グラフ上の離散凸関数とその応用

Discrete convex function on graph structure and its application

平井広志 *

東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology
The University of Tokyo

Abstract: In this paper, we explain basics in submodular optimization and discrete convex analysis, and its recent development in connection with valued constraint satisfaction problem (VCSP). We also explain discrete convex functions on graph structures with applications to minimum 0-extension problem and multicommodity flow.

1 はじめに

本稿では，劣モジュラ最適化･離散凸解析の基本事項の説明から始めて，低アリティ関数和最小化問題 (Valued CSP) との関わりと最近の展開を，ラベル割当問題を例に取りながら紹介する．関連する話題として，筆者が導入したグラフ上の離散凸関数についても紹介する．

$\mathbf{R}$ を実数全体の集合とし，$\bar{\mathbf{R}} := \mathbf{R} \cup \{\infty\}$ とする．$\mathbf{Z}$ を整数全体 $\{\ldots, -2, -1, 0, 1, 2, 3, \ldots\}$ の集合とする．

2 劣モジュラ関数とは

定義 1. 劣モジュラ関数とは，有限集合 V の部分集合に対して値をとる集合関数 $f : 2^V \to \bar{\mathbf{R}}$ であって，

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y) \quad (X, Y \subseteq V)$$

を満たすもののことである．

以下は，劣モジュラ関数が現れる代表的な離散最適化問題である．

問題 1 (最大フロー問題)**.** $G = (V \cup \{s, t\}, E)$ を無向グラフとして，枝に非負の容量が与えられている．入口 s から出口 t へ，できる限りたくさん「フロー」を流したい．

フローの上限を与えるものとしてカット関数というものを考える．$X \subseteq V$ に対して，$f(X)$ の値を $\{s\} \cup X$ から出る枝の容量の総和と定義する．このようにして得られる集合関数 f は，カット関数と呼ばれる．

*連絡先：東京大学大学院情報理工学系研究科
〒 113-8656 東京都文京区本郷 7-3-1
E-mail: hirai@mist.i.u-tokyo.ac.jp

補題 2. カット関数は劣モジュラである．

カット関数の値は，流せるフローの上界を与えることが分かる．実は，フローの最大流量とカット関数の最小値は一致する．

定理 3 (最大フロー最小カット定理)**.** フローの最大流量とカット関数の最小値は等しい．

特に，最大フロー値は劣モジュラ関数の最小値である．劣モジュラ関数は，他の離散最適化問題の文脈においても現れ，重要な役割を果たしてきた．特に，解きたい問題が劣モジュラ関数の最小値を求めることに帰着することがよくある．

上記の最大フロー問題には，最大フローを求める「高速な」アルゴリズムがあり，同時に最小カットを返してくれる．ここで「高速な」の意味を述べると，グラフのサイズ (頂点数 $n+$ 枝数 m) の多項式のステップ回数で最大フローと最小カットを返してくれるということである．このように，入力のサイズの多項式のステップ回数で正しく動作するアルゴリズムを多項式時間アルゴリズムという．問題 P に対して，P を解く多項式時間アルゴリズムがあるとき，P は多項式時間で解ける，または，多項式時間可解であるという．

定理 4 (Grötschel-Lovász-Schrijver 81)**.** 劣モジュラ関数の最小化は，オラクルモデルの下で，多項式時間で解ける．

オラクルモデルとは，「X を与えると $f(X)$ を返してくれるサブルーチン（オラクル）が入力として与えられる．その呼び出しにかかる手間は 1 ステップとみる」という問題設定である．大域最小解が台集合 V のサイズの多項式回のサブルーチン呼び出しで求まるということである．

この定理の背景には，劣モジュラ関数を「離散的な凸関数と見る」という考えがある．それを述べる．そのまえに集合関数を $0,1$ ベクトル上の関数と同一視するやり方を述べる．台集合を $V=\{1,2,\ldots,n\}$ として，$X\subseteq V$ に対して，$\mathbf{R}^n$ 中の $0,1$ ベクトル $x=(x_1,x_2,\ldots,x_n)$ であって，$i\in X$ ならば $x_i=1$，$i\notin X$ なら $x_i=0$ となるものを対応させる．逆に $0,1$ ベクトル x に，$x_i=1$ となっているインデックス i からなる部分集合 X を対応させる．すると集合関数 $f:2^V\to\bar{\mathbf{R}}$ と $0,1$ 格子上の関数 $f:\{0,1\}^n\to\bar{\mathbf{R}}$ は同一視される．改めて，劣モジュラ関数の定義をする．

定義 5. 劣モジュラ関数とは $\{0,1\}^n$ 上の関数 $f:\{0,1\}^n\to\bar{\mathbf{R}}$ であって

$$f(x)+f(y)\geq f(x\wedge y)+f(x\vee y)\quad(x,y\in\{0,1\}^n)$$

を満たすものである．

ここで $\wedge,\vee$ は $\{0,1\}^n$ 上の 2 項演算で $(x\wedge y)_i:=\min\{x_i,y_i\}$, $(x\vee y)_i:=\max\{x_i,y_i\}$ と定義される．

劣モジュラ関数 $f:\{0,1\}^n\to\bar{\mathbf{R}}$ は，Lovász 拡張と呼ばれる線形補間によって，凸関数 $\hat{f}:[0,1]^n\to\bar{\mathbf{R}}$ に拡張される．そして，この凸関数 $\hat{f}$ の各点の値と劣勾配は，高速に (n 回の関数呼び出しで) 計算することができ，楕円体法と呼ばれる手法によって多項式時間で $\hat{f}$ の $[0,1]^n$ 上の最小化が実現され，結果として，f の $\{0,1\}^n$ 上の最小化が実現できる．

楕円体法は多項式時間アルゴリズムであるものの非常に遅く，実用性からは，程遠いものであった．楕円体法を経由しない「組合せ的な」多項式時間劣モジュラ関数最小化アルゴリズムは，1999 年に岩田-Fleischer-藤重と Schrijver により独立同時に提案された．また，劣モジュラ関数を離散的な凸関数と見るという考え方は，室田による離散凸解析の提唱につながった．このあたりは，教科書 [3, 20, 21, 22, 23] を参考にされたい．本稿では，これらの教科書とは少し異なる立場からの最近の展開を述べる．

3　画像処理への応用：グラフカット

その導入として，次のような画像処理の問題に対するグラフカットと呼ばれる最大フロー最小カットアルゴリズムを用いた手法 [13] について述べる．

問題 2. 画像がノイズが発生する回線を通って送られてきたとする．この画像からノイズを取り除きたい．

画像は白黒画像としよう．この問題を次のように離散最適化問題へ定式化する．まず画像のピクセルの集合を $V=\{1,2,\ldots,n\}$ と番号を付けて，$0,1$ ベクトル $x=(x_1,x_2,\ldots,x_n)$ をピクセル i が白色のとき $x_i=0$，黒色のとき $x_i=1$ と対応させる．今，送られてきた画像に対応する $0,1$ ベクトルを $y=(y_1,y_2,\ldots,y_n)$ としよう．これからノイズが除かれた画像に対応する $0,1$ ベクトル $x=(x_1,x_2,\ldots,x_n)$ を推定したい．そこで，次の最適化問題の (大域的) 最小解 x を推定画像とするのである．

$$\begin{array}{lll}\text{P1:} & \text{Min.} & \displaystyle\sum_{1\leq i\leq n}b_i|y_i-x_i|+\sum_{1\leq i<j\leq n}c_{ij}|x_i-x_j| \\ & \text{s.t.} & x=(x_1,x_2,\ldots,x_n)\in\{0,1\}^n.\end{array}$$

ここで b_i,c_{ij} は非負のパラメータである．目的関数の第一項は，x_i と y_i が異なるとペナルティがかかる関数の和で，b_i は，i の位置の y_i の信頼度を表していて，信頼度が高い場合は b_i は大きく設定する．第 2 項は，x_i と x_j が異なるとペナルティがかかる関数の和で，例えば，隣り合うピクセル間の色は似ていると期待されるので，i が j が隣り合っている場合は c_{ij} を大きくする．そして総ペナルティを最小化するのである．

適切なパラメータチューニングの下で，最小解 x に対応する画像には，白ばかりのところにポツンと黒のようなノイズが取り除かれ，ノイズに起因する黒白のギザギザの境界が滑らかになる (と期待される) のである．実験結果についても [13] を参照のこと．

実は，問題 P1 の目的関数は，劣モジュラであり，多項式時間で大域最小解が求まる．さらに，実は，以下のようにカット関数として表すことができるので，最大フロー最小カットアルゴリズム（例えば Goldberg-Tarjan のアルゴリズム）によって実際的な意味で高速に最小解が求まる．

目的関数をカット関数として表現するためのネットワークを以下に構成する．頂点集合はピクセル集合 V と黒色に対応する点 s，白色に対応する点 t からなるものとし，枝はすべての頂点間にあるものとする．ピクセル i と j の間の枝 ij の容量を c_{ij} と設定し，ピクセル i と黒色点 s の間の枝容量を $y_i=1$(y_i が黒) なら 0，そうでないなら b_i，ピクセル i と白色点 t の容量は，その逆，$y_i=0$(y_i が白) なら 0，そうでないなら b_i と設定する．すると，画像に対応する $0,1$ ベクトル x に対して，$x_i=1$ となる頂点 i からなる集合 $X\subseteq V$ に対するカット関数の値は，確かに目的関数の x での値に等しくなる．なので，このネットワークに対する最小カットを求めることによって，問題 P1 の大域最小解が得られる．このように，適切なネットワークを構成して，最小カットを求めることで問題が解けるとき,「グラフカットで解ける」などということにする．

より一般的な問題設定を考えたくなる．目的関数が次のような形をしている場合はどうか．

$$\sum_i h_i(x_i)+\sum_{i,j}h_{ij}(x_i,x_j)+\sum_{i,j,k}h_{ijk}(x_i,x_j,x_k)+\cdots$$

$h_{ijk\cdots}$ は，ピクセル $i,j,k,\cdots$ の (高次の) 相互ペナルティを表現する関数である．引数の個数が k 個以下の関数の和でかける関数を，k 次の関数と呼ぶことにする．P1 において，$|y_i - x_i|$ を任意の 1 次関数 $g(x_i)$，$|x_i - x_j|$ を一般の 2 次劣モジュラ関数 $h(x_i, x_j)$ にした場合も，実は，対応するネットワークが存在して，グラフカットで解ける．さらに，3 次の劣モジュラ関数もグラフカットで解けることが知られている (例えば [18])．ちなみに，4 次の劣モジュラ関数には，グラフで表現できないものが存在することが示されている [1]．

4　劣モジュラ関数の仲間たち

4.1　L$^\natural$ 凸関数

劣モジュラ関数の $\{0,1\}^n$ から整数格子 $\mathbf{Z}^n$ への拡張といえるものが L$^\natural$ 凸関数である [3, 20, 21, 23]．

定義 6. L$^\natural$ 凸関数とは $\mathbf{Z}^n$ 上の関数 $g : \mathbf{Z}^n \to \bar{\mathbf{R}}$ であって，次の不等式

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

を満たすものである．

ここで $\lfloor\cdot\rfloor$($\lceil\cdot\rceil$) は各成分の小数部を切り下げる (上げる) 演算である．次の性質が，特に重要である．

性質 7. (1) $x \in \mathbf{Z}^n$ が L$^\natural$ 凸関数 g の大域最小解かどうかは，劣モジュラ関数最小化によって判定でき，大域最小解でないときは，$g(x') < g(x)$ となる x' が見つかる．

(2) 1 次元凸関数 $h : \mathbf{Z} \to \mathbf{R}$ に対して，$g(x) := h(x_i)$ や $g(x) := h(x_i - x_j)$ は L$^\natural$ 凸関数である．また，L$^\natural$ 凸関数の非負結合も L$^\natural$ 凸関数である．

性質 (1) より，劣モジュラ関数最小化を繰り返すことで大域最小解を得る降下アルゴリズムが構成できる．

再び，画像処理の問題 2 に戻る．いま画像がグレースケールであるとしよう．各ピクセルの色の濃さが K 段階に分かれているとすると画像は成分が 1 から K までの値をとる n 次元の整数ベクトル x と表現できる．問題 P1 において変数 x の取り得る範囲を $x \in \{1, 2, \ldots, K\}^n$ にした問題を P2 とする．性質 (2) より，P2 は，L$^\natural$ 凸関数の最小化となることが分かる．目的関数は，2 次なので，最適性・降下方向のチェックは 2 次劣モジュラ関数最小化となり，グラフカットで解ける．そして，その繰り返しで大域最小解が求まる．論文 [16] では，繰り返しの回数は $O(K)$ に出来ることが示され，画像処理への応用も述べられている．

なお，L$^\natural$ 凸関数にも，Lovász 拡張の類似があって，$\mathbf{R}^n$ 上の凸関数へと拡張することができる．

4.2　双劣モジュラ関数

定義 8. 双劣モジュラ関数とは $\{-1, 0, 1\}^n$ 上の関数 $f : \{-1, 0, 1\}^n \to \bar{\mathbf{R}}$ であって，次の不等式

$$f(x) + f(y) \geq f(x \wedge y) + f(x \sqcup y) \quad (x, y \in \{-1, 0, 1\}^n)$$

を満たすものである．

2 項演算 $\wedge, \sqcup$ について説明する．半順序を $-1 \succ 0 \prec 1$ と定義する (-1 と 1 は比較不能)．$(x \wedge y)_i$ は，x_i, y_i が比較可能なときは (この半順序で)$\min\{x_i, y_i\}$ と定義し，そうでなければ 0 と定義する．$(x \sqcup y)_i$ は，min を max に変えて定義する．

双劣モジュラ関数にも，Lovász 拡張の類似があり，$[-1, 1]^n$ 上の凸関数へ拡張され，楕円体法によってオラクルモデルの下での多項式時間で最小化できる．組合せ的な多項式時間アルゴリズムは，[4, 19] で提案されている．

双劣モジュラ関数の代表的な例として，

$$\sum_i a_i x_i + b_i |x_i| + \sum_{i,j} c_{ij} |x_i - x_j| + \sum_{i,j} d_{ij} |x_i + x_j|$$

がある ($b_i, c_{ij}, d_{ij} \geq 0$)．この関数は，グラフカットで最小化できる．実は，任意の 2 次双劣モジュラ関数はグラフカットで最小化できる (石井勇太，東京大学修士論文，2014 年)．

4.3　k-劣モジュラ関数

k-劣モジュラ関数は Huber-Kolmogorov [11] によって導入された双劣モジュラ関数の拡張である．いま，S_k を $k+1$ 個の要素からなる集合で，0 という特別な要素を含むものとする．S_k 上の半順序を，$0 \prec a$ ($a \in S_k \setminus \{0\}$)，それ以外は比較不能と定義する．そして ${S_k}^n$ 上の 2 項演算 $\wedge$，$\sqcup$ を $(x_i \wedge y_i)$ は，x_i, y_i が比較可能なときは (この半順序で)$\min\{x_i, y_i\}$ と定義し，そうでなければ 0 と定義する．$(x \sqcup y)_i$ は，min を max に変えて定義する．

定義 9. k-劣モジュラ関数とは，${S_k}^n$ 上の関数 $f : {S_k}^n \to \bar{\mathbf{R}}$ がであって，次の不等式

$$f(x) + f(y) \geq f(x \wedge y) + f(x \sqcup y) \quad (x, y \in {S_k}^n)$$

を満たすものである．

$k = 1$ のときは，劣モジュラ関数，$k = 2$ のときは，双劣モジュラ関数に一致する．

k-劣モジュラ関数は難しい問題 (NP 困難な問題) を緩和すると現れることがある．再び，画像処理の問題 2 を取り上げる．今度は，画像が赤 (R)，緑 (G)，青 (B)

の 3 色からなる場合を考える．画像は $\{R, G, B\}^n$ の点と対応する．今までの類推から，次のように最適化問題を定式化してみる．

$$\begin{array}{lll} \text{P3:} & \text{Min.} & \sum_{1\le i\le n} b_i\delta(y_i, x_i) + \sum_{1\le i<j\le n} c_{ij}\delta(x_i, x_j) \\ & \text{s.t.} & x = (x_1, x_2, \ldots, x_n) \in \{R, G, B\}^n \end{array}$$

δ は，2 つの引数が同じとき 0 をとり，そうでないとき 1 をとる関数である．

この問題は，多分割カット問題として知られる NP 困難な問題と等価である．この問題を緩和するために新しい色 0 を付け加え，$\delta(0, R) = \delta(0, B) = \delta(0, R) := 1/2$, $\delta(0, 0) = 0$ と δ を拡張して次の緩和問題を考える．

$$\begin{array}{lll} \text{P3}': & \text{Min.} & \sum_{1\le i\le n} b_i\delta(y_i, x_i) + \sum_{1\le i<j\le n} c_{ij}\delta(x_i, x_j) \\ & \text{s.t.} & x = (x_1, x_2, \ldots, x_n) \in \{0, R, G, B\}^n \end{array}$$

$\{0, R, G, B\}$ を S_3 とみると，P3$'$ の目的関数は，3-劣モジュラ関数になる．この問題 P3$'$ は，グラフカットで解ける．P1 のときと同様にネットワークを構成する．ただし s, t のかわりに 3 色 R, G, B に対応する r, g, b を (同様にして) 取り付ける．そして，このネットワークにおいて，r を含み g, b を含まない極小な最小カット X_r，g を含み r, b を含まない極小な最小カット X_g，b を含み r, g を含まない極小な最小カット X_b を求める．すると X_r, X_g, X_b は，互いに交わりがない．X_r, X_g, X_b に入っている点 i について，それぞれ $x_i := R, G, B$ と割り当てる．その他は $x_i := 0$ と割り当てる．得られる x は問題 P3$'$ の大域最小解である．もしも 0 を割り当てられた x_i がなければ，元の問題 P3 の大域最小解でもある．0 が割り当られた x_i はすべて同じ色 (例えば R) に割り当てると得られる解の目的関数値は，元の問題 P3 の最適値の 2 倍以内にある．これは多分割カットに対する古典的な 2 近似アルゴリズムである [25]．このように新しい点を追加することで k-劣モジュラ関数に拡張する手法は，劣モジュラ緩和と呼ばれ，その有用性が認識され始めている [6, 12, 26]．

k-劣モジュラ関数は，オラクルモデルの下で，多項式時間で最小化できるかどうかは現時点で未解決であり，理論的な関心事である [5, 11]．一方，次章に述べる Valued CSP という枠組みの下では，多項式時間で最小化が可能である．

5　低アリティ関数和最小化問題

今まで扱った最適化問題は，目的関数が引数が少ない関数のいくつかの和で書かれているという特徴があった．このような最適化問題を統一的に扱う Valued CSP(Valued Constraint Satisfaction Problem) という枠組み [27] を紹介する．

D を有限集合とする．D 上のコスト関数 f とは，ある k に対して $f : D^k \to \bar{\mathbf{R}}$ となるもののことを指す．この f の引数の個数 $k = k_f$ を f のアリティと呼ぶ．今，変数の集合を $x_1, x_2, \ldots, x_n$ とする．コスト関数 f と引数の割当 $\sigma : \{1, 2, \ldots, k_f\} \to \{1, 2, \ldots, n\}$ の組 (f, σ) を制約と呼ぶ．D 上の Valued CSP とは，制約の (有限) 集合 $\mathcal{C}$ を入力とする以下の最適化問題である．

$$\begin{array}{lll} \text{VCSP:} & \text{Min.} & \sum_{(f,\sigma)\in\mathcal{C}} f(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(k_f)}) \\ & \text{s.t.} & x = (x_1, x_2, \ldots, x_n) \in D^n. \end{array}$$

この問題設定では，$\mathcal{C}$ のコスト関数 f のアリティk_f は 2 とか 3 とか小さいものを想定していて，各コスト関数 f は $\underbrace{D \times D \times \cdots \times D}_{k_f}$ のテーブルとして保持していると考える．従って，入力のサイズは $|\mathcal{C}||D|^K$ と n である．K はアリティの最大値とする．オラクルモデルとの相違に注意する．ちなみに，各 f の値が 0 か ∞ をとる場合が CSP である．

Valued CSP は，以下のように整数計画問題として定式化できる．

$$\begin{array}{lll} \text{IP:} & \text{Min.} & \sum_{(f,\sigma)\in\mathcal{C}} \sum_{p\in D^{k_f}} f(p)\lambda^p_{f,\sigma} \\ & \text{s.t.} & \sum_{p\in D^{k_f}: p_j = u} \lambda^p_{f,\sigma} = \mu^u_{\sigma(j)} \\ & & \qquad ((f,\sigma)\in\mathcal{C}, 1\le j\le k_f, u\in D), \\ & & \sum_{u\in D} \mu^u_i = 1 \quad (1\le i\le n), \\ & & \lambda^p_{f,\sigma} \in \{0,1\} \quad ((f,\sigma)\in\mathcal{C},\ p\in D^{k_f}), \\ & & \mu^u_i \in \{0,1\} \quad (1\le i\le n,\ u\in D). \end{array}$$

これが，VCSP と同値であることは，よく眺めると分かる (ヒント：$\mu^u_i = 1$ なら x_i を u に割り当てる)．重要な点は，IP は入力の多項式のサイズであるということである (変数の個数が $O(|\mathcal{C}||D|^K + n|D|)$，制約式の個数が $O(|\mathcal{C}||D|K + n)$)．

Basic-LP とは，IP の変数の $\{0, 1\}$ 制約を $[0, 1]$ に緩和して得られる線形計画問題のことある．これは LP ソルバで解ける．最近になって，この Basic-LP が厳密になる，すなわち，IP と最適値が等しくなるための驚くべき十分条件が得られた．その説明をしよう．

定義 10. D 上のコスト関数集合 $\mathcal{L}$ に対する分数的ポリモルフィズムとは，D 上の 2 項演算の形式的凸結合

$\sum_j \alpha_j \odot_j$ で，$\mathcal{L}$ 内のすべてのコスト関数 f に対して

$$\frac{1}{2}\left(f(x)+f(y)\right) \geq \sum_j \alpha_j f(x \odot_j y) \quad (x, y \in D^{k_f})$$

をみたすものである．

ここで D 上の 2 項演算 $\odot$ は D^k 上に $(x \odot y)_i := x_i \odot y_i$ と作用するものとする．例えば，$\frac{1}{2}\wedge + \frac{1}{2}\vee$ は劣モジュラ関数に対する分数的ポリモルフィズムである．$\mathrm{L}^\natural$ 凸関数や双劣モジュラ，k-劣モジュラ関数も分数的ポリモルフィズムで定義されていると解釈できる．

定理 11 (Thapper-Živný 2012 [24]; [17])**.** 制約集合 $\mathcal{C}$ に現れるコスト関数の集合に対して半束演算を項としてもつ分数的ポリモルフィズムが存在すると，Basic-LP は厳密となり，VCSP は多項式時間で解ける．

ここで半束演算とは，2 項演算 $\odot$ であって対称律 $x \odot y = y \odot x$，べき等律 $x \odot x = x$，結合律 $(x \odot y) \odot z = x \odot (y \odot z)$ を満たすものである．例えば，$\wedge$ や $\vee$ は，半束演算である．この結果によって，k-劣モジュラな Valued CSP は Basic-LP で解けることが明らかになった．

系 12. k-劣モジュラなコスト関数を入力とする Valued CSP は多項式時間で解ける．

他にも，一般の束上の劣モジュラ関数に対する Valued CSP なども Basic-LP で解ける．

6　グラフ上の離散凸関数

ここでは，最近，筆者が [9, 10] で導入したグラフ構造上の L 凸関数について紹介する．定義には，かなり準備がいるので，その動機と重要な性質についてのみ述べる．再び，画像の問題 2 を考える．画像は，l 色からなるカラー画像である状況を考える．今度は，色の間に距離が与えられている．その距離関数は，あるグラフ Γ のグラフ距離 d_Γ でかけるとしよう．そうして，我々は，グラフ Γ 上の最小ゼロ拡張問題 [14] と呼ばれる問題に辿り着く．

$$\begin{array}{lll} \text{0-EXT:} & \text{Min.} & \displaystyle\sum_{i,k} b_{ik} d_\Gamma(v_k, x_i) + \sum_{i,j} c_{ij} d_\Gamma(x_j, x_i) \\ & \text{s.t.} & x = (x_1, x_2, \ldots, x_n) \in {V_\Gamma}^n. \end{array}$$

ここで $V_\Gamma = \{v_1, v_2, \ldots, v_l\}$ とする．これも Valued CSP の一種で，グラフ Γ が 1 本の枝からなるときは白黒問題 P1(最小カット問題) に対応し，Γ が長さ K のパスのときは，グレースケール問題 P2 に，Γ が 3 角形のときは，3 色 (RGB) 問題 P3(多分割カット問題) であり，Γ が 3 つの葉からなるスターのときが緩和問題 P3$'$ である．最小ゼロ拡張問題は，Γ 上に施設を配置する施設配置問題の変種という見方も出来る．Γ が木のときは，グラフカット (の繰り返し) で解けることが古くから知られている [15]．画像処理への応用が [2, 6] にある．

最小ゼロ拡張問題は，Γ によってＮＰ困難になったり多項式時間可解になったりする．Karzanov は，どのようなグラフなら多項式時間可解になるかという問題を提起した．この問題は次のように解決された．

定理 13 (Karzanov 1998 [14])**.** Γ が向き付け可能モジュラグラフでないなら，Γ 上の最小ゼロ拡張問題は NP 困難である．

定理 14 (Hirai 2012 [9])**.** Γ が向き付け可能モジュラグラフなら，Γ 上の最小ゼロ拡張問題は多項式時間で解ける．

モジュラグラフとは，任意の 3 点 v_1, v_2, v_3 に対して，$d_\Gamma(v_i, v_j) = d_\Gamma(v_i, m) + d_\Gamma(m, v_j)$ $(1 \leq i < j \leq 3)$ を満たす点 m(メディアンと呼ばれる) が存在するようなグラフであり，(モジュラグラフの) 向き付けとは，枝の向き付けであって，任意の 4 サイクル uv, vz, zw, wu に対して，$u \to v$ なら $w \to z$ となるもののことである．向き付け可能なモジュラグラフは，モジュラ束 (のハッセ図) の貼り合わせになっている．例としては，パス，木，グリッドグラフ，そしてそれらの直積としてかけるようなグラフがある．特に直積に関して閉じていることに注意する．

定理 14 の証明のために，[9] では，新しい離散凸関数である「モジュラ半束上の劣モジュラ関数」と「向き付けられたモジュラグラフ (の頂点集合) 上の L 凸関数」を導入している．モジュラ半束とは，モジュラ束の半束への一般化であり，ハッセ図がモジュラグラフになるものである．定理 14 は，次の性質の系として導かれる．

性質 15.

(1) モジュラ半束上の劣モジュラ関数に対する Valued CSP は多項式時間で解ける．
(定理 11 の応用)

(2) 点 x が L 凸関数 g の大域最小解かどうかは，あるモジュラ半束上の劣モジュラ関数最小化によって判定できる．大域最小解でないときは，$g(x') < g(x)$ となる x' が見つかる．

(3) (同じ Γ 上の)L 凸関数の非負結合も L 凸関数である．

(4) 向き付け可能なモジュラグラフ Γ に対して，距離関数 d_Γ は $\Gamma \times \Gamma$ 上の L 凸関数であり，Γ 上の最小ゼロ拡張問題は，$\Gamma \times \Gamma \times \cdots \times \Gamma$ 上の L 凸関数最小化問題になる．

整数格子 $\mathbf{Z}^n$ を (有向) パス P の n 直積 P^n と同一視したとき，P^n 上の L 凸関数は $\mathrm{L}^\natural$ 凸関数と一致する．また，k-劣モジュラ関数は，モジュラ半束 ${S_k}^n$ の劣モジュラ関数とも，k 個の葉をもつスターの n 直積上の L 凸関数とも見なせる．

最後に，多品種フローとの関連を述べる．多品種フローは，複数の種類のフローが流れる状況をモデリングしたもので，通常のネットワークフローの拡張である．多品種フローにも，最大フロー最小カット定理の類似の定理がいくつか知られており，双対問題は，最小ゼロ拡張型の問題となる [7, 8]．最大フロー問題の双対は最小カット問題であり，それは劣モジュラ関数最小化であった．このアナロジーが部分的に成立する．

定理 16 ([10])**.** 最大多品種フロー問題が離散的な双対問題をもつならば，それは，ある向き付けられたモジュラグラフ上の L 凸関数最小化となる．

3 色 (RGB) 問題 P3 の緩和 P3′ を解くために構成したネットワークを思い出す．r, g を結ぶフロー，g, b を結ぶフロー，b, r を結ぶフロー，の 3 種類のフローからなる多品種フローを考え，その総流量の最大化を考える．実は，最大流量は，P3′ の最小値に一致する．これは，多品種フロー理論で Lovász-Cherkassky の定理として知られているものである．P3′ は k-劣モジュラ関数最小化であった．これが定理 16 の一例である．

謝辞

発表の機会を与えてくださった大阪大学の河原吉伸准教授に感謝します．詳細なコメントをくださった東京大学の室田一雄教授に感謝します．本研究は，科研費（23740068）と最先端研究開発支援プログラム（FIRST 合原最先端数理モデルプロジェクト）により助成されたものである．

参考文献

[1] D. Cohen, P. Jeavons, and S. Živný, The expressive power of binary submodular functions, *Discrete Appl. Math.* **157**, (2009), 3347-3358.

[2] P. Felzenszwalb, G. Pap, E. Tardos, and R. Zabih, Globally optimal pixel labeling algorithms for tree metrics. CVPR'10, 3153–3160.

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

[4] S. Fujishige and S. Iwata, Bisubmodular function minimization. *SIAM J. Discrete Math.* **19**, (2006),1065–1073.

[5] S. Fujishige and S. Tanigawa: A min-max theorem for transversal submodular functions and its implications, RIMS-1790, 2013.

[6] I. Gridchyn and V. Kolmogorov, Potts model, parametric maxflow and k-submodular functions, ICCV'13, `arXiv:1310.1771`.

[7] 平井広志，多品種流の話，数学入門公開講座，京都大学数理解析研究所，(2009), 57–82.

[8] 平井広志，多品種フロー理論，第 23 回 RAMP シンポジウム論文集, (2011), 17–40.

[9] H. Hirai, Discrete convexity and polynomial solvability in minimum 0-extension problems, *SODA'13*, pp.1770-1788.

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

[11] A. Huber and V. Kolmogorov, Towards minimizing k-submodular functions, ISCO'12, 451–462.

[12] Y. Iwata, K. Oka, and Y. Yoshida, Linear-time FPT algorithm via network flow, SODA'14, `arXiv:1307.4927v1`.

[13] 石川博，グラフカット，コンピュータビジョン最先端ガイド 1, 2008, 39–74.

[14] A. V. Karzanov, Minimum 0-extensions of graph metrics, *Europ. J. Combin.* **19** (1998), 71–101.

[15] A. J. W. Kolen, *Tree Network and Planar Rectilinear Location Theory*, CWI Tracts. CWI, 1986.

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

[17] V. Kolmogorov, J. Thapper and S. Živný, The power of linear programming for general-valued CSPs, 2013, `arXiv:1311.4219`.

[18] V. Kolmogorov and R. Zabih, What energy functions can be minimized via graph cuts ? PAMI **26** (2004), 147–159.

[19] S. T. McCormick and S. Fujishige, Strongly polynomial and fully combinatorial algorithms for bisubmodular function minimization. *Math. Program.* **122** (2010), 87–120.

[20] 室田一雄，離散凸解析，共立出版，東京，2001.

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

[22] 室田一雄, 離散凸解析の考えかた, 共立出版, 東京, 2007.

[23] 室田一雄，塩浦昭義，離散凸解析と最適化アルゴリズム，朝倉書店，東京，2013.

[24] J. Thapper and S. Živný, The power of linear programming for valued CSPs, FOCS'12, 669-678.

[25] V. V. Vazirani, *Appoximation Algorithms*, Springer, 2001 (浅野孝夫訳：近似アルゴリズム, シュプリンガー, 東京，2002)

[26] M. Wahlström, Half-integrality, LP-branching and FPT Algorithms, SODA'14, `arXiv:1310.2841v1`.

[27] S. Živný, *The Complexity of Valued Constraint Satisfaction Problems,* Springer, Heidelberg, 2012.