}]$ having no leaving arc.

/* Let Y be the set in $\mathcal{F}$ corresponding to w, w_{j_0} be the node in W corresponding to X_{j_0} such that $\beta(Y, 0) \in \Delta X_{j_0}$ and let $w_{j_0} \to w_{j_1} \to \cdots \to w_{j_l} (= w)$ be a directed path in $T[\tilde{W}]$. */

(3-I) $x^*(\beta(Y, 0)) := \sum_{i=0}^{l} \Delta d(X_{j_i})$.

(3-II) $\tilde{W} := \tilde{W} \setminus \{w_{j_0}, \cdots, w_{j_l}\}$.

Step 4. Output x^* and halt. □

Here $T[\tilde{W}]$ denotes the subtree of T induced by $\tilde{W}$.

We now analyze the complexity of Algorithm DP. Steps 0, 2, 3, and 4 require $\mathrm{O}(n)$ time. For Step 1, if $Y \in \mathcal{F}$ is a minimal set in $\mathcal{F}$ (i.e., w is a leaf of T), then we compute $\alpha(Y, k)$ and $\beta(Y, k)$ $(k = 0, \cdots, h(Y) - 1)$ in $\mathrm{O}(h(Y)|\Delta Y|q)$ time. On the other hand, if Y is not minimal, then $\alpha(Y, k)$ and $\beta(Y, k)$ $(k = 0, \cdots, h(Y) - 1)$ can be computed in $\mathrm{O}(h(Y)(|\mathcal{S}(Y)| +$

$|\Delta Y|q)$) time, since $\alpha(Y,k)$ is obtained by taking the minimum among $|\mathcal{S}(Y)|+|\Delta Y|$ values. Hence Step 1 requires

$$\mathrm{O}\left(\sum_{Y\in\mathcal{F}} h(Y)(|\mathcal{S}(Y)|+|\Delta Y|q)\right) \quad = \quad \mathrm{O}(n^2q)$$

time. Here, note that $h(Y)\leq n$, $\sum_{Y\in\mathcal{F}}|\Delta Y|=|V|=n$, and $\sum_{Y\in\mathcal{F}}|\mathcal{S}(Y)|\leq|\mathcal{F}|-1\leq 2n-2$, since $\mathcal{F}$ is laminar. Therefore, we have the following theorem.

Theorem 6 *Algorithm* DP *computes an optimal solution for Problem* $(\mathbf{P}_1)$ *in* $\mathrm{O}(n^2q)$ *time.* □

4.3 The Lower Bound for the Time Complexity When F is Given by an Oracle

In this section we consider a lower bound for the time complexity of our problem when F is given by an oracle. We shall show that the oracle has to be invoked $\Omega(n^2)$ times even if we know in advance that F is given in the form of (3), i.e., $F=\sum_{v\in V}f_v(x(v))$. This, together with Theorem 6, implies that Algorithm DP is optimal if F is given by an oracle.

Suppose n is a positive even number. Let $g_0:\mathbb{R}_+\to\mathbb{R}_+$ be a monotone increasing and strictly concave function (e.g., $g_0(x)=\frac{-1}{x+1}+1$ $(x\geq 0)$), and for each $i=\frac{n}{2}+1,\frac{n}{2}+2,\cdots,n$ define $g_i:\mathbb{R}_+\to\mathbb{R}_+$ by

$$g_i(\xi) \quad = \quad \begin{cases} g_0(\frac{n}{2}+1)-g_0(i-\frac{n}{2}) & (\xi>0)\\ 0 & (\xi=0).\end{cases}$$

Then, let $V=\{v_1,\cdots,v_n\}$ and consider a problem instance I obtained by

$$\begin{aligned} &\text{a laminar family } \mathcal{F}=\left\{X_i=\{v_1,\cdots,v_{\frac{n}{2}+i}\}\ \middle|\ i=0,\cdots,\tfrac{n}{2}\right\},\\ &\text{a demand function } \ d:d(X_i)=i+1 \quad (i=0,1,\cdots,\tfrac{n}{2}),\\ &\text{a cost function } F(x)=\sum_{v\in V}f_v(x(v)),\end{aligned} \tag{35}$$

where

$$f_{v_i}(\xi) \quad = \quad \begin{cases} g_0(\xi) & (v_i\in X_0)\\ g_i(\xi) & (v_i\in V-X_0).\end{cases}$$

From Theorem 5, we can get an optimal solution of this instance I as follows.

$$x_{(v_i,k)}(v) \quad = \quad \begin{cases} k & (v=v_i)\\ \frac{n}{2}+1-k & (v=v_{\frac{n}{2}+k})\\ 0 & (v\in V\setminus\{v_i,v_{\frac{n}{2}+k}\})\end{cases} \tag{36}$$

for some $v_i \in X_0$ and $k \in \{1, 2, \cdots, \frac{n}{2}+1\}$. Here, when $k = \frac{n}{2}+1$, we mean $x_{(v_i,k)}(v) = k$ if $v = v_i$, and 0 otherwise. Note that the optimal value for the instance I is $g_0(\frac{n}{2}+1)$, and any optimal solution can be represented as (36) because $g_0 = f_{v_i}(v_i \in X_0)$ is strictly concave and for each $v \in V \setminus X_0$ $f_v(x(v))$ is identical for $x(v) > 0$. For the instance I, suppose that F is given by an oracle and that an algorithm A can compute an optimal solution by invoking the oracle for $F(x)$ for each $x \in S$ with $|S| \leq \frac{n}{2}(\frac{n}{2}+1) - 1$. Then we are led to a contradiction as shown below.

For each $x \in S$, we know the value of $F(x) = \sum_{v \in V} f_v(x(v))$ after executing Algorithm A. Furthermore, let us assume that we know that F is separable, and $f_v(k)$ for all the pairs $(v, k) \notin X_0 \times \{1, \cdots, \frac{n}{2}+1\}$ such that $x(v) = k$ for some $x \in S$. Then, in order to compute the values $f_v(k)$ with $(v, k) \in X_0 \times \{1, \cdots, \frac{n}{2}+1\}$, we solve a linear system $B\eta = b$ of equations, one for each $x \in S$ that corresponds to $\sum_{v \in V} f_v(x(v)) = F(x)$. Here, B is a $|S| \times \frac{n}{2}(\frac{n}{2}+1)$ 0-1 matrix, b is a column $|S|$-vector, and η is an unknown $\frac{n}{2}(\frac{n}{2}+1)$-column vector $\eta = (\eta_v(k) \mid v \in X_0, k = 1, \cdots, \frac{n}{2}+1)$, where $\eta_v(k)$ denotes the unknown variable representing the possible value of $f_v(k)$. Since the number of the rows in B ($\leq \frac{n}{2}(\frac{n}{2}+1) - 1$) is less than the number of the columns, it follows from elementary linear algebra that there exists a non-zero vector $\varphi \in \mathbb{R}^{\frac{n}{2}(\frac{n}{2}+1)}$ such that

$$\eta^{\varepsilon} = (f_v(k) \mid v \in X_0, k \in \{1, \cdots, \tfrac{n}{2}+1\}) + \varepsilon\varphi \quad (-\infty < \varepsilon < +\infty)$$

is a solution of $B\eta = b$. Note that the $\varphi \in \mathbb{R}^{\frac{n}{2}(\frac{n}{2}+1)} \setminus \{\mathbf{0}\}$ must satisfy

$$\varphi_{v'}(k') > 0, \qquad \varphi_{v''}(k'') < 0 \tag{37}$$

for some v', v'', k', and k'', since B is a 0-1 matrix. For a sufficiently small $\varepsilon(> 0)$ let $f_v^+, f_v^- : \mathbb{R}_+ \to \mathbb{R}_+$ $(v \in X_0)$ be monotone concave functions such that $f_v^+(k) = \eta_v^{+\varepsilon}(k)$ and $f_v^-(k) = \eta_v^{-\varepsilon}(k)$. Since $|S|$ is finite and g_0 is strictly concave, f_v^+ and f_v^- are well-defined.

For the instance I and any $S \subseteq \mathbb{R}_+^V$ with $|S| \leq \frac{n}{2}(\frac{n}{2}+1) - 1$, we define two instances $\mathrm{I}^{\pm}$, each of which has a laminar family $\mathcal{F}$ and a demand function d as given in (35), but has a cost function $F^{\pm}$ different from F defined by

$$\begin{aligned} F^+ &= \sum_{v \in X_0} f_v^+ + \sum_{v \in V \setminus X_0} f_v, \\ F^- &= \sum_{v \in X_0} f_v^- + \sum_{v \in V \setminus X_0} f_v. \end{aligned}$$

Let

$$C = \{(v^*, k^*) \mid \varphi_{v^*}(k^*) = \max\{\varphi_v(k) \mid v \in X_0,\ k \in \{1, 2, \cdots, \tfrac{n}{2}\}\}.$$

Then it follows from (37) that if $(v, k) \notin C$, then $x_{(v,k)}$ given by (36) is not an optimal solution of I^- and otherwise ($(v, k) \in C$), $x_{(v,k)}$ is not an optimal solution of I^+.

Now, let y be an optimal solution of I obtained by Algorithm A. Since g_0 is strictly concave, we have $y = x_{(v,k)}$ for some (v, k). Note that

$$F(x) = F^-(x) = F^+(x) \quad (x \in S).$$

Hence we cannot distinguish any of the three functions F, F^+, and F^- from the others based on the function values on S, so that y should also be an optimal solution of problem instances I^+ and I^-. However, y cannot be an optimal solution of both I^+ and I^- as shown above, which is a contradiction.

Summing up the arguments, we get

Lemma 7 *There exists a problem instance of* $(\mathbf{P}_1)$ *in* (20) *that requires at least* $\frac{n}{2}(\frac{n}{2}+1)$ *calls to the oracle for* F *even if* F *is a separable monotone concave function (i.e.,* $F = \sum_{v \in V} f_v$ *with monotone concave functions* $f_v : \mathbb{R}_+ \to \mathbb{R}_+$ $(v \in V)$*).*

This implies the following theorem.

Theorem 8 *If* F *is given by an oracle, then Problem* (20) *requires* $\Omega(n^2 q)$ *time.*

We can easily see that Lemma 7 still holds even if each f_v is given by an oracle.

Theorem 9 *Let* F *be a separable monotone concave function (i.e.,* $F = \sum_{v \in V} f_v$ *with monotone concave functions* $f_v : \mathbb{R}_+ \to \mathbb{R}_+$ $(v \in V)$*). If each* f_v *is given by an oracle, then Problem* (20) *requires* $\Omega(n^2 q)$ *time.*

Notice that Algorithm DP given in Section 4.2 is optimal, due to this theorem.

Theorem 10 *If the cost function* $F = \sum_{X \in \mathcal{F}} f_{\Delta X}$ *is given by an oracle, then Problem* (20) *requires* $\Theta(n^2 q)$ *time.*

5 The Linear Cost Case

In this section we consider the problem whose cost function is given as F_3, i.e.,

$$\begin{array}{llll} (\mathbf{P}_3) & \text{Minimize} & \displaystyle\sum_{v \in V : x(v) > 0} (a_v x(v) + b_v) & \\ & \text{subject to} & x(X) \geq d(X) \qquad (X \in \mathcal{F}), & (38) \\ & & x(v) \geq 0 \qquad (v \in V), & \end{array}$$

where a_v and b_v are nonnegative constants. Namely, the cost function F is the sum of f_v $(v \in V)$ represented by

$$f_v(x) = \begin{cases} a_v x + b_v & (x > 0) \\ 0 & (x = 0). \end{cases}$$

Note that Problem ($\mathbf{P}_3$) in (38) can be solved in $O(n^2q)$ time by using Algorithm DP in Section 4.2. We shall show that it admits an $O(n\log^2 n)$ time algorithm. We remark that our problem requires $O(n^2q)$ time even if F is separable.

5.1 An Algorithm Based on Lower Envelope Computation

For $i = 1, 2, \cdots, k$, let $f_i : \mathbb{R}_+ \to \mathbb{R}_+$ be given by

$$f_i(x) \;=\; \begin{cases} a_i x + b_i & (x > 0) \\ 0 & (x = 0). \end{cases}$$

The *lower envelope* of $f_1, \cdots, f_k$ is given as $f(x) = \min_i f_i(x)$, and note that it is piecewise linear for $x > 0$, and hence it can be represented as

$$f(x) \;=\; \begin{cases} 0 & (x = 0) \\ g_j(x) & (P_j < x \le P_{j+1},\ j = 0, \cdots, l-1), \\ g_l(x) & (P_l < x) \end{cases}$$

where $P_0(= 0) < P_1 < \cdots < P_l$, each g_j is one of the f_is, and for the slopes α_j of g_js we have $\alpha_0 > \alpha_1 > \cdots > \alpha_l$. We say that g_j *attains* f *at* x if either ($P_j < x \le P_{j+1}$ and $j \in \{0, \cdots, l-1\}$) or ($P_j < x$ and $j = l$).

It is well known that the lower envelope of $f_1, \cdots, f_{j+1}$ can be computed from that of $f_1, \cdots, f_j$ and f_{j+1} in $O(\log j)$ time. Hence the lower envelope of $f_1, \cdots, f_k$ can be constructed in $O(k \log k)$ time.

Our algorithm is similar to Algorithm DP in Section 4.2, but, for each $Y \in \mathcal{F}$, it constructs the lower envelope corresponding to $\alpha(Y, k)$ ($k = 0, \cdots, h(Y)-1$). This implicit computation of $\alpha(Y, k)$ ($k = 0, \cdots, h(Y)-1$) makes the algorithm faster.

Algorithm ENVELOPE

Step 0. $\tilde{T}(= (\tilde{W}, \tilde{A})) := T$.

Step 1. (Make the lower envelopes) While $\tilde{W} \neq \{w_{i_0}\}$ **do**

Choose an arbitrary leaf $w \in \tilde{W}$ of $T[\tilde{W}]$.

/* Let Y be the set in $\mathcal{F}$ corresponding to w */

(1-I) If w is a leaf of T, then compute the lower envelope f_Y of $f_v(x + d(Y))$ ($v \in Y$). Otherwise, compute the lower envelope of

$$f_X(x + \Delta d(Y)) + \sum_{\substack{Z \in \mathcal{S}(Y): \\ Z \neq X}} f_Z(0) \qquad (X \in \mathcal{S}(Y)),$$

$$f_v(x + \Delta d(Y)) + \sum_{Z \in \mathcal{S}(Y)} f_Z(0) \qquad (v \in \Delta Y).$$

(1-II) $\tilde{W} := \tilde{W} \setminus \{w\}$.

Step 2. $\tilde{W} := W \setminus \{w_{i_0}\}$, and $x^*(v) := 0$ for all $v \in V$.

Step 3. (Compute an optimal solution x^*) While $\tilde{W} \neq \emptyset$ do

Choose an arbitrary node w of $T[\tilde{W}]$ having no leaving arc.

/* Let Y be the set in $\mathcal{F}$ corresponding to w. Assume that f_Y is attained at 0 by a function obtained from f_v. Let w_{j_0} be the node in $\tilde{W}$ corresponding to X_{j_0} such that $v \in \Delta X_{j_0}$, and let $w_{j_0} \to w_{j_1} \to \cdots \to w_{j_l}(= w)$ be a directed path in $T[\tilde{W}]$. */

(3-I) $x^*(v) := \sum_{i=0}^{l} \Delta d(X_{j_i})$.

(3-II) $\tilde{W} := \tilde{W} \setminus \{w_{j_0}, \cdots, w_{j_l}\}$.

Step 4. Output x^* and halt. □

Let us consider more precisely Step 1 for computing the lower envelope f_Y. If w is a leaf of T, then it follows from the above discussion that f_Y can be computed in $\mathrm{O}(|Y| \log |Y|)$ time. On the other hand, if w is not a leaf of T, then let X^* be a set in $\mathcal{S}(Y)$ with the maximum $|X^*|$. Then we construct f_Y by successively adding all lines appearing in $f_X(x + \Delta d(Y))(x \geq 0)$ for $X \in \mathcal{S}(Y) \setminus \{X^*\}$ or $f_v(x + \Delta d(Y))(x \geq 0)$ for $v \in \Delta Y$ to $f_{X^*}(x + \Delta d(Y))$. Since $|Y| \geq 2|X|$ for any $X \in \mathcal{S}(Y) \setminus \{X^*\}$, we have $\mathrm{O}(n \log n)$ additions in total. Since each addition can be done in $\mathrm{O}(\log n)$ time, Step 1 can be executed in $\mathrm{O}(n \log^2 n)$ time.

Lemma 11 *Algorithm* ENVELOPE *correctly computes an optimal solution in* $\mathrm{O}(n \log^2 n)$ *time if all the values of* a_v *and* b_v *are given.*

Proof. Since the correctness can be proved similarly as for Algorithm DP, it suffices to analyze the running time.

Clearly, Steps 0, 2, 3, and 4 can be executed in $\mathrm{O}(n)$ time. Moreover, the above argument shows that Step 1 can be carried out in $\mathrm{O}(n \log^2 n)$ time. Hence the total running time of Algorithm ENVELOPE is $\mathrm{O}(n \log^2 n)$. □

Theorem 12 *Problem* $(\mathbf{P}_3)$ *in* (38) *can be solved in* $\mathrm{O}(n \log^2 n)$ *time if* F *is given explicitly in a functional form, and in* $\mathrm{O}(n(q + \log^2 n))$ *time if* F *is given implicitly by an oracle.*

Proof. The former statement follows from Lemma 11. The latter follows from Lemma 11 and the fact that all a_v and b_v $(v \in V)$ can be computed in $\mathrm{O}(nq)$ time by invoking the oracle. □

6 The General Cost Case

In this section we consider Problem (**P**) in (6) $\sim$(8) whose cost function F is general monotone concave. We show that it requires $\Omega(2^{\frac{n}{2}}q)$ time to solve the problem if F is given implicitly by an oracle, and that it is NP-hard if F is given explicitly in a functional form.

6.1 A Lower Bound When F is Given by an Oracle

Now we consider Problem (**P**) whose cost function F is given by an oracle. To get a lower bound we consider the following problem instance I. Let $V = \{v_1, \cdots, v_n\}$, where n is even. Suppose $\mathcal{F} = \{\{v_{2i-1}, v_{2i}\} \mid i = 1, \cdots, \frac{n}{2}\}$, $d(X) = 1$ for all $X \in \mathcal{F}$, and $F(x) = \sum_{\sigma\in\Pi} g_\sigma(x)$, where Π is the set of all permutations of $\{1, 2, \cdots, n\}$ and

$$g_\sigma(x) = g(x(v_{\sigma_1}) \cdot 2^{n-1} + x(v_{\sigma_2}) \cdot 2^{n-2} + \cdots + x(v_{\sigma_{n-1}}) \cdot 2 + x(v_{\sigma_n}))$$

where $g : \mathbb{R}_+ \to \mathbb{R}_+$ is a strictly concave and monotone function, and $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_n)$ is a permutation of order n. Clearly g_σ and F are strictly concave and monotone as g is.

We can easily see that all the optimal solutions x satisfy

$$\{x(v_{2i-1}), x(v_{2i})\} = \{0, 1\} \qquad \text{for } i = 1, \cdots, \frac{n}{2} \tag{39}$$

and we thus have $2^{\frac{n}{2}}$ optimal solutions.

For an $S \subseteq \mathbb{R}^V_+$ with $|S| \le 2^{\frac{n}{2}} - 1$, we construct two problem instances I^- and I^+ as follows. I^- and I^+ both have $\mathcal{F}$ and d defined as above, but they have different cost functions F^- and F^+. For $y \in \mathbb{R}^V_+$ let $y_\sigma = \sum_{i=1}^n y(v_{\sigma_i}) \cdot 2^{n-i}$. Then, for any permutation $\sigma^* \in \Pi$, we have $|\{y_{\sigma^*} \mid y \in S\}| \le 2^{\frac{n}{2}} - 1$. In addition, since $y_{\sigma^*} \ne z_{\sigma^*}$ holds for two distinct vectors y and z in $\mathbb{R}^V_+$ satisfying (39), there is a vector x^* of form (39) such that $x^*_{\sigma^*} \ne y_{\sigma^*}$ for all $y \in S$. Choosing any such vector x^*, we define two monotone concave functions g^- and g^+ by

$$g^\pm(\alpha) = \begin{cases} g(\alpha) & \alpha \in \{y_{\sigma^*} \mid y \in S\} \\ g(\alpha) \pm \varepsilon & \alpha = x^*_{\sigma^*}, \end{cases}$$

where ε is a sufficiently small positive number. Since g is strictly concave, $|S|$ is finite, and $\varepsilon\,(> 0)$ is sufficiently small, g^- and g^+ are well-defined. Then we define F^- and F^+ by

$$F^\pm \;=\; g^\pm_{\sigma^*} + \sum_{\sigma\in\Pi:\sigma\ne\sigma^*} g_\sigma,$$

where $g^\pm_{\sigma^*}(x) = g^\pm(x_{\sigma^*})$.

Let x^- and x^+ be any optimal solutions of I^- and I^+, respectively. Then we can see that

$$x^- = x^*, \qquad x^+ \neq x^*. \tag{40}$$

Hence we have

Theorem 13 *Problem* (**P**) *requires at least* $2^{\frac{n}{2}}$ *calls to the oracle in the worst case when the cost function is given by an oracle.*

Proof. Assume to the contrary that Algorithm A computes an optimal solution y by invoking the oracle for the values of $x \in S$ with $|S| \leq 2^{\frac{n}{2}} - 1$. For this S, we construct I^- and I^+ defined as above. Then we have $F(x) = F^-(x) = F^+(x)$ for all $x \in S$. Hence y must be an optimal solution to all I, I^- and I^+. However, this contradicts (40). □

Corollary 14 *It takes* $\Omega(2^{\frac{n}{2}})$ *time to solve Problem* (**P**) *if* F *is given by an oracle.*

6.2 NP-Hardness when F is Given in a Functional Form

We now consider Problem (**P**) when the cost function is given explicitly in a functional form, We show that the problem is NP-hard, by reducing to it the following 3 SAT, which is known to be NP-hard.

> **3 SAT**
> Input: A 3-CNF $\varphi = \bigwedge_{c_j \in \mathcal{C}} c_j$, where $c_j = (l_{j_1} \vee l_{j_2} \vee l_{j_3})$.
> Question: Is φ satisfiable, i.e., does there exist an assignment $y^* \in \{0,1\}^N$ such that $\varphi(y^*) = 1$?

Here c_j is a clause containing three literals l_{j_1}, l_{j_2} and l_{j_3} in $\{y_1, \cdots, y_N, \bar{y}_1, \cdots, \bar{y}_N\}$.

Given a problem instance I of 3 SAT, we construct the corresponding instance J of Problem (**P**) as follows.

Let $V = \{v_1, \cdots, v_{2N}\}$ and $\mathcal{F} = \{X_i = \{v_{2i-1}, v_{2i}\} \mid i = 1, 2, \cdots, N\}$. Let d be a demand function defined by $d(X_i) = 1$ for each $X_i \in \mathcal{F}$, and let F be a cost function given by

$$F(x) = \alpha \left(\sum_{i=1}^{N} \min\{x(v_{2i-1}), x(v_{2i})\} + \sum_{c_j \in \mathcal{C}} \min\{x(v^{(j_1)}), x(v^{(j_2)}), x(v^{(j_3)})\} \right) \tag{41}$$

where $\alpha > 0$, and $v^{(j_k)} = v_{2i-1}$ if $l_{j_k} = \bar{y}_i$ and $v^{(j_k)} = v_{2i}$ if $l_{j_k} = y_i$. Note that F is a monotone concave function, and hence $\mathcal{F}$, d, and F defined above give a problem instance of (**P**). Intuitively, v_{2i-1} and v_{2i} correspond to literals y_i and $\bar{y}_i$, respectively, $\min\{x(v_{2i-1}), x(v_{2i})\}$ in F produces an assignment of y_i, and $\min\{x(v^{(j_1)}),\ x(v^{(j_2)}),\ x(v^{(j_3)})\}$ in F imposes $c_j(y) = 1$. More formally, we have the following lemma.

Lemma 15 *Let* J *denote the problem instance of* (**P**) *constructed as above from a problem instance* I *of* 3SAT*. Let* $OPT(\mathrm{J})$ *denote the optimal value of* J*. Then* I *is satisfiable if and only if* $OPT(\mathrm{J}) = 0$*, and unsatisfiable if and only if* $OPT(\mathrm{J}) \geq \alpha\,(> 0)$*.*

Proof. Let us first show the first statement, i.e., I is satisfiable if and only if $OPT(\mathrm{J}) = 0$. Let y^* be a satisfying assignment of I, i.e., $\varphi(y^*) = 1$. Then we define $x^* \in \mathbb{R}_+^V$ by $x^*(v_{2i-1}) = 1$ and $x^*(v_{2i}) = 0$ if $y_i^* = 1$, and $x^*(v_{2i-1}) = 0$ and $x^*(v_{2i}) = 1$ otherwise. Clearly, x^* is a feasible solution of J, and we have $F(x^*) = 0$. Since $F(x) \geq 0$ for all $x \in \mathbb{R}_+^V$, $OPT(\mathrm{J}) = 0$. If $OPT(\mathrm{J}) = 0$, then any optimal solution x^* of J satisfies $\{x^*(v_{2i-1}), x^*(v_{2i})\} = \{0, 1\}$ for all i and at least one of $x^*(v^{(j_1)})$, $x^*(v^{(j_2)})$ and $x^*(v^{(j_3)})$ is 0. Thus by letting $y_i^* = 1$ if $x^*(v_{2i-1}) = 1$ and 0 otherwise, we have a satisfying assignment y^* of φ.

Let us finally show that the unsatisfiability of I implies $OPT(\mathrm{J}) \geq \alpha$. Since $OPT(\mathrm{J}) \neq 0$ implies the unsatisfiability of I from the equivalence shown above, this completes the proof. Since F is concave monotone and the constraints are $x(v_{2i-1})+x(v_{2i}) \geq 1$ for all i, we restrict optimal solutions x^* of J to those satisfying $\{x^*(v_{2i-1}), x^*(v_{2i})\} = \{0, 1\}$ for all i. This implies $F(x^*) \geq \alpha$, since I is not satisfiable. □

Note that the value of α in (41) can be arbitrarily large. This means that our problem cannot be approximated unless P=NP.

Theorem 16 *It is* NP*-hard to approximate Problem* (**P**) *within any constant* α*. In particular, Problem* (**P**) *is* NP*-hard.*

7 Concluding Remarks

We have considered the problem of minimizing monotone concave functions with laminar covering constraints. We have shown an $\mathrm{O}(n^2 q)$ algorithm when the objective function F can be decomposed into monotone concave functions by the partition of V that is induced by the laminar family $\mathcal{F}$. Our algorithm is optimal when F is given by an oracle. We have also given a faster algorithm when F is the sum of linear cost functions with fixed setup costs. We have further shown that in general our problem requires $\Omega(2^{\frac{n}{2}} q)$ time when F is given implicitly by an oracle, and that it is NP-hard in general if F is given explicitly in a functional form.

In this paper we have assumed that the objective function F is monotone nondecreasing. It should be noted that this monotonicity assumption can be removed if we impose that the sum $x(V)$ be equal to a constant, since the monotonicity is only used to have an optimal solution x with the minimum $x(V)$.

Acknowledgments: The present work is supported by a Grant-in-Aid from Ministry of Education, Culture, Sports, Science and Technology of Japan.

The authors would like to express their appreciation to Hiro Ito (Kyoto Univ.), Hiroshi Nagamochi (Kyoto Univ.), András Frank (Eötvös Univ.), and Tibor Jordán (Eötvös Univ.) for their valuable comments.

References

[1] P. K. Ahuja, T. L. Magnanti and J. B. Orlin: *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, (1993).

[2] K. Arata, S. Iwata, K. Makino, and S. Fujishige: Locating sources to meet flow demands in undirected networks, *J. Algorithms*, **42** (2002), 54–68.

[3] A. A. Benczúr and D. R. Karger: Augmenting undirected edge connectivity in $\tilde{O}(n^2)$ time, *J. Algorithms*, **37** (2000), 2–36.

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

[5] A. Frank: Augmenting graphs to meet edge-connectivity requirements, *SIAM J. Discrete Mathematics*, **5** (1992), 25-53.

[6] S. Fujishige: *Submodular Functions and Optimization: Second Edition*, Annals of Discrete Mathematics, **58**, Elsevier (2005).

[7] H. Ito and M. Yokoyama: Edge connectivity between nodes and node-subsets, *Networks*, **31** (1998), 157–164.

[8] H. Ito, K. Makino, K. Arata, S. Honami, Y. Itatsu, and S. Fujishige: Source location problem with flow requirements in directed networks, *Optimization Methods and Software*, **18** (2003), 427–435.

[9] L. Lovász: *Combinatorial Problems and Exercises*, North-Holland (1979).

[10] W. Mader: A reduction method for edge-connectivity in graphs, *Ann. Discrete Mathematics*, **3** (1978), 145–164.

[11] W. Mader: Konstruktion aller n-fach kantenzusammenhangenden Digraphen, *European J. Combin.*, **3** (1982), 63–67.

[12] H. Nagamochi: Computing extreme sets in graphs and its applications, *Proc. of the 3rd Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications* (January 21–24, 2003, Tokyo, Japan) 349–357.

[13] H. Nagamochi and T. Ibaraki: A note on minimizing submodular functions, *Information Processing Letters*, **67** (1998), 239–244,

[14] H. Nagamochi and T. Ibaraki: Augmenting edge-connectivity over the entire range in $\tilde{\mathrm{O}}(nm)$ time, *J. Algorithms*, **30** (1999), 253–301.

[15] D. Naor, D. Gusfield and C. Martel: A fast algorithm for optimally increasing the edge connectivity, *SIAM J. Computing*, **26** (1997), 1139–1165.

[16] M. Sakashita, K. Makino, and S. Fujishige: Minimizing a monotone concave function with laminar covering constraints, *ISAAC2005*, LNCS **3827** (2005), 71–81.

[17] H. Tamura, M. Sengoku, S. Shinoda, and T. Abe: Some covering problems in location theory on flow networks, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **E75-A** (1992), 678–683.

[18] H. Tamura, H. Sugawara, M. Sengoku, and S. Shinoda: Plural cover problem on undirected flow networks, *IEICE Trans.*, **J81-A** (1998), 863–869 (in Japanese).

[19] T. Watanabe and A. Nakamura: Edge-connectivity augmentation problems, *Computer and System Sciences* **35** (1987) 96–144.