L 拡張可能関数と最小費用多品種フロー問題に対する近接スケーリングアルゴリズム

平井 広志
東京大学大学院 情報理工学系研究科
数理情報学専攻
hirai@mist.i.u-tokyo.ac.jp

January 8, 2016

Abstract

本論文では，木の直積上の定義される L 拡張可能関数と交代的 L 凸関数と呼ぶ新しい離散凸関数のクラスを導入し，その理論を展開する．我々は，その最適化問題に関する, 大域最適性基準，k-劣モジュラ関数最小化による最急降下アルゴリズム，持続性定理，近接性定理, 等の基本的な性質を確立する．

我々の理論は，最小費用多品種フロー問題のアルゴリズム設計に動機付けられている．この問題に対しては， Goldberg と Karzanov が容量スケーリングとコストスケーリングに基づく 2 つの組合せ的弱多項式時間アルゴリズムを与えているが，その解析は容易でなく，陽な計算量は不明である．我々の理論の応用として，この問題を解く新しいシンプルな弱多項式時間近接スケーリングアルゴリズムを与える．その計算量は，$O(n \log(nAC) \mathrm{MF}(kn, km))$ 時間である．ここで，n は頂点数，m は枝数，k はターミナルの個数，C は容量の総和，A は最大コスト，$\mathrm{MF}(n', m')$ は，頂点数 n'，枝数 m' のネットワークの最大流問題を解くための時間計算量である．我々のアルゴリズムは，より一般化された問題である最小費用点需要多品種フロー問題を同じ計算量で解くように設計されており，このクラスの問題に対する最初の組合せ的多項式時間アルゴリズムである．

1 序論

L$^\natural$ 凸関数 (Favati-Tardella [7], Murota [34], Fujishige-Murota [11]) とは，離散中点凸性と呼ばれる不等式

$$(1.1) \qquad g(x) + g(y) \geq g(\lfloor (x+y)/2 \rfloor) + g(\lceil (x+y)/2 \rceil) \quad (x, y \in \mathbf{Z}^n)$$

を満たす整数格子 $\mathbf{Z}^n$ 上の関数 g である．ここで，$\lfloor \cdot \rfloor$ ($\lceil \cdot \rceil$) は，各成分の小数部分の切り下げ (切り上げ) オペレータである．

L$^\natural$ 凸関数は，劣モジュラ集合関数の $\mathbf{Z}^n$ への拡張であり，離散凸解析 (Murota [35]) における基本的な離散凸関数のクラスを成している．L$^\natural$ 凸関数の代表的な例としては，以下のように表される関数がある：

$$(1.2) \qquad g(x) = \sum_i g_i(x_i) + \sum_{i,j} h_{ij}(x_i - x_j) \quad (x = (x_1, x_2, \ldots, x_n) \in \mathbf{Z}^n).$$

ここで， g_i や h_{ij} は，1 次元凸関数である．このような関数の最小化は，理論的にも実用的にも興味がある．例えば，最小費用ネットワークフローの双対問題がそれであり，また，コンピュータビジョンにも応用がある [31]. したがって L$^\natural$ 凸関数の理論は，これらの重要な最適化問題に統一的な視点を与えるものである．

L$^\natural$ 凸関数 g のいくつかの重要な性質を述べる．一つ目は，局所最適性が大域最適性を保証する L 最適性規準 [35, Theorem 7.14] である．各点 $x \in \mathbf{Z}^n$ において，x の離散的な「近傍」と

対応する局所的な最適化問題が定義され，x が近傍の中で最適 (局所最適) と大域最適であることが同値となる．さらに，この局所最適化問題は，劣モジュラ集合関数の最小化となり，多項式時間で解くことができる [14, 23, 37]．特に，x が局所最適でないと，$g(y) < g(x)$ となる別の点 y が求まる．そのような近傍点のなかで最も小さい $g(y)$ を持つものが自然に求まる．結果として得られる降下アルゴリズム「最急降下アルゴリズム」は，g の大域最適解を正しく求める [35, Section 10.3.1]．ここで，降下の回数は，初期解から大域最適解へのある種の l_∞ 距離に一致する [31, 36]．二つ目は，近接性定理 (Iwata-Shigeno [24]; [35, Theorem 7.18] も参照) である．$(2\mathbf{Z})^n$ 上での最小解 x に対して，その l_∞-半径 n 以内に $\mathbf{Z}^n$ 上の大域最適解が存在する．この興味深い性質は，L$^\natural$ 凸関数最小化に対する近接スケーリングアルゴリズムの基礎となる [35, Section 10.3.2]．

最近になって，L 凸性は，整数格子 $\mathbf{Z}^n$ をより一般化したグラフ構造においても考えられ始めている．まず，$\mathbf{Z}$ は無限の長さの有向パスの頂点集合と自然に同一視できることに注意する．すると， $\mathbf{Z}^n$ 上の関数は，これらのパスの n 直積上の関数とみなせる．さらに切り上げ切り下げオペレータ $\lceil , \rceil$, $\lfloor , \rfloor$ や離散中点凸性もこの同一視のもとで well-defined であることに注意する．したがって，L$^\natural$ 凸関数は，有向パスの直積グラフ上の関数として well-defined である．Kolmogorov [30] は，この観察に基づき，根付き木の直積上にツリー劣モジュラ関数 (tree-submodular function) という L$^\natural$ 凸関数の拡張を導入した．一方で，Hirai [16, 17] は，施設配置問題の計算複雑度分類と多品種フロー問題の組合せ的双対性理論に動機付けられて，モジュラ束が張り合わさったモジュラ複体 (modular complex) という離散構造上に L$^\natural$ 凸関数の類似物を定義した．

本論文は，この研究の流れに沿うものである．我々は，木 T の n 直積上に定義された関数に対し，L 拡張可能性 (L-extendability) という概念を導入する．この概念は，劣モジュラ緩和 (submodular relaxation) [13, 25, 29] や頂点被覆やマルチウェイカットなどの NP 困難問題の半整数緩和に動機付けられている．まず，我々は交代的 L 凸関数 (alternating L-convex function) というツリー劣モジュラ関数の変種を離散中点凸性の変種によって定義する．この関数クラスは，T がパスで $\mathbf{Z}$ と同一視するときは，Fujishige [10] による UJ 凸関数 (UJ-convex function) と一致する．さらに，半整数格子 $(\mathbf{Z}/2)^n$ の類似として，T の枝細分 T^* の n 直積 $(T^*)^n$ を考え，T^n 上の L 拡張可能関数 (L-extendable function) g を $(T^*)^n$ 上のある交代的 L 凸関数 $\bar{g}$ の T^n への制限によって得られる関数として定義する．ここで $\bar{g}$ を g の L 凸緩和 (L-convex relaxation) と呼ぶ．我々の貢献の最初の半分は，交代的 L 凸関数と L 拡張可能関数に対する以下のような基本的性質を確立したことである:

- 交代的 L 凸関数に L 最適性基準が類似が成立する．ここで，局所最適化問題は，k-劣モジュラ関数 (k-submodular function) の最小化となる．k-劣モジュラ関数は，Huber and Kolmogorov [22] によって導入された劣モジュラ集合関数と双劣モジュラ関数の拡張である．

- 交代的 L 凸関数に対する最急降下アルゴリズムには l_∞ 測地性が成立する．これは，降下の回数が初期解から最小解への l_∞ 距離に一致するというものである．

- L 拡張可能関数に対して以下のような近接定理が成り立つ．まず T を 2 つのカラークラス B, W を持つ 2 部グラフとみる．すると，B^n 上の最小解に対して，x を中心とする l_∞ 半径 $O(n)$ 以内に $T^n = (B \cup W)^n$ 上の大域最適解が存在する，というものである．

- L 凸緩和には，持続性定理 (persistency theorem) が成り立つ．これは，L 拡張可能関数の最小解がその L 凸緩和の任意の最小解 (つまり緩和解) をラウンディングすることで得られる，というものである．この性質は，双劣モジュラ緩和や k-劣モジュラ緩和において知られていた [13, 25, 29]([20] も参照)．

- 2 分離凸関数 (2-separable convex function) という L 拡張可能関数の便利な部分クラスを導入する．これは，(2.2) のように表現できる関数クラスの類似である．その自然な L 凸緩和に対する最急降下アルゴリズムは最大フローアルゴリズムによって実現できる．実際，局所最適化問題は，Iwata, Wahlström, and Yoshida [25] によって導入された基本 k-劣モジュラ関数の和としてかける特殊な k-劣モジュラ関数となる．彼らは，このクラスの k-

劣モジュラ関数の最小化が，最大フロー問題に帰着することを示している．したがって2分離凸関数のL凸緩和は効率的に最小化が可能である．あるケースにおいては，緩和解から2近似解が求まる．この近似アルゴリズムは，マルチウェイカット問題に対する古典的な2近似アルゴリズム [6] の拡張とみなせる．

以上の結果は，本論文の後半の貢献である最小費用多品種フロー問題に対する新しいシンプルな多項式時間スケーリングアルゴリズムの設計とその計算量評価に使用される．ターミナル集合 S を持つ整数枝容量をもつ無向ネットワークに対して，マルチフロー (多品種フロー) とは，異なるターミナルを結ぶパスの集合とその上の流量値関数の対であって容量条件を満たすものとする．最大一様マルチフローとは，マルチフローであって流量の総和が最大となるものである．いまネットワークの各枝に非負コストが与えられているとする．最小費用多品種フロー問題とは，最大一様マルチフローであって最小費用を持つものを求める問題である．Karzanov [27] は，常に半整数の最小費用最大一様マルチフローが存在することを示し，それを求める擬多項式時間アルゴリズムを与えた．その後，Karzanov [28] は，LP の多項式時間アルゴリズム (楕円体法や内点法) を用いる強多項式時間アルゴリズムを与えた．現在のところ，この問題に対する「組合せ的」強多項式時間アルゴリズムは知られていない．Goldberg and Karzanov [12] は，容量スケーリングとコストスケーリングに基づく2つの「組合せ的」弱多項式時間アルゴリズムを与えたが，そのアルゴリズムの記述と計算量の解析は容易ではなく，陽な計算量評価は与えられていない．

我々は，L拡張可能関数の理論の応用として，最小費用多品種フロー問題に対し，新しいシンプルな組合せ的弱多項式時間スケーリングアルゴリズムを与える．まず，この問題の双対問題を細分星 T の直積 T^n 上の2分離L凸関数の最小化として定式化できることを示す．ここで細分星 (subdivided star) とは星グラフに枝細分を繰り返して得られる木である．そして，近接スケーリング手法を適用する．各スケーリングステージで，スケールされた問題は，再び2分離L凸関数の最小化であり，上述のように最急降下アルゴリズムが最大フロー計算によって実装される．降下の反復回数は，近接定理，持続性定理，l_∞ 測地性定理によって評価される．最終ステージで，双対最適解が求まり，それにより，目標であった (半整数) 最小費用最大一様マルチフローが構成される．我々のアルゴリズムは，最小費用流 (の双対) に対する近接スケーリングアルゴリズム [1, 31] の多品種フローバージョンといえるものであり，陽な計算量評価を持つ最初の組合せ的多項式時間アルゴリズムである．アルゴリズムの計算量は， $O(n \log(nAC)\mathrm{MF}(kn, km))$ で，n は頂点数，m は枝数，k はターミナルの個数，C は容量の総和，は最大コスト，$\mathrm{MF}(n', m')$ は，頂点数 n'，枝数 m' のネットワークの最大流問題を解くための時間計算量である．また，我々のアルゴリズムは，より一般化された問題である最小費用点需要多品種フロー問題を同じ計算量で解くように設計されており，このクラスの問題に対する最初の組合せ的多項式時間アルゴリズムである．この多品種フロー問題は，ターミナルバックアップ問題 [2, 3, 41] と呼ばれるネットワークデザイン問題の LP 緩和として現れる．最近，Fukunaga [8] は容量制約付きターミナルバックアップ問題に対して (楕円体法などに得られた)LP 解を丸める 4/3 近似アルゴリズムを与えている．我々のアルゴリズムにより，この LP 解を効率的に求めることができるようになり，彼の近似アルゴリズムが効率的に実現できることになる．以下，第2節にL拡張可能関数の理論とその結果を述べ，第3節に多品種フローへの応用を述べる．本論文は，[18] の日本語版ダイジェストであり，省略した説明や証明は，そちらを参考にされたい．

Notation. 実数，非負実数，整数，非負整数の集合をそれぞれ $\mathbf{R}$, $\mathbf{R}_+$, $\mathbf{Z}$, $\mathbf{Z}_+$ で表す．拡大された実数の集合を $\overline{\mathbf{R}} := \mathbf{R} \cup \{\infty\}$ とする．ここで ∞ は無限大要素で，$x < \infty$ $(x \in \mathbf{R})$ が $\infty + x = \infty$ $(x \in \overline{\mathbf{R}})$ が成り立つものとする．関数 $f : E \to \overline{\mathbf{R}}$ に対し， $\mathrm{dom}\, f$ を $f(x) \neq \infty$ となる $x \in E$ からなる E の部分集合とする．また，部分集合 $X \subseteq E$ に対し，$\sum_{x \in X} f(x)$ を $f(X)$ で表す．

グラフ $G = (V, E)$ に対し，頂点 i から j への枝を ij と書く．頂点部分集合 X に対し， $i \in X$ かつ $j \notin X$ となる枝 ij から成る集合を δX と書く．頂点 s, t に対し，$s \in X \not\ni t$ となる頂点部分集合 X を (s, t)-カットという．頂点部分集合 A に対して， $s \in X \subseteq V \setminus A$ となるとき X を (s, A)-カットという．

2 L 拡張可能関数

2.1 準備: $\boldsymbol{k}$-劣モジュラ関数

非負整数 k に対し，特殊な元 0 を持つ $(k+1)$ 個の要素からなる集合を S_k とする．S_k 上の半順序 $\preceq$ を $0 \prec u$ $(u \in S_k \setminus \{0\})$ かつ他の関係はないものと定義する．すると S_k は 0 を最小元として持つ結び半束 (meet semilattice) になる．オペレータ $\sqcup$ を u, v が比較可能なら $u \sqcup v := u \vee v$，そうでないなら，$u \sqcup v := 0$ と定義する．非負整数の n 組 $\boldsymbol{k} = (k_1, k_2, \ldots, k_n)$ に対し，結び半束 $S_{\boldsymbol{k}}$ を直積 $S_{k_1} \times S_{k_2} \times \cdots \times S_{k_n}$ で定義する．$S_{\boldsymbol{k}}$ のオペレータ $\wedge, \sqcup$ を $x \wedge y := (x_1 \wedge y_1, x_2 \wedge y_2, \ldots, x_n \wedge y_n)$, $x \sqcup y := (x_1 \sqcup y_1, x_2 \sqcup y_2, \ldots, x_n \sqcup y_n)$ で定義する．

定義 2.1 ([22])**.** $f : S_{\boldsymbol{k}} \to \overline{\mathbf{R}}$ が不等式

$$f(x) + f(y) \geq f(x \wedge y) + f(x \sqcup y) \quad (x, y \in S_{\boldsymbol{k}})$$

満たすとき，$\boldsymbol{k}$-劣モジュラであるであるという．

もし，$\boldsymbol{k} = (k, k, \ldots, k)$ なら，$\boldsymbol{k}$-劣モジュラ関数を単に k-劣モジュラ関数という．1-劣モジュラ関数は通常の劣モジュラ (集合) 関数であり，2-劣モジュラ関数は双劣モジュラ関数である．劣モジュラ (集合) 関数と双劣モジュラ関数の場合と異なり，一般の $\boldsymbol{k}$-劣モジュラ関数が多項式時間で最小化できるかどうかはわかっていない．しかし，$\boldsymbol{k}$-劣モジュラ関数が次数の低い関数の和で与えられる場合は，Thapper and Živný [40] によって，多項式時間で最小化できることが示されている ([32] も参照).

ここでは，さらに特殊な $\boldsymbol{k}$-劣モジュラ関数を扱う．k, k' を非負整数とする．$a \in S_k$ に対し，ϵ_a と θ_a を以下で定義される 1 次元 k-劣モジュラ関数とする:

$$\epsilon_a(u) := \begin{cases} 1 & \text{if } u = a \neq 0, \\ 0 & \text{otherwise}, \end{cases} \qquad \theta_a(u) := \begin{cases} -1 & \text{if } u = a \neq 0, \\ 0 & \text{if } u = 0, \\ 1 & \text{otherwise}. \end{cases} \qquad (u \in S_k).$$

任意の 1 次元 k-劣モジュラ関数が以下のように ϵ_a と θ_a と定数の和でかけることをみるのは難しくない:

$$f = f(0) + (f(0) - f(a))\theta_a + \sum_{b \in S_k \setminus \{0,a\}} (f(b) - 2f(0) + f(a))\epsilon_b. \tag{2.1}$$

ここで a は f の最小値を達成する元である．$(a, a') \in S_k \times S_{k'}$ に対し，$\mu_{a,a'}$ を次で定義される $S_k \times S_{k'}$ の関数とする：

$$\mu_{a,a'}(u, v) := \begin{cases} 0 & \text{if } u = a \neq 0 \text{ or } v = a' \neq 0 \text{ or } u = v = 0, \\ 1 & \text{if } v = 0 \neq u \neq a \text{ or } u = 0 \neq v \neq a', \\ 2 & \text{otherwise}, \end{cases} \qquad ((u, v) \in S_k \times S_{k'}).$$

半順序同型写像 $\sigma : S_k \to S_{k'}$ に対し，δ_σ を次で定義される $S_k \times S_{k'}$ 上の関数とする．

$$\delta_\sigma(u, v) := \begin{cases} 0 & \text{if } v = \sigma(u), \\ 1 & \text{if } |\{u, v\} \cap \{0\}| = 1, \\ 2 & \text{otherwise}, \end{cases} \qquad ((u, v) \in S_k \times S_{k'}).$$

もしも σ が恒等写像 id ならば，δ_{id} を単に δ と書く．$\mu_{a,a'}$ と δ_σ はともに $\boldsymbol{k}$-劣モジュラ関数であることに注意する．基本 $\boldsymbol{k}$-劣モジュラ関数 f とは $S_{\boldsymbol{k}}$ $(\boldsymbol{k} = (k_1, k_2, \ldots, k_n))$ 上の関数であって以下のどれかのようにかけるものである：

型 I: ある i と 1 次元 k-劣モジュラ関数 $f' : S_{k_i} \to \overline{\mathbf{R}}$ に対して，$f(x) = f'(x_i)$.

型 II: 異なる i, j と同型写像 $\sigma : S_{k_i} \to S_{k_j}$ に対して，$f(x) = \delta_\sigma(x_i, x_j)$.

型 III: 異なる i,j と $(a,a') \in S_{k_i} \times S_{k_j}$ に対して，$f(x) = \mu_{a,a'}(x_i, x_j)$.

Iwata, Wahlström, and Yoshida [25] は，基本 k-劣モジュラ関数の和でかけるような $\boldsymbol{k}$-劣モジュラ関数は，最大フローアルゴリズムによって効率的に最小化できることを示した．

定理 2.2 ([25, Section 6])**.** $\boldsymbol{k} = (k_1, k_2, \ldots, k_n)$ として，m 個の基本 $\boldsymbol{k}$-劣モジュラ関数の非負結合は，$O(\mathrm{MF}(kn, km))$ 時間で最小化できる．ここで $k := \max k_i$.

この定理は，関数値と (s,t)-カット値が等しくなるようなネットワークを構成することによって示される．

2.2 交代的 L 凸関数

T を木として，T の頂点集合も T で表す．T を 2 部グラフとみて，2 つのカラークラスを B と W とする．B の頂点を黒頂点，W の頂点を白頂点という．T 上の半順序 $\preceq$ を，u と v が隣接し，かつ，$(u,v) \in W \times B$ であるときに $u \prec v$ と定義する．結果として得られる半順序集合には，長さ 2 のチェインは存在せず，B が極大元，W が極小元の集合となる．d を T 上の最短路距離とする．すなわち，$d(u,v)$ は，u と v を結ぶ唯一のパスの枝の個数である．2 つの頂点 $u, v \in T$ に対して，以下を満たす頂点対 (a,b) が唯一存在する: $d(u,v) = d(u,a) + d(a,b) + d(b,v)$, $d(u,a) = d(b,v)$, かつ，$d(a,b) \leq 1$. 特に $a = b$ であるか，a と b は隣接している．そして，$u \bullet v$ と $u \circ v$ を $\{u \bullet v, u \circ v\} = \{a, b\}$ と $u \circ v \preceq u \bullet v$ で定義する．すなわち， $a = b$ なら $u \bullet v = u \circ v = a = b$ であり，そうでないなら $u \bullet v$ は $\{a,b\}$ の黒頂点，$u \circ v$ は $\{a,b\}$ の白頂点である．n を正整数とする．T の n 直積 T^n を考える．T^n 上の l_∞ 距離 d を以下で定義する:

$$d(x,y) := \max_{1 \leq i \leq n} d(x_i, y_i) \quad (x, y \in T^n).$$

定義 2.3. $g : T^n \to \overline{\mathbf{R}}$ が不等式

$$g(x) + g(y) \geq g(x \bullet y) + g(x \circ y) \quad (x, y \in T^n) \tag{2.2}$$

を満たすとき交代的 L 凸という．

定義不等式 (2.2) は離散中点凸性 (1.1) の明らかなバリエーションである (注意 2.12 も参照). T^n を半順序集合 T の直積半順序集合とみる．$x = (x_1, x_2, \ldots, x_n) \in T^n$ の主イデアル $\mathcal{I}(x) := \{y \in T^n \mid y \preceq x\}$ と主フィルター $\mathcal{F}(x) := \{y \in T^n \mid y \succeq x\}$ は以下で与えられる:

$$\begin{aligned}
\mathcal{I}(x) &= \mathcal{I}(x_1) \times \mathcal{I}(x_2) \times \cdots \times \mathcal{I}(x_n), \\
\mathcal{F}(x) &= \mathcal{F}(x_1) \times \mathcal{F}(x_2) \times \cdots \times \mathcal{F}(x_n), \\
\mathcal{I}(x_i) &= \begin{cases} \{x_i\} \cup \{x_i \text{ の隣接点 }\} & \text{if } x_i \in B, \\ \{x_i\} & \text{if } x_i \in W, \end{cases} \\
\mathcal{F}(x_i) &= \begin{cases} \{x_i\} \cup \{x_i \text{ の隣接点 }\} & \text{if } x_i \in W, \\ \{x_i\} & \text{if } x_i \in B. \end{cases}
\end{aligned}$$

$\mathcal{I}(x_i)$ を半束 S_{k_i} $(k_i := |\mathcal{I}(x_i)| - 1)$ とみる．ここで最小元は x_i である．同様に $\mathcal{F}(x_i)$ を半束 $S_{k'_i}$ $(k'_i := |\mathcal{F}(x_i)| - 1)$ とみる．すると $\mathcal{I}(x) \simeq S_{\boldsymbol{k}}$ $(\boldsymbol{k} = (k_1, k_2, \ldots, k_n))$, $\mathcal{F}(x) \simeq S_{\boldsymbol{k}}$ $(\boldsymbol{k} = (k'_1, k'_2, \ldots, k'_n))$ である．この対応の下で，$\mathcal{F}(x)$ においては $\bullet = \sqcup$ と $\circ = \wedge$ が成り立ち，$\mathcal{I}(x)$ においては $\circ = \sqcup$ と $\bullet = \wedge$ が成り立つ．したがって次が成り立つ：

補題 2.4. 任意の $x \in T^n$ に対して，交代的 L 凸関数 g は $\mathcal{I}(x)$ 上で $\boldsymbol{k}$-劣モジュラであり，$\mathcal{F}(x)$ 上で $\boldsymbol{k}$-劣モジュラである．

次は，L$^\natural$ 凸関数の L 最適性基準 [35, Theorem 7.5] のアナログである．

定理 2.5. g を T^n 上の交代的 L 凸関数とする．もし，$x \in \operatorname{dom} g$ が g の最小解でないなら，$x' \in \mathcal{I}(x) \cup \mathcal{F}(x)$ であって $g(x') < g(x)$ となるものが存在する．

Proof. まず $g(y) < g(x)$ となる y が存在する．そのような y のうち $d(x,y)$ が最小になるものをとる．不等式 (2.2) によって，$2g(x) > g(x)+g(y) \geq g(x \bullet y) + g(x \circ y)$ を得る．したがって $g(x \bullet y) < g(x)$ または $g(x \circ y) < g(x)$ である．最小性と $d(y, x \bullet y) \leq \lceil d(x,y)/2 \rceil$, $d(y, x \circ y) \leq \lceil d(x,y)/2 \rceil$ より，$d(x,y) \leq 1$ を得る．すると $x \bullet y \in \mathcal{F}(x)$, $x \circ y \in \mathcal{I}(x)$ となり，主張を得る． $\square$

この定理によって，次のような交代的 L 凸関数の最小化アルゴリズムが得られる．これは，$\mathrm{L}^\natural$ 凸関数に対する最急降下アルゴリズム [35, Section 10.3.1] のアナログである．

最急降下アルゴリズム:

Input: 交代的 L 凸関数 $g : T^n \to \overline{\mathbf{R}}$, 初期点 $x \in \operatorname{dom} g$.

Step 1: y を $\mathcal{I}(x) \cup \mathcal{F}(x)$ における g の最小解とする

Step 2: もし $g(x) = g(y)$ なら停止; x は大域最小解． そうでないなら $x := y$ として step 1 に戻る．

定理 2.5 によって，アルゴリズムが step 2 で停止するとき，x は g の大域的最小解である．また step 1 は，g を $\mathcal{I}(x)$ 上と $\mathcal{F}(x)$ 上で最小化することで実行できる．補題 2.4 によって, g は $\mathcal{I}(x)$ 上と $\mathcal{F}(x)$ 上で, $\boldsymbol{k}$-劣モジュラであるので，step 1 は，$\boldsymbol{k}$-劣モジュラ関数最小化アルゴリズムによって実行される．特に，g が上に述べた特殊な形であれば，各反復は多項式時間で実行できる．時間計算量のバウンドを得るために反復回数を評価する．$\mathrm{L}^\natural$ 凸関数の場合は，反復回数は実行可能領域の l_∞ 直径の定数倍で抑えられることが示されている [31]. 最近の解析 [36] によると，反復回数は，初期点から最小解までのある種の l_∞ 距離に厳密に一致することが示されている ([38] も参照).

類似の結果が，交代的 L 凸関数に対しても成立する．$\mathrm{opt}(g)$ を g の最小解からなる集合とする．明らかに，反復回数は少なくとも初期点 x から $\mathrm{opt}(g)$ までの最小 l_∞ 距離 $d(\mathrm{opt}(g), x) := \min\{d(y,x) \mid y \in \mathrm{opt}(g)\}$ によって下から抑えられる．次の定理は，この下界はほとんどタイトであることを示している．

定理 2.6. 交代的 L 凸関数 $g : T^n \to \overline{\mathbf{R}}$ と初期点 $x \in \operatorname{dom} g$ に対し，最急降下アルゴリズムの総反復回数 m は，高々$d(\mathrm{opt}(g), x) + 2$ である．もしも $g(x) = \min_{y \in \mathcal{F}(x)} g(y)$ あるいは $g(x) = \min_{y \in \mathcal{I}(x)} g(y)$ なら，m は $d(\mathrm{opt}(g), x)$ に一致する．

2.3 L 拡張可能関数

定義 2.7. $h : B^n \to \overline{\mathbf{R}}$ が L 拡張可能であるとは，T^n 上のある交代的 L 凸関数 $g : T^n \to \overline{\mathbf{R}}$ によって h が g の B^n への制限となることである．

枝細分を利用して，T^n 上の関数に対しても L 拡張可能性を定義する．T の枝細分 T^* とは，T の任意の枝 $e = uv$ を 2 つの枝 $uw_e, w_e v$ に置き換えて得られるグラフである．この新しい頂点 w_e を uv の中点と呼ぶ．するともとの T は T^* の部分集合であり，T^* のカラークラスの 1 つである．T^* においては，T の頂点を黒点とする．中点は白点となる．

定義 2.8. $h : T^n \to \overline{\mathbf{R}}$ が (中点)L 拡張可能であるとは，$(T^*)^n$ 上のある交代的 L 凸関数 $g : (T^*)^n \to \overline{\mathbf{R}}$ によって h が g の T^n への制限となることである．

特に，この g を h の L 凸緩和と呼ぶ．もしも $\min_{x \in T^n} g(x) = \min_{x \in (T^*)^n} h(x)$ なら，g を厳密 L 凸緩和と呼ぶ．実は，交代的 L 凸関数は厳密 L 凸緩和を持つので L 拡張可能関数は，交代的 L 凸関数より (真に) 広いクラスである [19](注意 2.12 も参照)．頂点被覆問題やマルチウェイカット問題 (の目的関数) は，L 凸緩和を持つ．従って，L 拡張可能関数の最小化は一般に NP 困難である．しかし，L 拡張可能関数は以下に述べるいくつかの便利な性質を持つ．これらは，第 3 章に述べる最小費用多品種フロー問題に対する近接スケーリングアルゴリズムの設計と計算量解析で重要な役割を演ずる．最初の性質は，定理 2.5 に似た最適性規準である．

定理 2.9. $h : B^n \to \overline{\mathbf{R}}$ を L 拡張可能関数とする．もしも $x \in \mathrm{dom}\, h$ が最小解でないなら，$d(x,y) \leq 2$ となる $y \in B^n$ が存在する．

2 つ目の性質は，持続性 (persistency) と呼ばれるものである．この性質は，双劣モジュラ緩和に対して Kolmogorov [29] に定義され，k-劣モジュラ緩和にも拡張されている [13, 25]. この性質は，緩和 g の最小解をラウンディングすることで h の最小解 y が得られるというものである．

定理 2.10. $h : B^n \to \overline{\mathbf{R}}$ を L 拡張可能関数とする．そして $g : T^n \to \overline{\mathbf{R}}$ をその L 凸緩和とする．このとき g の最小解 x に対して，h の最小解 y であって $y \in \mathcal{F}(x) \cap B^n$ となるものが存在する．

3 つ目は近接性定理である．L$^\natural$ 凸関数 $g : \mathbf{Z}^n \to \overline{\mathbf{R}}$ の場合は，g の $(2\mathbf{Z})^n$ 上の最小解 x に対して x の l_∞ 半径 n 以内に g の $\mathbf{Z}$ 上の最適解 y が存在するというものである (Iwata-Shigeno [24]; [35, Theorem 7.6] と [9, Theorem 20.10] も参照). このアナロジーが L 拡張可能関数に対しても成り立つ．

定理 2.11. $h : T^n \to \overline{\mathbf{R}}$ を (中点)L 拡張可能関数とする．x を h の B^n 上での最小解とする．このとき，$d(x,y) \leq 2n$ となる h の最小解 y が存在する．特に h が厳密 L 凸緩和を持つときは，$d(x,y) \leq n$ となる h の最小解 y が存在する．

注意 2.12. 離散中点凸性には様々な変種が考えられ，それに対応する離散凸関数のクラスが考えられる．いま T の各枝に向きが付けられているとする．中点オペレータ $\mathrm{mid} : T \times T \to T^*$ を $\mathrm{mid}(p,q)$ が $d(p,u) = d(u,q)$ と $d(p,u) + d(u,q) = d(p,q)$ を満たす唯一の点 $u \in T^*$ となるよう定義する．$u \in T^* \setminus T$ に対し，$\overline{u}$ と $\underline{u}$ を T の頂点であって $\overline{u}\underline{u}$ が T の (有向) 枝を成し u がその中点となるものとする．$u \in T$ に対しては $\overline{u} = \underline{u} := u$ と定義する．これらのオペレータを T^n, $(T^*)^n$ 上に各成分毎に適用することで拡張する．T 上の関数 g であって，以下の不等式を満たすものを考える:

$$g(x) + g(y) \geq g(\overline{\mathrm{mid}(x,y)}) + g(\underline{\mathrm{mid}(x,y)}) \quad (x, y \in T^n). \tag{2.3}$$

T がパスで $\mathbf{Z}$ と同一視し，i と $i+1$ を結ぶ枝に $i \to i+1$ と向きをつけると，上の不等式 (2.3) は離散中点凸性に一致し，g は．L$^\natural$ 凸関数である．T が唯一のシンクを持つとき，つまり，T が根付き木であるとき，オペレータ $\overline{\mathrm{mid}}$ と $\underline{\mathrm{mid}}$ はそれぞれ Kolmogorov [30] の意味での $\sqcup$ と $\sqcap$ に一致する．そして g はツリー劣モジュラである．もしも，向き付けがジグザグ向き付けであるとき，g は交代的 L 凸関数である．したがって異なる T の向き付けは異なるクラスの離散凸関数を定義することになる．

L 拡張可能関数の理論は，これらすべてクラスを次の意味で扱うことが出来る．

(2.4) $g : T^n \to \overline{\mathbf{R}}$ が (2.3) を満たすと，g は (中点)L 拡張可能であり，厳密 L 凸緩和 $\bar{g} : (T^*)^n \to \overline{\mathbf{R}}$ が以下で与えられる：

$$\bar{g}(u) := (g(\overline{u}) + g(\underline{u}))/2 \quad (u \in (T^*)^n).$$

証明は [19] で与えられる．特に，g の T^n 上での最小化を考えるとき，代わりに $\bar{g}$ の $(T^*)^n$ 上での最小化を考えればよい．そして上述の結果が $\bar{g}$ に対して適用可能となる．この事実は，交代的 L 凸関数と L 拡張可能関数をセットで導入したことの 1 つの理由でもある．

2.4 2 分離凸関数

ここでは，2 分離凸関数という L 拡張可能関数の特殊クラスを導入する．これは，次のように表される $\mathbf{Z}^n$ 上の関数 f のアナロジーである：

$$\sum_i f_i(x_i) + \sum_{i,j} g_{ij}(x_i - x_j) + \sum_{i,j} h_{ij}(x_i + x_j) \quad (x \in \mathbf{Z}^n). \tag{2.5}$$

ここで f_i, g_{ij}, h_{ij} は $\mathbf{Z}$ 上の 1 次元凸関数である．Hochbaum [21] は，このような関数の最小化を考え，半整数緩和と 2 近似を持つ NP 困難な問題に対する統一的な視点を与えている．実際，(1.2) の関数 f は，自然な半整数緩和を有し，その最小化は最大フローアルゴリズムによって効率的におこなえる．

2 分離凸関数に対する類似の結果が成り立つことを示す．まずいくつかの準備をする．$\mathbf{Z}$ 上の関数 h が非減少であるとは，$\Delta h(t) := h(t) - h(t-1) \geq 0$ となることであり， 凸であるとは $\Delta^2 h(t) := h(t+1) - 2h(t) + h(t-1) \geq 0$ となることである．さらに偶であるとは任意の奇整数 t に対して $(h(t-1) + h(t+1))/2 = h(t)$ となることである．木 T 上には，自然に凸関数概念が導入される．このような概念は施設配置問題の古い文脈で考察されていた [5, 26, 39]．$u, v \in T$ と $0 \leq t \leq d(u,v)$ となる整数 t に対し，$[u,v]_t$ を $d(u,v) = d(u,s) + d(s,v)$ と $t = d(u,s)$ を満たすユニークな点 s とする．T 上の関数 h が凸であるとは， 任意の $u, v \in T$ に対し，$t \mapsto h([u,v]_t)$ で決まる $\mathbf{Z}$ 上の関数が凸であることとする．

補題 2.13. T 上の関数に対して凸性，交代的 L 凸性，L 拡張可能性は同値である．T 上の 2 つの凸関数 f, g と $\alpha, \beta \in \mathbf{R}_+$ に対して $\alpha f + \beta g$ は凸関数，$u \mapsto \max\{f(u), g(u)\}$ で決まる関数も凸関数．

h を $\mathbf{Z}$ 上の関数とする．3 つの関数 $h_T, h_{T;z}, h_{T;z,w}$ $(z, w \in T)$ を以下で定義する:

$$\begin{aligned} h_T(u,v) &:= h(d(u,v)) \quad (u,v \in T), \\ h_{T;z}(u) &:= h(d(u,z)) \quad (u \in T), \\ h_{T;z,w}(u,v) &:= h(d(u,z) + d(v,w)) \quad (u,v \in T). \end{aligned}$$

定理 2.14. h を $\mathbf{Z}$ 上の非減少凸関数とする．z, w を T の頂点とする．

(1) $h_{T;z}$ は T 上の凸関数である．

(2) h は偶とする．このとき h_T は交代的 L 凸である．さらに，$(u,v) \in T \times T$ に対して，h_T は，$\mathcal{F}(u) \times \mathcal{F}(v)$ 上で，基本 $\boldsymbol{k}$-劣モジュラ関数の和である．具体的には，

$$h_T(s,t) - h(D) = \begin{cases} \Delta h(1)\delta(s,t) & \text{if } u = v \in W, \\ \Delta h(D)\theta_a(s) & \text{if } u \in W, v \in B, \\ \Delta h(D)\theta_b(t) & \text{if } u \in B, v \in W, \\ \Delta h(D)(\theta_a(s) + \theta_b(t)) + \Delta^2 h(D)\mu_{a,b}(s,t) & \text{if } u, v \in W : u \neq v, \\ 0 & \text{if } u, v \in B \end{cases}$$

となる．ここで $D := d(u,v)$，a は $\mathcal{F}(u)$ 上で v に最も近い点で，b は $\mathcal{F}(v)$ 上で u に最も近い点である．

(3) h を偶として，z, w が同じカラークラスに属するとする．このとき $h_{T;z,w}$ は交代的 L 凸である．さらに，$(u,v) \in T \times T$ に対して，$h_{T;z,w}$ は $\mathcal{F}(u) \times \mathcal{F}(v)$ 上で基本 $\boldsymbol{k}$-劣モジュラ関数の和である．具体的には

$$h_{T;z,w}(s,t) - h(D) = \begin{cases} \Delta h(D)\theta_a(s) & \text{if } u \in W, v \in B, \\ \Delta h(D)\theta_b(t) & \text{if } u \in B, v \in W, \\ \Delta h(D)(\theta_a(s) + \theta_b(t)) + \Delta^2 h(D)\mu_{a,b}(s,t) & \text{if } u, v \in W, \\ 0 & \text{if } u, v \in B \end{cases}$$

となる．ここで $D := d(u,z) + d(v,w)$, a は，$\mathcal{F}(u)$ 上で z に最も近い点で b は，$\mathcal{F}(v)$ 上で w に最も近い点である．

$\mathcal{I}(u) \times \mathcal{I}(v)$ 上の局所的な表現は，B と W の役割を入れ替えて得られる．T^n 上の関数 ω が定理 2.14 の関数の和でかけるとき 2 分離 L 凸という．2 分離 L 凸関数とは，次のようにかける関数である：

$$\omega(x) := \sum_i f_i(x_i) + \sum_{i,j} g_{ij}(d(x_i, x_j)) + \sum_{i,j} h_{ij}(d(x_i, z_i) + d(x_j, w_j)) \quad (x \in T^n). \tag{2.6}$$

ここで f_i は T 上の凸関数, g_{ij} と h_{ij} は $\mathbf{Z}$ 上の非減少で偶な凸関数，z_i と w_j は同じカラークラスに属する頂点である．B^n 上の関数 ω が 2 分離凸であるとは，ω が (2.6) のようにかけることをいう．g_{ij} と h_{ij} が (偶とは限らない) 非減少凸関数でよい．2 分離凸関数は L 拡張可能である．L 凸緩和 $\bar{\omega}$ は,

$$\bar{\omega}(x) := \sum_i f_i(x_i) + \sum_{i,j} \bar{g}_{ij}(d(x_i, x_j)) + \sum_{i,j} \bar{h}_{ij}(d(x_i, z_i) + d(x_j, w_j)) \quad (x \in T^n) \tag{2.7}$$

で与えられる．ここで $\bar{g}_{ij}$ と $\bar{h}_{ij}$ は g_{ij} と h_{ij} の各奇数 z の値を $(g_{ij}(z-1) + g_{ij}(z+1))/2$ と $(h_{ij}(z-1) + h_{ij}(z+1))/2$ に置き換えて得られる偶な関数である．T^n 上の関数 ω が 2 分離凸であるとは，ω が (2.6) のようにかけることをいう．ここで g_{ij} と h_{ij} は非減少凸関数で z_i と w_j は T の任意の頂点である．すると ω は (中点)L 拡張可能である．定理 2.14 によって，L 凸緩和は，局所的に基本 k-劣モジュラ関数の和である．従って，$\bar{\omega}$ は最大フロー問題を逐次的に解くことで最小化できる．また，以下のように表せる 2 分離凸関数を凸施設配置関数 (convex multifacility location function) という：

$$\omega(x) = \sum_{i,j} f_{ij}(d(x_i, z_j)) + \sum_{i,j} g_{ij}(d(x_i, x_j)) \quad (x \in B^n). \tag{2.8}$$

ここで f_{ij} と g_{ij} は非負非減少凸関数であり，z_j は黒点とする．このときは，L 凸緩和として

$$\bar{\omega}(x) = \sum_{i,j} \bar{f}_{ij}(d(x_i, z_j)) + \sum_{i,j} \bar{g}_{ij}(d(x_i, x_j)) \quad (x \in T^n) \tag{2.9}$$

がとれる．

定理 2.15. $\omega : B^n \to \mathbf{R}$ を 2 分離凸関数で m 個の項からなるものとする．$\bar{\omega} : T^n \to \mathbf{R}$ をその L 凸緩和とする．

(1) 初期点 $x \in B^n$ に対して，$O(d(\mathrm{opt}(\bar{\omega}), x)\mathrm{MF}(kn, km))$ 時間で $\bar{\omega}$ の T^n 上の大域最小解 x^* が求まる．ここで k は T の最大次数である．

(2) ω を凸施設配置関数とする．$y \in B$ に対し，丸め解 $(x^*)_{\to y}$ は ω の 2 近似解である．

ここで T^n から B^n への丸めは以下で与えられる．$y \in B$ と $x \in T^n$ に対して $x_{\to y}$ を $z \in B^n$ であって $x_i \in B$ なら $z_i = x_i$，$x_i \in W$ なら z_i を y へ近づく隣接点となるものとする．(1) は定理 2.2, 2.6, 2.14 の帰結である．(2) は，マルチウェイカット問題に対する 2 近似アルゴリズム [6] を拡張ともみなせる．

3 最小費用多品種フロー

本節では，前節の結果の多品種フロー理論への応用を述べる．$\mathcal{N}$ を頂点集合 V，枝集合 E，ターミナル集合 $S \subseteq V$，枝容量 $c : E \to \mathbf{Z}_+$ の無向ネットワークとする．S-パスとは S の異なるターミナルを結ぶパスのことである．マルチフロー (多品種フロー)f とは，S-パスの集合 $\mathcal{P}$ とその上の流量値関数 $\lambda : \mathcal{P} \to \mathbf{R}_+$ の対 $(\mathcal{P}, \lambda)$ であって容量条件

$$f(e) := \sum \{\lambda(P) \mid P \in \mathcal{P}\text{: } P \text{ は } e \text{ を含む }\} \leq c(e) \quad (e \in E) \tag{3.1}$$

を満たすものである．2λ が整数値のとき，f は半整数であるという．異なるターミナル $s, t \in S$ に対して，$f(s,t)$ を f 内の (s,t)-フローの総流量，すなわち，

$$f(s,t) := \sum \{\lambda(P) \mid P \in \mathcal{P}\text{: } P \text{ は } (s,t)\text{-パス }\}$$

とする．さらに $f(s)$ を s に流れ込む総流量，つまり， $f(s) := \sum_{t\in S\setminus\{s\}} f(s,t)$ とする．総流量 v_f を以下で定義する．

$$v_f := \sum_{P\in\mathcal{P}} \lambda(P) = \sum_{s,t\in S: s\neq t} f(s,t) = \frac{1}{2}\sum_{s\in S} f(s).$$

最大一様マルチフローとは，マルチフロー f であって総流量 v_f が最大であるものである．最大一様マルチフロー問題 (MF) とは，それを求める問題である．

(MF) 最大一様マルチフローを求めよ．

$f(s)$ は f 内の $(s, S\setminus\{s\})$-フローの総流量であり，それは $(s, S\setminus\{s\})$-カットの最小値 κ_s で上から抑えられる．したがってマルチフローの総流量は $\sum_{s\in S}\kappa_s/2$ で上から抑えられる．Lovász [33] と Cherkassky [4] による古典的定理は，この上界が半整数マルチフローで達成されるというものである．

定理 3.1 ([4, 33])**.** 最大一様マルチフローの流量は，

$$\frac{1}{2}\sum_{s\in S}\kappa_s$$

に一致する．さらに半整数最大一様マルチフローが存在する．

Karzanov [27] は (MF) の最小費用バージョンを考えた．いまネットワーク $\mathcal{N}$ に非負枝費用 $a: E\to\mathbf{Z}_+$ が与えられているとする．マルチフロー f の総費用 a_f を $a_f := \sum_{e\in E} a(e)f(e)$ と定義する．最小費用最大一様マルチフロー問題とは以下の問題である：

(MCMF) 最大一様マルチフローであって最小費用を持つものを求めよ．

ここでは，この問題に対し，[12, 15, 27, 28] とは異なる一般化された定式化を考える．今，さらに非負の需要 $r: S\to\mathbf{Z}_+$ がターミナル各点に与えられているとする．マルチフロー f が r に対して実行可能であるとは，$f(s)\geq r(s) (s\in S)$ を満たすこととする．最小費用実行可能マルチフロー問題を以下に導入する：

(N) 実行可能マルチフローであって最小費用を持つものを求めよ．

問題 (MCMF) が (N) に帰着することをみるのは難しくない．$r(s) := \kappa_s$ と設定すればよい．

問題 (N) は自然なものであるが，最近まで研究されてこなかったようである．Fukunaga [8] は，ターミナルバックアップ問題 [2, 3, 41] というネットワークデザイン問題の LP 緩和として (N) を導入した．[8] でも指摘されているように，問題 (N) は次のカット被覆線形計画問題とも定式化できる．

$$\begin{array}{lll}
\textbf{(L)} & \text{Min.} & \displaystyle\sum_{e\in E} a(e)x(e) \\
& \text{s.t.} & x(\delta X)\geq r(s) \quad (s\in S, X\in\mathcal{C}_s), \\
& & 0\leq x(e)\leq c(e) \quad (e\in E).
\end{array}$$

ここで，$\mathcal{C}_s$ は，$(s, S\setminus\{s\})$-カットの集合である．

補題 3.2 (see [8])**.** 2つの問題 (N) と (L) の最適値は一致し，以下が成立する．

(1) (N) の最適解 f に対して，$x(e) := f(e)$ $(e\in E)$ で決まるフローサポート $x: E\to\mathbf{R}_+$ は，(L) の最適解である．

(2) (L) の最適解 $x: E\to\mathbf{R}_+$ に対して，x を枝容量とした $\mathcal{N}$ の実行可能マルチフロー f が存在して，(N) の最適解である．

Fukunaga [8] は，(N) と (L) の半整数性を証明した．

定理 3.3 ([8])**.** (N) と (L) には半整数最適解が存在する．それらは強多項式時間で求まる．

この多項式時間可解性は，(L) を解く多項式時間 LP ソルバーに依存する．(L) の実行可能領域は，最小カット計算による多項式時間分離オラクルをもつ．したがって楕円体法で (L) を多項式時間で解くことができる．

前節の結果の応用として，(N), (L), そして (MCMF) の半整数最適解を求める組合せ的な多項式時間スケーリングアルゴリズムが構築される．

定理 3.4. $O(n \log(nAC)\,\mathrm{MF}(kn, km))$ 時間で (N), (L), そして (MCMF) の半整数最適解が求まる．ここで，n は頂点数，m は枝数，k はターミナルの個数，C は容量の総和，A は最大コストである．

我々の知る限り，このアルゴリズムは，(N) と (L) を解く最初の組合せ的多項式時間アルゴリズムであり，(MCMF) を解く陽な計算量評価を持つ最初の組合せ的多項式時間アルゴリズムである．

3.1 双対性

まず，(N) の双対問題を木の上の配置問題にする．ここでの木は位相的な意味での木，つまり，可縮な 1 次元複体のことである．$\mathbf{R}^S$ を S 上の関数の集合とする．ターミナル $s \in S$ に対して，e_s を $e_s(s) := 1$, $e_s(t) := 0$ $(t \neq s)$ となる関数とする．つまり s 番目の単位ベクトルである．$\mathcal{T}_s := \mathbf{R}_+ e_s$ とする．そして $\mathcal{T} := \bigcup_{s \in S} \mathcal{T}_s \subseteq \mathbf{R}^S$ とする．$\mathcal{T}$ 上の距離 D を

$$D(p,q) := \begin{cases} |p(s) - q(s)| & \text{if } p, q \in \mathcal{T}_s \text{ for } s \in S, \\ |p(s)| + |q(t)| & \text{if } p \in \mathcal{T}_s, q \in \mathcal{T}_t \text{ for distinct } s, t \in S, \end{cases} \qquad (p, q \in \mathcal{T})$$

と定義する．$\mathcal{T}$ は，半直線 $\mathbf{R}_+$ を原点で貼り合わせた星状の空間である．D は $\mathbf{R}^S$ から誘導される距離とは異なることに注意する．

$V = \{1, 2, \ldots, n\}$ とする．次の $\mathcal{T}$ 上の連続的な配置問題 (D) を考える．

$$\begin{aligned} \textbf{(D)}\text{: Max.} \quad & \sum_{s \in S} r(s) D(0, p_s) - \sum_{ij \in E} c(ij)(D(p_i, p_j) - a(ij))^+ \\ \text{s.t.} \quad & p = (p_1, p_2, \ldots, p_n) \in \mathcal{T} \times \mathcal{T} \times \cdots \times \mathcal{T}, \\ & p_s \in \mathcal{T}_s \quad (s \in S). \end{aligned}$$

ここで $(z)^+ := \max(0, z)$. (D) の実行可能解 p をポテンシャルという，そして，各成分 p_i が $\mathbf{R}^S$ の半整数ベクトルのとき，半整数という．ポテンシャル p が任意の $s \in S$ と $p_i \in \mathcal{T}_s$ となる $i \in V$ に対し，$D(0, p_i) \leq D(0, p_s)$ となるとき，このポテンシャルをプロパーであるという．

命題 3.5. (N) の最小値と (D) の最大値は一致する．さらに (D) には，プロパーな半整数最適ポテンシャルが存在する．

なぜ，このような双対性が成立するのか手短に説明する．実は，(D) は (L) の LP 双対問題の同値な別表現である．(L) の LP 双対問題を素朴に考えると，それは $\mathcal{C}$ の分数カット詰め込み問題となる．ここでよく知られる uncrossing 手法によって，非ゼロサポートがラミナーになる最適解の存在が示される．このラミナー最適解にラミナー族の木表現を考える．この木は星状になり (D) のポテンシャルを対応させることができる．目的関数値も D を用いて解釈される．半整数性は定理 3.3 から従う．

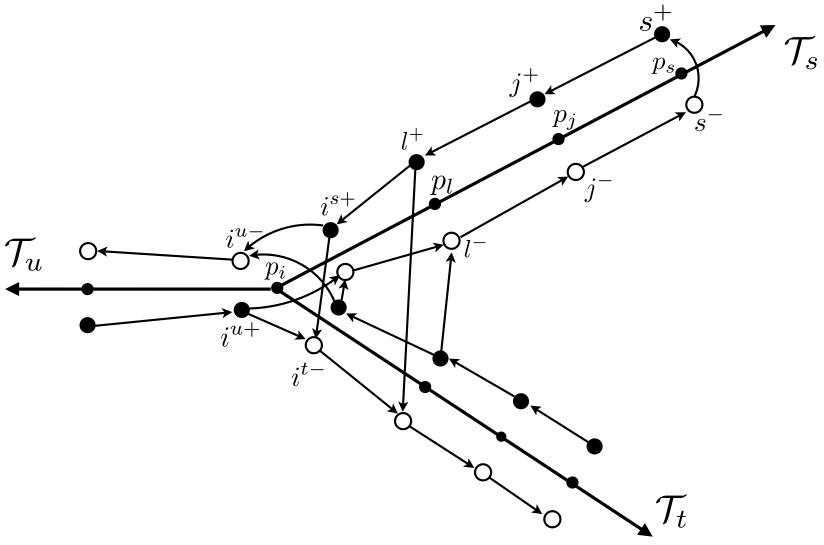

Figure 1: ２重被覆ネットワーク

3.2　2 重被覆ネットワーク

ここでは，(D) の最適ポテンシャルから (N) の最適マルチフローを求めるアルゴリズムを述べる．ここで次の仮定をおく．

(CP) すべての $e \in E$ の費用は正，すなわち，$a(e) > 0$.

a を適切に摂動することによって，この仮定は常に満たされる．すると，最適ポテンシャル p に対し，2 重被覆ネットワーク $\mathcal{D}_p$ と呼ぶ上下限枝容量付き有向ネットワークが対応し，その循環フローから (N) の最適マルチフローが得られる．これは Karzanov [28] が (MCMF) を解くときに行った手法のマイナーバリエーションである．

p を (プロパーな) ポテンシャルとする．U_0 を $p_i = 0$ を満たす非ターミナル頂点 i からなる集合である．ターミナル $s \in S$ に対して，U_s をターミナル s と $p_i \in \mathcal{T}_s \setminus \{0\}$ を満たす非ターミナル頂点 i からなる集合である．V は，U_0 と $U_s (s \in S)$ の素な和集合である．$E_=$ を $D(p_i, p_j) = a(ij)$ となる枝 ij の集合で，$E_>$ を $D(p_i, p_j) > a(ij)$ となる枝 ij の集合とする．p に対する 2 重被覆ネットワーク $\mathcal{D}_p$ を以下のように構成する．各ターミナル s に対し，2 つの頂点 s^+, s^- を考える．各非ターミナル頂点 i で U_0 に属してないものに対し，2 つの頂点 i^+, i^- を考える．U_0 の各非ターミナル頂点 i に対し，$2|S|$ 個の頂点 i^{s+}, i^{s-} $(s \in S)$ を考える．$\mathcal{D}_p$ の頂点集合はこれらの頂点からなる．次に $\mathcal{D}_p$ の枝集合 A を定義する．各枝 $ij \in E_= \cup E_>$, 枝集合 A_{ij} を以下で定義する：

$$A_{ij} := \begin{cases} \{j^+i^+, i^-j^-\} & \text{if } i, j \in U_s, D(0, p_i) < D(0, p_j), \\ \{j^+i^{s+}, i^{s-}j^-\} & \text{if } i \in U_0, j \in U_s, \\ \{i^+j^-, j^+i^-\} & \text{if } i \in U_s, j \in U_t, s \neq t. \end{cases}$$

ここで，$ij \in E_= \cup E_>$ に対し，端点のポテンシャル p_i,p_j は $\mathcal{T}$ の異なる点であることに注意する．これは $a(ij)$ が正であるという仮定 (CP) のおかげである．A_{ij} の 2 つの枝の容量上限を $c(ij)$ とする．容量下限を $ij \in E_=$ なら 0，$ij \in E_>$ なら $c(ij)$ とする．U_0 の各点 i に対して，枝集合 B_i を $\{i^{s+}i^{t-} \mid s, t \in S, s \neq t\}$ と定義する．各枝の容量上限を ∞，容量下限を 0 とする．各ターミナル $s \in S$ に対し，枝 s^-s^+ を考え，容量下限を $r(s)$，上限を $p_s = 0$ なら ∞，そうでないなら $r(s)$ とする．$\mathcal{D}_p$ の枝集合 A を A_{ij} $(ij \in E_= \cup E_>)$, B_i $(i \in U_0)$, $\{s^-s^+\}$ $(s \in S)$ の和集合とする．図 1 のように $\mathcal{D}_p$ が写像 $i^\pm \mapsto p_i$ によって $\mathcal{T}$ に埋め込まれていると考える．

さて，$\mathcal{D}_p$ の整数循環フロー $\phi : A \to \mathbf{Z}_+$ を (それが存在するとして) 考える．ϕ を有向サイクル $C_1, C_2, \ldots, C_{m'}$ の特徴ベクトルの非負結合として表す．ここで結合係数 $q_1, q_2, \ldots, q_{m'}$ を

して正整数としてよい．m' は高々$\mathcal{D}_p$ の枝数である．$\mathcal{D}_p$ の構成法より，各有向サイクルはあるターミナル s に対する枝 s^-s^+ と交わり，次に別のターミナル t の t^-t^+ と交わる．すべてのターミナル枝 s^-s^+ を各 C_l から取り除く，すると，有向パス $P_l^1, P_l^2, \ldots, P_l^{n_l}$ $(n_l \leq |S|)$ が得られる．各 P_l^j は異なるターミナル s,t に対する s^+ から t^- へのパスである．$\bar{P}_l^r$ を元のネットワーク $\mathcal{N}$ で P_l^r に対応する S-パスとする．それは P_l^r において $i^{\pm}$ や $i^{s\pm}$ を i と置き換え，無駄な枝 $i^{s+}i^{t-}$ を取り除くことで得られる．$\mathcal{P}$ を S-パス $\bar{P}_l^r$ からなる集合とし，$\lambda(\bar{P}_l^r) := q_l^r/2$ とおく．すると $f_\phi := (\mathcal{P}, \lambda)$ が半整数の最適マルチフローになる：

命題 3.6. ポテンシャル p が最適であることと $\mathcal{D}_p$ に循環フローが存在することは同値である．さらに任意の (整数) 循環フロー ϕ に対し， f_ϕ は，(N) に対する (半整数) 最適マルチフローである．

この命題の証明の概略を述べる．命題 3.5 より，マルチフロー f とポテンシャル p がそれぞれ最適になるための最適性条件が得られるが，それは，f のフローサポートに対する kilta 条件と f の各パスが p によって $\mathcal{T}$ 上の測地線を誘導するという条件からなる．kilta 条件は，$\mathcal{D}_p$ の枝の上下限容量制約に対応し，測地線条件は，$\mathcal{D}_p$ の構成法より自動的に満たされる．

3.3 近接スケーリングアルゴリズム

ここでは定理 3.4 を証明するアルゴリズムを与える．前節の議論によって，(D) を解けば十分である．$\omega : \mathcal{T}^n \to \overline{\mathbf{R}}$ を

$$\omega(p) := \sum_{ij \in E} c(ij)(D(p_i, p_j) - a(ij))^+ + \sum_{s \in S} I_s(p_s) - r(s)D(0, p_s) \quad (p \in \mathcal{T}^n) \tag{3.2}$$

で定義する．ここで I_s は $\mathcal{T}_s$ の支持関数

$$I_s(q) := \begin{cases} 0 & \text{if } q \in \mathcal{T}_s, \\ \infty & \text{otherwise,} \end{cases} \quad (q \in \mathcal{T})$$

とする．すると (D) は ω の最小化と同値である．最適解が存在する範囲は以下で与えられる．ここで $A := \max\{a(e) \mid e \in E\}$.

補題 3.7. プロパーな半整数最適ポテンシャル p であって $D(0, p_i) \leq nA$ $(i = 1, 2, \ldots, n)$ となるものが存在する．

$L := \lceil \log nA \rceil$ とする．$\mathcal{T}'$ を $\mathcal{T}$ の部分集合で $D(0, q) \leq 2^L$ となる点 q からなるものとする．上の補題のよって，(D) は， $(\mathcal{T}')^n$ 上での ω の最小化問題となる．$\sigma = -1, 0, 1, 2, \ldots, L$ に対し，T_σ を頂点集合 $\mathcal{T}' \cap (2^\sigma \mathbf{Z}^S)$ をもつ (グラフの意味での) 木であって $D(u, v) = 2^\sigma$ となる u, v が隣接するものとする．特に，T_σ は $\mathcal{T}'$ の離散化である．T_σ のグラフ距離を d_σ とかく．このとき

$$2^\sigma d_\sigma(u, v) = D(u, v)$$

がなりたつ．T_σ の 2 つのカラークラスを B_σ と W_σ とかく．$0 \in B_\sigma$ とする．すると $T_{\sigma-1}$ は T_σ の枝細分と自然にみなせる．従って

$$T_{\sigma-1} = (T_\sigma)^*, \quad B_{\sigma-1} = T_\sigma. \tag{3.3}$$

ターミナル $s \in S$ に対し，$f_{s,\sigma} : T_\sigma \to \overline{\mathbf{R}}$ を

$$f_{s,\sigma}(p) := I_s(p) - r(s) 2^\sigma d_\sigma(0, p) \quad (p \in T_\sigma)$$

と定義する．各枝 $ij \in E$ に対し，$g_{ij,\sigma} : \mathbf{Z} \to \mathbf{R}$ を

$$g_{ij,\sigma}(z) := c(ij)(2^\sigma z - a(ij))^+ \quad (z \in \mathbf{Z})$$

と定義する．$\omega_\sigma : T_\sigma{}^n \to \overline{\mathbf{R}}$ を ω の $T_\sigma{}^n$ への制限とする．それは

$$\omega_\sigma(p) = \sum_{s \in S} f_{s,\sigma}(p_s) + \sum_{ij \in E} g_{ij,\sigma}(d_\sigma(p_i, p_j)) \quad (p \in T_\sigma{}^n)$$

ともかける．各枝 ij に対し，偶な関数 $\bar{g}_{ij,\sigma} : \mathbf{Z} \to \mathbf{R}$ を 2.4 節のように定義する．すなわち z が奇数なら $\bar{g}_{ij,\sigma}(z) := (g_{ij,\sigma}(z-1)+g_{ij,\sigma}(z+1))/2$，$z$ が偶数なら $\bar{g}_{ij,\sigma}(z) := g_{ij,\sigma}(z)$．$\bar{\omega}_\sigma : T_\sigma{}^n \to \overline{\mathbf{R}}$ を

$$\bar{\omega}_\sigma(p) = \sum_{s \in S} f_{s,\sigma}(p_s) + \sum_{ij \in E} \bar{g}_{ij,\sigma}(d_\sigma(p_i, p_j)) \quad (p \in T_\sigma{}^n)$$

と定義する．これらの定義の仕方により，以下が成り立つ．

補題 **3.8.** (1) ω_σ は，$T_\sigma{}^n$ 上の 2 分離凸関数である．$\bar{\omega}_\sigma$ は，2 分離 L 凸関数である．

(2) $\bar{\omega}_\sigma$ は，$\omega_{\sigma+1}$ の L 凸緩和である．

(3) $\bar{\omega}_{-1}$ の最小解は (D) の最適解である．

我々のゴールは，交代的 L 凸関数 $\bar{\omega}_{-1}$ を最小化することである．

近接スケーリングアルゴリズム:

Step 0: $\sigma := L = \lceil \log nA \rceil$, $p^{\sigma+1} := (0, 0, \ldots, 0) \in B_\sigma{}^n$.

Step 1: $\bar{\omega}_\sigma$ の最小解 $p^\sigma \in T_\sigma{}^n$ を初期点 $p^{\sigma+1}$ から最急降下アルゴリズムで求める．

Step 2: もしも $\sigma = -1$ なら，$p = p^{-1}$ が (D) の最適解であり，step 3 へ進む．そうでなければ $\sigma \leftarrow \sigma - 1$ として step 1 に戻る．

Step 3: $\mathcal{D}_p$ を構成して，整数循環フロー ϕ を求める．すると f_ϕ が (N) の半整数最適マルチフローである．

近接性定理 2.11 と持続性定理 2.10 によって，step 1 における初期点 $p^{\sigma+1}$ と p^σ が距離が以下のようにバウンドされる．

補題 **3.9.** $d_\sigma(p^{\sigma+1}, p^\sigma) \leq 6n + 4$.

したがって，定理 2.6 によって, 最急降下アルゴリズムの反復回数は $6n+4$ で上から抑えられる．$\bar{\omega}_\sigma$ は $O(m)$ 項からなる 2 分離 L 凸関数で T_σ の最大次数はターミナルの個数 k なので，定理 2.15 によって，$O(n\mathrm{MF}(kn, km))$ 時間で p^σ が求まる．トータルで $O(nL\,\mathrm{MF}(kn, km))$ 時間である．枝費用がすべて正のときは，$L = \lceil \log nA \rceil$ で，もしも費用がゼロの枝を含むときは a を適切に摂動して摂動後には (CP) が満たされ，枝費用最大値を AC でおさえることができる．よって定理 3.4 が証明された．

謝辞

本研究は，JSPS 科研費 25280004, 26330023, 26280004 の助成を受けたものです．

References

[1] R. K. Ahuja, D. S. Hochbaum, and J. B. Orlin, A cut-based algorithm for the nonlinear dual of the minimum cost network flow problem, *Algorithmica* **39** (2004), 189–208.

[2] E. Anshelevich and A. Karagiozova, Terminal backup, 3D matching, and covering cubic graphs, *SIAM Journal on Computing* **40** (2011), 678–708.

[3] A. Bernáth, Y. Kobayashi, and T. Matsuoka, The generalized terminal backup problem, *SIAM Journal on Discrete Mathematics* **29** (2015), 1764–1782.

[4] B. V. Cherkasski, A solution of a problem of multicommodity flows in a network. *Ekonomika i Matematicheskie Metody* **13** (1977), 143–151 (in Russian).

[5] P. M. Dearing, R. L. Francis, and T. J. Lowe, Convex location problems on tree networks, *Operations Research* **24** (1976), 628–642.

[6] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, The complexity of multiterminal cuts, *SIAM Journal on Computing* **23** (1994), 864–894.

[7] P. Favati and F. Tardella, Convexity in nonlinear integer programming, *Ricerca Operativa* **53** (1990), 3–44.

[8] T. Fukunaga, Approximating the generalized terminal backup problem via half-integral multiflow relaxation, In: *32nd International Symposium on Theoretical Aspects of Computer Science (STACS'15)* (2015), pp. 316–328. `arXiv:1409.5561`.

[9] S. Fujishige, *Submodular Functions and Optimization, 2nd Edition*, Elsevier, Amsterdam, 2005.

[10] S. Fujishige, Bisubmodular polyhedra, simplicial divisions, and discrete convexity, *Discrete Optimization* **12** (2014) 115–120.

[11] S. Fujishige and K. Murota, Notes on L-/M-convex functions and the separation theorems, *Mathematical Programming, Series A* **88** (2000), 129–146.

[12] A. V. Goldberg and A. V. Karzanov, Scaling methods for finding a maximum free multiflow of minimum cost, *Mathematics of Operations Research* **22** (1997), 90–109.

[13] I. Gridchyn and V. Kolmogorov, Potts model, parametric maxflow and k-submodular functions, In: *Proceedings of IEEE International Conference on Computer Vision (ICCV'13)*, 2013, 2320–2327. `arXiv:1310.1771`.

[14] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.

[15] H. Hirai, Half-integrality of node-capacitated multiflows and tree-shaped facility locations on trees, *Mathematical Programming, Series A* **137** (2013), 503–530.

[16] H. Hirai, Discrete convexity and polynomial solvability in minimum 0-extension problems, *Mathematical Programming, Series A*, to appear (the extended abstract appeared SODA'13).

[17] H. Hirai, Discrete convexity for multiflows and 0-extensions, in: *Proceeding of 8th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications*, 2013, pp. 209–223.

[18] H. Hirai, L-extendable functions and a proximity scaling algorithm for minimum cost multiflow problem, *Discrete Optimization* **18** (2015), 1–37.

[19] H. Hirai, L-convexity on graph structures, in preparation.

[20] H. Hirai and Y. Iwamasa, On k-submodular relaxation, 2015, `arXiv:1504.07830`.

[21] D. S. Hochbaum, Solving integer programs over monotone inequalities in three variables: a framework for half integrality and good approximations, *European Journal of Operational Research* **140** (2002), 291–321.

[22] A. Huber and V. Kolmogorov, Towards minimizing k-submodular functions, in: *Proceedings of the 2nd International Symposium on Combinatorial Optimization (ISCO'12)*, LNCS 7422, Springer, Berlin, 2012, pp. 451–462.

[23] S. Iwata, L. Fleischer, and S. Fujishige, A combinatorial strongly polynomial algorithm for minimizing submodular functions, *Journal of the ACM* **48** (2001), 761–777.

[24] S. Iwata and M. Shigeno, Conjugate scaling algorithm for Fenchel-type duality in discrete convex optimization, *SIAM Journal on Optimization* **13** (2002), 204–211.

[25] Y. Iwata, M. Wahlström, and Y. Yoshida, Half-integrality, LP-branching and FPT Algorithms, `arXiv:1310.2841`, the conference version appeared in SODA'14.

[26] A. W. J. Kolen, *Tree Network and Planar Rectilinear Location Theory*, CWI Tract 25, Center for Mathematics and Computer Science, Amsterdam, 1986.

[27] A. V. Karzanov, A minimum cost maximum multiflow problem, in: *Combinatorial Methods for Flow Problems*, Institute for System Studies, Moscow, 1979, pp. 138–156 (Russian).

[28] A. V. Karzanov, Minimum cost multiflows in undirected networks, *Mathematical Programming, Series A* **66** (1994), 313–324.

[29] V. Kolmogorov, Generalized roof duality and bisubmodular functions, *Discrete Applied Mathematics* **160** (2012), 416–426.

[30] V. Kolmogorov, Submodularity on a tree: Unifying $\mathrm{L}^\natural$-convex and bisubmodular functions, in: *Proceedings of the 36th International Symposium on Mathematical Foundations of Computer Science (MFCS'11)*, LNCS 6907, Springer, Berlin, 2011, pp. 400–411

[31] V. Kolmogorov and A. Shioura, New algorithms for convex cost tension problem with application to computer vision, *Discrete Optimization* **6** (2009), 378–393.

[32] V. Kolmogorov, J. Thapper, and S. Živný, The power of linear programming for general-valued CSPs, *SIAM Journal on Computing* **44** (2015), 1–36.

[33] L. Lovász, On some connectivity properties of Eulerian graphs, *Acta Mathematica Academiae Scientiarum Hungaricae* **28** (1976), 129–138.

[34] K. Murota, Discrete convex analysis, *Mathematical Programming* **83** (1998), 313–371.

[35] K. Murota, *Discrete Convex Analysis*, SIAM, Philadelphia, 2003.

[36] K. Murota and A. Shioura, Exact bounds for steepest descent algorithms of L-convex function minimization, *Operations Research Letters* **42** (2014), 361–366.

[37] A. Schrijver, A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *Journal of Combinatorial Theory, Series B* **80** (2000), 346–355.

[38] 塩浦昭義, L 凸関数の最小化アルゴリズム：離散凸解析と諸分野との繋がり, 第 27 回 RAMP シンポジウム予稿, 2015.

[39] B. C. Tansel, R. L. Francis, and T. J. Lowe, Location on networks I, II, *Management Science* **29** (1983), 498–511.

[40] J. Thapper and S. Živný, The power of linear programming for valued CSPs, in: *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'12)*, 2012, pp. 669–678.

[41] D. Xu, E. Anshelevich, and M. Chiang, On survivable access network design: complexity and algorithms, in: *Proceedings of the 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'08)*, 2008, pp. 186–190.