ラミナー被覆制約を持つ単調凹関数最小化問題

坂下　麻里子 †　牧野　和久 †　藤重　悟 ‡

† 〒 560-8531 大阪府豊中市待兼山町 1-3 大阪大学大学院 基礎工学研究科 社会システム数理領域
‡ 〒 606-8502 京都市左京区北白川追分町 京都大学 数理解析研究所

概要　V を $|V| = n$ である有限の集合とする．V の部分集合族 $\mathcal{F} \subseteq 2^V$ は，任意の集合 $X, Y \in \mathcal{F}$ に対して，$X \cap Y = \emptyset, X \subseteq Y, X \supseteq Y$ のいずれかが成立するとき，ラミナー族と呼ばれる．本研究では，ラミナー族 $\mathcal{F} \subseteq 2^V$，ラミナー族上の要求関数 $d : \mathcal{F} \to \mathbf{R}_+$，コストを表す単調凹関数 $F : \mathbf{R}_+^V \to \mathbf{R}_+$ が与えられたとき，ラミナー被覆制約と非負制約を満たしながらコスト $F(x)$ を最小化する問題を考察する．本研究では，コスト関数 F がラミナー族 $\mathcal{F}$ に基づく V の分割上の単調凹関数に分離できるとき，上記の問題が $\mathrm{O}(n^2q)$ 時間で解けることを示す．ただし，q は $x \in \mathbf{R}_+^V$ から $F(x)$ を得るための計算量である．また，F がオラクルとして与えられるときは $\Omega(n^2q)$ 時間必要であり，このときには我々の提案するアルゴリズムが最適であることを示す．さらに F が固定費つきの線形関数 f_v $(v \in V)$ の和として表現できる場合に対する $\mathrm{O}(n \log^2 n)$ 時間アルゴリズムを構成する．一般の単調凹関数 F に対しては，F がオラクルとして与えられるならば，$\Omega(2^{\frac{n}{2}} q)$ 時間必要であり，F が明示的に与えられる場合でも NP 困難であることを示す．

Minimizing a Monotone Concave Function with Laminar Covering Constraints

Mariko Sakashita†　Kazuhisa Makino†　Satoru Fujishige‡

†Division of Mathematical Science for Social Systems, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, 560-8531, Japan.
‡Research Institute for Mathematical Sciences, Kyoto University, Kyoto, 606-8502, Japan.

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 $d : \mathcal{F} \to \mathbf{R}_+$ and a monotone concave cost function $F : \mathbf{R}_+^V \to \mathbf{R}_+$, we consider the problem of finding a minimum-cost $x \in \mathbf{R}_+^V$ such that $x(X) \geq d(X)$ for all $X \in \mathcal{F}$. In this paper, we show that the problem can be solved in $\mathrm{O}(n^2q)$ time, if F can be decomposed into monotone concave functions by the partition of V based on $\mathcal{F}$ where q is the time needed to compute $F(x)$ for any $x \in \mathbf{R}_+^V$. We also prove that if F is given as an oracle then it takes $\Omega(n^2q)$ time to solve the problem. This implies that the algorithm proposed in this paper is optimal. We give an $\mathrm{O}(n \log^2 n)$ algorithm if F is the sum of linear cost functions with fixed opening costs. Finally, we show that in general our problem needs $\Omega(2^{\frac{n}{2}} q)$ when F is given as an oracle, and it is NP-hard, if F is given explicitly.

1　序論

V を $|V| = n$ である有限の集合とする．V の部分集合族 $\mathcal{F} \subseteq 2^V$ は，任意の集合 $X, Y \in \mathcal{F}$ に対して，$X \cap Y = \emptyset$, $X \subseteq Y$, $X \supseteq Y$ のいずれかが成立するとき，ラミナー (laminar) 族と呼ばれる．本研究では，ラミナー族 $\mathcal{F} \subseteq 2^V$，$\mathcal{F}$ 上の要求関数 $d : \mathcal{F} \to \mathbf{R}_+$，コストを表す単調凹関数 $F : \mathbf{R}_+^V \to \mathbf{R}_+$ が与えられたとき，ラミナー被覆制約 $(x(X) \geq d(X), X \in \mathcal{F})$ と非負制約 $(x(v) \geq 0, v \in V)$ を満たしながら，コスト関数を最小化する問題について考察する．

この問題に対しては，会社などの組織の階層構造がラミナー族によって表現できることから分かるように，様々な応用が考えられる．また，一見するとラミナー構造を持たない問題でも，我々の問題として定式化可能なものがある．例えば，コストが単調凹関数で記述でき，需要量が一定である無向ネットワークにおけるフローに基づく施設配置問題が挙げられる．この施設配置問題は，特に，コストが各点での施設建設（供給量には依存しない）費用の和で記述される場合について研究が行われているが [1, 12]，本研究は，費用が各点の供給量にも単調凹に依存する一般的な場合も扱う．

また，無向ネットワークにおける枝連結度増大問題も，我々の問題への定式化が可能な例である．本研究では，コストが，新しく加えられた枝に関する各点の次数の重み和で記述される場合だけでなく [3, 10]，分離可能である単調凹関数として記述できる場合も扱う．

本研究では，コスト関数 F が一般の単調凹関数である場合に加えて，次の 3 通り，(i): $F_1(x) = \sum_{X \in \mathcal{F}} f_{\Delta X}(x[\Delta X])$，(ii): $F_2(x) = \sum_{v \in V} f_v(x(v))$，(iii): $F_1(x) = \sum_{v \in V: x(v)>0}(a_v x(v) + b_v)$ のように記述できる場合について考察を行う．ただし，$\Delta X = X - \sum_{Y \in \mathcal{F}: Y \subsetneq X} Y$，$f_{\Delta X}$ は ΔX 上の非負単調凹関数，$x[\Delta X]$ は変数 x の ΔX 上への制限，f_v は $\{v\}$ 上の 1 変数非負単調凹関数，a_v, b_v は非負定数とする．定義から明らかに，F_1 の特殊形が F_2 であり，さらに F_2 の特殊形が F_3 である．

本研究では，コスト関数 F が V の分割上の単調凹関数の和，すなわち F_1 により記述できるとき，我々の問題が $\mathrm{O}(n^2 q)$ 時間で解け，加えて F がオラクルとして与えられるときは $\Omega(n^2 q)$ 時間必要であり，この場合は我々の提案するアルゴリズムが最適であることを示す．ここで F のオラクルとは，任意の $x \in \mathbf{R}_+^V$ に対して，その関数値 $F(x)$ を出力するものであり，F がオラクルで与えられるかどうかに関わらず，x から $F(x)$ を得るための計算量を $\mathrm{O}(q)$ とする．また，この結果より，コスト関数が式 F_1 で表される無向ネットワーク $\mathcal{N}$ におけるフローに基づく施設配置問題，及び枝連結度増大問題も効率的に解ける．さらに F が固定費つきの線形関数 f_v $(v \in V)$ の和，すなわち式 F_3 により表現できる場合は $\mathrm{O}(n \log^2 n)$ 時間で解くことができることを示す．一方，一般の単調凹関数 F に対しては，F がオラクルとして与えられるならば $\Omega(2^{\frac{n}{2}} q)$ 時間必要であり，F が直接与えられる場合でも我々の問題が NP 困難であることを示す．以下，まとめた結果を表 1 に示す．

表 1: 本研究での結果

	明示的	オラクル
F_1	$\mathrm{O}(n^2 q)$	$\Theta(n^2 q)$
F_2	$\mathrm{O}(n^2 q)$	$\Theta(n^2 q)$
F_3	$\mathrm{O}(n \log^2 n)$	$\mathrm{O}(n(\log^2 n + q))$
一般	NP 困難	$\Omega(2^{\frac{n}{2}})$

2　諸定義

V を $|V| = n$ である有限の集合とする．V の部分集合族 $\mathcal{F} \subseteq 2^V$ と，任意の集合 $X, Y \in \mathcal{F}$ に対して，

$$X \cap Y,\ X - Y,\ Y - X$$

の少なくとも 1 つが空集合になるとき，$\mathcal{F}$ をラミナー (laminar) 族と呼ぶ．すなわち，任意の集合 $X, Y \in \mathcal{F}$ に対して，$\mathcal{F}$ がラミナー族であるのは，$X \cap Y = \emptyset$，$X \subseteq Y$，$X \supseteq Y$ のいずれかが成立するときである．

ラミナー族 $\mathcal{F} = \{X_i \mid i \in I\}$ に対して，以下のように有向グラフ $T = (W, A)$ を構成する．ただし，i_0 は $i_0 \notin I$ である新しい添字とする．

$$\begin{aligned} W &= \{w_i \mid i \in I \cup \{i_0\}\} \\ A &= \{a_i = (w_i, w_j) \mid X_i \subsetneq X_j, \text{かつ}, \\ &\qquad X_i \subsetneq Y \subsetneq X_j \text{である } Y \in \mathcal{F} \text{ が存在しない} \} \\ &\qquad \cup \{a_i = (w_i, w_{i_0}) \mid X_i \text{が } \mathcal{F} \text{ において極大} \}. \end{aligned}$$

このようにして得られた有向グラフ $T = (W, A)$ は $\mathcal{F}$ がラミナーであることから w_{i_0} を根とする内向木となり，T を $\mathcal{F}$ の木表現と呼ぶ．根 w_{i_0} に対応する集合を仮想的に X_{i_0} と記す．

$X_i \in \mathcal{F}$ に対して，以下のように子集合族 $\mathcal{S}(X_i)$, 差分 ΔX_i, 深さ $h(X_i)$ を定義する．

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

関数 $F : \mathbf{R}^V \to \mathbf{R}$ が，任意の変数 $x, y \in \mathbf{R}^V$ に対して，$x \leq y$ ならば $F(x) \leq F(y)$ を満たすとき F を単調 (非減少)，また，任意の変数 $x, y \in \mathbf{R}^V$ 及び $0 \leq \alpha \leq 1$ である任意の α に対して，

$$\alpha F(x) + (1 - \alpha) F(y) \ \leq \ F(\alpha x + (1 - \alpha) y) \qquad (1)$$

を満たすとき，F は凹関数であるという．

本研究で扱う，ラミナー被覆制約を持つ凹関数最小化問題は，ラミナー族 $\mathcal{F} \subseteq 2^V$，単調 (非減少) 凹関数 $F : \mathbf{R}_+^V \to \mathbf{R}_+$ と $\mathcal{F}$ 上の関数 $d : \mathcal{F} \to \mathbf{R}_+$ が与えられたとき，以下の最適化問題として表現できる．

$$\begin{array}{lll} \text{Minimize} & F(x) & \\ \text{subject to} & x(X) \geq d(X) & (X \in \mathcal{F}) \\ & x(v) \geq 0 & (v \in V) \end{array} \qquad (2)$$

ただし，$\mathbf{R}_+$ は非負実数の集合，$x(X) = \sum_{v \in X} x(v)$ とし，また，d をラミナー族 $\mathcal{F}$ 上の要求関数 (demand function) と呼ぶ．さらに，一般性を失うことなく，$F(0) = 0$ を仮定する．

関数 $d:\mathcal{F}\to\mathbf{R}$ に対して，差分 Δd を $\Delta d(X)=d(X)-\sum_{Y\in\mathcal{S}(X)}d(Y)$ により定義する．定義から明らかに，$\Delta d(X)\le 0$ であるような $X\in\mathcal{F}$ を制約条件から除いても問題 (2) は不変である．従って，今後，

$$\Delta d(X)>0 \qquad (X\in\mathcal{F}) \tag{3}$$

と仮定する．

我々はラミナー被覆制約をもつ凹関数最小化問題に対して，コスト関数 F が陽に与えられる場合，及びオラクルとして与えられる場合について考察する．どちらの場合も，$x\in\mathbf{R}_+^V$ に対してその関数値 $F(x)$ を得るための計算量を q とする．本研究では，特に目的関数 F が以下の 3 つの場合について扱う．

$$F_1(x) = \sum_{X\in\mathcal{F}} f_{\Delta X}(x[\Delta X]) \tag{4}$$

$$F_2(x) = \sum_{v\in V} f_v(x(v)) \tag{5}$$

$$F_3(x) = \sum_{v\in V:x(v)>0} (a_v x(v)+b_v). \tag{6}$$

ここで，$f_{\Delta X}$ は ΔX 上の非負単調凹関数，$x[\Delta X]$ は変数 x の ΔX 上への制限，f_v は $\{v\}$ 上の非負単調凹関数，a_v, b_v は非負定数とする．ラミナー族 $\mathcal{F}$ は V の分割を与える．F_1 は関数がこの分割上で定義される単調凹関数の和として表される場合を示す．F_2 は各 $v\in V$ に対する単調凹関数 f_v に分解可能な場合，F_3 はその f_v が固定費つきの線形関数として与えられる場合を示す．ここで，F_3 を構成する関数 f_v は $f_v(0)=0$ であり，b_v が非負であることから f_v の凹性は保証される．これらの定義から，F_2 を構成する各 f_v をより具体的に与えたものが F_3 であること，F_1 を与える分割 $\Delta X(X\in\mathcal{F})$ をさらに細分化することによって F_2 が得られることは明らかである．すなわち F_1 の特殊形が F_2 であり，その F_2 の特殊形が F_3 である．

2.1 応用例

ラミナー被覆制約を持つ最小化問題に対する例となる 2 つのネットワーク問題を紹介する．

2.1.1 無向ネットワークにおけるフローに基づく施設配置問題

$\mathcal{N}=(V,E,u)$ を点集合 V，枝集合 E，容量関数 $u:E\to\mathbf{R}_+$ を持つ無向ネットワークとする．記述を簡単にするため，今後，$\mathcal{N}$ のかわりに，各枝 $e\in E$ を双方向化して得られる対称な有向ネットワーク $\hat{\mathcal{N}}=(\hat{G}=(V,\hat{E}),\hat{u})$ を用いる．ただし，$\hat{E}=\{(v,w),(w,v)\mid\{v,w\}\in E\}$, $\hat{u}(v,w)=\hat{u}(w,v)=u(\{v,w\})$ $(\{v,w\}\in E)$ とする．

ネットワーク $\hat{\mathcal{N}}$ 中のフロー $\varphi:\hat{E}\to\mathbf{R}_+$ が供給量 $x:V\to\mathbf{R}_+$ に対して，実行可能 (feasible) であるとは，各点 v から流れ出るフロー量 $\partial\varphi(v)(=\sum_{(v,w)\in\hat{E}}\varphi(v,w)-\sum_{(w,v)\in\hat{E}}\varphi(w,v))$ が v 自身の供給量 $x(v)$ 以下であるという制約と，容量制約 $(0\le\varphi(e)\le\hat{u}(e),\, e\in\hat{E})$ が満たされることをいう．

無向ネットワーク $\mathcal{N}$，需要量 $k>0$，供給量に対するコスト関数 $F:\mathbf{R}_+^V\to\mathbf{R}_+$ が与えられたとき，施設配置問題 [1] は各点 v 毎に需要量 k 以上流すことのできる実行可能フロー φ_v が存在するというフロー制約を満たすような最小コスト供給量 x を求める問題である．この施設配置問題は，特に，

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

すなわち，コストが各点での供給施設建設（供給量には依存しない）費用の和で記述される場合について研究が行われている [1, 12]．その他，需要量が各点で異なる場合，有向ネットワークの場合など様々な研究がされている [1, 13, 6, 5]．

非空である $X\subseteq V$ は，X の任意の真部分集合 $Y\neq\emptyset$ に対して，$\kappa(Y)>\kappa(X)$ を満たすとき，極集合 (extreme set) と呼ばれる．ただし，$\kappa(X)=\sum_{v\in X,w\in V-X}u(v,w)$ で，X のカット容量と呼ぶ．$\mathcal{F}$ を $\mathcal{N}$ の全ての極集合の族とする．κ の正モジュラ (posi-modular) 性より，$\mathcal{F}$ はラミナー族となる [10]．

$X\in\mathcal{F}$ の不足度 (deficiency) $d(X)$ を

$$d(X)=\max\{k-\kappa(X),0\} \tag{7}$$

により定義すると，この施設配置問題は我々の問題 (2) として記述できる．無向ネットワーク $\mathcal{N}=(G=(V,E),u)$ と需要量 $k(>0)$ が与えられると，全ての極集合の族 $\mathcal{F}$，及び，要求 $d:\mathcal{F}\to\mathbf{R}_+$ は $\mathrm{O}(n(m+n\log n))$ 時間で求めることができる [10]．従って，コストが各点上の単調凹関数に分離可能な場合，例えば，各点の施設建設費と凹である運転費用の和で記述できる場合は，この施設配置問題は $\mathrm{O}(nm+n^2(q+\log n))$ 時間で解ける．

2.1.2 無向ネットワークにおける枝連結度増大問題

$\mathcal{N} = (G = (V,E), u)$ を無向ネットワーク，$u : E \to \mathbf{R}_+$ をその容量関数とする．任意の $v, w \in V$ に対して，v-w 間の最大フロー量が k 以上であるとき，$\mathcal{N}$ を k 枝連結と呼ぶ．無向ネットワーク $\mathcal{N}$，正実数 k，点に関するコスト関数 $F : \mathbf{R}_+^V \to \mathbf{R}_+$ が与えられるとき，枝連結度増大問題 [3, 11] は，$\mathcal{N}' = (G' = (V, E \cup D), u \oplus \mu_D)$ が k 枝連結，かつ，コスト $F(\kappa_D)$ が最小となるような新枝集合 D とその容量 $\mu_D : D \to \mathbf{R}_+$ を求める問題である．ただし，$\kappa_D(v) = \sum_{w \in V \setminus \{v\}} \mu_D(v, w)$，$u \oplus \mu_D$ は u と μ_D の直和である．$x = \kappa_D$ とすると，任意の $X \subsetneq V$ に対して $\kappa(X) + x(X) \geq k$ ならば，

$$x(X) \geq d(X) \qquad (X \in \mathcal{F}) \tag{8}$$

となる．ただし，$d(X)$ は (7) により与えられ，$\mathcal{F}$ は $\mathcal{N}$ の極集合族とする．逆に，最大フロー最小カット定理より，(8) を満たす任意の x から $\mathcal{N}'$ を k 枝連結にする $\mu_D : D \to \mathbf{R}_+$ を構成できる [7, 8, 9]．さらに，この操作は $\mathrm{O}(n(m + n \log n))$ 時間で実行できる [11]．

枝連結度増大問題は，特にコスト関数が $F(\kappa_D) = \sum_{v \in V} c(v)\kappa_D(v)$ で与えられる場合について研究が行われている [3, 11]．$c : V \to \mathbf{R}_+$ は各点のコストを与える関数で，全ての点で $c(v) = 1/2$ とすれば，$F(\kappa_D)$ は D のサイズに等しい．本研究で提案するアルゴリズムは，[10, 11] の結果と合わせて，F が式 (4) で与えられる場合，この枝連結度増大問題を $\mathrm{O}(nm + n^2(q + \log n))$ 時間で解く．

3 分離可能なコスト最小化

本節では，コスト関数が F_1 で表現できる場合，すなわち，

$$\begin{array}{lll} \text{Minimize} & \displaystyle\sum_{X \in \mathcal{F}} f_{\Delta X}(x[\Delta X]) & \\ \text{subject to} & x(X) \geq d(X) & (X \in \mathcal{F}) \\ & x(v) \geq 0 & (v \in V) \end{array} \tag{9}$$

で与えられる場合を考察する．ただし $f_{\Delta X}$ は ΔX 上の非負単調凹関数で，$f_{\Delta X}(0) = 0$ を満たす．

我々は $\mathrm{O}(n^2 q)$ 時間アルゴリズムを提案し，さらに目的関数 F がオラクルとして与えられるときは，F が $F = \sum_{v \in V} f_v$ と分解可能である（すなわち，式 (5) の F_2 として記述できる）ときでも，我々の問題を解くために，$\Omega(n^2 q)$ 時間必要であることを示す．

3.1 最適解の構造的性質

本節では，(9) の最適解の構造的性質を明らかにし，それに基づいてアルゴリズムを開発する．$\mathcal{F}$ をラミナーな V の部分集合族，$T = (W, A)$ を $\mathcal{F}$ の木表現とする．

まず，$Y \in \mathcal{F}$ 上に制限した以下の問題を考える．

$$\begin{array}{lll} \text{Minimize} & \displaystyle\sum_{X \in \mathcal{F} : X \subseteq Y} f_{\Delta X}(x[\Delta X]) & \\ \text{subject to} & x(X) \geq d(X) & (X \in \mathcal{F}, X \subseteq Y) \\ & x(v) \geq 0 & (v \in V) \end{array} \tag{10}$$

部分問題の最適解の性質を明らかにすることで，原問題の最適解の構造を示す．

補題 3.1 極小な $Y \in \mathcal{F}$ 対する部分問題 (10) は，次を満たす最適解 $x = z_v$ を持つ．

ある $v \in Y$ に対して，

$$z_v(t) = \begin{cases} d(Y)\ (= \Delta d(Y)) & (t = v) \\ 0 & (t \in V - \{v\}). \end{cases} \tag{11}$$

補題 3.2 極小でない $Y \in \mathcal{F}$ 上の部分問題 (10) において，以下の $z_v\ (v \in \Delta Y)$

$$\begin{aligned} z_v(t) &= \begin{cases} \Delta d(Y) & (t = v) \\ 0 \ (t \in (V \setminus Y) \cup (\Delta Y \setminus \{v\})) \end{cases} \\ z_v(X) &= d(X) \qquad (X \in \mathcal{S}(Y)) \end{aligned} \tag{12}$$

あるいは，$y_X\ (X \in \mathcal{S}(Y))$

$$\begin{aligned} y_X(Z) &= \begin{cases} d(X) + \Delta d(Y) & (Z = X) \\ d(Z) \quad (Z \neq X, Z \in \mathcal{S}(Y)) \end{cases} \\ y_X(v) &= 0 \qquad (v \in (V \setminus Y) \cup \Delta Y) \end{aligned} \tag{13}$$

のいずれかが最適解となる．

$W^* = \{w_i \mid X_i \in \mathcal{F}\}$ とする．$\mathcal{P} = \{P_1, \cdots, P_k\} \subseteq 2^{W^*}$ は，次を満たすとき，W^* のパス分割 (path-partition) と呼ばれる．

- $\bigcup_i P_i = W^*$，$i \neq j$ ならば $P_i \cap P_j = \emptyset$,
- 各 P_j が $T = (W, A)$ 上の有向パス $w_{j_0} \to w_{j_1} \to \cdots \to w_{j_{r_j}}$ であり $\Delta X_{j_0} \neq \emptyset$

定理 3.3 問題 (10) は W^* のパス分割 $\mathcal{P} = \{P_1, \cdots, P_k\}(v_j \in \Delta X_{j_0})$ に基づく次のような最適解 x^* を持つ．

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

証明 $T = (W, A)$ の高さ h の帰納法により証明する．

$h = 1$ のとき，補題 3.1 より各 $Y \in \mathcal{F}$ に対する問題 (10) は (11) で表される最適解 x_Y を持ち，F の単調性，及び分離可能性より $x = \sum_{Y \in \mathcal{F}} x_Y$ は (9) の最適解である．また，明らかにこの x は定理に示される性質を持つ．

次に，ある ℓ に対して $h \le \ell$ のとき定理が成立すると仮定して，$\ell + 1$ の場合を考える．補題 3.2 より，極大な $Y \in \mathcal{F}$ に対する問題 (10) は式 (12) もしくは (13) で表される最適解を持つ．まず，式 (12) の z_v が最適解となる場合は，帰納法の仮定より，$X \in \mathcal{S}(Y)$ に対する問題 (10) はあるパス分割 $\mathcal{P}_X$ に基づく最適解 x_X を持っており，次のように定めた x_Y はパス分割 $\mathcal{P}_Y$ に基づく (9) の最適解となる．

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

ただし，w は Y に対応する W^* の点，e_v は $t = v$ で $\Delta d(Y)$，それ以外で 0 をとるベクトルとする．

一方，式 (13) の y_X が (10) の最適解となる場合は，$Z \neq X$ である $Z \in \mathcal{S}(Y)$ に対する部分問題 (10) と X に対する次の問題を考える．

$$\begin{aligned} \text{Minimize} \quad & \sum_{Z \in \mathcal{F}: Z \subseteq X} f_{\Delta Z}(x[\Delta Z]) \\ \text{subject to} \quad & x(X) \ge d(X) + \Delta d(Y) \\ & x(Z) \ge d(Z) \quad (Z \in \mathcal{F}, Z \subsetneq X) \\ & x(v) \ge 0 \qquad (v \in V) \end{aligned} \tag{14}$$

帰納法の仮定より，$Z \in \mathcal{S}(Y), Z \neq X$ に対する問題 (10) はそれぞれパス分割 $\mathcal{P}_Z$ とそれに基づく最適解 x_Z，また，問題 (14) もパス分割 $\mathcal{P}_X$ とそれに基づく最適解 x_X を持つ．今，この X に対応する点を含む集合を $P \in \mathcal{P}_X$ とし，Y に対応する点を w とすると，次に定める $\mathcal{P}_Y$ に基づく最適解 $x_Y = \sum_{Z \in \mathcal{S}(Y)} x_Z$ を構成できる．

$$\mathcal{P}_Y = \bigcup_{Z \in \mathcal{S}(Y): Z \neq X} \mathcal{P}_Z \cup (\mathcal{P}_X - \{P\} \cup \{P \cup \{w\}\})$$

以上，$x = \sum_{Y \in \mathcal{S}(X_{i_0})} x_Y$，$\mathcal{P} = \bigcup_{Y \in \mathcal{S}(X_{i_0})} \mathcal{P}_Y$ と定めることにより，定理に従う最適解 x を得る． □

3.2 アルゴリズムとその計算量

本節では問題 (9) を解く多項式時間アルゴリズムを構成する．

提案するアルゴリズムは，$\mathcal{F}$ の木表現 $T = (W, A)$ の葉から根に向かい各点 $w_i \in W^*$ で最適なパス分割を順次計算するという動的計画法を用いることにより，最適解 x^* を求める．

任意の $Y \in \mathcal{F}$ に対して，対応する $w \in W$ から根 w_{i_0} までの有向パスを $w_{j_0}(= w), w_{j_1}, \cdots, w_{j_{h(Y)-1}}$, $w_{j_{h(Y)}}(= w_{i_0})$ とする．この $Y \in \mathcal{F}$ と $k\,(0 \le k \le h(Y) - 1)$ に対して以下のように定義される問題について考える．

$$\begin{aligned} \text{Minimize} \quad & \sum_{X \in \mathcal{F}: X \subseteq Y} f_{\Delta X}(x[\Delta X]) \\ \text{subject to} \quad & x(Y) \ge d(Y) + \sum_{i=1}^{k} \Delta d(X_{j_i}) \\ & x(X) \ge d(X) \quad (X \in \mathcal{F}, X \subsetneq Y) \\ & x(v) \ge 0 \qquad (v \in V) \end{aligned} \tag{15}$$

問題 (15) に定理 3.3 を用いることで，この問題が $\{w_i \mid X_i \in \mathcal{F}, X_i \subseteq Y\}$ のパス分割 $\mathcal{P}$ とそれに基づく最適解を持つことが分かる．今，Y に対応する点 $w \in W$ を含む集合を $P_j \in \mathcal{P}$ とする．$Y \in \mathcal{F}$ に対するテーブル g_Y は $\{0, 1, \cdots, h(Y) - 1\}$ から $\mathbf{R}_+ \times V$ への関数であり，$g_Y(k)_1$ は (15) の最適値，$g_Y(k)_2$ は上記の集合 $P_j \subseteq W$ に対する（定理 3.3 の）$v_j \in \Delta X_{j_0}$ を与えるものとする．以下ではこのテーブルの作成法を考える．

T の葉に対応する，すなわち極小な $Y \in \mathcal{F}$ に対しては，問題 (15) に補題 3.1 を用いることにより，$v \in \Delta Y (= Y)$ に対して

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

のいずれかが最適解になることから（$d(X_{j_0}) = \Delta d(X_{j_0})$ に注意），$k = 0, 1, \cdots, h(Y) - 1$ に対して，

$$g_Y(k) = \left(\min_{v \in Y} f_Y(z_v^k),\ \arg\min_{v \in Y} f_Y(z_v^k) \right) \tag{16}$$

となる．ただし，$\arg\min_{v \in Y} f_Y(z_v^k)$ は $f_Y(z_{v^*}^k) = \min_{v \in Y} f_Y(z_v^k)$ となるような $v^* \in Y$ を表す．

極小でない $Y \in \mathcal{F}$ については，定理 3.3 より

$$g_Y(k)_1 = \min\left\{\min_{X\in\mathcal{S}(Y)}\left\{g_X(k+1)_1 + \sum_{\substack{Z\in\mathcal{S}(Y):\\ Z\neq X}} g_Z(0)_1\right\},\right.$$
$$\left.\min_{v\in\Delta Y}\left\{f_{\Delta Y}(z_v^k) + \sum_{X\in\mathcal{S}(Y)} g_X(0)_1\right\}\right\},\ (17)$$

また，$g_Y(k)_2$ は

$$g_Y(k)_1 = g_X(k+1)_1 + \sum_{Z\in\mathcal{S}(Y):Z\neq X} g_Z(0)_1$$

のときは $g_X(k+1)_2$，

$$g_Y(k)_1 = f_{\Delta Y}(z_v^k) + \sum_{X\in\mathcal{S}(Y)} g_X(0)_1$$

のときは v となる．

以下のアルゴリズムでは，まず T の葉から式 (16),(17) を用いて，根に向かって各 $w \in W^*$ に対応する集合 $Y \in \mathcal{F}$ のテーブル g_Y を作成する．このとき $\sum_{Y\in\mathcal{S}(X_{i_0})} g_Y(0)_1$ が原問題 (9) の最適値となる．次に $g_Y(0)_2$ の情報を利用して，パス分割 $\mathcal{P}$ とそれに基づく解 x^* を根から葉に向かい計算する．

アルゴリズム テーブル

ステップ 0: $\tilde{T}(=(\tilde{W},\tilde{A})) := T$.

ステップ 1: $\tilde{T}$ の葉 w を一つ選んで，

(1-I) 対応する $Y \in \mathcal{F}$ のテーブル g_Y を式 (16)，式 (17) に基づき作成する．

(1-II) $\tilde{W} := \tilde{W} - \{w\}$.
$\tilde{W} = \{w_{i_0}\}$ ならばステップ 2 へ．
そうでなければステップ 1 へ戻る．

ステップ 2: $\tilde{T} := T$， $x^*(v) := 0\,(v \in V)$.

ステップ 3: $\tilde{T}$ における $Y \in \mathcal{S}(X_{i_0})$ を一つ選んで，
$(g_Y(0)_2 \in \Delta X_{j_0}, w_{j_0} \to w_{j_1} \to \cdots \to w_{j_{l+1}}(=w_{i_0})$ は $\tilde{T}$ 上の有向パス.)

(3-I) $x^*(g_Y(0)_2) := \sum_{i=0}^{l} \Delta d(X_{j_i})$.

(3-II) $\tilde{T}$ からこのパスを除去し，更新．
$\tilde{W} = \{w_{i_0}\}$ ならばステップ 4 へ．
そうでなければステップ 3 へ戻る．

ステップ 4: x^* を出力し，終了する．

このアルゴリズムの計算量について考察する．まず，テーブル g_X $(X \in \mathcal{F})$ の作成，すなわちステップ 1 にかかる計算量を求める．極小な $X \in \mathcal{F}$ に対する g_X は，式 (16) より $\mathrm{O}(h(X)|\Delta X|q)$ 時間，非極小 $X \in \mathcal{F}$ に対する g_X は $\mathrm{O}(h(X)(|\mathcal{S}(X)|+|\Delta X|q))$ 時間で求められる．従って

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

となる．また，その他のステップ 0, 2, 3, 4 の処理は線形時間 $\mathrm{O}(n)$ で可能である．以上をまとめて，

定理 3.4 アルゴリズム テーブルは，問題 (9) を $\mathrm{O}(n^2q)$ 時間で解く．

3.3 問題の下界

本節では，コスト関数 F が式 (5) の F_2 で記述できる場合の問題 (9) の下界を示す．より正確には，目的関数 F のオラクルが与えられたとき，問題を解くために必要なオラクル呼び出し回数の下界を情報量理論に基づき示す．この下界によって上節のアルゴリズムの最適性も示される．

次の問題例について考える．

問題例 I

ラミナー族 : $\mathcal{F} = \{X_0, X_1, \cdots, X_{\frac{n}{2}}\}$,
$(X_i = \{v_1, \cdots, v_{\frac{n}{2}+i}\}\ (i=0,1,\cdots,\frac{n}{2}))$

要求関数 : $d(X_i) = i+1\ \ (i=0,\cdots,\frac{n}{2})$

コスト関数 : $F(x) = \sum_{v\in V} f_v(x(v))$

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

ただし，$g_0 : \mathbf{R}_+ \to \mathbf{R}_+$ は厳密に凹である単調増加関数である．例えば，$g_0(x) = \log(x+1)\ (x \ge 0)$ とする．g_i は

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

により与える．また，n は偶数とする．

補題 3.5 式 (5) で記述されるコスト関数 F がオラクルとして与えられる問題 (9) では，少なくとも $\frac{n}{2}(\frac{n}{2}+1)$ 回以上のオラクル呼び出しを必要とする場合がある．

略証 この問題例に定理 3.3 を用いると，次のように記述できる最適解が存在することが分かる．

ある $v_i \in X_0, k = 1,\cdots,\frac{n}{2}+1$ に対して，

$$x_{(v_i,k)}(v) = \begin{cases} k & (v = v_i) \\ \frac{n}{2}-k+1 & (v = v_{\frac{n}{2}+k}) \\ 0 & (v \in V - \{v_i, v_{\frac{n}{2}+k}\}) \end{cases} \qquad (18)$$

この問題例の最適値は $g_0(\frac{n}{2}+1)$ となり，(18) 以外に最適解を持たない．この問題例 I と $|S| \leq \frac{n}{2}(\frac{n}{2}+1)-1$ である $S \subseteq \mathbf{R}_+^V$ に対して，I と同じラミナー族 $\mathcal{F}$ と要求関数 d を持つが，それぞれのコスト関数 F^- と F^+ が I のコスト関数 F とは異なるような問題例 I^- と I^+ を次式を満たすように作成できる．

$$F^{\pm}(x) = F(x) \qquad (x \in S)$$

また，式 (18) で表されるベクトルで S に属さない x^* に対して，$x = x^*$ のとき，

$$F^-(x) = F(x) - \varepsilon^- \tag{19}$$

$$F^+(x) = F(x) + \varepsilon^+ \tag{20}$$

ただし，$\varepsilon^-, \varepsilon^+$ は十分小さい正数とする．このとき，I^- の最適解は式 (19) となるような x' であり，この x' は I^+ では最適解とはならない．

ここで，あるアルゴリズム A が問題例 I に対して，上記の S に属するベクトルに対してのみオラクル呼び出しを行い，最適解として y を出力したとすると，$F(x) = F^-(x) = F^+(x)$ $(x \in S)$ より y は問題例 I, I^-, I^+ すべてにおいて最適解となるはずであるが，これは上記の議論より矛盾している． □

補題 3.5 より，このとき問題 (9) を解くためには，$\Omega(n^2 q)$ 時間必要であることが分かる．従って，3.2 節で示したアルゴリズムは最適である．

4　固定費を含む線形分離関数最小化

本章では，ラミナー被覆制約を持つ凹関数最小化問題のうち目的関数が F_3 で表現できる場合，すなわち，

$$\begin{array}{lll} \text{Minimize} & \displaystyle\sum_{v \in V: x(v) > 0} (a_v x(v) + b_v) & \\ \text{subject to} & x(X) \geq d(X) \quad (X \in \mathcal{F}) & (21) \\ & x(v) \geq 0 \qquad\quad (v \in V) & \end{array}$$

で与えられる場合を考察する．ただし，a_v, b_v は非負定数である．

この問題は，問題 (9) に含まれるため前章で提案したアルゴリズムを用いて，$\mathrm{O}(n^2 q)$ で解くことができる．前節において，この問題の目的関数が，例え $F = \sum_{v \in V} f_v$ と分解可能であってもオラクルとして与えられる場合は $\Omega(n^2)$ 時間必要であることを示したが，この場合，$\mathrm{O}(n \log^2 n)$ 時間で解ける．前章で提案したアルゴリズムは問題 (9) の最適値及び定理 3.3 中の $v_j \in \Delta X_{j_0}$ を与えるテーブル g_X を作成するものである．このテーブル g_X は f_v $(v \in X)$ を予め定められた長さだけ平行移動させたもの $\hat{f}_v$ の下側エンベロープ $l_X(x) = \min_{v \in X} \hat{f}_v(x)$ を離散的に実現したものとみなすことができる．本節ではコスト関数 F が F_3 で与えられる場合，下側エンベロープを用いてテーブル g_X を表現することにより高速なアルゴリズムを開発する．

よく知られているように $f_1, \cdots, f_j$ の下側エンベロープと f_{j+1} が与えられたときそれらの下側エンベロープは $\mathrm{O}(\log j)$ 時間で計算できる．また $f_1, \cdots, f_n$ からそれらの下側エンベロープは $\mathrm{O}(n \log n)$ 時間で求められる．

3.2 節のアルゴリズム テーブルを以下のように改良して (ステップ０, １の (1-II), ２, ３の (3-II), ４は同じ)，テーブル g_X の代わりに下側エンベロープ l_X を作成し，問題を解く．

アルゴリズム　エンベロープ

ステップ１: $\tilde{T}$ の各葉 w と対応する $Y \in \mathcal{F}$ に対して，

(1-I) w が T の葉であるとき，$f_v (v \in \Delta Y)$ の下側エンベロープ l_Y を作成．
そうでないとき，

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

$$l_v(x + \Delta d(Y)) + \sum_{Z \in \mathcal{S}(Y)} l_Z(0) \qquad (v \in \Delta Y)$$

の下側エンベロープ l_X を作成．

ステップ３: $\tilde{T}$ における各 $Y \in \mathcal{S}(X_{i_0})$ に対して，
関数 f_{v_j} が $l_Y(0)$ を与えるとする．
($v_j \in \Delta X_{j_0}, w_{j_0} \to w_{j_1} \to \cdots \to w_{j_{(l+1)}}$
$(= w_{i_0})$ を $\tilde{T}$ 上の有向パスとする．)

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

ステップ１で w が T の葉でないとき，下側エンベロープを計算する際には，すでに計算されている下側エンベロープ $l_X(X \in \mathcal{S}(Y))$ 中で $|X|$ が最も大きい X^* の $l_{X^*}(x + \Delta d(Y)) + \sum_{Z \in \mathcal{S}(Y): Z \neq X^*} f_Z(0)$ にその他の半直線を足し込むことにより，全体の下側エンベロープを求めるものとする．

補題 4.1 アルゴリズム エンベロープは $\mathrm{O}(n \log^2 n)$ 時間で正しい解を出力する．

証明 アルゴリズムの正当性は，アルゴリズム テーブルと同様に示されるため，計算量のみについて議論する．

ステップ０, ２, ３, ４は，明らかに T の線形時間 $\mathrm{O}(n)$ で解くことができる．ステップ１の計算量は，w

が T の葉でないときは，$X \in \mathcal{S}(Y)$ の中で $|X|$ が最大である X^* の $l_{X^*}(x+\Delta d(Y))+\sum_{Z\in\mathcal{S}(Y):Z\neq X^*} l_Z(0)$ に他の半直線を足し込むことで l_Y を計算する．ここで，w が T の葉であるときも含めたアルゴリズム全体のステップ 1 における半直線を足し込む操作の回数 K を考える．半直線を足し込む側の集合，すなわち前述の X^* とならない集合全体からなる族を $\mathcal{G}$ とすると，回数 K は

$$K \leq \sum_{X\in\mathcal{G}} |X| + \sum_{X\in\mathcal{F}} |\Delta X|$$

を満たす．また，この $\mathcal{G}$ は，任意の $X, Y \in \mathcal{G}$ に対して $|X| \leq n$，かつ，$Y \subset X \Rightarrow 2|Y| \leq |X|$ を満たす．よって，$K \leq n\log n + n$ となり，ステップ 1 の計算量は

$$K \cdot \mathrm{O}(\log n) = \mathrm{O}(n\log^2 n).$$

まとめて，全体の計算量は $\mathrm{O}(n\log^2 n)$ となる． □

F がオラクルで与えられる場合は，各 v に対して，2 回のオラクル呼び出しにより a_v, b_v を求められる．従って，以下の定理が導かれる．

定理 4.2 コスト関数 F が F_3 で表されるとき，問題 (21) は $\mathrm{O}(n\log^2 n)$ 時間で解くことができる．また，F がオラクルで与えられる場合でも $\mathrm{O}(n(\log^2 n+q))$ 時間で解くことができる．

5 一般の凹コスト関数最小化

コスト関数 F が非負単調（非減少）凹関数であるが，式 (4)(5)(6) のいずれによっても記述できない一般の場合，次の否定的な結果を得る．

定理 5.1 コスト関数 F が一般の単調凹関数であるとき，

(I) F がオラクルとして与えられるとき，問題 (2) を解くためには $\Omega(2^{\frac{n}{2}}q)$ 時間必要である．

(II) F が明示的に与えられるときでも，問題 (2) は NP 困難である．

頁数制限のため，ここでは定理の詳しい証明は省略するが，(I) の場合，すなわち，F がオラクルとして与えられるときは，少なくとも $2^{\frac{n}{2}}$ 回以上のオラクル呼び出しを必要とする問題例が存在することから，定理が導かれる．また，(II) の場合，すなわち，F が明示的に与えられるときでも，NP 困難であると知られている充足可能性問題 (SAT) を我々の問題に帰着させることで，我々の問題 (2) が NP 困難であること，ならびに近似困難性が示される．

参考文献

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

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

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

[4] S. Fujishige: *Submodular Functions and Optimization* (North-Holland, 1991).

[5] 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.

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

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

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

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

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

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

[12] 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.

[13] 田村, 菅原, 仙石, 篠田: 無向フローネットワークにおける総合被覆問題について，電子情報通信学会論文誌, **J81-A** (1998), 863-869.