離散凸最適化の
アルゴリズムとソフトウェアの研究

土村 展之

概要

本研究では、離散凸解析の分野に現れる、離散凸関数の最小化問題を取り上げ、その効率的な解法を考察する。関数の最小化問題とは、関数と制約条件が与えられたときに、関数値を最も小さくする変数の値を求める問題である。最小化の対象となる関数は、連続変数の関数だけでなく、離散変数の関数の場合もある。特に離散変数の関数については、ある望ましい “凸性” を持つクラス「離散凸関数」に対して、既に変数の次元数に関する多項式時間アルゴリズムが考案されている。本研究は、この離散凸関数の最小化問題に対して、より効率的な解法を提案する。その基本的なアイデアは、与えられた離散凸関数が連続変数上に拡張できることを前提として、まず連続変数上の最小化問題を解き、次にその解をもとにして離散変数上の最小化問題を解く、というものである。この手法の効率性は、連続変数上の解と離散変数上の解の距離に左右されるが、本研究では、これら 2 つの解が近くにあることを示し、提案手法が高速であることを理論的に明らかにした。また実際に計算実験を行い、現実的な規模の問題で提案手法が従来手法よりも約 10 倍高速に動作することを確かめた。そして、提案手法を含めた離散凸関数最小化アルゴリズムを実装して公開し、関連する研究の利便性を高めた。

目次

第 1 章

序論

1.1 問題設定と研究の成果

非線形最小化問題の分野では、大規模な多変数凸関数の最小化問題の解が数値的に求められることからわかるように、凸性が望ましい性質であると考えられている。離散変数の関数についても同様の性質が模索されてきた。1980 年代には劣モジュラ性と凸性の関係が明らかになり、1990 年代後半には L 凸と M 凸という 2 種類の離散凸性が提唱され、これにより離散凸解析の分野が確立した [62]。

L 凸関数の概念は、劣モジュラ集合関数の Lovász 拡張 (Lovász extension) [46] を一般化したものである。一方、M 凸関数の概念は、Dress–Wenzel による付値マトロイド (valuated matroid) [12] を一般化したものとして生まれた。L 凸関数と M 凸関数は、このようにそれぞれ異なる概念の自然な拡張であるが、Legendre 変換によって互いに移り合うという共役性を持ち、離散凸解析の分野において対となって中心的な役割を果たす。L 凸関数と M 凸関数は、それぞれ離散分離定理や最大・最小定理など、凸関数としての望ましい性質を持つ。そして 2 つの離散凸性は、非線形組合せ最適化問題のあるクラスに対して有用な枠組みを与える。すなわち、局所最適性によって大域最適性が得られること、最適化問題が降下法や貪欲法のような局所的探索法によって効率的に求解できることを保証する。具体的には、離散凸関数最小化問題に対する最急降下法が擬多項式時間アルゴリズムであること、さらにスケーリング技法を用いることで多項式時間アルゴリズムに高速化できることが知られている [52, 62, 88, 91]。

離散凸性は、身近な場面でも目にすることができる。例えば、電気回路における、電流源をつないだときの消費電力を表す関数が M 凸関数に対応し、電圧源をつないだときの消費電力を表す関数が L 凸関数に対応する [59, 82]。在庫管理理論と離散凸関数とのかかわりは深く、在庫管理における様々な問題設定において、L 凸性と等価なマルチモジュラ性が重要な役割を果たす [3, 27]。他にも、離散凸性の応用は理論経済学の不可分財市場の分析 [78] や, 混合多項式行列によるシステム解析 [59] などでも見られる。

また、離散凸性を次のようにとらえることもできる。世の中の様々な場面に現れる離散最適化問題の中には、効率的に解くことのできる、良い性質をもつものがある。このような問題に共通する、良い性質の本質を取り出して抽象化したものが離散凸解析の離散凸性である。それ

と同時に離散凸性を利用した効率的な最適化アルゴリズムが開発されている。このことから、ある離散最適化問題が離散凸性をもつならば、汎用のアルゴリズムによって最適化できることがわかる。もちろん、問題固有の強い性質を利用すれば、さらに高速なアルゴリズムを開発できる余地はあるが、固有の性質に深く立ち入らずとも、統一的な性質だけで効率的に最適化できるのが離散凸解析の利点である。また、個々の問題の性質を利用した専用のアルゴリズムとして知られているものが、離散凸解析の観点から考察すると、離散凸関数最小化問題における汎用のアルゴリズムの特殊例として説明できる場合がある。例えば、最小木問題におけるクラスカル法は、M 凸関数最小化アルゴリズムの特殊例とみることができる [75]。同様に、最短路問題におけるダイクストラ法は、L 凸関数最小化アルゴリズムの特殊例とみることができる [76]。

このように、離散凸解析の分野においては、離散関数の幅広い問題クラスに共通して存在する望ましい性質を取り上げ、抽象化して美しい理論を構築することに成功していると言えるが、一方では、理論的に扱いやすい性質ばかりが注目されがちだという問題がある。離散凸解析の汎用アルゴリズムと関係の深いアルゴリズムは基礎的あるいは古典的なものが多く、このようなアルゴリズムの性質を統一的に説明できることは確かに興味深いが、離散凸解析の実用面での貢献は必ずしも大きくない。近似解法や発見的解法のようなアプローチをとって、実際的な問題を実用上高速に解こうとする研究は少ない傾向にあると言える。

そこで本論文では、離散凸解析の分野で、より現実的なアプローチを行うことを狙い、周辺分野でも用いられる連続緩和手法を、広い問題クラスに対して統一的に適用することを試みる。連続緩和手法は、例えば数理計画法の分野では，整数計画問題を解く場合に、線形計画問題に緩和した問題から最適解の上界あるいは下界を得るというような形で用いられ、一定の成果を上げている。しかしながら、この手法を一般の離散最適化問題に適用しようとすると、次のような困難が予想される。離散変数の問題と連続変数の問題の関係は、単純な関係にあるわけではなく、いつでも連続緩和が行えて緩和問題が得られるというわけではない。そして緩和問題が得られ、緩和問題の解（緩和解）が得られたとしても、元の離散最適化問題との関係性がいつも保たれるわけではなく、2 つの最適化問題の解が遠く離れる例もある。整数計画問題の場合には、求めたい最適解が緩和解からすぐに構成できるわけではなく、緩和解から求められる上下界値を利用して、次の手順である切除平面法や分枝限定法での探索を効率化するなどの工夫がなされている。このように、連続緩和手法は理論的に深い考察に基づいて開発されている [10, 16, 32, 40, 42, 43, 86]。

本論文では、離散凸解析の離散変数の関数と連続変数の関数の関係について論じ、離散最適化問題に連続緩和手法を適用する。対象の問題クラスとしては、離散凸解析の離散 L 凸/M 凸関数最小化問題とする。連続緩和手法の手順としては、連続変数に緩和した凸関数最小化問題を解き、得られた緩和解を整数に丸め、離散変数の降下法の初期解として用いることを考える。問題例によっては、この連続緩和手法が効率的に働くことは十分に予想されることではあるが、効率的に働く問題例の条件や緩和手法の具体的な手順を明らかにするために、次の 3 つの条件を検討する必要がある。第 1 の条件は、与えられた離散最適化問題から、緩和問題である連続最適化問題が生成できること、第 2 の条件は、生成した連続最適化問題が、元の問題に

比べて十分に効率的に解くことができること、第 3 の条件は、2 つの最適化問題の解の距離が十分に近いことである。一般の離散最適化問題ではこれらの 3 つの条件が満たされるわけではないが、本論文で対象とする離散凸解析の離散 L 凸/M 凸関数最小化問題については、次のようになる。まず第 1 の条件について、離散変数の関数の L 凸性と M 凸性の概念は、連続変数の関数にも拡張されており [70, 73]、緩和問題の存在が明らかにされている。ただし、その生成方法はそれほど明らかではない。次に第 2 の条件について、緩和問題である連続変数の L 凸/M 凸関数最小化問題は、通常の凸関数の最小化問題の特殊例にあたるので、連続最適化分野の成果により、大規模であっても現実的な時間で数値的な解を求めることができる [21, 93]。最後の第 3 の条件について、離散 L 凸/M 凸関数最小化問題の解と、その緩和問題の解の関係は、既存研究からは明らかではない。このような 3 つの条件の状況から、緩和解が元の問題の解に十分に近いことを示せれば、連続緩和手法の有用性を明らかにできることがわかる。

本論文の 1 つめの成果は、離散 L 凸/M 凸関数最小化問題の解とその緩和問題の解との距離に関する近接定理の証明を与えることである。この近接定理は、2 つの解の距離が十分に近いことを示すものであり、離散 L 凸/M 凸関数最小化問題に対するはじめての連続緩和の近接定理である。これより、提案する離散 L 凸/M 凸関数最小化問題に対する連続緩和手法のアルゴリズムが、実用上だけでなく理論的にも高速であることが明らかになる。そして、提案する連続緩和法を M 凸関数最小化問題の特殊例である資源配分問題に適用し、先行研究による連続緩和手法と提案手法との計算量の比較を行う。その結果、提案手法が既知の最良の結果と同じ計算量を達成し、また限定された問題クラスでは計算量を改善する場合があることを示す。

本論文のもう 1 つの成果は、離散凸解析の最適化ソルバを開発し、オープンソースソフトウェアとして公開したことである。最適化の他の分野に目を向けると、例えば線形計画問題に対しては、商用の ILOG CPLEX[*1], Gurobi Optimizer[*2], 非商用の GLPK (GNU Linear Programming Kit), lp_solve をはじめとする数々のソルバが開発されている。また、半正定値計画問題に対しても、SDPA や SeDuMi といったソルバが公開されている。このようなソフトウェアを用いることで、問題の性質や求解のアルゴリズムの詳細を理解していない人でも、実問題を効率的に解くことができる。離散凸解析の分野についても、理論的な成果が応用分野に浸透するためには、同様のソフトウェアが必要であると考えられるが、これまで公開されていなかった。そこで本論文では、離散凸最適化ソルバを開発して Web 上に公開した。このソルバは、C 言語によって記述されており、離散凸関数最小化アルゴリズムの複数の実装を含む。中でも、提案する連続緩和法の実装は、素朴なアルゴリズムに比べて約 10 倍の規模の問題まで解くことのできる、性能の高いものである。同時に、実際上の問題を扱ったデモンストレーションソフトウェアも公開し、Web 上から簡単に動作を観察することができる。配布するソフトウェアには、他の研究者の開発されたソフトウェアを含めているが、ライセンスには注意を払い、そのまま配布することはもちろん、利用者が改変再配布することもできるように調整した。これらにより、関連分野の研究者が離散凸解析関連の研究や応用事例研究が行う際

[*1] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/
[*2] http://www.octobersky.jp/products/gurobi/

や、実務家が離散凸関数の現れる実問題を解く際の助けとなる。

1.2　本論文の構成

本論文の構成は以下の通りである.

2 章では離散凸解析の基礎事項をまとめる．L 凸関数と M 凸関数の定義を示し、この 2 つの離散凸関数に対応する離散凸集合を説明し、離散凸関数の例を挙げる。そして、2 つの離散凸性を連続変数の関数に拡張して得られる性質を述べる。

3 章では，L 凸関数最小化問題を取り上げる。既存のアルゴリズムについて述べ、連続緩和手法を提案する。提案手法の効率性の裏付けとなる近接定理を証明し、計算実験を行い従来手法よりも高速であることを示す。

4 章では，M 凸関数最小化問題を取り上げる。3 章と同様に、既存のアルゴリズムについて述べ、連続緩和手法を提案する。近接定理の証明と計算実験による速度比較も同様に行い、従来手法より改善されていることを示す。そして、M 凸関数最小化問題の特殊例である資源配分問題について、問題固有の特性を利用しながら、提案する連続緩和手法を適用し、計算量を解析する。特定の問題クラスで先行研究の連続緩和手法とほぼ同じ計算量を達成し、またいくつかの問題クラスでは先行手法の計算量より改善されていることを述べる。

5 章においては、開発した離散凸最適化ソルバソフトウェアについて述べる。ソルバによって最小化することのできる離散凸関数の種類と、実装したアルゴリズムを説明する。また、最小化ルーチンを利用したデモンストレーションソフトウェアの公開や、在庫管理問題への応用についても述べる。

6 章では、結論と今後の課題を述べる.

第 2 章

離散凸解析の基礎

2.1 用語・記号

この節では、本論文で用いる記号や用語を説明する。

2.1.1 記号

まず記号の定義を行う。n を 2 以上の正の整数とし，$N = \{1, 2, \ldots, n\}$ とする．$\mathbb{R}$ は実数の集合を、$\mathbb{R}_+$ は非負の実数の集合を表す。同様に，$\mathbb{Z}$ は整数の集合を，$\mathbb{Z}_+$ は非負の整数の集合を表す。

ベクトル $\boldsymbol{x} \in \mathbb{R}^n$ の第 i 成分を $x(i)$ と表す。すなわち

$$\boldsymbol{x} = (x(1), x(2), \ldots, x(n))$$

である。あるいは、ベクトル $\boldsymbol{x}$ の第 i 成分を x_i と表す場合もある。すなわち

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$$

である。ベクトル $\boldsymbol{x}$ の ℓ_∞-ノルム $\|\boldsymbol{x}\|_\infty$ と ℓ_1-ノルム $\|\boldsymbol{x}\|_1$ を

$$\|\boldsymbol{x}\|_\infty = \max_{i \in N} |x(i)|,$$
$$\|\boldsymbol{x}\|_1 = \sum_{i \in N} |x(i)|$$

と定義する．ベクトル $\mathbf{1}$ とベクトル $\mathbf{0}$ を

$$\mathbf{1} = (1, 1, \ldots, 1) \in \mathbb{R}^n,$$
$$\mathbf{0} = (0, 0, \ldots, 0) \in \mathbb{R}^n$$

と定義する．

ベクトル $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ に対する等式や不等式は，成分ごとの等式や不等式を意味する．例えば，$\boldsymbol{x} \leq \boldsymbol{y}$ はすべての $i \in N$ に対して $x(i) \leq y(i)$ であることを表す．

ベクトル $\boldsymbol{x} \in \mathbb{R}^n$ と実数 $k \in \mathbb{R}$ に対して、$k\boldsymbol{x}$ は $\boldsymbol{x}$ を成分ごとに k 倍したベクトルを表し、$\frac{\boldsymbol{x}}{k}$ は $\boldsymbol{x}$ を成分ごとに $1/k$ 倍したベクトルを表す。したがって

$$k\boldsymbol{x} = (kx(1), kx(2), \ldots, kx(n)),$$
$$\frac{\boldsymbol{x}}{k} = \left(\frac{x(1)}{k}, \frac{x(2)}{k}, \ldots, \frac{x(n)}{k}\right)$$

である.

ベクトル $\boldsymbol{x} \in \mathbb{R}^n$ に対して、$\lceil \boldsymbol{x} \rceil \in \mathbb{Z}^n$ は $\boldsymbol{x}$ を成分ごとに切り上げによって整数に丸めたベクトルを表し、$\lfloor \boldsymbol{x} \rfloor \in \mathbb{Z}^n$ は $\boldsymbol{x}$ を成分ごとに切り捨てによって整数に丸めたベクトルを表す。したがって

$$\lceil \boldsymbol{x} \rceil = (\lceil x(1) \rceil, \lceil x(2) \rceil, \ldots, \lceil x(n) \rceil),$$
$$\lfloor \boldsymbol{x} \rfloor = (\lfloor x(1) \rfloor, \lfloor x(2) \rfloor, \ldots, \lfloor x(n) \rfloor)$$

である.

ベクトル $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ に対して, $\boldsymbol{x} \vee \boldsymbol{y}$ は成分ごとに $\boldsymbol{x}$ と $\boldsymbol{y}$ の最大値をとって得られるベクトルを表し、$\boldsymbol{x} \wedge \boldsymbol{y}$ は成分ごとに $\boldsymbol{x}$ と $\boldsymbol{y}$ の最小値をとって得られるベクトルを表す。したがって

$$\boldsymbol{x} \vee \boldsymbol{y} = (\max\{x(1), y(1)\}, \max\{x(2), y(2)\}, \ldots, \max\{x(n), y(n)\}),$$
$$\boldsymbol{x} \wedge \boldsymbol{y} = (\min\{x(1), y(1)\}, \min\{x(2), y(2)\}, \ldots, \min\{x(n), y(n)\})$$

である.

ベクトル $\boldsymbol{\ell}, \boldsymbol{u} \in \mathbb{R}^n$ に対して、**区間** $[\boldsymbol{\ell}, \boldsymbol{u}]_{\mathbb{R}}$ は $\boldsymbol{\ell}$ 以上 $\boldsymbol{u}$ 以下なるベクトルの集合を表す。したがって

$$[\boldsymbol{\ell}, \boldsymbol{u}]_{\mathbb{R}} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{\ell} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$$

である。同様に、ベクトル $\boldsymbol{\ell}, \boldsymbol{u} \in \mathbb{Z}^n$ に対して、**整数区間** $[\boldsymbol{\ell}, \boldsymbol{u}]_{\mathbb{Z}}$ は $\boldsymbol{\ell}$ 以上 $\boldsymbol{u}$ 以下なる整数ベクトルの集合を表す。したがって

$$[\boldsymbol{\ell}, \boldsymbol{u}]_{\mathbb{Z}} = \{\boldsymbol{x} \in \mathbb{Z}^n \mid \boldsymbol{\ell} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$$

である。

ベクトル $\boldsymbol{x} \in \mathbb{R}^n$ と部分集合 $Y \subseteq N$ に対して,

$$x(Y) = \sum_{i \in Y} x(i)$$

とする. すなわち $x(Y)$ は Y に対応する成分の和を表す。

ベクトル $\boldsymbol{x} \in \mathbb{R}^n$ の**正の台** $\mathrm{supp}^+(\boldsymbol{x}) \subseteq N$ と**負の台** $\mathrm{supp}^-(\boldsymbol{x}) \subseteq N$ を、それぞれ

$$\mathrm{supp}^+(\boldsymbol{x}) = \{i \in N \mid x(i) > 0\},$$
$$\mathrm{supp}^-(\boldsymbol{x}) = \{i \in N \mid x(i) < 0\}$$

で表される集合であると定義する.

整数（番号）$i \in N$ に対して、第 i **単位ベクトル** $\boldsymbol{\chi}_i \in \mathbb{R}^n$ とは、各 $j \in N$ について

$$\boldsymbol{\chi}_i(j) = \begin{cases} 1 & (j = i \text{ のとき}) \\ 0 & (\text{それ以外のとき}) \end{cases}$$

なるベクトルである．また $i = 0$ に対応する場合として、

$$\boldsymbol{\chi}_0 = \mathbf{0}$$

と定義する。

部分集合 $Y \subseteq N$ に対して、Y の**特性ベクトル** $\boldsymbol{\chi}_Y \in \mathbb{R}^n$ とは、各 $j \in N$ について

$$\boldsymbol{\chi}_Y(j) = \begin{cases} 1 & (j \in Y \text{ のとき}) \\ 0 & (\text{それ以外のとき}) \end{cases}$$

なるベクトルである。このとき、

$$\boldsymbol{\chi}_Y = \sum_{i \in Y} \boldsymbol{\chi}_i$$

が成り立つ。

ベクトル $\boldsymbol{x}$ に対して、**転置**したベクトルを $\boldsymbol{x}^\top$ で表す。したがって、$\boldsymbol{x}$ が行ベクトルのとき $\boldsymbol{x}^\top$ は列ベクトルであり、

$$\boldsymbol{x}^\top = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{2.1}$$

である。

ベクトル $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ に対して，その**内積**を $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ で表す．すなわち

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{n} x_i y_i \tag{2.2}$$

である．

2.1.2 凸関数

次に凸集合と凸関数に関する用語の定義を行う。

$\mathbb{R}^n$ の部分集合 S が**凸集合**であるというのは、任意の 2 点 $\boldsymbol{x}, \boldsymbol{y} \in S$ に対して、$\boldsymbol{x}$ と $\boldsymbol{y}$ を結ぶ線分が S に含まれるときであると定義する。

$\mathbb{R}^n$ の空でない任意の部分集合 S に対して、S の**閉凸包**とは S を含む最小の閉凸集合のことである。S の閉凸包は一意に定まり、それを $\overline{S}$ と表す。閉凸包の厳密な定義は以下の通りである。S を含む閉凸集合（閉かつ凸である集合）の全体を $\mathcal{C}$ とする、すなわち $\mathcal{C} = \{X \mid X \text{ は閉凸集合}, X \supseteq S\}$ とする．このとき

$$X, Y \in \mathcal{C} \implies X \cap Y \in \mathcal{C}$$

が成り立つ。すべての $X \in \mathcal{C}$ の共通部分

$$A = \bigcap \{X \mid X \in \mathcal{C}\}$$

を考えると、A は閉集合であり、かつ凸集合である。したがって、$A \in \mathcal{C}$ であり、A は集合 $\mathcal{C}$ の包含関係に関して最小な元である。これを S の**閉凸包**という．

$F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ を連続変数の関数とする．$F(\boldsymbol{x})$ が有限値をとる点 $\boldsymbol{x} \in \mathbb{R}^n$ の集合を F の**実効定義域**と呼ぶ。また F の最小値を与える点 $\boldsymbol{x}$ の集合を**最小化集合**と呼ぶ。グラフ $y = F(\boldsymbol{x})$ の上側にある点すべてからなる集合を**エピグラフ**と呼ぶ。すなわち、F の実効定義域 $\mathrm{dom}_{\mathbb{R}} F$、最小化集合 $\mathrm{argmin}_{\mathbb{R}} F$、エピグラフ $\mathrm{epi}\, F$ は，それぞれ

$$\begin{aligned}
\mathrm{dom}_{\mathbb{R}} F &= \{\boldsymbol{x} \in \mathbb{R}^n \mid F(\boldsymbol{x}) < +\infty\}, \\
\mathrm{argmin}_{\mathbb{R}} F &= \{\boldsymbol{x} \in \mathbb{R}^n \mid F(\boldsymbol{x}) \leq F(\boldsymbol{y})\ (\forall \boldsymbol{y} \in \mathbb{R}^n)\}, \\
\mathrm{epi}\, F &= \{(\boldsymbol{x}, \alpha) \in \mathbb{R}^n \times \mathbb{R} \mid \alpha \geq F(\boldsymbol{x})\}
\end{aligned}$$

と定義される。

関数 F が**凸である**というのは、任意の $\boldsymbol{x}, \boldsymbol{y} \in \mathrm{dom}_{\mathbb{R}} F$、および $0 \leq t \leq 1$ なる任意の実数 t に対して、F が

$$tF(\boldsymbol{x}) + (1-t)F(\boldsymbol{y}) \geq F(t\boldsymbol{x} + (1-t)\boldsymbol{y})$$

を満たすときであると定義される。関数 F が凸であることと、そのエピグラフ $\mathrm{epi}\, F$ が凸集合であることは同値である。連続変数の凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ は，$\mathrm{dom}_{\mathbb{R}} F \neq \emptyset$ であるときに**真 (proper)** であるといい，$\mathrm{epi}\, F$ が閉集合であるときに F は**閉 (closed)** であるという． 閉真凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ について，$\mathrm{dom}_{\mathbb{R}} F$ が有界であれば $\mathrm{argmin}_{\mathbb{R}} F \neq \emptyset$ である。

関数 F が点 $\boldsymbol{x} \in \mathbb{R}^n$ の付近で微分可能であれば，**勾配**

$$\nabla F(\boldsymbol{x}) = \left(\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \ldots, \frac{\partial F}{\partial x_n} \right)$$

が定義できる。この勾配を計算すれば極小点であるかどうかの判定ができる．つまり、任意の方向ベクトル $\boldsymbol{d} \in \mathbb{R}^n$ $(\|\boldsymbol{d}\|_1 = 1)$ と十分小さい正実数 $\alpha > 0$ に対して，近似式

$$F(\boldsymbol{x} + \alpha\boldsymbol{d}) \approx F(\boldsymbol{x}) + \alpha\langle \nabla F(\boldsymbol{x}), \boldsymbol{d} \rangle$$

が成り立つので、F がすべての点 $\boldsymbol{x} \in \mathbb{R}^n$ で微分可能な凸関数のときには，点 $\boldsymbol{x}$ が極小点であることと，任意の $\boldsymbol{d}$ に対して

$$\langle \nabla F(\boldsymbol{x}), \boldsymbol{d} \rangle = 0$$

となることは同値である。また，これは

$$\nabla F(\boldsymbol{x}) = \boldsymbol{0}$$

とも同値である．

凸関数 F は一般には微分可能とは限らないが，F が微分可能でない場合でも，$\boldsymbol{x}$ が $\mathrm{dom}_{\mathbb{R}} F$ の内点ならば、**方向微分**

$$F'(\boldsymbol{x};\boldsymbol{d}) = \lim_{\alpha \searrow 0} \frac{F(\boldsymbol{x}+\alpha\boldsymbol{d}) - F(\boldsymbol{x})}{\alpha} \tag{2.3}$$

が存在する．ただし、$\alpha \searrow 0$ は α が正の側から 0 に近づくことを表す．点 $\boldsymbol{x}$ が極小点であるための条件を方向微分を用いて表すと、

$$\boldsymbol{x} \text{ は } F \text{ の極小点} \iff \text{任意の } \boldsymbol{d} \text{ に対して } F'(\boldsymbol{x};\boldsymbol{d}) \geq 0 \tag{2.4}$$

となる．しかしながら，この条件を計算によって確かめるには、無限個の方向 $\boldsymbol{d}$ について調べる必要がある。

2.1.3 離散凸関数

離散変数の関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ についても、同様に実効定義域 $\mathrm{dom}_{\mathbb{Z}} f$ と最小化集合 $\mathrm{argmin}_{\mathbb{Z}} f$ を考えることができ，それぞれ

$$\mathrm{dom}_{\mathbb{Z}} f = \{\boldsymbol{x} \in \mathbb{Z}^n \mid f(\boldsymbol{x}) < +\infty\},$$
$$\mathrm{argmin}_{\mathbb{Z}} f = \{\boldsymbol{x} \in \mathbb{Z}^n \mid f(\boldsymbol{x}) \leq f(\boldsymbol{y})\ (\forall \boldsymbol{y} \in \mathbb{Z}^n)\}$$

と定義される。なお、離散関数は離散点上でのみ定義されるので、連続関数のエピグラフに対応するものを考えることは不自然である。

離散変数の関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が**凸拡張可能** (convex extensible) というのは，ある連続変数の凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が存在して、

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) \qquad (\forall \boldsymbol{x} \in \mathbb{Z}^n) \tag{2.5}$$

を満たすときであると定義される。

離散凸解析では離散変数の **L 凸関数**と **M 凸関数**が重要な役割を担う。この 2 つの関数の定義は後に述べるが、関数クラスの包含関係は図 2.1（左）のようになっている。L 凸関数と M 凸関数はともに凸拡張可能であるが、共通部分を持たない。

離散変数の L 凸関数と M 凸関数の概念を、連続変数の関数に拡張することもできる。したがって、連続変数の L 凸関数と連続変数の M 凸関数を考えることができる。この 2 つの関数の定義も後に述べるが、関数クラスの包含関係は図 2.1（右）のようになっている。連続変数の L 凸関数と M 凸関数はともに凸関数であるが、共通部分を持たない。

離散変数の L 凸関数と M 凸関数について、概念をより一般化した関数クラスがそれぞれ **$\mathrm{L}^\natural$ 凸関数**と **$\mathrm{M}^\natural$ 凸関数**である。なお、$\mathrm{L}^\natural$ は「エル・ナチュラル」、$\mathrm{M}^\natural$ は「エム・ナチュラル」と読む。図 2.1（左）のように、$\mathrm{L}^\natural$ 凸関数と $\mathrm{M}^\natural$ 凸関数はそれぞれ L 凸関数と M 凸関数を包含する。2 つの関数の共通部分、すなわち $\mathrm{L}^\natural$ 凸関数でも $\mathrm{M}^\natural$ 凸関数でもある関数は、**分離凸関数**と呼ばれる、1 変数の凸関数の和で表される関数である。連続変数の場合も、図 2.1（右）のように、$\mathrm{L}^\natural$ 凸関数と $\mathrm{M}^\natural$ 凸関数、そして分離凸関数の関係は同様である。

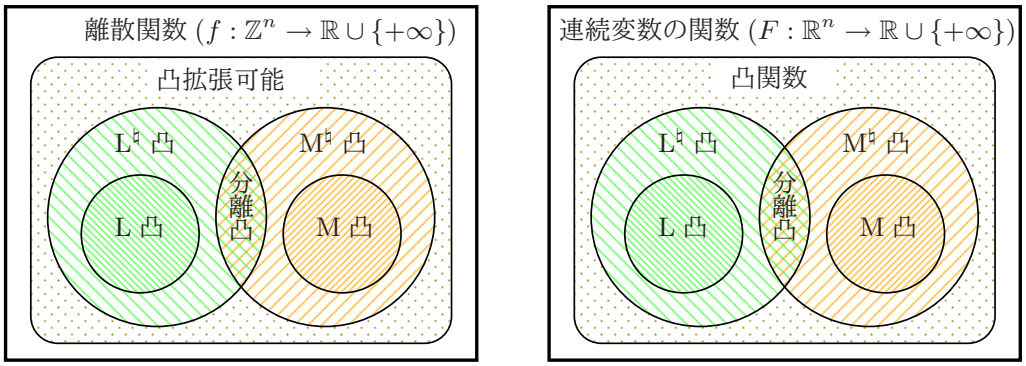


図 2.1. 離散凸関数（左）と凸関数（右）のクラスの包含関係

表 2.1. 用語の使い方

		離散変数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$	連続変数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$
L 凸関数 (広義)	L$^\natural$ 凸関数	離散 L$^\natural$ 凸関数	連続 L$^\natural$ 凸関数
	L 凸関数	離散 L 凸関数	連続 L 凸関数

		離散変数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$	連続変数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$
M 凸関数 (広義)	M$^\natural$ 凸関数	離散 M$^\natural$ 凸関数	連続 M$^\natural$ 凸関数
	M 凸関数	離散 M 凸関数	連続 M 凸関数

本論文中では、L 凸関数の場合には、関数名に g, G を、変数名に $\boldsymbol{p}, \boldsymbol{q}$ を用いることが多い。同様に、M 凸関数の場合には、関数名に f, F を、変数名に $\boldsymbol{x}, \boldsymbol{y}$ を用いることが多い。また、大文字の関数名は連続変数の場合に用いることが多い。

L$^\natural$ 凸関数は L 凸関数を一般化した概念ではあるが、本質的には L 凸関数と等価であるので、本論文では「L 凸関数」という用語を「L$^\natural$ 凸関数」の意味を含めて用いることがある。しかし「L$^\natural$ 凸関数」という用語は「L 凸関数」の意味を含むことはない。「M 凸関数」と「M$^\natural$ 凸関数」の関係も同様である（表 2.1 参照）。

「L 凸関数」という用語で、離散変数と連続変数のどちらの L 凸関数を指すのかは文脈によるが、多くの場合は離散変数である。とくに離散変数であることを明示したい場合には「離散 L 凸関数」と表記する。連続変数の場合には「連続 L 凸関数」と表記することもあるが、連続関数（関数値の連続）と紛らわしいので頻度は少ない。「L$^\natural$ 凸関数」「M 凸関数」「M$^\natural$ 凸関数」についてもそれぞれ同様である（表 2.1 参照）。

2.2 L 凸関数

ここでは L$^\natural$ 凸関数と L 凸関数の定義と基本的な性質について述べる [17, 58, 65, 66, 75]。

2.2.1 L$^\natural$ 凸関数

L$^\natural$ 凸関数の等価な定義はいくつかあるが、まず離散中点凸性を用いたものについて述べる。連続変数の関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ の**中点凸性**は

$$G(\boldsymbol{p}) + G(\boldsymbol{q}) \geq 2G\left(\frac{\boldsymbol{p}+\boldsymbol{q}}{2}\right) \quad (\forall \boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^n) \tag{2.6}$$

と定義される。連続関数 G に対しては、中点凸性と凸性は等価であり、中点凸性を満たすときに G は**凸関数**であると定義することができる。 これに対して、離散中点凸性では、$\frac{\boldsymbol{p}+\boldsymbol{q}}{2}$ を整数格子点上にとるために $\left\lceil \frac{\boldsymbol{p}+\boldsymbol{q}}{2} \right\rceil$ と $\left\lfloor \frac{\boldsymbol{p}+\boldsymbol{q}}{2} \right\rfloor$ の 2 点に分離する。すなわち、離散関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の**離散中点凸性**は

$$g(\boldsymbol{p}) + g(\boldsymbol{q}) \geq g\left(\left\lceil \frac{\boldsymbol{p}+\boldsymbol{q}}{2} \right\rceil\right) + g\left(\left\lfloor \frac{\boldsymbol{p}+\boldsymbol{q}}{2} \right\rfloor\right) \quad (\forall \boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^n) \tag{2.7}$$

と表され、この離散中点凸性を満たすときに、g は **L$^\natural$ 凸関数**であると定義される（図 2.2 参照）.

次に、劣モジュラ性に着目した定義について述べる。第 2.1.1 節に述べたように、ベクトル $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^n$ に対して, 成分ごとに最大値, 最小値をとって得られるベクトルを $\boldsymbol{p} \vee \boldsymbol{q}$, $\boldsymbol{p} \wedge \boldsymbol{q}$ と表す（図 2.3 参照）. すなわち、

$$\boldsymbol{p} \vee \boldsymbol{q} = (\max\{p_1, q_1\}, \max\{p_2, q_2\}, \ldots, \max\{p_n, q_n\}), \tag{2.8}$$
$$\boldsymbol{p} \wedge \boldsymbol{q} = (\min\{p_1, q_1\}, \min\{p_2, q_2\}, \ldots, \min\{p_n, q_n\}) \tag{2.9}$$

である。 関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が**劣モジュラ** (submodular) であるというのは,

(SBF$[\mathbb{Z}]$) 任意の $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^n$ に対して

$$g(\boldsymbol{p}) + g(\boldsymbol{q}) \geq g(\boldsymbol{p} \vee \boldsymbol{q}) + g(\boldsymbol{p} \wedge \boldsymbol{q}) \tag{2.10}$$

を満たすときであると定義される。ただし，$g(\boldsymbol{p})$ と $g(\boldsymbol{q})$ の少なくとも一方が $+\infty$ であるときには，不等式 (2.10) は満たされているものとみなす.

関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が**並進劣モジュラ** (translation-submodular) であるというのは、

(SBF$^\natural[\mathbb{Z}]$) 任意の $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^n$ と任意の $\alpha \in \mathbb{Z}_+$ に対して

$$g(\boldsymbol{p}) + g(\boldsymbol{q}) \geq g((\boldsymbol{p} - \alpha\mathbf{1}) \vee \boldsymbol{q}) + g(\boldsymbol{p} \wedge (\boldsymbol{q} + \alpha\mathbf{1})) \tag{2.11}$$

を満たすときであると定義される。ただし，$g(\boldsymbol{p})$ と $g(\boldsymbol{q})$ のいずれかが $+\infty$ であるときには，不等式 (2.11) は満たされているものとみなす．なお、式 (2.11) で $\alpha = 0$ とすると、劣モジュ

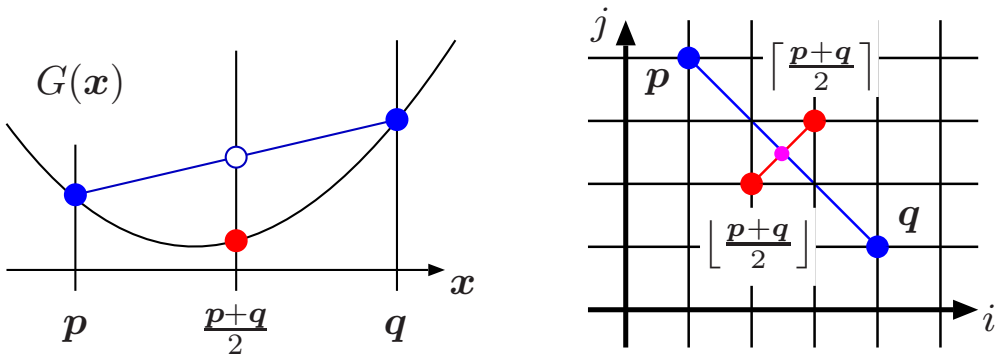


図 2.2. 連続関数と離散関数の中点凸性

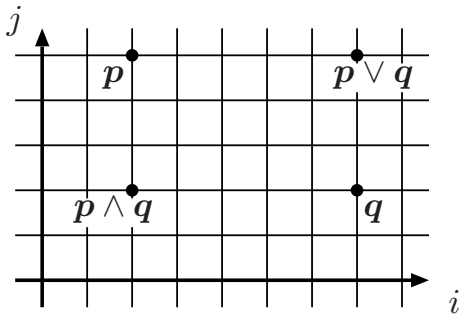


図 2.3. $\boldsymbol{p} \vee \boldsymbol{q}$ と $\boldsymbol{p} \wedge \boldsymbol{q}$ の定義

ラ性の式 (2.10) と一致する。したがって、並進劣モジュラ性の方が劣モジュラ性よりも強い条件である。

これらの定義のもとで、次の命題が成り立つ.

命題 2.1 ([62, 定理 7.7]). 関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ に対して，並進劣モジュラ性 (2.11) は離散中点凸性 (2.7) と同値である. ■

n 変数の関数 g が与えられたとき，$n+1$ 変数の関数 $\tilde{g}$ を

$$\tilde{g}(\boldsymbol{p}, p_{n+1}) = g(\boldsymbol{p} - p_{n+1}\mathbf{1}) \qquad (\forall \boldsymbol{p} \in \mathbb{Z}^n, \forall p_{n+1} \in \mathbb{Z}) \tag{2.12}$$

によって定義する．このとき，次の命題が成り立つ.

命題 2.2 ([62, 定理 7.1]). 関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の並進劣モジュラ性 (2.11) は，関数 $\tilde{g} : \mathbb{Z}^{n+1} \to \mathbb{R} \cup \{+\infty\}$ の劣モジュラ性 (2.10) と同値である. ■

上の 2 つの命題から，$\mathrm{L}^\natural$ 凸関数の必要十分条件が得られる.

定理 2.3. 関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ に関して，次の 3 つの条件はいずれも同値である。

(a) 離散中点凸性 (2.7).
(b) 並進劣モジュラ性 (2.11).
(c) 式 (2.12) で定義される $\tilde{g}$ の劣モジュラ性 (2.10).
したがって、(a), (b), (c) のいずれもが g が $\mathrm{L}^\natural$ 凸関数であるための必要十分条件である. ■

この定理により, $\mathrm{L}^\natural$ 凸関数とは並進劣モジュラ性をもつ関数である, と定義し直すことができる. なお、離散関数は（並進劣モジュラ性よりも条件の弱い）劣モジュラ性をもったとしても、必ずしも $\mathrm{L}^\natural$ 凸関数であるわけではない。それは次の例からわかる。

例 2.1. 任意の 1 変数の離散関数は、劣モジュラ性 (2.10) をもつ。なぜなら、$p \leq q$ なる任意の $p, q \in \mathbb{Z}$ に対して、

$$p \vee q = q, \quad p \wedge q = p$$

であるので、1 変数関数 $g : \mathbb{Z} \to \mathbb{R} \cup \{+\infty\}$ に対して、

$$g(p) + g(q) = g(p \vee q) + g(p \wedge q)$$

が成り立ち、g は劣モジュラの不等式 (2.10) を等号で満たすからである。
しかしながら、1 変数関数には凸でないものがある。例えば $g(p) = (-1)^p$ は劣モジュラでありながら $\mathrm{L}^\natural$ 凸性をもたない関数の例である。 ■

凸拡張可能性に関して, 次の定理が成り立つ.

定理 2.4 ([62, 定理 7.20]). $\mathrm{L}^\natural$ 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は凸拡張可能である. ■

大域最小性に関して, 次の定理が成り立つ. なお、この性質は**最小性規準**と呼ばれる。

定理 2.5 ([62, 定理 7.14(2)]). 関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が $\mathrm{L}^\natural$ 凸関数のとき、$\boldsymbol{p} \in \mathrm{dom}_{\mathbb{Z}}\, g$ が g の最小点であるためには, 任意の $\boldsymbol{q} \in \{0, 1\}^n$ に対して

$$g(\boldsymbol{p}) \leq \min\{g(\boldsymbol{p} - \boldsymbol{q}), g(\boldsymbol{p} + \boldsymbol{q})\}$$

となることが必要十分である. ■

離散関数における「凸性」を定義するにあたって、定義の正当性を連続変数の凸関数の性質との類似性に求めるならば、この 2 つの定理の存在は重要である。前者、すなわち凸拡張可能であることは、$\mathrm{L}^\natural$ 凸関数が連続変数の凸関数に対応づけられることを示す。後者、すなわち最小性規準を持つことは、連続変数の凸関数の重要な性質の 1 つである、局所最小性と大域最小性が一致するという性質が、$\mathrm{L}^\natural$ 凸関数で実現されていることを示す。このような観点から、2 つの定理の存在によって、$\mathrm{L}^\natural$ 凸関数は「離散凸関数」としての資格を有していると言える。

2.2.2 L 凸関数

関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が **L 凸関数**であるというのは、g が $\mathrm{L}^\natural$ 凸関数であり, かつ,

(TRF[$\mathbb{Z}$])　ある実数 r が存在して，任意の $\boldsymbol{p} \in \mathbb{Z}^n$ に対して

$$g(\boldsymbol{p} + \mathbf{1}) = g(\boldsymbol{p}) + r \tag{2.13}$$

を満たすときであると定義される。この条件 (TRF[$\mathbb{Z}$]) は、**1 方向の線形性**と呼ばれる。

L 凸関数の必要十分条件として，次の定理が知られている．

定理 2.6 ([62, 定理 7.3])**.** 関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が L 凸関数であるためには，g が劣モジュラ性 (2.10) と $\mathbf{1}$ 方向の線形性 (2.13) をもつことが必要十分である．■

あるいは、劣モジュラ性 (2.10) と $\mathbf{1}$ 方向の線形性 (2.13) の条件を L 凸関数の定義とすることもできる。

この定理により，L 凸関数は $\mathrm{L}^\natural$ 凸関数の特殊ケースである。しかし両者は本質的には等価な概念であり，次のように互いに変換しあうことができる。n 変数の $\mathrm{L}^\natural$ 凸関数 g が与えられたとき，$n+1$ 変数の関数 $\tilde{g} : \mathbb{Z}^{n+1} \to \mathbb{R} \cup \{+\infty\}$ を

$$\tilde{g}(\boldsymbol{p}, p_{n+1}) = g(\boldsymbol{p} - p_{n+1}\mathbf{1}) \qquad (\forall \boldsymbol{p} \in \mathbb{Z}^n, \forall p_{n+1} \in \mathbb{Z}) \tag{2.14}$$

によって定義すると，$\tilde{g}$ は L 凸関数になる（式 (2.13) を $r = 0$ に対して満たす）．逆に，n 変数の L 凸関数 g が与えられたとき，$n-1$ 変数の関数 $\hat{g} : \mathbb{Z}^{n-1} \to \mathbb{R} \cup \{+\infty\}$ を

$$\hat{g}(\boldsymbol{q}) = g(\boldsymbol{q}, 0) \qquad (\forall \boldsymbol{q} \in \mathbb{Z}^{n-1}) \tag{2.15}$$

によって定義すると，$\hat{g}$ は $\mathrm{L}^\natural$ 凸関数になる．このような意味で両者は本質的に等価である．

ただし、両者が 1 対 1 対応をしているわけではなく、L 凸関数には式 (2.13) の r の自由度があることに注意されたい。つまり、$\mathrm{L}^\natural$ 凸関数から式 (2.14) によって作り出した L 凸関数では常に $r = 0$ であり、逆に r の異なる L 凸関数から同じ $\mathrm{L}^\natural$ 凸関数が作り出される。

凸拡張可能性と最小性規準に関して，次の定理が成り立つ．この定理は、$\mathrm{L}^\natural$ 凸関数に対する定理（定理 2.4，定理 2.5）と本質的に等価な内容である．

定理 2.7 ([62, 定理 7.19])**.** L 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は凸拡張可能である．■

定理 2.8 ([62, 定理 7.14(1)])**.** 関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が L 凸関数で，式 (2.13) で $r = 0$ であるとする．$\boldsymbol{p} \in \mathrm{dom}_{\mathbb{Z}}\, g$ が g の最小点であるためには，任意の $\boldsymbol{q} \in \{0, 1\}^n$ に対して

$$g(\boldsymbol{p}) \leq g(\boldsymbol{p} + \boldsymbol{q})$$

となることが必要十分である．■

2.2.3　$\mathrm{L}^\natural$ 凸集合と L 凸集合

$\mathrm{L}^\natural$ 凸関数と L 凸関数に対応する離散凸集合の概念を説明する。一般に、集合 $S \subseteq \mathbb{Z}^n$ に対して

$$\delta_S(\boldsymbol{p}) = \begin{cases} 0 & (\boldsymbol{p} \in S) \\ +\infty & (\boldsymbol{p} \notin S) \end{cases}$$

で定義される関数 $\delta_S : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ を集合 S の**標示関数**という。集合 $S \subseteq \mathbb{Z}^n$ が **$\mathrm{L}^\natural$ 凸集合**であるというのは，その標示関数 $\delta_S : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が $\mathrm{L}^\natural$ 凸関数であるときであると定義される。同様に，標示関数 δ_S が L 凸関数のとき，集合 S は **L 凸集合**であると定義される。すなわち，

$$S \text{ が } \mathrm{L}^\natural \text{ 凸集合} \iff \delta_S \text{ が } \mathrm{L}^\natural \text{ 凸関数}, \tag{2.16}$$

$$S \text{ が L 凸集合} \iff \delta_S \text{ が L 凸関数} \tag{2.17}$$

である．

$\mathrm{L}^\natural$ 凸集合について、定義 (2.16) と $\mathrm{L}^\natural$ 凸関数の必要十分条件（定理 2.3）から次の定理が成り立つ。

定理 2.9 ([65, 定理 4.8])**.** 集合 $S \subseteq \mathbb{Z}^n$ に関して、次の 3 つの条件 (a)， (b)， (c) は同値である．

(a) 離散中点凸性

$$\forall \boldsymbol{p},\ \boldsymbol{q} \in S \implies \left\lceil \frac{\boldsymbol{p}+\boldsymbol{q}}{2} \right\rceil,\ \left\lfloor \frac{\boldsymbol{p}+\boldsymbol{q}}{2} \right\rfloor \in S. \tag{2.18}$$

(b) 任意の非負整数 α に対して，

$$\forall \boldsymbol{p},\ \boldsymbol{q} \in S \implies (\boldsymbol{p} - \alpha\mathbf{1}) \vee \boldsymbol{q},\ \boldsymbol{p} \wedge (\boldsymbol{q} + \alpha\mathbf{1}) \in S. \tag{2.19}$$

(c) 任意の $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^n$ と任意の $\alpha, \beta \in \mathbb{Z}$ に対して、

$$\boldsymbol{p} - \alpha\mathbf{1},\ \boldsymbol{q} - \beta\mathbf{1} \in S \implies (\boldsymbol{p} \vee \boldsymbol{q}) - (\alpha \vee \beta)\mathbf{1},\ (\boldsymbol{p} \wedge \boldsymbol{q}) - (\alpha \wedge \beta)\mathbf{1} \in S. \tag{2.20}$$

この (a)， (b)， (c) のいずれもが，S が $\mathrm{L}^\natural$ 凸集合であるための必要十分条件である． ■

例 2.2. 一般の n に対して，$\boldsymbol{a} \in \{\mathbb{Z} \cup \{-\infty\}\}^n$, $\boldsymbol{b} \in \{\mathbb{Z} \cup \{+\infty\}\}^n$ の定める整数区間

$$[\boldsymbol{a}, \boldsymbol{b}]_{\mathbb{Z}} = \{\boldsymbol{p} \in \mathbb{Z}^n \mid a_i \le p_i \le b_i\ (i = 1, 2, \ldots, n)\} \tag{2.21}$$

は $\mathrm{L}^\natural$ 凸集合である。 ■

L 凸集合について、定義 (2.17) と L 凸関数の必要十分条件（定理 2.6）から、次の定理が成り立つ。

定理 2.10 ([65, 定理 4.10])**.** 集合 $S \subseteq \mathbb{Z}^n$ が L 凸集合であるためには，条件

$$\boldsymbol{p},\ \boldsymbol{q} \in S \Longrightarrow \boldsymbol{p} \vee \boldsymbol{q},\ \boldsymbol{p} \wedge \boldsymbol{q} \in S, \tag{2.22}$$

$$\boldsymbol{p} \in S \Longrightarrow \boldsymbol{p} + \mathbf{1},\ \boldsymbol{p} - \mathbf{1} \in S \tag{2.23}$$

がともに成り立つことが必要十分である． ■

上のように定義された $\mathrm{L}^\natural$ 凸集合と L 凸集合は、次の定理のように、簡単な不等式で記述することもできる。

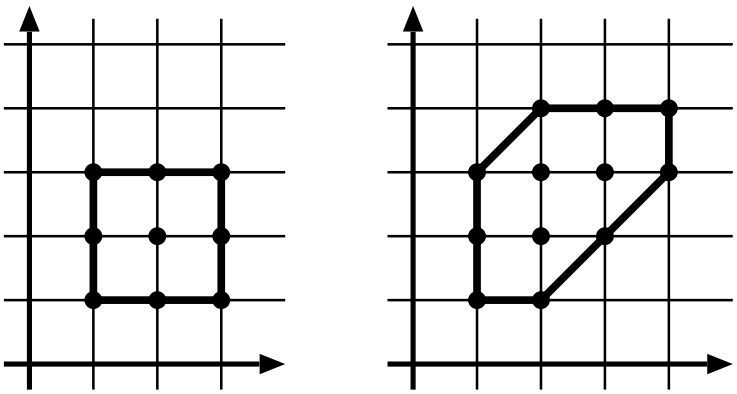


図 2.4. $\mathrm{L}^\natural$ 凸集合の例

定理 2.11 ([65, 定理 4.12])**.**

(1) 集合 $S \subseteq \mathbb{Z}^n$ が $\mathrm{L}^\natural$ 凸集合であるためには，ある整数 $\alpha_i \in \mathbb{Z} \cup \{-\infty\}$, $\beta_i \in \mathbb{Z} \cup \{+\infty\}$, $\gamma_{ij} \in \mathbb{Z} \cup \{+\infty\}$ $(i, j = 1, 2, \ldots, n; i \neq j)$ が存在して

$$\begin{aligned} S &= \{\boldsymbol{p} \in \mathbb{Z}^n \mid \alpha_i \leq p_i \leq \beta_i,\ p_j - p_i \leq \gamma_{ij}\ (i, j = 1, 2, \ldots, n; i \neq j)\} \\ &= \{\boldsymbol{p} \in \mathbb{Z}^n \mid \alpha_i \leq p_i \leq \beta_i,\ p_j - \gamma_{ij} \leq p_i \leq p_j + \gamma_{ji}\ (i, j = 1, 2, \ldots, n; i \neq j)\} \end{aligned}$$

と表されることが必要十分である．

(2) 集合 $S \subseteq \mathbb{Z}^n$ が L 凸集合であるためには，ある整数 $\gamma_{ij} \in \mathbb{Z} \cup \{+\infty\}$ $(i, j = 1, 2, \ldots, n; i \neq j)$ が存在して

$$\begin{aligned} S &= \{\boldsymbol{p} \in \mathbb{Z}^n \mid p_j - p_i \leq \gamma_{ij}\ (i, j = 1, 2, \ldots, n; i \neq j)\} \\ &= \{\boldsymbol{p} \in \mathbb{Z}^n \mid p_j - \gamma_{ij} \leq p_i \leq p_j + \gamma_{ji}\ (i, j = 1, 2, \ldots, n; i \neq j)\} \end{aligned}$$

と表されることが必要十分である． ■

定理 2.11 より，集合 $S \subseteq \mathbb{Z}^n$ が $\mathrm{L}^\natural$ 凸集合、あるいは L 凸集合ならば、S の閉凸包 $\overline{S} \subseteq \mathbb{R}^n$ は

$$S = \overline{S} \cap \mathbb{Z}^n$$

を満足することがわかる。

連続変数の凸関数について、実効定義域と最小化集合はどちらも凸集合であるが、それと同様に、離散 $\mathrm{L}^\natural$ 凸/L 凸関数については、次の 2 つの定理が成り立つ。

定理 2.12 ([62, 命題 7.8])**.**

(1) $\mathrm{L}^\natural$ 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の実効定義域 $\mathrm{dom}_{\mathbb{Z}}\, g$ は $\mathrm{L}^\natural$ 凸集合である．

(2) L 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の実効定義域 $\mathrm{dom}_{\mathbb{Z}}\, g$ は L 凸集合である．

定理 2.13 ([62, 命題 7.16])**.**

(1) $\mathrm{L}^\natural$ 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の最小化集合 $\mathrm{argmin}_{\mathbb{Z}}\, g$ は $\mathrm{L}^\natural$ 凸集合である．

(2) L 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の最小化集合 $\operatorname{argmin}_{\mathbb{Z}} g$ は L 凸集合である．

2.2.4 2 次関数

2 次関数が $\mathrm{L}^\natural$ 凸関数であるための条件を考える．一般の n 変数（離散変数）の 2 次関数は

$$g(\boldsymbol{p}) = \boldsymbol{p}^\top A \boldsymbol{p} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} p_i p_j \qquad (\boldsymbol{p} = (p_i)_{i=1}^n \in \mathbb{Z}^n) \tag{2.24}$$

とあらわせる．ただし，係数行列 A の成分 a_{ij} は実数である。ここで任意の i, j について $a_{ij} = a_{ji}$ であるとしてよい。すなわち A は対称行列としてよい。

定理 2.14 ([62])**.** 式 (2.24) の 2 次関数 g が $\operatorname{dom}_{\mathbb{Z}} g = \mathbb{Z}^n$ の仮定の下で $\mathrm{L}^\natural$ 凸関数であるための必要十分条件は,

$$a_{ij} \le 0 \quad (i \ne j), \qquad \sum_{j=1}^{n} a_{ij} \ge 0 \quad (i = 1, 2, \ldots, n) \tag{2.25}$$

で与えられる．

対称行列 A に対する条件 (2.25) を書き換えると、

$$a_{ij} \le 0 \quad (i \ne j), \qquad a_{ii} \ge \sum_{j \ne i} |a_{ij}| \quad (i = 1, 2, \ldots, n)$$

となる。この条件を満たす行列は優対角対称 M 行列と呼ばれるものである。第 2 の条件が, 優対角性を示している。優対角な対称 M 行列は半正定値であることが知られており、凸性との対応関係が見られる。

L 凸関数の条件は，次のようになる．

定理 2.15 ([62])**.** 式 (2.24) の 2 次関数 g が $\operatorname{dom}_{\mathbb{Z}} g = \mathbb{Z}^n$ の仮定の下で L 凸関数であるための必要十分条件は,

$$a_{ij} \le 0 \quad (i \ne j), \qquad \sum_{j=1}^{n} a_{ij} = 0 \quad (i = 1, 2, \ldots, n) \tag{2.26}$$

で与えられる． ■

2.2.5 $\mathrm{L}^\natural$ 凸/L 凸関数の例

$\mathrm{L}^\natural$ 凸/L 凸関数の基本的な例を挙げる．実効定義域については、$\mathrm{L}^\natural$ 凸関数の場合は $\mathrm{L}^\natural$ 凸集合であること，L 凸関数の場合は L 凸集合であることを仮定する（定理 2.12)。$\mathbb{Z}^n$ は $\mathrm{L}^\natural$ 凸集合でもあり、L 凸集合でもある。$\boldsymbol{p} = (p_1, p_2, \ldots, p_n) \in \mathbb{Z}^n$ とする．

1 次関数

ベクトル $\boldsymbol{\alpha} \in \mathbb{R}^n$ と実数 $\beta \in \mathbb{R}$ によって定義される 1 次関数

$$g(\boldsymbol{p}) = \langle \boldsymbol{\alpha}, \boldsymbol{p} \rangle + \beta \tag{2.27}$$

は、実効定義域が $\mathrm{L}^\natural$ 凸集合であれば $\mathrm{L}^\natural$ 凸関数であり、実効定義域が L 凸集合であれば L 凸関数である。

2 次関数

実数係数の 2 次関数

$$g(\boldsymbol{p}) = \boldsymbol{p}^\top A \boldsymbol{p} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} p_i p_j \qquad (a_{ij} = a_{ji} \in \mathbb{R}) \tag{2.28}$$

は、先に示したように、係数行列 $A = (a_{ij})$ が

$$a_{ij} \leq 0 \quad (i \neq j), \qquad \sum_{j=1}^{n} a_{ij} \geq 0 \quad (i = 1, 2, \ldots, n) \tag{2.29}$$

を満たすとき $\mathrm{L}^\natural$ 凸関数であり（定理 2.14），

$$a_{ij} \leq 0 \quad (i \neq j), \qquad \sum_{j=1}^{n} a_{ij} = 0 \quad (i = 1, 2, \ldots, n) \tag{2.30}$$

を満たすとき L 凸関数である（定理 2.15）．

分離凸関数

1 変数の凸関数 ψ_i $(i = 1, 2, \ldots, n)$ の和の形で定義される**分離凸関数**

$$g(\boldsymbol{p}) = \sum_{i=1}^{n} \psi_i(p_i) \tag{2.31}$$

は $\mathrm{L}^\natural$ 凸関数であり，1 変数の凸関数 ψ_{ij} $(i, j = 1, 2, \ldots, n; i \neq j)$ の和の形で定義される関数

$$g(\boldsymbol{p}) = \sum_{i \neq j} \psi_{ij}(p_i - p_j) \tag{2.32}$$

は L 凸関数である．さらに

$$g(\boldsymbol{p}) = \sum_{i=1}^{n} \psi_i(p_i) + \sum_{i \neq j} \psi_{ij}(p_i - p_j) \tag{2.33}$$

は $\mathrm{L}^\natural$ 凸関数である．

劣モジュラ集合関数

劣モジュラ集合関数は、$\mathrm{dom}_{\mathbb{Z}}\, g \subseteq \{0, 1\}^n$ である $\mathrm{L}^\natural$ 凸関数 g と同一視できる（第 2.2.6 節参照）．

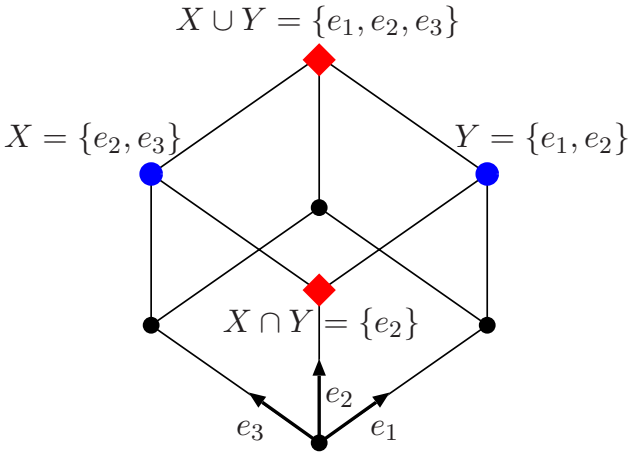


図 2.5. 劣モジュラ（優モジュラ）集合関数の構造

2.2.6 劣モジュラ集合関数

L$^\natural$ 凸関数と劣モジュラ集合関数は、関係が深いのでここで取り上げる。

集合関数とは，部分集合を入力として受け取り、それに対応して数値を出力する関数である。出力する数値は、整数や実数などが考えられる。例えば集合 $N = \{1, 2, \ldots, n\}$ とおき，N の部分集合の全体を 2^N と表すとき，N 上の集合関数は $\mu : 2^N \to \mathbb{R} \cup \{+\infty\}$ のような形式で表される。

集合関数 $\mu : 2^N \to \mathbb{R} \cup \{+\infty\}$ は，不等式

$$\mu(X) + \mu(Y) \geq \mu(X \cup Y) + \mu(X \cap Y) \qquad (\forall X, Y \subseteq N) \tag{2.34}$$

を満たすとき，**劣モジュラ関数**と呼ばれる（図 2.5 参照）．また、逆向きの不等式

$$\mu(X) + \mu(Y) \leq \mu(X \cup Y) + \mu(X \cap Y) \qquad (\forall X, Y \subseteq N) \tag{2.35}$$

を満たすとき，**優モジュラ関数**と呼ばれる．（あるいは、$-\mu$ が劣モジュラ関数のときに μ は優モジュラ関数だと定義してもよい。）

例 2.3. $\mu(X) = |X|$ (=集合 X の要素の個数) に対して、

$$|X| + |Y| = |X \cup Y| + |X \cap Y|$$

であるので、$\mu(X)$ は不等式 (2.34) を等号で満たす劣モジュラ関数であり、不等式 (2.35) を等号で満たす優モジュラ関数でもある。このように、劣モジュラかつ優モジュラであるものは、モジュラ関数と呼ばれる。 ■

例 2.4.

(1) $\mu(X) = |X|^2$ は優モジュラ関数である．
(2) $\mu(X) = \sqrt{|X|}$ は劣モジュラ関数である． ■

証明.
(1) $\mu(X) = |X|^2$ のとき、

$$\mu(X) + \mu(Y) \leq \mu(X \cup Y) + \mu(X \cap Y)$$

を証明する。$a = |X \cap Y|$, $x = |X| - a$, $y = |Y| - a$ とすると、a, x, y はいずれも非負整数である。

$$\begin{aligned}
&(\text{左辺}) - (\text{右辺}) = \mu(X) + \mu(Y) - \mu(X \cup Y) - \mu(X \cap Y) \\
&= (a+x)^2 + (a+y)^2 - (a+x+y)^2 - a^2 \\
&= (a^2 + 2ax + x^2) + (a^2 + 2ay + y^2) - \{a^2 + 2a(x+y) + (x+y)^2\} - a^2 \\
&= -2xy \leq 0
\end{aligned}$$

より証明される。
(2) $\mu(X) = \sqrt{X}$ のとき、

$$\mu(X) + \mu(Y) \geq \mu(X \cup Y) + \mu(X \cap Y)$$

は、$a = |X \cap Y|$, $x = |X| - a$, $y = |Y| - a$ とした時に

$$\sqrt{a+x} + \sqrt{a+y} \geq \sqrt{a+x+y} + \sqrt{a} \qquad (a, x, y \geq 0)$$

であることから、成り立つことがわかる。 ■

特性ベクトルについては第 2.1.1 節でも述べたが、ここではより詳しく説明する。任意のベクトル $\boldsymbol{p} \in \{0,1\}^n$ は，$p_i = 1$ である番号 i の全体を X とすれば，第 i 単位ベクトル $\boldsymbol{\chi}_i$ を用いて

$$\boldsymbol{p} = \sum_{i \in X} \boldsymbol{\chi}_i$$

と表現される．例えば，$n = 5$ で $\boldsymbol{p} = (1,0,0,1,1)$ のとき，$X = \{1,4,5\}$ である．このように，ベクトル $\boldsymbol{p} \in \{0,1\}^n$ と部分集合 X は 1 対 1 に対応がとれる。部分集合 X に対応するベクトルを X の**特性ベクトル**と呼び，記号 $\boldsymbol{\chi}_X$ で表す．この例では，$(1,0,0,1,1) = \boldsymbol{\chi}_{\{1,4,5\}}$ である．

この対応関係により，$\mathrm{dom}_{\mathbb{Z}}\, g \subseteq \{0,1\}^n$ である関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ と，N 上の集合関数 $\mu : 2^N \to \mathbb{R} \cup \{+\infty\}$ の間に 1 対 1 対応が得られる．すなわち

$$\mu(X) = g(\boldsymbol{\chi}_X) \qquad (\forall X \subseteq N) \tag{2.36}$$

による μ と g の対応である．例えば，$n = 5$ のとき，$\mu(\{1,4,5\}) = g(1,0,0,1,1)$ である．

$\{0,1\}$ ベクトルの演算と、集合の包含関係に対応がとれる。2 つの $\{0,1\}^n$ ベクトルの平均値を切り上げたものと切り捨てたものが、2 つの集合の和集合と積集合に対応する。つまり

$$\boldsymbol{p} = \boldsymbol{\chi}_X,\ \boldsymbol{q} = \boldsymbol{\chi}_Y \quad \Longrightarrow \quad \left\lceil \frac{\boldsymbol{p}+\boldsymbol{q}}{2} \right\rceil = \boldsymbol{\chi}_{X \cup Y},\ \left\lfloor \frac{\boldsymbol{p}+\boldsymbol{q}}{2} \right\rfloor = \boldsymbol{\chi}_{X \cap Y} \tag{2.37}$$

が成り立つ.

式 (2.37) により，g の離散中点凸性 (2.7) と μ の劣モジュラ性 (2.34) が対応する．したがって，次の定理が成り立つ.

定理 2.16 ([62, 命題 7.4])**.** 式 (2.36) の対応関係の下で，g が $\mathrm{L}^\natural$ 凸関数であることと，μ が劣モジュラ関数であることは同値である. ■

劣モジュラ集合関数が最適化の分野で着目され始めたのは、1960 年代の終わり頃からである。J. Edmonds らによって組合せ最適化の分野が大きく発展する過程で、劣モジュラ関数が意識されるようになったが、当時はまだ劣モジュラ関数が凸関数か凹関数かのどちらであるか、定まった見解はなかった。

1980 年代はじめになって，S. Fujishige，A. Frank，L. Lovász らにより，集合関数の劣モジュラ性と凸性との関係が明確になった．Fujishige は劣モジュラ関数のフェンシェル型双対定理 [15] を示し，Frank は劣モジュラ関数の離散分離定理 [14] を示した．Lovász は，ある集合関数が劣モジュラであることと，その集合関数に関連する連続変数の関数が凸関数であることが同値であることを示した [46]．これらの成果により、劣モジュラ関数は凸関数であるとの見解で一致し、劣モジュラ関数の双対性は凸解析における双対性に整数性を加えたものである、との認識が受け入れられることになった。そして、1990 年代後半からの K. Murota による離散凸解析の進展へとつながる [19, 57, 58, 69] 。

劣モジュラ関数の最小化を高速に行うアルゴリズムの開発は、1970 年頃から研究対象として取り組まれてきた。まず、1981 年に楕円体法に基づく弱多項式時間アルゴリズムが M. Grötschel， Lovász, A. Schrijver によって提案された [26]。1988 年には同じグループによって強多項式時間アルゴリズムに改良された。そして 1999 年、組合せ的な強多項式時間アルゴリズムが、S. Iwata，L. Fleischer，Fujishige [35] の 3 人のグループと Schrijver [85] によって、ほぼ同時に提案された。その後もアルゴリズムの改良が続けられている [81]。

なお，劣モジュラ集合関数の最大化は困難な問題であり，最大カット問題 (Maximum Cut Problem) を含んでいることから、P≠NP 予想のもとでは多項式時間のアルゴリズムが存在しないことがわかる。(P≠NP 予想を用いずに多項式時間のアルゴリズムが存在しないことを証明することもできる [37, 45, 46].）近年は近似アルゴリズムの設計が活発に行われている [1, 8, 9, 13, 80, 89, 97]。一方，$\{0,1\}$ ベクトル上の $\mathrm{M}^\natural$ 凹関数は劣モジュラであり（後述する定理 2.23 参照），$\mathrm{M}^\natural$ 凹関数の最大化（= $\mathrm{M}^\natural$ 凸関数の最小化）には効率的なアルゴリズムが知られている．したがって，$\{0,1\}$ ベクトル上の $\mathrm{M}^\natural$ 凹関数は、容易に最大化のできる劣モジュラ集合関数の部分族となっている [67]。

2.2.7 連続変数の $\mathrm{L}^\natural$ 凸/L 凸関数

ここでは、離散 $\mathrm{L}^\natural$ 凸/L 凸関数の概念を連続関数上に拡張する [70, 73, 74]。

連続変数の $\mathrm{L}^\natural$ 凸関数

凸関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が

($\mathbf{SBF}^\natural[\mathbb{R}]$)　任意の $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^n$ と任意の $\alpha \in \mathbb{R}_+$ に対して

$$G(\boldsymbol{p}) + G(\boldsymbol{q}) \geq G((\boldsymbol{p} - \alpha\mathbf{1}) \vee \boldsymbol{q}) + G(\boldsymbol{p} \wedge (\boldsymbol{q} + \alpha\mathbf{1})) \tag{2.38}$$

を満たすとき，関数 G は **$\mathbf{L}^\natural$ 凸関数**と定義される．ただし，$G(\boldsymbol{p})$ と $G(\boldsymbol{q})$ のいずれかが $+\infty$ であるときには，不等式 (2.38) は満たされているものとみなす．

性質 ($\mathrm{SBF}^\natural[\mathbb{R}]$) は**並進劣モジュラ性** (translation-submodularity) と呼ばれる。さらに $\alpha = 0$ とした場合、つまり

($\mathbf{SBF}[\mathbb{R}]$)　任意の $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^n$ に対して

$$G(\boldsymbol{p}) + G(\boldsymbol{q}) \geq G(\boldsymbol{p} \vee \boldsymbol{q}) + G(\boldsymbol{p} \wedge \boldsymbol{q}) \tag{2.39}$$

は**劣モジュラ性** (submodularity) と呼ばれる。ただし，$G(\boldsymbol{p})$ と $G(\boldsymbol{q})$ のいずれかが $+\infty$ であるときには，不等式 (2.39) は満たされているものとみなす．この劣モジュラ性を表す不等式は、$\boldsymbol{p}, \boldsymbol{q}$ が連続変数になっていることを除いて、離散変数の関数に関する劣モジュラ性の不等式 (2.10) と同じである。

$\mathrm{L}^\natural$ 凸関数を、並進劣モジュラ性を用いることなく、劣モジュラ性だけで定義することもできる。そのためには、変数の数を 1 つ追加する。

定理 2.17 ([65, 定理 6.3])**.** 関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が $\mathrm{L}^\natural$ 凸関数であるためには,

$$\tilde{G}(\boldsymbol{p}, p_{n+1}) = G(\boldsymbol{p} - p_{n+1}\mathbf{1}) \qquad (\forall \boldsymbol{p} \in \mathbb{R}^n, \forall p_{n+1} \in \mathbb{R})$$

で定義される $\tilde{G} : \mathbb{R}^{n+1} \to \mathbb{R} \cup \{+\infty\}$ が劣モジュラ性 (2.39) をもつことが必要十分である．

■

なお、離散 $\mathrm{L}^\natural$ 凸関数であるための必要十分条件として、定理 2.3 には (a)， (b)， (c) の 3 通りあった。このうち、(b) は式 (2.38) に、(c) は定理 2.17 にそれぞれ対応する。しかしながら (a) の離散中点凸性については、連続変数に置き換えると、通常の凸関数を定義する中点凸性となるだけで、$\mathrm{L}^\natural$ 凸性に対応する性質は得られない。

連続変数の L 凸関数

関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が **$\mathbf{L}$ 凸関数**であることは，G が $\mathrm{L}^\natural$ 凸関数であり，さらに，$\mathbf{1}$ 方向の線形性

($\mathbf{TRF}[\mathbb{R}]$)　ある実数 r が存在して，任意の $\boldsymbol{p} \in \mathbb{R}^n$ と任意の $\alpha \in \mathbb{R}$ に対して

$$G(\boldsymbol{p} + \alpha\mathbf{1}) = G(\boldsymbol{p}) + \alpha r \tag{2.40}$$

をもつことと定義される．

L 凸関数の必要十分条件として，次の定理が知られている．

定理 2.18 ([62, 定理 7.30])**.** 関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が L 凸関数であるためには，G が劣モジュラ性 (2.39) と $\mathbf{1}$ 方向の線形性 (2.40) をもつことが必要十分である． ■

定義により，L 凸関数は $\mathrm{L}^\natural$ 凸関数の特殊ケースである。しかし両者は本質的には等価な概念であり，次のように互いに変換しあうことができる。n 変数の $\mathrm{L}^\natural$ 凸関数 G が与えられたとき，$n+1$ 変数の関数 $\tilde{G} : \mathbb{R}^{n+1} \to \mathbb{R} \cup \{+\infty\}$ を

$$\tilde{G}(\boldsymbol{p}, p_{n+1}) = G(\boldsymbol{p} - p_{n+1}\mathbf{1}) \qquad (\forall \boldsymbol{p} \in \mathbb{R}^n, \forall p_{n+1} \in \mathbb{R}) \tag{2.41}$$

によって定義すると，$\tilde{G}$ は L 凸関数になる．逆に，n 変数の L 凸関数 G が与えられたとき，$n-1$ 変数の関数 $\hat{G} : \mathbb{R}^{n-1} \to \mathbb{R} \cup \{+\infty\}$ を

$$\hat{G}(\boldsymbol{q}) = G(\boldsymbol{q}, 0) \qquad (\forall \boldsymbol{q} \in \mathbb{R}^{n-1}) \tag{2.42}$$

によって定義すると，$\hat{G}$ は $\mathrm{L}^\natural$ 凸関数になる．このように $\mathrm{L}^\natural$ 凸関数と L 凸関数は両者は本質的に等価であり、離散変数の $\mathrm{L}^\natural$ 凸関数と L 凸関数が等価であることとも同じ構造をしている。

最小性規準

$\mathrm{L}^\natural$ 凸/L 凸関数は凸関数であるから，局所最小解が大域最小解となる。局所最小解であることは，式 (2.4) の「$\boldsymbol{p}$ は G の極小点 $\iff$ 任意の $\boldsymbol{d}$ に対して $G'(\boldsymbol{p}; \boldsymbol{d}) \geq 0$」によって確認できるが，方向の離散性があるので，特定の方向 $\boldsymbol{d}$ への方向微分を調べればよい．

定理 2.19 ([62, 定理 7.33])**.**

(1) 関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が $\mathrm{L}^\natural$ 凸関数のとき，$\boldsymbol{p} \in \mathrm{dom}_{\mathbb{R}} G$ が G の最小点であるためには，任意の $\boldsymbol{q} \in \{0, 1\}^n$ に対して

$$G'(\boldsymbol{p}; \boldsymbol{q}) \geq 0, \qquad G'(\boldsymbol{p}; -\boldsymbol{q}) \geq 0$$

となることが必要十分である．

(2) 関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が L 凸関数のとき，$\boldsymbol{p} \in \mathrm{dom}_{\mathbb{R}} G$ が G の最小点であるためには，$G'(\boldsymbol{p}; \mathbf{1}) = 0$，かつ，任意の $\boldsymbol{q} \in \{0, 1\}^n$ に対して

$$G'(\boldsymbol{p}; \boldsymbol{q}) \geq 0$$

となることが必要十分である． ■

2.2.8 連続変数の $\mathrm{L}^\natural$ 凸/L 凸集合

連続変数の $\mathrm{L}^\natural$ 凸/L 凸関数に対応する凸集合について説明する。

集合 $S \subseteq \mathbb{R}^n$ が **$\mathbf{L}^\natural$ 凸集合**であるというのは，その標示関数 $\delta_S : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が $\mathrm{L}^\natural$ 凸関数であるときであると定義される。同様に，標示関数 δ_S が L 凸関数のとき，集合 S は **L**

凸集合であると定義される。すなわち,

$$S \text{ が } \mathrm{L}^\natural \text{ 凸集合} \iff \delta_S \text{ が } \mathrm{L}^\natural \text{ 凸関数}, \tag{2.43}$$
$$S \text{ が } \mathrm{L} \text{ 凸集合} \iff \delta_S \text{ が } \mathrm{L} \text{ 凸関数} \tag{2.44}$$

である．式 (2.43) と式 (2.44) で δ_S が閉真凸関数ならば、$\mathrm{L}^\natural$ 凸/L 凸集合は閉集合であることがわかる。

$\mathrm{L}^\natural$ 凸集合について、定義 (2.43) と $\mathrm{L}^\natural$ 凸関数の必要十分条件（式 (2.38) や定理 2.17）から次の定理が成り立つ。

定理 2.20 ([65, 定理 6.6(1)])**.** 閉集合 $S \subseteq \mathbb{R}^n$ に対して次の 2 つの条件 (a), (b) は同値である．

(a) 任意の非負実数 α に対して,

$$\forall \boldsymbol{p},\ \boldsymbol{q} \in S \implies (\boldsymbol{p} - \alpha\mathbf{1}) \vee \boldsymbol{q},\ \boldsymbol{p} \wedge (\boldsymbol{q} + \alpha\mathbf{1}) \in S. \tag{2.45}$$

(b) 任意の $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^n$ と任意の $\alpha, \beta \in \mathbb{R}$ に対して、

$$\boldsymbol{p} - \alpha\mathbf{1},\ \boldsymbol{q} - \beta\mathbf{1} \in S \implies (\boldsymbol{p} \vee \boldsymbol{q}) - (\alpha \vee \beta)\mathbf{1},\ (\boldsymbol{p} \wedge \boldsymbol{q}) - (\alpha \wedge \beta)\mathbf{1} \in S. \tag{2.46}$$

この (a), (b) のいずれもが，S が $\mathrm{L}^\natural$ 凸集合であるための必要十分条件である． ■

L 凸集合について、定義 (2.44) と L 凸関数の必要十分条件（定理 2.18）から次の定理が成り立つ。

定理 2.21 ([65, 定理 6.6(2)])**.** 閉集合 $S \subseteq \mathbb{R}^n$ が L 凸集合であるためには，任意の $\alpha \in \mathbb{R}$ に対して、2 つの条件

$$\forall \boldsymbol{p},\ \boldsymbol{q} \in S \Longrightarrow \boldsymbol{p} \vee \boldsymbol{q},\ \boldsymbol{p} \wedge \boldsymbol{q} \in S, \tag{2.47}$$
$$\forall \boldsymbol{p} \in S \Longrightarrow \boldsymbol{p} + \alpha\mathbf{1} \in S \tag{2.48}$$

がともに成り立つことが必要十分である． ■

$\mathrm{L}^\natural$ 凸集合と L 凸集合の特徴付けは、不等式を用いて表すこともできる。その表現は、定理 2.11 を $\mathbb{R}^n$ 上に拡張した、次の形になる。

定理 2.22 ([65, 定理 6.7])**.**

(1) 閉集合 $S \subseteq \mathbb{R}^n$ が $\mathrm{L}^\natural$ 凸集合であるためには, ある実数 $\alpha_i \in \mathbb{R} \cup \{-\infty\}$, $\beta_i \in \mathbb{R} \cup \{+\infty\}$, $\gamma_{ij} \in \mathbb{R} \cup \{+\infty\}$ $(i, j = 1, 2, \ldots, n; i \neq j)$ によって

$$\begin{aligned} S &= \{\boldsymbol{p} \in \mathbb{R}^n \mid \alpha_i \leq p_i \leq \beta_i,\ p_j - p_i \leq \gamma_{ij}\ (i, j = 1, 2, \ldots, n; i \neq j)\} \\ &= \{\boldsymbol{p} \in \mathbb{R}^n \mid \alpha_i \leq p_i \leq \beta_i,\ p_j - \gamma_{ij} \leq p_i \leq p_j + \gamma_{ji}\ (i, j = 1, 2, \ldots, n; i \neq j)\} \end{aligned}$$

と表されることが必要十分である．

(2) 閉集合 $S \subseteq \mathbb{R}^n$ が L 凸集合であるためには，ある実数 $\gamma_{ij} \in \mathbb{R} \cup \{+\infty\}$ $(i, j = 1, 2, \ldots, n; i \neq j)$ によって

$$\begin{aligned} S &= \{\boldsymbol{p} \in \mathbb{R}^n \mid p_j - p_i \leq \gamma_{ij} \ (i, j = 1, 2, \ldots, n; i \neq j)\} \\ &= \{\boldsymbol{p} \in \mathbb{R}^n \mid p_j - \gamma_{ij} \leq p_i \leq p_j + \gamma_{ji} \ (i, j = 1, 2, \ldots, n; i \neq j)\} \end{aligned}$$

と表されることが必要十分である． ■

$\mathrm{L}^\natural$ 凸集合と L 凸集合はどちらも凸多面体であるので、$\mathrm{L}^\natural$ 凸集合を **$\mathrm{L}^\natural$ 凸多面体**、L 凸集合を **L 凸多面体**とも呼ぶ。

2.2.9 連続変数の $\mathrm{L}^\natural$ 凸/L 凸関数の例

連続変数の $\mathrm{L}^\natural$ 凸/L 凸関数の基本的な例を挙げる．実効定義域の閉包については、$\mathrm{L}^\natural$ 凸関数の場合は $\mathrm{L}^\natural$ 凸多面体であること，L 凸関数の場合は L 凸多面体であることを仮定する。$\boldsymbol{p} = (p_1, p_2, \ldots, p_n) \in \mathbb{R}^n$ とする．

1 次関数

ベクトル $\boldsymbol{\alpha} \in \mathbb{R}^n$ と実数 $\beta \in \mathbb{R}$ によって定義される 1 次関数

$$G(\boldsymbol{p}) = \langle \boldsymbol{\alpha}, \boldsymbol{p} \rangle + \beta \tag{2.49}$$

は、実効定義域が $\mathrm{L}^\natural$ 凸多面体であれば $\mathrm{L}^\natural$ 凸関数であり、実効定義域が L 凸多面体であれば L 凸関数である。

2 次関数

実数係数の 2 次関数

$$G(\boldsymbol{p}) = \boldsymbol{p}^\top A \boldsymbol{p} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} p_i p_j \qquad (a_{ij} = a_{ji} \in \mathbb{R}) \tag{2.50}$$

は、係数行列 $A = (a_{ij})$ が

$$a_{ij} \leq 0 \quad (i \neq j), \qquad \sum_{j=1}^{n} a_{ij} \geq 0 \quad (i = 1, 2, \ldots, n) \tag{2.51}$$

を満たすとき $\mathrm{L}^\natural$ 凸関数であり，

$$a_{ij} \leq 0 \quad (i \neq j), \qquad \sum_{j=1}^{n} a_{ij} = 0 \quad (i = 1, 2, \ldots, n) \tag{2.52}$$

を満たすとき L 凸関数である [71]．これらの条件は、離散変数の場合の条件（定理 2.14、定理 2.15）と同じである。

分離凸関数

1 変数の凸関数 ψ_i $(i = 1, 2, \ldots, n)$ の和の形で定義される分離凸関数

$$G(\boldsymbol{p}) = \sum_{i=1}^{n} \psi_i(p_i) \tag{2.53}$$

は $\mathrm{L}^\natural$ 凸関数であり，1 変数の凸関数 ψ_{ij} $(i, j = 1, 2, \ldots, n; i \neq j)$ の和の形で定義される関数

$$G(\boldsymbol{p}) = \sum_{i \neq j} \psi_{ij}(p_i - p_j) \tag{2.54}$$

は L 凸関数である．さらに

$$G(\boldsymbol{p}) = \sum_{i=1}^{n} \psi_i(p_i) + \sum_{i \neq j} \psi_{ij}(p_i - p_j) \tag{2.55}$$

は $\mathrm{L}^\natural$ 凸関数である．

2.3　M 凸関数

ここでは $\mathrm{M}^\natural$ 凸関数と M 凸関数の定義と基本的な性質について述べる [57, 65, 66, 69, 75]。

2.3.1　$\mathrm{M}^\natural$ 凸関数

関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は、交換公理

($\mathbf{M}^\natural$-EXC[$\mathbb{Z}$])　任意の $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^n$ と任意の $i \in \mathrm{supp}^+(\boldsymbol{x} - \boldsymbol{y})$ に対して，ある $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y}) \cup \{0\}$ が存在して

$$f(\boldsymbol{x}) + f(\boldsymbol{y}) \geq f(\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j) + f(\boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_j) \tag{2.56}$$

を満たすときに **$\mathbf{M}^\natural$ 凸関数**と定義される。ただし，第 2.1.1 節に述べたように、一般にベクトル $\boldsymbol{v}$ に対して

$$\mathrm{supp}^+(\boldsymbol{v}) = \{i \mid v_i > 0\}, \qquad \mathrm{supp}^-(\boldsymbol{v}) = \{i \mid v_i < 0\}$$

であり，$\boldsymbol{\chi}_i$ は第 i 単位ベクトル $(i > 0)$ を表し，

$$\boldsymbol{\chi}_0 = \mathbf{0} = (0, 0, \ldots, 0) \in \mathbb{Z}^n$$

であるとする。

関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が**優モジュラ** (supermodular) であるというのは,

任意の $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^n$ に対して

$$f(\boldsymbol{x}) + f(\boldsymbol{y}) \leq f(\boldsymbol{x} \vee \boldsymbol{y}) + f(\boldsymbol{x} \wedge \boldsymbol{y}) \tag{2.57}$$

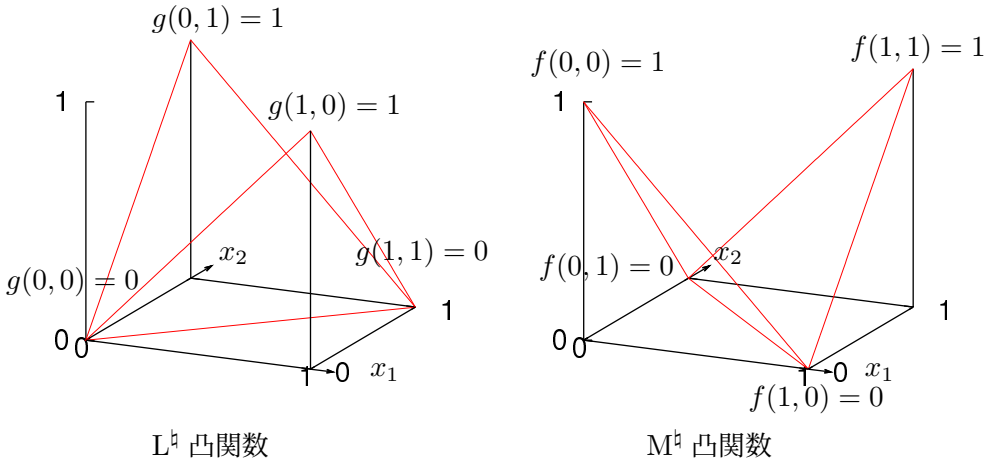


図 2.6. 凸拡張可能な劣モジュラ関数（左）と優モジュラ関数（右）

を満たすときであると定義される。ただし，$f(\boldsymbol{x})$ と $f(\boldsymbol{y})$ の少なくとも一方が $+\infty$ であるときには，不等式 (2.57) は満たされているものとみなす．なお、優モジュラの不等式 (2.57) は劣モジュラの不等式 (2.10) の不等号の向きを逆にしたものである。したがって、f が優モジュラであれば $-f$ は劣モジュラである。

f が M♮ 凸関数ならば，優モジュラの不等式 (2.57) を満たす。

定理 2.23 ([62, 定理 6.19])**.** M♮ 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は優モジュラである．ただし、逆は成り立たない。つまり、優モジュラである離散関数が M♮ 凸関数であるとは限らない。 ■

上の定理から、M♮ 凸関数の場合には、凸性と優モジュラ性に関連があるように見える。L♮ 凸関数の場合に、凸性と劣モジュラ性に関連があった（L♮ 凸性と並進劣モジュラ性が同値）こととは対照的である。

優モジュラ関数最小化（=劣モジュラ関数最大化）は困難なことが知られているが、その中で M♮ 凸関数は最小化が効率良く行える優モジュラ関数の部分族となっている。

次の例に示す通り、劣モジュラ性と凸拡張可能性は互いに独立な性質である。

例 2.5. 図 2.6 のように、$n = 2$, $\mathrm{dom}_{\mathbb{Z}}\, g = \{0,1\}^2$ の場合の L♮ 凸関数（左）と、$n = 2$, $\mathrm{dom}_{\mathbb{Z}}\, f = \{0,1\}^2$ の場合の M♮ 凸関数（右）を考える。$\boldsymbol{x} = (1,0)$, $\boldsymbol{y} = (0,1)$ とすると $\boldsymbol{x} \vee \boldsymbol{y} = (1,1)$, $\boldsymbol{x} \wedge \boldsymbol{y} = (0,0)$ であるので、左の例では $g(\boldsymbol{x}) = g(\boldsymbol{y}) = 1$, $g(\boldsymbol{x} \vee \boldsymbol{y}) = g(\boldsymbol{x} \wedge \boldsymbol{y}) = 0$ となり、g は劣モジュラの不等式 (2.10) を満たす。同様に右の例では $f(\boldsymbol{x}) = f(\boldsymbol{y}) = 0$, $f(\boldsymbol{x} \vee \boldsymbol{y}) = f(\boldsymbol{x} \wedge \boldsymbol{y}) = 1$ となり、f は優モジュラの不等式 (2.57) を満たす。したがって g と $-f$ は劣モジュラである。一方で g と f はともに凸拡張可能であるので、劣モジュラ性と凸拡張可能性は独立した性質であることがわかる。 ■

凸拡張可能性と最小性規準に関して，次の定理が成り立つ。

定理 2.24 ([62, 定理 6.42])**.** $\mathrm{M}^\natural$ 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は凸拡張可能である. ■

定理 2.25 ([62, 定理 6.26(2)])**.** 関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が $\mathrm{M}^\natural$ 凸関数のとき，$\boldsymbol{x} \in \mathrm{dom}_{\mathbb{Z}} f$ が f の最小点であるためには，任意の $i, j \in \{0, 1, \ldots, n\}$ に対して

$$f(\boldsymbol{x}) \leq f(\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j)$$

となることが必要十分である. ■

$\mathrm{L}^\natural$ 凸関数にも同様の定理（定理 2.4，定理 2.5）が存在するが、$\mathrm{M}^\natural$ 凸関数においても、離散関数における「凸性」を定義するにあたって、定義の正当性を連続変数の凸関数の性質との類似性に求めるならば、この 2 つの定理の存在は重要である。前者、すなわち凸拡張可能であることは、$\mathrm{M}^\natural$ 凸関数が連続変数の凸関数に対応づけられることを示す。後者、すなわち最小性規準を持つことは、連続変数の凸関数の重要な性質の 1 つである、局所最小性と大域最小性が一致するという性質が、$\mathrm{M}^\natural$ 凸関数で実現されていることを示す。このような観点から、2 つの定理の存在によって、$\mathrm{M}^\natural$ 凸関数は「離散凸関数」としての資格を有していると言える。

2.3.2　M 凸関数

M 凸関数は、$\mathrm{M}^\natural$ 凸関数の特殊形で、交換公理における j を選ぶ範囲を $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y})$ に限定した（0 を含めない）ものとして定義される。すなわち，$f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が **M 凸関数**であるためには，次の交換公理

> **(M-EXC[$\mathbb{Z}$])**　任意の $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^n$ と任意の $i \in \mathrm{supp}^+(\boldsymbol{x} - \boldsymbol{y})$ に対して，ある $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y})$ が存在して
>
> $$f(\boldsymbol{x}) + f(\boldsymbol{y}) \geq f(\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j) + f(\boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_j) \tag{2.58}$$

を満たすことと定義される。この条件から，M 凸関数 f の実効定義域は成分和が一定の超平面の上に乗っていること，すなわち，ある整数 r に対して

$$\mathrm{dom}_{\mathbb{Z}} f \subseteq \{\boldsymbol{x} \in \mathbb{Z}^n \mid \sum_{i=1}^{n} x_i = r\} \tag{2.59}$$

が成り立つことが導かれる.

定義により，M 凸関数は $\mathrm{M}^\natural$ 凸関数の特殊ケースである。しかし両者は本質的には等価な概念であり，次のように互いに変換しあうことができる。n 変数の $\mathrm{M}^\natural$ 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が与えられたとき，$n+1$ 変数の関数 $\tilde{f} : \mathbb{Z}^{n+1} \to \mathbb{R} \cup \{+\infty\}$ を

$$\tilde{f}(\boldsymbol{x}, x_{n+1}) = \begin{cases} f(\boldsymbol{x}) & (x_{n+1} = -\sum_{i=1}^{n} x_i \text{ のとき}) \\ +\infty & (\text{その他}) \end{cases} \qquad (\boldsymbol{x} \in \mathbb{Z}^n, x_{n+1} \in \mathbb{Z}) \tag{2.60}$$

によって定義すると，$\tilde{f}$ は M 凸関数になる．逆に，n 変数の M 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が与えられたとき，式 (2.59) の $r \in \mathbb{Z}$ を用いて、$n-1$ 変数の関数 $\hat{f} : \mathbb{Z}^{n-1} \to \mathbb{R} \cup \{+\infty\}$ を

$$\hat{f}(\boldsymbol{y}) = f(\boldsymbol{y}, r - \sum_{i=1}^{n-1} y_i) \qquad (\boldsymbol{y} \in \mathbb{Z}^{n-1}) \tag{2.61}$$

によって定義すると，$\hat{f}$ は $\mathrm{M}^\natural$ 凸関数になる．このように両者は相互に変換ができる。

ただし、両者が 1 対 1 対応をしているわけではなく、M 凸関数には式 (2.59) の r の自由度があることに注意されたい。つまり、$\mathrm{M}^\natural$ 凸関数から式 (2.60) によって作り出した M 凸関数では常に $r = 0$ であり、逆に r の異なる M 凸関数から同じ $\mathrm{M}^\natural$ 凸関数が作り出される。

このような $\mathrm{M}^\natural$ 凸関数と M 凸関数の関係（等価ではあるが、対応には自由度がある）は、$\mathrm{L}^\natural$ 凸関数と L 凸関数の関係と同様である。

凸拡張可能性と最小性規準に関して，次の定理が成り立つ．この定理は、$\mathrm{M}^\natural$ 凸関数に対する定理（定理 2.24，定理 2.25）と本質的に等価な内容である．

定理 2.26 ([62, 定理 6.42])**.** M 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は凸拡張可能である． ■

定理 2.27 ([62, 定理 6.26(1)])**.** 関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が M 凸関数のとき，$\boldsymbol{x} \in \mathrm{dom}_{\mathbb{Z}} f$ が f の最小点であるためには，任意の $i, j \in \{1, 2, \ldots, n\}$ に対して

$$f(\boldsymbol{x}) \leq f(\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j)$$

となることが必要十分である． ■

2.3.3 $\mathrm{M}^\natural$ 凸集合と M 凸集合

$\mathrm{M}^\natural$ 凸関数と M 凸関数に対応する離散凸集合について説明する。

集合 $S \subseteq \mathbb{Z}^n$ が **$\mathrm{M}^\natural$ 凸集合**であるというのは，その標示関数 $\delta_S : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が $\mathrm{M}^\natural$ 凸関数であるときであると定義される。同様に，標示関数 δ_S が M 凸関数のとき，集合 S は **M 凸集合**と定義される．すなわち，

$$S \text{ が } \mathrm{M}^\natural \text{ 凸集合} \iff \delta_S \text{ が } \mathrm{M}^\natural \text{ 凸関数}, \tag{2.62}$$
$$S \text{ が M 凸集合} \iff \delta_S \text{ が M 凸関数} \tag{2.63}$$

である。

$\mathrm{M}^\natural$ 凸集合について、定義 (2.62) と $\mathrm{M}^\natural$ 凸関数の交換公理 ($\mathrm{M}^\natural$-EXC[$\mathbb{Z}$]) から次の定理が成り立つ。まず、集合に関する交換公理を定義する。

> **($\mathrm{B}^\natural$-EXC[$\mathbb{Z}$])** 任意の $\boldsymbol{x}, \boldsymbol{y} \in S$ と任意の $i \in \mathrm{supp}^+(\boldsymbol{x} - \boldsymbol{y})$ に対して，ある $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y}) \cup \{0\}$ が存在して
>
> $$\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j \in S \quad \text{かつ} \quad \boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_j \in S \tag{2.64}$$
>
> を満たす。

定理 2.28 ([65, 定理 5.4])**.** 集合 $S \subseteq \mathbb{Z}^n$ が $\mathrm{M}^\natural$ 凸集合であるためには，上記の交換公理 ($\mathrm{B}^\natural$-EXC[$\mathbb{Z}$]) を満たすことが必要十分である． ■

例 2.6. 一般の n に対して，整数区間 $[\boldsymbol{\ell}, \boldsymbol{u}]_{\mathbb{Z}}$（ただし，$\boldsymbol{\ell} \in \{\mathbb{Z} \cup \{-\infty\}\}^n$, $\boldsymbol{u} \in \{\mathbb{Z} \cup \{+\infty\}\}^n$）は $\mathrm{M}^\natural$ 凸集合である． ■

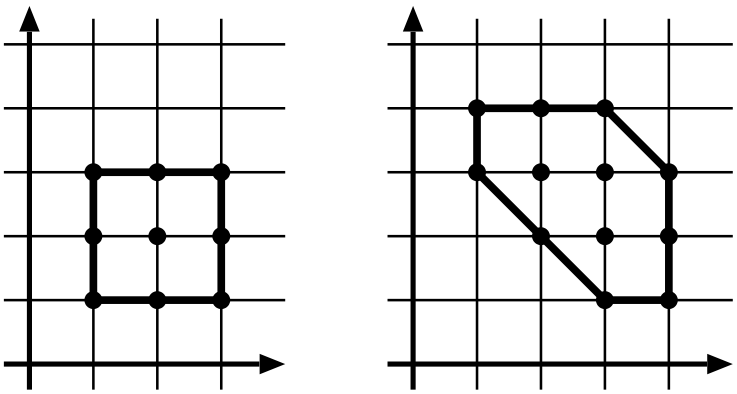


図 2.7. $\mathrm{M}^\natural$ 凸集合の例

M 凸集合について、定義 (2.63) と M 凸関数の交換公理 (M-EXC[$\mathbb{Z}$]) から次の定理が成り立つ。まず、$\mathrm{M}^\natural$ 凸集合の交換公理 ($\mathrm{B}^\natural$-EXC[$\mathbb{Z}$]) において，j を選ぶ範囲を $j \in \mathrm{supp}^-(\boldsymbol{x}-\boldsymbol{y})$ に限定した次の交換公理を定義する。

> **(B-EXC[$\mathbb{Z}$])**　任意の $\boldsymbol{x}, \boldsymbol{y} \in S$ と任意の $i \in \mathrm{supp}^+(\boldsymbol{x}-\boldsymbol{y})$ に対して，ある $j \in \mathrm{supp}^-(\boldsymbol{x}-\boldsymbol{y})$ が存在して
>
> $$\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j \in S \quad \text{かつ} \quad \boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_j \in S \tag{2.65}$$
>
> を満たす。

定理 2.29 ([65, 定理 5.6])**.** 集合 $S \subseteq \mathbb{Z}^n$ が M 凸集合であるためには，上記の交換公理 (B-EXC[$\mathbb{Z}$]) を満たすことが必要十分である．■

交換公理 (B-EXC[$\mathbb{Z}$]) から，M 凸集合 S は成分和が一定の超平面の上に乗っていること，すなわち，ある整数 r に対して

$$S \subseteq \{\boldsymbol{x} \in \mathbb{Z}^n \mid \sum_{i=1}^{n} x_i = r\}$$

が成り立つことがわかる。

例 2.7. $n = 2$ の場合の M 凸集合は，ある $l \in \mathbb{Z} \cup \{-\infty\}, u \in \mathbb{Z} \cup \{+\infty\}, r \in \mathbb{Z}$ によって

$$\{(x_1, x_2) \in \mathbb{Z}^2 \mid x_1 + x_2 = r, l \le x_1 \le u\}$$

と表される集合である．■

$\mathrm{M}^\natural$ 凸集合と M 凸集合は簡単な不等式で記述できる。不等式の係数は 1 と 0 のどちらかであり，右辺には $N = \{1, 2, \ldots, n\}$ 上の劣モジュラ集合関数が現れる．第 2.1.1 節に述べたよ

うに、ベクトル $\boldsymbol{x} \in \mathbb{Z}^n$ と部分集合 $X \subseteq N$ に対して,

$$x(X) = \sum_{i \in X} x_i \tag{2.66}$$

とする。$\mathrm{M}^\natural$ 凸集合は劣モジュラ関数 ρ と優モジュラ関数 μ の組によって記述され、M 凸集合は劣モジュラ関数 ρ によって記述される.

定理 2.30 ([65, 定理 5.9, 定理 5.8])**.**

(1) 集合 $S \subseteq \mathbb{Z}^n$ が $\mathrm{M}^\natural$ 凸集合であるためには，ある整数値劣モジュラ集合関数 $\rho : 2^N \to \mathbb{Z} \cup \{+\infty\}$ と整数値優モジュラ集合関数 $\mu : 2^N \to \mathbb{Z} \cup \{-\infty\}$ が存在して

$$S = \{\boldsymbol{x} \in \mathbb{Z}^n \mid \mu(X) \leq x(X) \leq \rho(X)\ (\forall X \subseteq N)\}$$

を満たすことが必要十分である．ただし $\rho(\emptyset) = \mu(\emptyset) = 0$ であって,

$$X, Y \subseteq N \implies \rho(X) - \mu(Y) \geq \rho(X \setminus Y) - \mu(Y \setminus X)$$

を満たすとする.

(2) 集合 $S \subseteq \mathbb{Z}^n$ が M 凸集合であるためには，ある整数値劣モジュラ集合関数 $\rho : 2^N \to \mathbb{Z} \cup \{+\infty\}$ が存在して

$$S = \{\boldsymbol{x} \in \mathbb{Z}^n \mid x(X) \leq \rho(X)\ (\forall X \subset N),\ x(N) = \rho(N)\}$$

を満たすことが必要十分である．ただし $\rho(\emptyset) = 0$, $\rho(N) < +\infty$ とする． ■

整数値劣モジュラ集合関数 $\rho : 2^N \to \mathbb{Z} \cup \{+\infty\}$ によって定められる多面体

$$B(\rho) = \{\boldsymbol{x} \in \mathbb{Z}^n \mid x(X) \leq \rho(X)\ (\forall X \subset N),\ x(N) = \rho(N)\} \tag{2.67}$$

は**基多面体**と呼ばれるが、定理 2.30(2) で示したように、M 凸集合と本質的に同じものである。

なお，定理 2.30 より，$\mathrm{M}^\natural$ 凸集合，M 凸集合にはくぼみがない、すなわち

$$S = \overline{S} \cap \mathbb{Z}^n$$

が成り立つことが分かる.

連続変数の凸関数の実効定義域と最小化集合は凸集合であるが、それと同様に、離散 $\mathrm{M}^\natural$ 凸/M 凸関数の場合は次のようになる。

定理 2.31 ([62, 命題 6.7])**.**

(1) $\mathrm{M}^\natural$ 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の実効定義域 $\mathrm{dom}_{\mathbb{Z}} f$ は $\mathrm{M}^\natural$ 凸集合である.

(2) M 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の実効定義域 $\mathrm{dom}_{\mathbb{Z}} f$ は M 凸集合である.

定理 2.32 ([62, 命題 6.29])**.**

(1) $\mathrm{M}^\natural$ 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の最小化集合 $\mathrm{argmin}_{\mathbb{Z}} f$ は $\mathrm{M}^\natural$ 凸集合である.

(2) M 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の最小化集合 $\mathrm{argmin}_{\mathbb{Z}} f$ は M 凸集合である.

2.3.4　2 次関数

2 次関数が $\mathrm{M}^\natural$ 凸関数であるための条件を考える．一般の n 変数（離散変数）の 2 次関数は

$$f(\boldsymbol{x}) = \boldsymbol{x}^\top A\boldsymbol{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_i x_j \qquad (\boldsymbol{x} = (x_i)_{i=1}^n \in \mathbb{Z}^n) \tag{2.68}$$

とあらわせる．ただし，係数行列 A の成分 a_{ij} は実数である。ここで任意の i, j について $a_{ij} = a_{ji}$ であるとしてよい。すなわち A は対称行列としてよい。

関数 $f(\boldsymbol{x})$ が $\mathrm{M}^\natural$ 凸であることを，係数行列 A に関する条件として表すと，次の定理のようになる．

定理 2.33 ([28, 65])**.** 式 (2.68) の 2 次関数 f が $\mathrm{dom}_{\mathbb{Z}} f = \mathbb{Z}^n$ の仮定の下で $\mathrm{M}^\natural$ 凸関数であるための必要十分条件は,

$$\{i, j\} \cap \{k\} = \emptyset \Longrightarrow a_{ij} \geq \min(a_{ik}, a_{jk}), \tag{2.69}$$

$$\text{任意の } (i, j) \text{ に対して} \qquad a_{ij} \geq 0 \tag{2.70}$$

で与えられる．この条件は $i = j$ の場合を含む。 ■

上の条件 (2.69), (2.70) を $n = 3$ の場合にあてはめると、対角要素については

$$\begin{array}{l} a_{11} \geq a_{12} = a_{21} \geq 0, \\ a_{11} \geq a_{13} = a_{31} \geq 0, \\ a_{22} \geq a_{21} = a_{12} \geq 0, \\ a_{22} \geq a_{23} = a_{32} \geq 0, \\ a_{33} \geq a_{31} = a_{13} \geq 0, \\ a_{33} \geq a_{32} = a_{23} \geq 0 \end{array}$$

のように、同じ行（または列）の非対角要素よりも大きく、非対角要素 a_{12}, a_{23}, a_{31} については

$$\begin{array}{l} a_{12} \geq a_{23} = a_{31} \text{ または} \\ a_{23} \geq a_{31} = a_{12} \text{ または} \\ a_{31} \geq a_{12} = a_{23} \end{array}$$

のように、3 つの値が等しいか、それとも最小値をとるものが 2 つ現れるかのどちらかになる。

$\mathrm{M}^\natural$ 凸 2 次関数は、層族を用いて平方和分解ができる．層族とは、次の性質をもつ集合族である。集合 $\{1, 2, \ldots, n\}$ の部分集合の族 $\mathcal{T}$ が、条件

$$X, Y \in \mathcal{T} \Longrightarrow X \cap Y = \emptyset \text{ または } X \subseteq Y \text{ または } X \supseteq Y \tag{2.71}$$

を満たすとき、集合族 $\mathcal{T}$ を層族という。

定理 2.34 ([28])**.** 2 次関数 $f(\boldsymbol{x})$ が $\mathrm{dom}_{\mathbb{Z}} f = \mathbb{Z}^n$ の仮定の下で $\mathrm{M}^\natural$ 凸関数であるためには, ある層族 $\mathcal{T}$ と正の実数 γ_X $(X \in \mathcal{T})$ によって

$$f(\boldsymbol{x}) = \sum_{X \in \mathcal{T}} \gamma_X (\sum_{i \in X} x_i)^2 \tag{2.72}$$

と表現されることが必要十分である．また，この表現は一意に定まる． ■

例 2.8. 2 次関数

$$f(\boldsymbol{x}) = (x_1 + x_2)^2 + (x_2 + x_3)^2 + (x_3 + x_1)^2$$

は $\mathrm{M}^\natural$ 凸関数である．$\{\{1,2\},\{2,3\},\{3,1\}\}$ は層族ではないが、

$$f(\boldsymbol{x}) = (x_1 + x_2 + x_3)^2 + x_1^2 + x_2^2 + x_3^2$$

と変形することができるので、$f(\boldsymbol{x})$ は層族 $\mathcal{T} = \{\{1,2,3\},\{1\},\{2\},\{3\}\}$ を用いて式 (2.72) の形式で表現できる。 ■

例 2.9. 2 次関数

$$f(\boldsymbol{x}) = (x_1 + x_2)^2 + (x_2 + x_3)^2 + (x_3 + x_4)^2 + (x_4 + x_1)^2$$

は $\mathrm{M}^\natural$ 凸関数ではない．これは式 (2.72) を満たすように見えるが、$\mathcal{T} = \{\{1,2\},\{2,3\},\{3,4\},\{4,1\}\}$ は層族ではない．また、別の層族の表現も存在しない。 ■

例 2.10. $a_1 > a_2 > \cdots > a_n > 0$ のとき，

$$a_{ij} = \min(a_i, a_j) \qquad (1 \le i, j \le n) \tag{2.73}$$

で定義される行列 $A = (a_{ij})$ は，条件 (2.69), (2.70) を満たす．例えば，$n = 5$ のとき，

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a_2 & a_2 & a_3 & a_4 & a_5 \\ a_3 & a_3 & a_3 & a_4 & a_5 \\ a_4 & a_4 & a_4 & a_4 & a_5 \\ a_5 & a_5 & a_5 & a_5 & a_5 \end{bmatrix} \tag{2.74}$$

であり，定理 2.34 における表現式 (2.72) は

$$\begin{aligned} f(\boldsymbol{x}) = {} & (a_1 - a_2){x_1}^2 + (a_2 - a_3)(x_1 + x_2)^2 + (a_3 - a_4)(x_1 + x_2 + x_3)^2 \\ & + (a_4 - a_5)(x_1 + x_2 + x_3 + x_4)^2 + a_5(x_1 + x_2 + x_3 + x_4 + x_5)^2 \end{aligned}$$

となる． ■

2 次関数が M 凸関数になる条件は次の定理で与えられる．これは $\mathrm{M}^\natural$ 凸関数の場合の定理 2.33 に対応する．

定理 2.35 ([28])**.** 式 (2.68) の 2 次関数 f が M 凸関数であるための必要十分条件は，

$$\{i,j\} \cap \{k,l\} = \emptyset \implies a_{ij} + a_{kl} \ge \min(a_{ik} + a_{jl}, a_{il} + a_{jk}) \tag{2.75}$$

で与えられる（$i = j$ や $k = l$ の場合を含む）．ただし，ある整数 r に対して

$$\mathrm{dom}_{\mathbb{Z}}\, f = \{\boldsymbol{x} \in \mathbb{Z}^n \mid \sum_{i=1}^{n} x_i = r\}$$

とする．

2.3.5　$\mathrm{M}^\natural$ 凸/M 凸関数の例

$\mathrm{M}^\natural$ 凸/M 凸関数の基本的な例を挙げる．実効定義域については、$\mathrm{M}^\natural$ 凸関数の場合は $\mathrm{M}^\natural$ 凸集合であること、M 凸関数の場合は M 凸集合であることを仮定する（定理 2.31）。$\mathbb{Z}^n$ の全体は $\mathrm{M}^\natural$ 凸集合ではあるが、M 凸集合ではないことに注意する（L 凸集合とは状況が異なる）。$\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{Z}^n$ とする．

1 次関数

ベクトル $\boldsymbol{p} \in \mathbb{R}^n$ と実数 $\alpha \in \mathbb{R}$ によって定義される 1 次関数

$$f(\boldsymbol{x}) = \alpha + \langle \boldsymbol{p}, \boldsymbol{x} \rangle \tag{2.76}$$

は、実効定義域が $\mathrm{M}^\natural$ 凸集合であれば $\mathrm{M}^\natural$ 凸関数であり、実効定義域が M 凸集合であれば M 凸関数である。

2 次関数

実数係数の 2 次関数

$$f(\boldsymbol{x}) = \boldsymbol{x}^\top A \boldsymbol{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j \qquad (a_{ij} = a_{ji} \in \mathbb{R}) \tag{2.77}$$

を考える。定理 2.33 に示したように、係数行列 $A = (a_{ij})$ が

$$\{i, j\} \cap \{k\} = \emptyset \Longrightarrow a_{ij} \geq \min(a_{ik}, a_{jk}), \tag{2.78}$$

$$\text{任意の } (i, j) \text{ に対して} \qquad a_{ij} \geq 0 \tag{2.79}$$

を満たすとき，$f(\boldsymbol{x})$ は $\mathrm{M}^\natural$ 凸関数である．ただし $\mathrm{dom}_{\mathbb{Z}}\, f = \mathbb{Z}^n$ とする [28, 65]．また，

$$\{i, j\} \cap \{k, l\} = \emptyset \implies a_{ij} + a_{kl} \geq \min(a_{ik} + a_{jl}, a_{il} + a_{jk}) \tag{2.80}$$

のとき，$f(\boldsymbol{x})$ は M 凸関数である（定理 2.35）．ただし，ある $r \in \mathbb{Z}$ に対して

$$\mathrm{dom}_{\mathbb{Z}}\, f = \{\boldsymbol{x} \in \mathbb{Z}^n \mid \sum_{i=1}^{n} x_i = r\}$$

とする [65].

分離凸関数

1 変数の凸関数 φ_i $(i = 1, 2, \ldots, n)$ によって定義される分離凸関数

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \varphi_i(x_i) \tag{2.81}$$

は、実効定義域が $\mathrm{M}^\natural$ 凸集合であれば $\mathrm{M}^\natural$ 凸関数であり、実効定義域が M 凸集合であれば M 凸関数である。

さらに，φ_0 も 1 変数の凸関数とするとき，**擬分離凸関数**

$$f(\boldsymbol{x}) = \varphi_0(\sum_{i=1}^{n} x_i) + \sum_{i=1}^{n} \varphi_i(x_i) \tag{2.82}$$

は，$\mathbb{Z}^n$ 上で $\mathrm{M}^\natural$ 凸関数である．

層凸関数

$\{1, 2, \ldots, n\}$ の部分集合からなる層族 $\mathcal{T}$ と，各 $X \in \mathcal{T}$ に対応する 1 変数の凸関数 φ_X から作られる

$$f(\boldsymbol{x}) = \sum_{X \in \mathcal{T}} \varphi_X(\sum_{i \in X} x_i) \tag{2.83}$$

の形の関数を層凸関数と呼ぶ．層凸関数は $\mathrm{M}^\natural$ 凸関数である．例えば，

$$f(\boldsymbol{x}) = \varphi_1(x_1 + \cdots + x_n) + \varphi_2(x_2 + \cdots + x_n) + \cdots + \varphi_n(x_n) \tag{2.84}$$

は $\mathrm{M}^\natural$ 凸関数である．これは $\mathcal{T} = \{\{1, 2, \ldots, n\}, \{2, 3, \ldots, n\}, \ldots, \{n\}\}$ の場合である．また，$\mathcal{T} = \{\{1, 2, \ldots, n\}, \{1\}, \{2\}, \ldots, \{n\}\}$ の場合が擬分離凸関数 (2.82) にあたる．

一般の $\mathrm{M}^\natural$ 凸関数は層凸関数とは限らないが、2 次関数に限れば，すべての $\mathrm{M}^\natural$ 凸関数は層凸関数である（定理 2.34）．

その他

M 凸関数や M 凸集合は、次のような場面にも現れる。最小木問題の目的関数は M 凸関数である。2 部グラフ上のマッチング問題や最小費用流問題の最適値は $\mathrm{M}^\natural$ 凸関数である。行列やマトロイドの独立集合族は $\mathrm{M}^\natural$ 凸集合，基族は M 凸集合である．

2.3.6 連続変数の $\mathrm{M}^\natural$ 凸/M 凸関数

離散 $\mathrm{M}^\natural$ 凸/M 凸関数の概念を、連続関数上に拡張する [70, 73, 74]。拡張の方法は、$\mathrm{L}^\natural$ 凸/L 凸関数のときとは異なり、変数の定義域が単純に連続になるだけではない場面があることに注意する。

連続変数の $\mathrm{M}^\natural$ 凸関数

凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が

($\mathbf{M}^\natural$-EXC[$\mathbb{R}$]) 任意の $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ と任意の $i \in \mathrm{supp}^+(\boldsymbol{x} - \boldsymbol{y})$ に対して，ある $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y}) \cup \{0\}$ と $\alpha_0 \in \mathbb{R}_+$ が存在し，すべての $\alpha \in [0, \alpha_0]_{\mathbb{R}}$ に対して

$$F(\boldsymbol{x}) + F(\boldsymbol{y}) \geq F(\boldsymbol{x} - \alpha(\boldsymbol{\chi}_i - \boldsymbol{\chi}_j)) + F(\boldsymbol{y} + \alpha(\boldsymbol{\chi}_i - \boldsymbol{\chi}_j)) \tag{2.85}$$

の条件を満たすとき，関数 F は **$\mathbf{M}^\natural$ 凸関数**と定義される。この条件は、連続変数の $\mathrm{M}^\natural$ 凸関数の交換公理と呼ばれる。

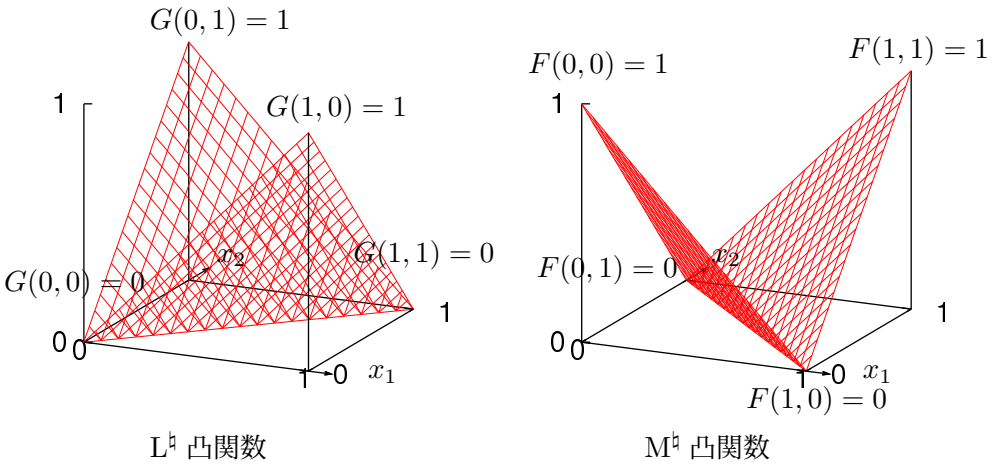


図 2.8. 連続変数の劣モジュラ関数（左）と優モジュラ関数（右）

離散変数の場合と同様に，M♮ 凸関数 F は優モジュラであり，不等式

$$F(\boldsymbol{x}) + F(\boldsymbol{y}) \leq F(\boldsymbol{x} \vee \boldsymbol{y}) + F(\boldsymbol{x} \wedge \boldsymbol{y}) \qquad (\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n) \tag{2.86}$$

が成り立つ．この優モジュラ性を表す不等式は、変数 $\boldsymbol{x}, \boldsymbol{y}$ が連続変数になっていることを除いて、離散変数の関数に関する優モジュラ性 (2.57) の不等式と同じである。

定理 2.36 ([62, 定理 6.51])**.** M♮ 凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ は優モジュラである．ただし、逆は成り立たない。つまり、優モジュラである連続変数の関数が M♮ 凸関数になるとは限らない。 ■

例 2.11. 離散変数の場合と同様に，凸性と劣モジュラ性は独立した性質である。$n = 2$ の場合の例を図 2.8 に示す。これは図 2.6 の凸拡張でもある。

$G(x_1, x_2) = |x_1 - x_2|$ は L♮ 凸関数であり、劣モジュラである。

$F(x_1, x_2) = |x_1 + x_2 - 1|$ は M♮ 凸関数であり、優モジュラである。 ■

連続変数の M 凸関数

M♮ 凸関数の交換公理 (M♮-EXC[$\mathbb{R}$]) において，j を選ぶ範囲を $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y})$ に狭めた条件を満たす F を **M 凸関数**という．すなわち，$F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が M 凸関数であるというのは，F は凸関数であって、交換公理

(M-EXC[$\mathbb{R}$])　任意の $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ と任意の $i \in \mathrm{supp}^+(\boldsymbol{x} - \boldsymbol{y})$ に対して，ある $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y})$ と $\alpha_0 \in \mathbb{R}_+$ が存在し，すべての $\alpha \in [0, \alpha_0]_{\mathbb{R}}$ に対して

$$F(\boldsymbol{x}) + F(\boldsymbol{y}) \geq F(\boldsymbol{x} - \alpha(\boldsymbol{\chi}_i - \boldsymbol{\chi}_j)) + F(\boldsymbol{y} + \alpha(\boldsymbol{\chi}_i - \boldsymbol{\chi}_j)) \tag{2.87}$$

を満たすときである．この条件から，M 凸関数 F の実効定義域は成分和が一定の超平面の上に乗っていること，すなわち，ある実数 r に対して

$$\mathrm{dom}_{\mathbb{R}} F \subseteq \{\boldsymbol{x} \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i = r\} \tag{2.88}$$

が成り立つことがわかる。

定義により，M 凸関数は $\mathrm{M}^\natural$ 凸関数の特殊ケースである。しかし両者は本質的には等価な概念であり，次のように互いに変換しあうことができる。n 変数の $\mathrm{M}^\natural$ 凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が与えられたとき，$n+1$ 変数の関数 $\tilde{F} : \mathbb{R}^{n+1} \to \mathbb{R} \cup \{+\infty\}$ を

$$\tilde{F}(\boldsymbol{x}, x_{n+1}) = \begin{cases} F(\boldsymbol{x}) & (x_{n+1} = -\sum_{i=1}^{n} x_i) \\ +\infty & (\text{その他}) \end{cases} \qquad (\boldsymbol{x} \in \mathbb{R}^n, x_{n+1} \in \mathbb{R}) \tag{2.89}$$

によって定義すると，$\tilde{F}$ は M 凸関数になる．逆に，n 変数の M 凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が与えられたとき，式 (2.88) を満たす $r \in \mathbb{R}$ を用いて、$n-1$ 変数の関数 $\hat{F} : \mathbb{R}^{n-1} \to \mathbb{R} \cup \{+\infty\}$ を

$$\hat{F}(\boldsymbol{y}) = F(\boldsymbol{y}, r - \sum_{i=1}^{n-1} y_i) \qquad (\boldsymbol{y} \in \mathbb{R}^{n-1}) \tag{2.90}$$

によって定義すると，$\hat{F}$ は $\mathrm{M}^\natural$ 凸関数になる．このような $\mathrm{M}^\natural$ 凸関数と M 凸関数が等価であるという構造は、離散変数と連続変数のどちらにも共通であり、$\mathrm{L}^\natural$ 凸関数/L 凸関数の場合にも共通である。

最小性規準

$\mathrm{M}^\natural$ 凸/M 凸関数は凸関数であるから，局所最小解が大域最小解となる。局所最小解であることは，式 (2.4) の「$\boldsymbol{x}$ は F の極小点 $\iff$ 任意の $\boldsymbol{d}$ に対して $F'(\boldsymbol{x}; \boldsymbol{d}) \geq 0$」によって確認できるが，方向の離散性があるので，特定の方向 $\boldsymbol{d}$ への方向微分を調べればよい．

定理 2.37 ([62, 定理 6.52])**.**

(1) 関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が $\mathrm{M}^\natural$ 凸関数のとき，$\boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F$ が F の最小点であるためには，任意の $i, j \in \{0, 1, \ldots, n\}$ に対して

$$F'(\boldsymbol{x}; -\boldsymbol{\chi}_i + \boldsymbol{\chi}_j) \geq 0$$

となることが必要十分である．

(2) 関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が M 凸関数のとき，$\boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F$ が F の最小点であるためには，任意の $i, j \in \{1, 2, \ldots, n\}$ に対して

$$F'(\boldsymbol{x}; -\boldsymbol{\chi}_i + \boldsymbol{\chi}_j) \geq 0$$

となることが必要十分である． ■

2.3.7　連続変数の M$^\natural$ 凸/M 凸集合

連続変数の M$^\natural$ 凸/M 凸関数に対応する凸集合について説明する。

集合 $S \subseteq \mathbb{R}^n$ が **M$^\natural$ 凸集合**であるというのは，その標示関数 $\delta_S : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が M$^\natural$ 凸関数であるときであると定義される。同様に，標示関数 δ_S が M 凸関数のとき，集合 S は **M 凸集合**と定義される．すなわち，

$$S \text{ が M}^\natural \text{ 凸集合} \iff \delta_S \text{ が M}^\natural \text{ 凸関数}, \tag{2.91}$$

$$S \text{ が M 凸集合} \iff \delta_S \text{ が M 凸関数} \tag{2.92}$$

である。式 (2.91) と式 (2.92) で δ_S が閉真凸関数ならば、M$^\natural$ 凸/M 凸集合は閉集合である。

M$^\natural$ 凸集合について、定義 (2.91) と M$^\natural$ 凸関数の交換公理 (M$^\natural$-EXC[$\mathbb{R}$]) から次の定理が成り立つ。まず、集合に関する交換公理を定義する。

> **(B$^\natural$-EXC[$\mathbb{R}$])**　任意の $\boldsymbol{x}, \boldsymbol{y} \in S$ と任意の $i \in \mathrm{supp}^+(\boldsymbol{x} - \boldsymbol{y})$ に対して，ある $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y}) \cup \{0\}$ と $\alpha_0 \in \mathbb{R}_+$ が存在し、すべての $\alpha \in [0, \alpha_0]_{\mathbb{R}}$ に対して
>
> $$\boldsymbol{x} - \alpha(\boldsymbol{\chi}_i - \boldsymbol{\chi}_j) \in S \quad \text{かつ} \quad \boldsymbol{y} + \alpha(\boldsymbol{\chi}_i - \boldsymbol{\chi}_j) \in S \tag{2.93}$$
>
> を満たす。

定理 2.38 ([65, 定理 6.23(1)])**.** 閉集合 $S \subseteq \mathbb{R}^n$ が M$^\natural$ 凸集合であるためには，上記の交換公理 (B$^\natural$-EXC[$\mathbb{R}$]) を満たすことが必要十分である． ■

M 凸集合について、定義 (2.92) と M 凸関数の交換公理 (M-EXC[$\mathbb{R}$]) から次の定理が成り立つ。まず、M$^\natural$ 凸集合の交換公理 (B$^\natural$-EXC[$\mathbb{R}$]) において，j を選ぶ範囲を $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y})$ に限定した次の交換公理を定義する。

> **(B-EXC[$\mathbb{R}$])**　任意の $\boldsymbol{x}, \boldsymbol{y} \in S$ と任意の $i \in \mathrm{supp}^+(\boldsymbol{x} - \boldsymbol{y})$ に対して，ある $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y})$ と $\alpha_0 \in \mathbb{R}_+$ が存在し、すべての $\alpha \in [0, \alpha_0]_{\mathbb{R}}$ に対して
>
> $$\boldsymbol{x} - \alpha(\boldsymbol{\chi}_i - \boldsymbol{\chi}_j) \in S \quad \text{かつ} \quad \boldsymbol{y} + \alpha(\boldsymbol{\chi}_i - \boldsymbol{\chi}_j) \in S \tag{2.94}$$
>
> を満たす。

定理 2.39 ([65, 定理 6.23(2)])**.** 閉集合 $S \subseteq \mathbb{R}^n$ が M 凸集合であるためには，上記の交換公理 (B-EXC[$\mathbb{R}$]) を満たすことが必要十分である． ■

M$^\natural$ 凸集合と M 凸集合の特徴付けは、不等式を用いて表すこともできる。その表現は、定理 2.30 を $\mathbb{R}^n$ 上に拡張した、次の形になる。

定理 2.40 ([65, 定理 6.24])**.**

(1) 集合 $S \subseteq \mathbb{R}^n$ が M$^\natural$ 凸集合であるためには, ある劣モジュラ集合関数 $\rho : 2^N \to \mathbb{R} \cup \{+\infty\}$ と優モジュラ集合関数 $\mu : 2^N \to \mathbb{R} \cup \{-\infty\}$ が存在して

$$S = \{\boldsymbol{x} \in \mathbb{R}^n \mid \mu(X) \le x(X) \le \rho(X)\ (\forall X \subseteq N)\}$$

と表現されることが必要十分である．ただし $\rho(\emptyset) = \mu(\emptyset) = 0$ であって，

$$X, Y \subseteq N \implies \rho(X) - \mu(Y) \geq \rho(X \setminus Y) - \mu(Y \setminus X)$$

を満たすとする．

(2) 集合 $S \subseteq \mathbb{R}^n$ が M 凸集合であるためには，ある劣モジュラ集合関数 $\rho : 2^N \to \mathbb{R} \cup \{+\infty\}$ が存在して

$$S = \{\boldsymbol{x} \in \mathbb{R}^n \mid x(X) \leq \rho(X)\ (\forall X \subset N),\ x(N) = \rho(N)\}$$

と表現されることが必要十分である．ただし $\rho(\emptyset) = 0$, $\rho(N) < +\infty$ とする． ■

$\mathrm{M}^\natural$ 凸集合と M 凸集合はどちらも凸多面体であるので、$\mathrm{M}^\natural$ 凸集合を $\mathbf{M}^\natural$ **凸多面体**、M 凸集合を $\mathbf{M}$ **凸多面体**とも呼ぶ。

2.3.8 連続変数の $\mathrm{M}^\natural$ 凸/M 凸関数の例

連続変数の $\mathrm{M}^\natural$ 凸/M 凸関数の基本的な例を挙げる．実効定義域の閉包については、$\mathrm{M}^\natural$ 凸関数の場合は $\mathrm{M}^\natural$ 凸多面体であること、M 凸関数の場合は M 凸多面体であることを仮定する。しかし $\mathrm{L}^\natural$ 凸/L 凸関数とは異なり、実行定義域を任意の $\mathrm{M}^\natural$ 凸/M 凸多面体に制限したときに $\mathrm{M}^\natural$ 凸/M 凸性が保たれるとは限らない。

$\mathbb{R}^n$ の全体は $\mathrm{M}^\natural$ 凸多面体ではあるが、M 凸多面体ではないことに注意する（L 凸多面体とは状況が異なる）。$\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ とする．

1 次関数

ベクトル $\boldsymbol{p} \in \mathbb{R}^n$ と実数 $\alpha \in \mathbb{R}$ によって定義される 1 次関数

$$F(\boldsymbol{x}) = \alpha + \langle \boldsymbol{p}, \boldsymbol{x} \rangle \tag{2.95}$$

は、実効定義域が $\mathrm{M}^\natural$ 凸多面体であれば $\mathrm{M}^\natural$ 凸関数であり、実効定義域が M 凸多面体であれば M 凸関数である。

2 次関数

実数係数の 2 次関数

$$F(\boldsymbol{x}) = \boldsymbol{x}^\top A \boldsymbol{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j \qquad (a_{ij} = a_{ji} \in \mathbb{R}) \tag{2.96}$$

は、係数行列 $A = (a_{ij})$ が

$$\{i, j\} \cap \{k\} = \emptyset \Longrightarrow a_{ij} \geq \min(a_{ik}, a_{jk}), \tag{2.97}$$

$$\text{任意の } (i, j) \text{ に対して} \qquad a_{ij} \geq 0 \tag{2.98}$$

を満たすとき，$F(\boldsymbol{x})$ は $\mathrm{M}^\natural$ 凸関数である．ただし $\mathrm{dom}_{\mathbb{R}} F = \mathbb{R}^n$ とする [65]．また，

$$\{i, j\} \cap \{k, l\} = \emptyset \implies a_{ij} + a_{kl} \geq \min(a_{ik} + a_{jl}, a_{il} + a_{jk}) \tag{2.99}$$

のとき，$F(\boldsymbol{x})$ は M 凸関数である．ただし，ある $r \in \mathbb{R}$ に対して

$$\mathrm{dom}_{\mathbb{R}} F = \{\boldsymbol{x} \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i = r\}$$

とする [65]．これらの条件は、離散変数の場合の必要十分条件（定理 2.33、定理 2.35）とも同じ形をしているが、連続変数の場合は十分条件であって必要条件ではないことに注意する。（次に述べる例 2.12 参照。）

なお A が正則のとき、$F(\boldsymbol{x})$ が $\mathrm{dom}_{\mathbb{R}} F = \mathbb{R}^n$ の仮定のもとで $\mathrm{M}^\natural$ 凸関数になるための必要十分条件は、A^{-1} が $\mathrm{L}^\natural$ 凸関数の条件 (2.29) を満たすことである [65, 71]。

例 2.12. 行列

$$A = \begin{bmatrix} 5 & 3 & 2 \\ 3 & 5 & 3 \\ 2 & 3 & 5 \end{bmatrix}$$

の逆行列は

$$A^{-1} = \frac{1}{51} \begin{bmatrix} 16 & -9 & -1 \\ -9 & 21 & -9 \\ -1 & -9 & 16 \end{bmatrix}$$

であり、$\mathrm{L}^\natural$ 凸関数の条件 (2.29) を満たすので、式 (2.96) の関数 $F : \mathbb{R}^3 \to \mathbb{R} \cup \{+\infty\}$ は連続変数の $\mathrm{M}^\natural$ 凸関数である。しかしながら行列 A は、連続変数の $\mathrm{M}^\natural$ 凸関数の十分条件 (2.97), (2.98) を満たしていない。またこの条件は離散変数の $\mathrm{M}^\natural$ 凸関数の条件でもあるので、$F(\boldsymbol{x})$ を $\boldsymbol{x} \in \mathbb{Z}^3$ に制限した離散関数は、$\mathrm{M}^\natural$ 凸関数ではない。 ■

分離凸関数

1 変数の凸関数 φ_i $(i = 1, 2, \ldots, n)$ によって定義される分離凸関数

$$F(\boldsymbol{x}) = \sum_{i=1}^{n} \varphi_i(x_i) \tag{2.100}$$

は、実効定義域が $\mathrm{M}^\natural$ 凸多面体であれば $\mathrm{M}^\natural$ 凸関数であり、実効定義域が M 凸多面体であれば M 凸関数である。

さらに，φ_0 も 1 変数の凸関数とするとき，擬分離凸関数

$$F(\boldsymbol{x}) = \varphi_0(\sum_{i=1}^{n} x_i) + \sum_{i=1}^{n} \varphi_i(x_i) \tag{2.101}$$

は，$\mathbb{R}^n$ 上で $\mathrm{M}^\natural$ 凸関数である．

層凸関数

$\{1, 2, \ldots, n\}$ の部分集合からなる層族 $\mathcal{T}$ と，各 $X \in \mathcal{T}$ に対応する 1 変数の凸関数 φ_X から作られる

$$F(\boldsymbol{x}) = \sum_{X \in \mathcal{T}} \varphi_X(\sum_{i \in X} x_i) \tag{2.102}$$

の形の層凸関数は $\mathrm{M}^\natural$ 凸関数である．例えば,

$$F(\boldsymbol{x}) = \varphi_1(x_1 + \cdots + x_n) + \varphi_2(x_2 + \cdots + x_n) + \cdots + \varphi_n(x_n) \tag{2.103}$$

は $\mathrm{M}^\natural$ 凸関数である．これは $\mathcal{T} = \{\{1, 2, \ldots, n\}, \{2, 3, \ldots, n\}, \ldots, \{n\}\}$ の場合である．また，$\mathcal{T} = \{\{1, 2, \ldots, n\}, \{1\}, \{2\}, \ldots, \{n\}\}$ の場合が擬分離凸関数 (2.82) にあたる．

離散変数の場合とは異なり、連続変数の場合は、例 2.12 のように、層凸関数ではない 2 次の $\mathrm{M}^\natural$ 凸関数が存在する。

第 3 章

L 凸関数の連続緩和と最小化

3.1 概要

離散凸解析においては、L$^\natural$ 凸/L 凸関数と M$^\natural$ 凸/M 凸関数という 2 種類の離散凸性が定義され、互いに共役な関係にあることが明らかにされている。このうち L$^\natural$ 凸/L 凸関数は、劣モジュラ集合関数と密接な関係にあり、また在庫管理理論における「Miller の離散凸関数」 [47] や、シフトスケジューリング問題にあらわれるマルチモジュラ性 [3, 41] とも等価な概念である [64]。この章では、このように実用上も重要な概念である離散 L$^\natural$ 凸/L 凸関数に対して、連続緩和による新しい最小化手法を提案する。

離散 L$^\natural$ 凸関数とは、関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ のうち、離散中点凸性（式 (2.7)）

$$g(\boldsymbol{p}) + g(\boldsymbol{q}) \geq g\left(\left\lceil \frac{\boldsymbol{p}+\boldsymbol{q}}{2} \right\rceil\right) + g\left(\left\lfloor \frac{\boldsymbol{p}+\boldsymbol{q}}{2} \right\rfloor\right) \quad (\forall \boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^n) \tag{3.1}$$

をもつものである。

離散 L 凸関数とは、L$^\natural$ 凸関数のうち、$\mathbf{1}$ 方向の線形性（式 (2.13)）

(TRF[$\mathbb{Z}$]) ある実数 r が存在して，任意の $\boldsymbol{p} \in \mathbb{Z}^n$ に対して

$$g(\boldsymbol{p} + \mathbf{1}) = g(\boldsymbol{p}) + r \tag{3.2}$$

をもつものであるが、最小化を考える上では $r = 0$ を前提とする。（そうでなければ、いくらでも関数値を小さくすることができるので、最小化することに意味がなくなる。）

L$^\natural$ 凸関数と L 凸関数は等価に書き換えができるので、アルゴリズムとしては離散 L$^\natural$ 凸関数について記述する。離散 L$^\natural$ 凸関数の最小化アルゴリズムは、大きく分けて、(a) 素朴な降下法、(b) スケーリング法、(c) 連続緩和法の 3 種類ある。このうち (c) が提案手法である。3 種類のアルゴリズムは、それぞれ (a) 最小性規準、(b) 格子間隔を粗くした場合の近接定理、(c) 格子間隔を無限に細かくした場合の近接定理に基づいたものである。

この章では、(c) の手法に必要な、格子間隔を無限に細かくした場合の近接定理を証明する。そして、この 3 種類のアルゴリズムについて、C 言語プログラムによる実装を行い、性能比較を行う。その結果、提案手法である (c) がもっとも効率的であることを述べる。

3.2　最小化アルゴリズムの基本形

3.2.1　離散変数関数の場合

離散 L$^\natural$ 凸/L 凸関数には大域的最小点を特徴づける局所的な最小性規準があることを、定理 2.5 と定理 2.8 で次のように述べた。

定理 3.1.
(1) 関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が L$^\natural$ 凸関数のとき、$\boldsymbol{p} \in \mathrm{dom}_{\mathbb{Z}}\, g$ が g の最小点であるためには，任意の $\boldsymbol{q} \in \{0,1\}^n$ に対して

$$g(\boldsymbol{p}) \leq \min\{g(\boldsymbol{p}-\boldsymbol{q}), g(\boldsymbol{p}+\boldsymbol{q})\} \tag{3.3}$$

となることが必要十分である．
(2) 関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が L 凸関数で，式 (3.2) で $r = 0$ であるとする．$\boldsymbol{p} \in \mathrm{dom}_{\mathbb{Z}}\, g$ が g の最小点であるためには，任意の $\boldsymbol{q} \in \{0,1\}^n$ に対して

$$g(\boldsymbol{p}) \leq g(\boldsymbol{p}+\boldsymbol{q}) \tag{3.4}$$

となることが必要十分である． ■

離散 L$^\natural$ 凸関数を最小化するアルゴリズムとしては，この定理 3.1(1) による大域最小性の局所的な特徴付けから，自然に最急降下法 [62, 第 10.3.1 節] を導くことができる．

L$^\natural$ 凸関数の最小化アルゴリズム（最急降下法）：SD$(g, \boldsymbol{p})$
Input: 離散 L$^\natural$ 凸関数 g と初期解 $\boldsymbol{p} \in \mathrm{dom}_{\mathbb{Z}}\, g$
Output: g の一つの最小解
Step 1: $g(\boldsymbol{p} + \varepsilon\boldsymbol{\chi}_X)$ を最小化する $\varepsilon \in \{1, -1\}$ と $X \subseteq \{1, 2, \ldots, n\}$ を探す．
Step 2: もし $g(\boldsymbol{p}) \leq g(\boldsymbol{p} + \varepsilon\boldsymbol{\chi}_X)$ であれば，$\boldsymbol{p}$ を返す（$\boldsymbol{p}$ は g の最小解の一つ）．
Step 3: $\boldsymbol{p} := \boldsymbol{p} + \varepsilon\boldsymbol{\chi}_X$ として Step 1 に戻る．

アルゴリズムの計算量解析のために、関数 g の実効定義域 $\mathrm{dom}_{\mathbb{Z}}\, g$ は有界であると仮定し、

$$K_\infty = \max\{\|\boldsymbol{p}-\boldsymbol{q}\|_\infty \mid \boldsymbol{p}, \boldsymbol{q} \in \mathrm{dom}_{\mathbb{Z}}\, g\} \tag{3.5}$$

とおく [39]．また g の関数値は、定数時間 T_{func} で得られるものと仮定する。

手順 Step 1，つまり定理 3.1(1) の検証は，劣モジュラ集合関数の最小化を 2 回実行することに相当する。これは多項式時間で行うことができる [35, 81, 85]．手順 Step 1 から Step 3 までの反復回数は、実効定義域の大きさに比例して $\mathrm{O}(K_\infty)$ である [20, 39, 75, 77]．したがって、劣モジュラ集合関数の最小解が $\mathrm{O}(S)$ 回の関数評価で求められるとすると，最急降下法では g の最小解は $\mathrm{O}(ST_{\mathrm{func}}K_\infty)$ の計算時間で求められる．つまり，最急降下法は（多項式時間ではなく）擬多項式時間アルゴリズムである．

3.2.2 連続変数関数の場合

連続変数の L$^\natural$ 凸/L 凸関数にも、離散関数と同様の最小性規準があることを、定理 2.19 で次のように述べた。

定理 3.2.
(1) 関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が L$^\natural$ 凸関数のとき，点 $\boldsymbol{p} \in \mathrm{dom}_{\mathbb{R}} G$ が G の最小点であるためには，任意の $\boldsymbol{q} \in \{0,1\}^n$ に対して

$$G'(\boldsymbol{p}; \boldsymbol{q}) \geq 0, \qquad G'(\boldsymbol{p}; -\boldsymbol{q}) \geq 0$$

となることが必要十分である．
(2) 関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が L 凸関数のとき，点 $\boldsymbol{p} \in \mathrm{dom}_{\mathbb{R}} G$ が G の最小点であるためには，$G'(\boldsymbol{p}; \boldsymbol{1}) = 0$，かつ，任意の $\boldsymbol{q} \in \{0,1\}^n$ に対して

$$G'(\boldsymbol{p}; \boldsymbol{q}) \geq 0$$

となることが必要十分である． ■

この最小性規準の存在にもかかわらず、連続 L$^\natural$ 凸/L 凸関数には特徴的な最小化アルゴリズムが確立されているわけではない。連続 L$^\natural$ 凸/L 凸関数の最小化においても、一般の凸関数の最小化で用いられる準ニュートン法のようなアルゴリズムが用いられる。ただし、関数のヘッセ行列が特殊な場合には、効率的な最小化アルゴリズムの開発も期待される。いずれにしても、離散 L$^\natural$ 凸/L 凸関数の最小化に比べると、連続 L$^\natural$ 凸/L 凸関数の最小化にかかる計算量は実際上は少ないと言える。

3.3 スケーリング

離散関数の最小解を求めるアルゴリズムを高速化する手法の 1 つに、スケーリング法がある。ここでは、まずスケーリングの基本的な考え方を述べ、次に離散 L$^\natural$ 凸/L 凸関数にどのように適用されるかを述べる。離散 L$^\natural$ 凸/L 凸関数にはこの手法が効率的に働き、容易に多項式時間の最小化アルゴリズムが実現される。

3.3.1 スケーリングの考え方

スケーリングの基本的なアイデアは、離散格子点の格子を間引くことにある。格子を間引くと精密さはなくなるが、関数の実効定義域が実質的に小さくなるので、最小化が高速に行えることが期待される。最小解の位置は、間引く前後で近いところにあることが多い。離散関数の性質によっては近さを理論的に解析できる場合があり、この距離を保証するのがスケーリングの近接定理である。

例えば、与えられた離散関数を最小化することを考える。同時に、偶数格子点だけに間引いた関数も考える。つまり、$f : \mathbb{Z}^n \to \mathbb{R}$ が与えられたとして、偶数格子点上の f を取り出した

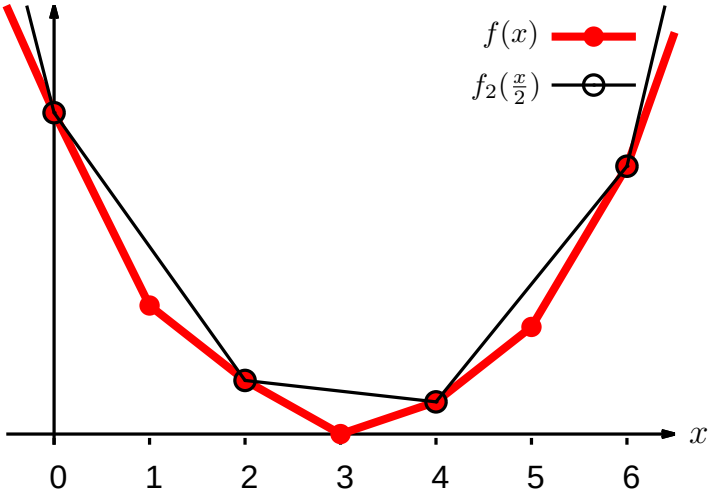


図 3.1. 離散変数のスケーリング

関数 f_2 を

$$f_2(\boldsymbol{x}) = f(2\boldsymbol{x})$$

として作る。

$\mathrm{dom}_{\mathbb{Z}}\, f_2$ は $\mathrm{dom}_{\mathbb{Z}}\, f$ に比べると、その要素の個数は $1/2^n$ ほどの大きさである。$\mathrm{argmin}_{\mathbb{Z}}\, f_2$ と $\mathrm{argmin}_{\mathbb{Z}}\, f$ に単純な関係性があれば、比較的容易に求められる $\mathrm{argmin}_{\mathbb{Z}}\, f_2$（に含まれる解）を利用することで、$\mathrm{argmin}_{\mathbb{Z}}\, f$（に含まれる解）を効率的に求めることができる。

このことを 1 変数の場合に描いたのが図 3.1 である。$f : \mathbb{Z} \to \mathbb{R}$ の最小化を次のようにして行う。まず $f_2(x)$ の最小化を行う。$f_2(\frac{4}{2})$ で最小となることがわかるので、次に $x = 4$ から $f(x)$ の最小化を行う。すると容易に $f(3)$ で最小化が完了する。更に再帰的に、最初の $f_2(x)$ の最小化にも、

$$f_4(x) = f_2(2x)$$

と定義した f_4 の最小化の結果を利用することもできる。

一般に，離散変数の関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ と正の整数 α および $\boldsymbol{b} \in \mathbb{Z}^n$ に対して，

$$f^\alpha(\boldsymbol{x}) = f(\alpha\boldsymbol{x} + \boldsymbol{b}) \qquad (\boldsymbol{x} \in \mathbb{Z}^n) \tag{3.6}$$

で定義される関数 $f^\alpha : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ はスケーリングと呼ばれる。スケーリングを再帰的に適用するアルゴリズムは，スケーリング法と呼ばれる。

関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ をスケーリングした関数 f^α の最小点（あるいは極小点）を $\boldsymbol{y}^\alpha$ とするとき，$\boldsymbol{y}^\alpha$ を用いて計算した f の近似最小点

$$\boldsymbol{x}^\alpha = \alpha\boldsymbol{y}^\alpha + \boldsymbol{b} \tag{3.7}$$

に関して，真の最小点 $\boldsymbol{x}^*$ との距離の近さを保証する定理は、近接定理と呼ばれる。1 変数の場合は、f が離散凸であれば容易に

$$x^\alpha - (\alpha - 1) \le x^* \le x^\alpha + (\alpha - 1) \tag{3.8}$$

が成り立つことがわかる。多変数の場合にも，後述するように、L$^\natural$ 凸/L 凸関数と M$^\natural$ 凸/M 凸関数に対して近接定理が成り立つ.

3.3.2 L 凸関数に対するスケーリング

関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ と正の整数 α および $\boldsymbol{b} \in \mathbb{Z}^n$ に対して，g のスケーリング (3.6) を

$$g^\alpha(\boldsymbol{p}) = g(\alpha\boldsymbol{p} + \boldsymbol{b})$$

とする.

L$^\natural$ 凸/L 凸関数は、スケーリングを行っても L$^\natural$ 凸/L 凸性を保持し続ける。これは（後述する）M$^\natural$ 凸/M 凸関数とは異なる、重要な性質である。

命題 3.3 ([62, 定理 7.11(1), 定理 7.10(2)])**.**
(1) 関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が L$^\natural$ 凸関数ならば，g^α も L$^\natural$ 凸関数である.
(2) 関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が L 凸関数ならば，g^α も L 凸関数である. ■

L$^\natural$ 凸/L 凸集合についても、同様にスケーリングの関係がある。集合 S と集合 S^α に関して、ある正の整数 α および $\boldsymbol{b} \in \mathbb{Z}^n$ が存在して

$$S^\alpha = \{\boldsymbol{p} \in \mathbb{Z}^n \mid \alpha\boldsymbol{p} + \boldsymbol{b} \in S\}$$

を満たすとする。このとき、それぞれの標示関数 $\delta_S : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ と $\delta_{S^\alpha} : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ の間には、

$$\delta_{S^\alpha}(\boldsymbol{p}) = \delta_S(\alpha\boldsymbol{p} + \boldsymbol{b})$$

の関係が成り立つので、次の命題の成り立つことがわかる。

命題 3.4.
(1) 集合 S が L$^\natural$ 凸集合ならば，集合 S^α も L$^\natural$ 凸集合である.
(2) 集合 S が L 凸集合ならば，集合 S^α も L 凸集合である. ■

3.3.3 スケーリングの近接定理

L$^\natural$ 凸/L 凸関数に対する近接定理は，スケーリングされた関数 g^α の極小点 $\boldsymbol{q}^\alpha$ から、(3.7) に従って

$$\boldsymbol{p}^\alpha = \alpha\boldsymbol{q}^\alpha + \boldsymbol{b}$$

として計算した点 $\boldsymbol{p}^\alpha$ の近くに，g の最小点 $\boldsymbol{p}^*$ が存在することを保証する．なお，命題 3.3, 定理 3.1 により，g^α の極小点は g^α の最小点である.

定理 3.5 ([62, 定理 7.18][36])**.** α を正整数とする.
(1) $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ を L$^\natural$ 凸関数とし，$\boldsymbol{p}^\alpha \in \mathrm{dom}_{\mathbb{Z}}\, g$ とする．任意の $\boldsymbol{q} \in \{0, 1\}^n$ に対して

$$g(\boldsymbol{p}^\alpha) \leq \min\{g(\boldsymbol{p}^\alpha - \alpha\boldsymbol{q}), g(\boldsymbol{p}^\alpha + \alpha\boldsymbol{q})\}$$

ならば，$\operatorname{argmin}_{\mathbb{Z}} g \neq \emptyset$ であって，

$$\boldsymbol{p}^{\alpha} - n(\alpha-1)\mathbf{1} \leq \boldsymbol{p}^{*} \leq \boldsymbol{p}^{\alpha} + n(\alpha-1)\mathbf{1}$$

を満たす $\boldsymbol{p}^{*} \in \operatorname{argmin}_{\mathbb{Z}} g$ が存在する．

(2) $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ を $g(\boldsymbol{p}) = g(\boldsymbol{p}+\mathbf{1})$ $(\forall \boldsymbol{p} \in \mathbb{Z}^n)$ を満たす L 凸関数とし，$\boldsymbol{p}^{\alpha} \in \operatorname{dom}_{\mathbb{Z}} g$ とする．任意の $\boldsymbol{q} \in \{0,1\}^n$ に対して

$$g(\boldsymbol{p}^{\alpha}) \leq g(\boldsymbol{p}^{\alpha} + \alpha\boldsymbol{q})$$

ならば，$\operatorname{argmin}_{\mathbb{Z}} g \neq \emptyset$ であって，

$$\boldsymbol{p}^{\alpha} \leq \boldsymbol{p}^{*} \leq \boldsymbol{p}^{\alpha} + (n-1)(\alpha-1)\mathbf{1}$$

を満たす $\boldsymbol{p}^{*} \in \operatorname{argmin}_{\mathbb{Z}} g$ が存在する． ■

3.3.4　スケーリング法

上の近接定理を利用して $\mathrm{L}^{\natural}$ 凸関数 g を最小化するアルゴリズム [62, 第 10.3.2 節] を示す．

$\mathrm{L}^{\natural}$ 凸関数の最小化アルゴリズム（スケーリング法）：SCALING$(g, \boldsymbol{p})$
Input: 離散 $\mathrm{L}^{\natural}$ 凸関数 g と初期解 $\boldsymbol{p} \in \operatorname{dom}_{\mathbb{Z}} g$
Output: g の一つの最小解
Step 0: $k := \lceil \log_2 K_{\infty} \rceil$, $\boldsymbol{q}_{k+1} := \mathbf{0}$ とおく．
Step 1: $\alpha_k := 2^k$ とおく。
$g_k(\boldsymbol{q}) := g(\boldsymbol{p} + \alpha_k \boldsymbol{q})$, $\operatorname{dom}_{\mathbb{Z}} g_k := \{\boldsymbol{q} \in \mathbb{Z}^n \mid 2\boldsymbol{q}_{k+1} - n\mathbf{1} \leq \boldsymbol{q} \leq 2\boldsymbol{q}_{k+1} + n\mathbf{1}\}$
として最急降下法 SD$(g_k, 2\boldsymbol{q}_{k+1})$ を呼び出し、得られた最小解を $\boldsymbol{q}_k$ とする。
Step 2: $k = 0$ ならば $\boldsymbol{p} + \boldsymbol{q}_0$ を返す（$\boldsymbol{p} + \boldsymbol{q}_0$ は g の最小解の一つ）．
Step 3: $k := k - 1$ として Step 1 に戻る．

手順 Step 1 で呼び出す最急降下法 SD$(g_k, 2\boldsymbol{q}_{k+1})$ では、関数の定義域が小さいので、必要な計算量は $\mathrm{O}(nST_{\mathrm{func}})$ ですむ。手順 Step 1 から Step 3 の反復は、ちょうど $(\lceil \log_2 K_{\infty} \rceil + 1)$ 回であるので、全体の計算量は $\mathrm{O}(nST_{\mathrm{func}} \lceil \log_2 K_{\infty} \rceil)$ となり、スケーリング法は多項式時間で終了することがわかる [62, 第 10.3.2 節].

3.4　連続緩和

一般に、離散最適化問題などで、離散変数の整数条件を緩めて連続変数に置き換えることは連続緩和と呼ばれ、連続緩和された問題は連続緩和問題、その解は連続緩和解と呼ばれる。まず連続緩和問題を解き、得られた連続緩和解を元に離散解を求め直すという工夫は、連続緩和法と呼ばれ、整数計画問題や混合整数計画問題でもよく用いられる手法である。一般に緩和した連続変数の問題のほうが解きやすく、高速化に役立つことが多い。

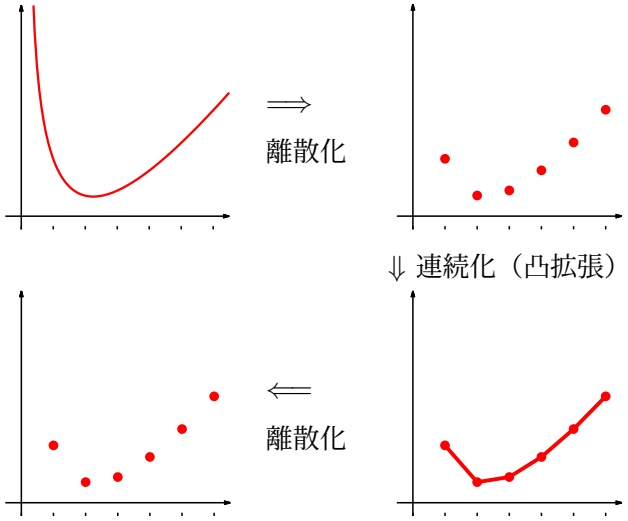


図 3.2. 関数の離散化と連続化

本節では、連続緩和法を L$^\natural$ 凸/L 凸関数の最小化に適用する。離散凸関数における連続緩和は、スケーリングとはいわば逆の操作、つまり格子を（間引くのではなく）詰めることを、無限に繰り返した末に行き着く姿ととらえることもできる。その意味では、スケーリングでは L$^\natural$ 凸/L 凸性が保存され近接定理が成立したように、連続緩和によっても類似する性質が見いだせることを期待する。

まず、連続変数の関数から離散変数の関数を作り出すことと、その逆の操作がどのように行われるのかを述べてから、提案手法である連続緩和法について説明する。

3.4.1 一般の凸関数に対する離散化と連続化

一般の連続変数の関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ の離散化は，定義域を $\mathbb{Z}^n$ に制限することによって定義される．すなわち，

$$\hat{f}(\boldsymbol{x}) = F(\boldsymbol{x}) \qquad (\boldsymbol{x} \in \mathbb{Z}^n) \tag{3.9}$$

で定められる関数 $\hat{f} : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が F の離散化である。この自然な定義によって、離散化は一意に定まる。また、元の関数の情報が失われていることに注意する（図 3.2 の上）。

これに対して、連続化には自由度が考えられるので、連続化の操作の定義には工夫を要する。離散変数の関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が凸拡張可能とは，条件 (2.5):

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) \qquad (\boldsymbol{x} \in \mathbb{Z}^n)$$

を満たす凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が存在することであるが、これを満たす関数は一意に定まらない。しかし、f の凸拡張の中で各点での値が最大のものは一意に定まり、

$$\overline{f}(\boldsymbol{x}) = \sup\{g(\boldsymbol{x}) \mid g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\} \text{ は閉真凸関数で } g(\boldsymbol{y}) \leq f(\boldsymbol{y}) \ (\forall \boldsymbol{y} \in \mathbb{Z}^n)\} \tag{3.10}$$

（ただし $\boldsymbol{x} \in \mathbb{R}^n$）で与えられ，$f$ の凸閉包と呼ばれる．ここでは，関数 f が凸拡張可能である場合に限って連続化の操作を考えることとし，一意に定まる f の凸閉包 $\overline{f}$ を f の連続化と定義することにする。

集合 S に対しては，S を含む最小の凸集合を S の凸包と呼んで $\overline{S}$ と表す。これが集合に対する連続化の操作にあたる．標示関数 δ_S を用いると，$\delta_{\overline{S}} = \overline{\delta_S}$ が成り立つ．

3.4.2　L 凸関数に対する離散化と連続化

$\mathrm{L}^\natural$ 凸/L 凸関数が凸拡張可能であることを、定理 2.4 と定理 2.7 で次のように述べた。したがって連続化が存在する。

定理 3.6.
(1) $\mathrm{L}^\natural$ 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は凸拡張可能である．
(2) L 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は凸拡張可能である． ■

$\mathrm{L}^\natural$ 凸/L 凸関数と $\mathrm{L}^\natural$ 凸/L 凸集合について，離散化と連続化が $\mathrm{L}^\natural$ 凸/L 凸性を保つかどうかを考える．結論としては、スケーリングと同様に、$\mathrm{L}^\natural$ 凸/L 凸性は離散化と連続化のどちらの場合にも保たれる。まず離散化について述べる。

定理 3.7.
(1) 連続変数の $\mathrm{L}^\natural$ 凸関数の離散化 (3.9) は，離散変数の $\mathrm{L}^\natural$ 凸関数である．
(2) 連続変数の L 凸関数の離散化 (3.9) は，離散変数の L 凸関数である．

次に，連続化について述べる。

定理 3.8.
(1) 離散変数の $\mathrm{L}^\natural$ 凸関数の連続化 (3.10) は，連続変数の $\mathrm{L}^\natural$ 凸関数である．
(2) 離散変数の L 凸関数の連続化 (3.10) は，連続変数の L 凸関数である． ■

これらの定理の特殊ケースとして、関数が標示関数である場合を考えると、集合の離散化と連続化に関する定理が導かれる．

定理 3.9.
(1) $\mathrm{L}^\natural$ 凸多面体に含まれる整数ベクトルの全体は，離散の $\mathrm{L}^\natural$ 凸集合である．
(2) L 凸多面体に含まれる整数ベクトルの全体は，離散の L 凸集合である． ■

定理 3.10.
(1) 離散の $\mathrm{L}^\natural$ 凸集合の凸包は，$\mathrm{L}^\natural$ 凸多面体である．
(2) 離散の L 凸集合の凸包は，L 凸多面体である． ■

3.4.3 連続緩和問題の定義

連続緩和問題の定義には、次の定理が有用である．凸拡張した関数として、閉真凸関数がとれる。

定理 3.11 ([62, 第 7.8 節])**.**
(1) 任意の離散 $\mathrm{L}^\natural$ 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ に対して，ある連続変数の閉真 $\mathrm{L}^\natural$ 凸関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が存在して，任意の $\boldsymbol{p} \in \mathbb{Z}^n$ に対して $G(\boldsymbol{p}) = g(\boldsymbol{p})$ を満たし，また $\mathrm{dom}_{\mathbb{R}} G$ が $\mathrm{dom}_{\mathbb{Z}} g$ の閉凸包となる．
(2) 任意の離散 L 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ に対して，ある連続変数の閉真 L 凸関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が存在して，任意の $\boldsymbol{p} \in \mathbb{Z}^n$ に対して $G(\boldsymbol{p}) = g(\boldsymbol{p})$ を満たし，また $\mathrm{dom}_{\mathbb{R}} G$ が $\mathrm{dom}_{\mathbb{Z}} g$ の閉凸包となる． ■

この定理の関数 G の例としては、式 (3.10) で定義した g の連続化（凸閉包）が挙げられる。
連続緩和問題の定義を行う。離散 $\mathrm{L}^\natural$ 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ に対して，

$$g(\boldsymbol{p}) = G(\boldsymbol{p}) \quad (\boldsymbol{p} \in \mathbb{Z}^n), \tag{3.11}$$

$$\mathrm{dom}_{\mathbb{R}} G \text{ が } \mathrm{dom}_{\mathbb{Z}} g \text{ の閉凸包} \tag{3.12}$$

の条件を満たす連続変数の閉真 $\mathrm{L}^\natural$ 凸関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ の最小化を連続緩和問題とする。このような関数 G の存在は定理 3.11 によって保証されている。以下では G を離散 $\mathrm{L}^\natural$ 凸関数 g に対する「連続緩和の $\mathrm{L}^\natural$ 凸関数」と表現する。

3.4.4 連続緩和の近接定理

離散 $\mathrm{L}^\natural$ 凸関数の最小解と、その連続緩和問題の最小解との近接性について述べる。どちらの最小解も唯一とは限らないので、近接定理が任意の最小解について成り立つのか、それともある最小解について成り立つのか、注意を払う必要がある。
ここでは 2 つの近接定理を示す。1 つめは、離散 $\mathrm{L}^\natural$ 凸関数の任意の離散最小解の近傍に，ある連続緩和解が存在することを示すものである。

定理 3.12. $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ を離散 $\mathrm{L}^\natural$ 凸関数，$G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ を連続緩和の $\mathrm{L}^\natural$ 凸関数とし，$\mathrm{argmin}_{\mathbb{R}} G \neq \emptyset$ を満たすとする．このとき，任意の $\boldsymbol{p}^* \in \mathrm{argmin}_{\mathbb{Z}} g$ に対して，ある $\bar{\boldsymbol{p}} \in \mathrm{argmin}_{\mathbb{R}} G$ が存在して，

$$\boldsymbol{p}^* - n\mathbf{1} \leq \bar{\boldsymbol{p}} \leq \boldsymbol{p}^* + n\mathbf{1} \tag{3.13}$$

を満たす.

定理 3.12 の証明は，後ほど第 3.5 節で与える．
提案する最小化アルゴリズムが本当に必要としているのは，定理 3.12 のいわば逆方向の定理である．すなわち，任意の連続緩和解の近傍に，ある離散最小解が存在することを示した

い．次に述べる 2 つめの近接定理はそのことを示すものであり，本章の主となる成果である．ここでは G の実効定義域が有界であると仮定する．なお閉真凸関数は、定義域が有界であれば最小化集合が非空となるので（第 2 章参照)、$\mathrm{argmin}_{\mathbb{R}} G \neq \emptyset$ となる。

定理 3.13. $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ を離散 $\mathrm{L}^\natural$ 凸関数，$G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ を連続緩和の $\mathrm{L}^\natural$ 凸関数とし、$\mathrm{dom}_{\mathbb{R}} G$ が有界であると仮定する．このとき，任意の $\bar{\boldsymbol{p}} \in \mathrm{argmin}_{\mathbb{R}} G$ に対して，ある $\boldsymbol{p}^* \in \mathrm{argmin}_{\mathbb{Z}} g$ が存在して，

$$\bar{\boldsymbol{p}} - n\mathbf{1} \leq \boldsymbol{p}^* \leq \bar{\boldsymbol{p}} + n\mathbf{1} \tag{3.14}$$

を満たす．

定理 3.13 の証明も，後ほど第 3.5 節で与える．

なお、g と G が $\mathrm{L}^\natural$ 凸関数ではなくて L 凸関数の場合は、式 (3.13) と式 (3.14) は、それぞれ

$$\boldsymbol{p}^* - (n-1)\mathbf{1} \leq \bar{\boldsymbol{p}} \leq \boldsymbol{p}^* + (n-1)\mathbf{1}, \tag{3.15}$$

$$\bar{\boldsymbol{p}} - (n-1)\mathbf{1} \leq \boldsymbol{p}^* \leq \bar{\boldsymbol{p}} + (n-1)\mathbf{1} \tag{3.16}$$

となる。

3.4.5　連続緩和法

ここでは連続緩和法のアルゴリズムを提案する．この手法は，初期解を連続緩和解とした最急降下法である．最小化したい離散 $\mathrm{L}^\natural$ 凸関数 $g : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ に対して，連続緩和の $\mathrm{L}^\natural$ 凸関数 $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が知られていて、かつ $\mathrm{argmin}_{\mathbb{R}} G \neq \emptyset$ を満たすことを仮定する．G が容易に最小化できれば，提案する連続緩和法によって g を効率的に最小化できる．

$\mathrm{L}^\natural$ 凸関数の最小化アルゴリズム（連続緩和法）：$\mathrm{RELAX}(g, G)$
Input: 離散 $\mathrm{L}^\natural$ 凸関数 g と連続緩和の $\mathrm{L}^\natural$ 凸関数 G
Output: g の一つの最小解
　Step 1: $\bar{\boldsymbol{p}} \in \mathrm{argmin}_{\mathbb{R}} G$ を見つける．
　Step 2: $\bar{\boldsymbol{p}}$ を整数ベクトルに丸めて $\boldsymbol{p} \in \mathrm{dom}_{\mathbb{Z}} g$ を得る．
　Step 3: 最急降下法 $\mathrm{SD}(g, \boldsymbol{p})$ の値を返す．

連続緩和解を高速に見つけることができれば，2 つめの近接定理 (定理 3.13) によって，提案する連続緩和法は効率的に実行できることがわかる．定義により連続 $\mathrm{L}^\natural$ 凸関数は凸関数であるので，手順 Step 1 で緩和解 $\bar{\boldsymbol{p}}$ を見つけるために，G に連続変数の凸関数最小化アルゴリズムを用いることができる．Step 3 の最急降下法 $\mathrm{SD}(g, \boldsymbol{p})$ の内部での反復回数は，定理 3.13 より，$\mathrm{O}(n)$ である．（初期解を特定しない最急降下法であれば，反復回数が $\mathrm{O}(K_\infty)$ となるところである．）したがって，Step 1 での緩和解を見つけるのに必要な計算量を T_{relax}、Step 2 での整数ベクトルに丸める操作に必要な計算量を T_{round} で表すと，連続緩和法が g の最小解を求めるのに必要な計算量は $\mathrm{O}(T_{\mathrm{relax}} + T_{\mathrm{round}} + nST_{\mathrm{func}})$ となる．

なお、第 3.4.3 節では連続緩和の $\mathrm{L}^\natural$ 凸関数 G の例として g の閉凸包による連続化を挙げたが、関数値の計算に多くの計算量がかかるため、連続緩和法には適さない。

3.5 近接定理の証明

連続緩和の 2 つの近接定理（定理 3.12 と定理 3.13）の証明を与える [54, 56].

3.5.1 連続緩和の近接定理の証明

定理 3.12 の証明. 整数 $s \geq 2$ について，$g_s : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ を

$$g_s(\boldsymbol{p}) := G(\frac{\boldsymbol{p}}{s}) \quad (\boldsymbol{p} \in \mathbb{Z}^n)$$

と定義する．このとき、式 (3.11) の

$$g(\boldsymbol{p}) = G(\boldsymbol{p}) \quad (\boldsymbol{p} \in \mathbb{Z}^n)$$

の関係を用いると、

$$g(\boldsymbol{p}) = g_s(s\boldsymbol{p}) \quad (\boldsymbol{p} \in \mathbb{Z}^n) \tag{3.17}$$

を満たすことがわかる。

任意の $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^n$ と $0 \leq \alpha \in \mathbb{Z}$ について,

$$\begin{aligned} g_s(\boldsymbol{p}) + g_s(\boldsymbol{q}) &= G(\frac{\boldsymbol{p}}{s}) + G(\frac{\boldsymbol{q}}{s}) \\ &\geq G((\frac{\boldsymbol{p}}{s} - \frac{\alpha}{s}\mathbf{1}) \vee \frac{\boldsymbol{q}}{s}) + G(\frac{\boldsymbol{p}}{s} \wedge (\frac{\boldsymbol{q}}{s} + \frac{\alpha}{s}\mathbf{1})) \\ &= G(\frac{(\boldsymbol{p} - \alpha\mathbf{1}) \vee \boldsymbol{q}}{s}) + G(\frac{\boldsymbol{p} \wedge (\boldsymbol{q} + \alpha\mathbf{1})}{s}) \\ &= g_s((\boldsymbol{p} - \alpha\mathbf{1}) \vee \boldsymbol{q}) + g_s(\boldsymbol{p} \wedge (\boldsymbol{q} + \alpha\mathbf{1})) \end{aligned}$$

が成り立つ．ただし，不等式は G の並進劣モジュラ性 ($\mathrm{SBF}^\natural[\mathbb{R}]$) より導かれた．これにより g_s が離散 $\mathrm{L}^\natural$ 凸関数であることがわかる.

$\boldsymbol{p}^*$ を g の最小解とする．g の最小性規準 (3.3) と式 (3.17) より,

$$g_s(s\boldsymbol{p}^*) \leq g_s(s\boldsymbol{p}^* \pm s\boldsymbol{\chi}_X) \quad (\forall X \subseteq \{1, 2, \ldots, n\})$$

が成り立つ．$\mathrm{L}^\natural$ 凸関数のスケーリングされた局所最適性についての近接定理 (定理 3.5(1)) を g_s, $s\boldsymbol{p}^*$ に適用すると，ある $\boldsymbol{p}_s \in \operatorname{argmin}_{\mathbb{Z}} g_s$ が存在して,

$$s\boldsymbol{p}^* - (s-1)n\mathbf{1} \leq \boldsymbol{p}_s \leq s\boldsymbol{p}^* + (s-1)n\mathbf{1} \tag{3.18}$$

を満たすことがわかる．辺々を s で割ると

$$\boldsymbol{p}^* - n\mathbf{1} \leq \boldsymbol{p}^* - \frac{s-1}{s}n\mathbf{1} \leq \frac{\boldsymbol{p}_s}{s} \leq \boldsymbol{p}^* + \frac{s-1}{s}n\mathbf{1} \leq \boldsymbol{p}^* + n\mathbf{1}$$

と変形できる.

集合

$$K := \{\boldsymbol{p} \in \mathbb{R}^n \mid \boldsymbol{p}^* - n\mathbf{1} \leq \boldsymbol{p} \leq \boldsymbol{p}^* + n\mathbf{1}\}$$

を考える．K はコンパクト集合であるので，K に含まれる任意の数列は，収束部分列をもち，極限点も K に含まれる．整数 $k \geq 1$ について，

$$s_k = 2^k, \quad \boldsymbol{p}_{s_k} \in \operatorname{argmin}_{\mathbb{Z}} g_{s_k}, \quad \frac{\boldsymbol{p}_{s_k}}{s_k} \in K$$

とおく．数列 $\{\frac{\boldsymbol{p}_{s_k}}{s_k}\}$ から収束部分列 $\{\frac{\boldsymbol{p}_{s_{k_i}}}{s_{k_i}}\}$ をとることができる。このとき

$$\lim_{i\to\infty} \frac{\boldsymbol{p}_{s_{k_i}}}{s_{k_i}} = \boldsymbol{p}' \in K$$

とおく．G の連続性より，

$$\lim_{i\to\infty} G(\frac{\boldsymbol{p}_{s_{k_i}}}{s_{k_i}}) = G(\lim_{i\to\infty} \frac{\boldsymbol{p}_{s_{k_i}}}{s_{k_i}}) = G(\boldsymbol{p}')$$

となる．このとき、粗い格子が細かい格子の部分集合を成すことから、$\{G(\frac{\boldsymbol{p}_{s_{k_i}}}{s_{k_i}})\}$ は単調減少数列

$$G(\frac{\boldsymbol{p}_{s_{k_1}}}{s_{k_1}}) \geq G(\frac{\boldsymbol{p}_{s_{k_2}}}{s_{k_2}}) \geq \cdots \geq G(\frac{\boldsymbol{p}_{s_{k_i}}}{s_{k_i}}) \geq \cdots$$

であり，

$$G(\boldsymbol{p}') \leq G(\frac{\boldsymbol{p}_{2^{k_i}}}{2^{k_i}}) = \min g_{2^{k_i}} \quad (i \in \mathbb{Z}, i \geq 1) \tag{3.19}$$

であることに注意する．（この事実は下で用いる。）

次に，上で定めた $\boldsymbol{p}'$ について $G(\boldsymbol{p}') = \min G$ つまり $\boldsymbol{p}' \in \operatorname{argmin}_{\mathbb{R}} G$ であることを、背理法を用いて証明する．そのために，$G(\boldsymbol{p}') > \min G$ であると仮定し，$\varepsilon_0 := G(\boldsymbol{p}') - \min G > 0$ とおく．$\boldsymbol{p}'' \in \operatorname{argmin}_{\mathbb{R}} G$ を任意に固定する．このとき、任意の $\delta > 0$ に対して、ある整数 $n' \in \{k_i \mid i = 1, 2, \ldots\}$ と

$$\boldsymbol{b}_0 = \lfloor \boldsymbol{p}'' \rfloor, \quad \boldsymbol{b}_k \in \{0,1\}^n \quad (k = 1, 2, \ldots, n')$$

を満たす数列 $\{\boldsymbol{b}_k\}$ が存在して，

$$\boldsymbol{q} := \sum_{k=0}^{n'} \frac{\boldsymbol{b}_k}{2^k} \tag{3.20}$$

が $|\boldsymbol{p}'' - \boldsymbol{q}| < \delta$ を満たす．ここで $2^{n'}\boldsymbol{q} \in \mathbb{Z}$ となることに注意する．一方、G の連続性により，

$$\forall \varepsilon' > 0, \exists \delta_{\varepsilon'} > 0 : |\boldsymbol{x} - \boldsymbol{y}| < \delta_{\varepsilon'} \Rightarrow |G(\boldsymbol{x}) - G(\boldsymbol{y})| < \varepsilon' \tag{3.21}$$

が成り立つ．さて，式 (3.21) において $\boldsymbol{x} = \boldsymbol{p}''$, $\varepsilon' = \frac{\varepsilon_0}{2}$ に対して定まる $\delta_{\varepsilon'}$ を δ とする。この δ に対して式 (3.20) で定まる $\boldsymbol{q}$ は、$|\boldsymbol{p}'' - \boldsymbol{q}| < \delta_{\varepsilon'}$ を満たすので $|G(\boldsymbol{p}'') - G(\boldsymbol{q})| < \frac{\varepsilon_0}{2}$ となり、$G(\boldsymbol{q}) < \min G + \frac{\varepsilon_0}{2}$ が成り立つ。したがって

$$\min g_{2^{n'}} \leq G(\boldsymbol{q}) < \min G + \frac{\varepsilon_0}{2} < \min G + \varepsilon_0 = G(\boldsymbol{p}')$$

となり，式 (3.19) に矛盾する．これより $G(\boldsymbol{p}') = \min G$ が証明された．すなわち、$\bar{\boldsymbol{p}}$ として $\boldsymbol{p}' = \lim_{i\to\infty} \frac{\boldsymbol{p}_{s_{k_i}}}{s_{k_i}}$ がとれる。 ■

3.5.2 連続緩和の近接定理（逆）の証明

定理 3.13 の証明. 定理 3.12 を逆方向に用いるため，まず $G : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が唯一の最小解 $\bar{\boldsymbol{p}} \in \mathbb{R}^n$ をもつ場合を考える．このとき，すべての $\boldsymbol{p}^* \in \operatorname{argmin}_{\mathbb{Z}} g$ が

$$\boldsymbol{p}^* - n\mathbf{1} \leq \bar{\boldsymbol{p}} \leq \boldsymbol{p}^* + n\mathbf{1}$$

を満たすことが，定理 3.12 より直ちにわかる．また，この条件を満たす $\boldsymbol{p}^* \in \operatorname{argmin}_{\mathbb{Z}} g$ が必ず存在することもわかる．

次に、G が唯一の最小解をもつとは限らない一般の場合を考える。この場合には、G を摂動して上の場合に帰着させる。最小解 $\bar{\boldsymbol{p}} \in \operatorname{argmin}_{\mathbb{R}} G$ を任意に 1 つ定めて，任意の $\varepsilon > 0$ に対して，関数 $G_\varepsilon : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ と関数 $g_\varepsilon : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ を

$$G_\varepsilon(\boldsymbol{p}) := G(\boldsymbol{p}) + \varepsilon \sum_{i=1}^{n} (p(i) - \bar{p}(i))^2 \quad (\boldsymbol{p} \in \mathbb{R}^n),$$

$$g_\varepsilon(\boldsymbol{p}) := G_\varepsilon(\boldsymbol{p}) \quad (\boldsymbol{p} \in \mathbb{Z}^n)$$

と定義する．関数 G_ε と関数 g_ε は，$\mathrm{L}^\natural$ 凸関数である。明らかに、G_ε は唯一の最小解 $\bar{\boldsymbol{p}}$ をもつ。さて，十分に小さな ε を定めたとき，g_ε の最小化元 $\boldsymbol{p}^*_\varepsilon \in \operatorname{argmin}_{\mathbb{Z}} g_\varepsilon$ が g の最小解となることを示そう．G の実効定義域は有界であると仮定したので,

$$\tilde{K}_\infty = \max\{\|\boldsymbol{p} - \boldsymbol{q}\|_\infty \mid \boldsymbol{p}, \boldsymbol{q} \in \operatorname{dom}_{\mathbb{R}} G\}$$

が存在する。

$$g(\boldsymbol{p}') = \min\{g(\boldsymbol{p}) \mid \boldsymbol{p} \in \operatorname{dom}_{\mathbb{Z}} g \setminus \operatorname{argmin}_{\mathbb{Z}} g\}$$

とする．すなわち、$g(\boldsymbol{p}')$ は g の最小値とは異なる 2 番目に小さい値である。摂動のパラメータ ε を

$$0 < \varepsilon < \frac{g(\boldsymbol{p}') - \min g}{n\tilde{K}_\infty^2}$$

を満たす値に定める．$\boldsymbol{p}^*_\varepsilon \in \operatorname{argmin}_{\mathbb{Z}} g_\varepsilon$ と $\boldsymbol{p} \in \operatorname{argmin}_{\mathbb{Z}} g$ に対して

$$g_\varepsilon(\boldsymbol{p}^*_\varepsilon) \leq g_\varepsilon(\boldsymbol{p})$$

であるので、g_ε の定義を適用すると

$$g(\boldsymbol{p}^*_\varepsilon) \leq g(\boldsymbol{p}) + \varepsilon \sum_{i=1}^{n} \{(p(i) - \bar{p}(i))^2 - (p^*_\varepsilon(i) - \bar{p}(i))^2\}$$

の成り立つことがわかる．ここで，$\boldsymbol{p} \in \operatorname{argmin}_{\mathbb{Z}} g$ より $g(\boldsymbol{p}^*_\varepsilon) \geq g(\boldsymbol{p})$ であるから、最後の和の項は非負である．したがって,

$$\begin{aligned} g(\boldsymbol{p}^*_\varepsilon) &\leq g(\boldsymbol{p}) + \frac{g(\boldsymbol{p}') - \min g}{n\tilde{K}_\infty^2} \sum_{i=1}^{n} \{(p(i) - \bar{p}(i))^2 - (p^*_\varepsilon(i) - \bar{p}(i))^2\} \\ &< g(\boldsymbol{p}) + g(\boldsymbol{p}') - \min g = g(\boldsymbol{p}') \end{aligned}$$

表 3.1. 実装した $\mathrm{L}^\natural$ 凸関数最小化アルゴリズム

表記	アルゴリズム
SD	最急降下法 (第 3.2.1 節)
SCALING	スケーリング法 (第 3.3.4 節)
RELAX	提案する連続緩和法 (第 3.4.5 節)

が成り立つ．ここで $g(\boldsymbol{p}')$ は 2 番目に小さい値であったから、$g(\boldsymbol{p}^*_\varepsilon)$ が最小値であること、すなわち $\boldsymbol{p}^*_\varepsilon \in \operatorname{argmin}_{\mathbb{Z}} g$ であることがわかる．

連続関数の最小解が唯一の場合の事実を適用することによって、$\boldsymbol{p}^*_\varepsilon \in \operatorname{argmin}_{\mathbb{Z}} g$ が

$$\boldsymbol{p}^*_\varepsilon - n\mathbf{1} \leq \bar{\boldsymbol{p}} \leq \boldsymbol{p}^*_\varepsilon + n\mathbf{1}$$

を満たすことが証明される。 ■

3.6　実験的評価

提案する連続緩和法の性能を、既存のアルゴリズムと比較した。この数値実験から、既存のアルゴリズムよりも提案手法がはるかに高速であることがわかった．

表 3.1 に示す 3 通りの離散 $\mathrm{L}^\natural$ 凸関数最小化アルゴリズムを C 言語で実装し，性能を比較した．以下のライブラリを利用した．

- Nocedal による ‘L-BFGS’ [*1] と，工藤による C++ 言語用インタフェース [*2] は準ニュートン法の実装で，連続関数最適化を行う [44]．このルーチンには目的関数の勾配が必要であるので，差分で近似した．1 つの差分を得るのに $n+1$ 回の関数評価が必要である．このライブラリは RELAX（提案する連続緩和法）でのみ用いた．
- 岩田による ‘SFM8’ は，Iwata–Fleischer–Fujishige [35] の実装で，劣モジュラ関数を高速に最小化する．必要な関数評価回数は，劣モジュラ関数の最大の絶対値を M として，$\mathrm{O}(n^5 \log_2 M)$ 回である．
- 斎藤・松本による ‘SIMD-oriented Fast Mersenne Twister’ [*3] は擬似乱数を生成する．問題例を生成するのに利用した．

問題例に用いた離散 $\mathrm{L}^\natural$ 凸関数は

$$g(\boldsymbol{p}) = \sum_{i=1}^{n} h_i(p(i)) + \sum_{1 \leq i < j \leq n} h_{ij}(p(i) - p(j)) \quad (\boldsymbol{p} \in \mathbb{Z}^n)$$

[*1] http://www.ece.northwestern.edu/~nocedal/lbfgs.html
[*2] http://chasen.org/~taku/software/misc/lbfgs/
[*3] http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/SFMT/

と表されるものである．ただし，

$$h_i(z) = a_i(z - c_i)^2 + b_i(z - c_i), \quad h_{ij}(z) = a_{ij}z^2 + b_{ij}z$$

は 1 変数関数である．連続緩和法では，連続緩和の $\mathrm{L}^\natural$ 凸関数として

$$G(\boldsymbol{p}) = \sum_{i=1}^{n} h_i(p(i)) + \sum_{1 \le i < j \le n} h_{ij}(p(i) - p(j)) \quad (\boldsymbol{p} \in \mathbb{R}^n)$$

を用いた．1 つの n につき 10 問の問題例を生成した．整数の係数は、

$$1 \le a_i,\, a_{ij} \le n,$$
$$-n^2 \le b_i,\, c_i,\, b_{ij} \le n^2$$

の範囲から一様ランダムに選んだ．初期解は，各問題例ごとに

$$-10n \le p_0(i) \le 10n$$

を満たす整数格子点からランダムに選んだ．

計算機環境は HP dx5150 SF/CT, AMD Athlon 64 3200+ processor (2.0GHz, 512KB L2 cache), 4GB memory, Vine Linux 4.1 (kernel 2.6.16), gcc 3.3.6 である．

ここで実装した 3 種類のアルゴリズム（SD, SCALING, RELAX）では，どれも最小化したい $\mathrm{L}^\natural$ 凸関数の値のみを用いており、関数の内部構造は利用していない．そして問題ごとに，最小化に必要となった関数評価回数と計算時間を測定した．数値計算の結果は図 3.3 にまとめた．図 3.3 の上のグラフは両対数グラフであり，縦軸は関数評価回数 C，横軸は関数の次元 n である．図 3.3 の下のグラフも両対数グラフであり，縦軸は計算時間 T，横軸は関数の次元 n である．これより，関数評価回数 C について、すべてのアルゴリズムで $\log C$ と $\log n$ が比例していることが読み取れる．つまり，ある l を用いて $C = \mathrm{O}(n^l)$ と表せることになる．計算時間 T についても同様に $T = \mathrm{O}(n^l)$ と表せる．l を最小二乗法であてはめて求めた値を、表 3.2 にまとめた．関数評価回数のオーダーでも RELAX がもっとも小さいことがわかる。

以上のように，計算機実験によって，ランダムなテスト問題に対して，連続緩和法は先行手法よりも高速に $\mathrm{L}^\natural$ 凸関数を最小化できることが確かめられた．

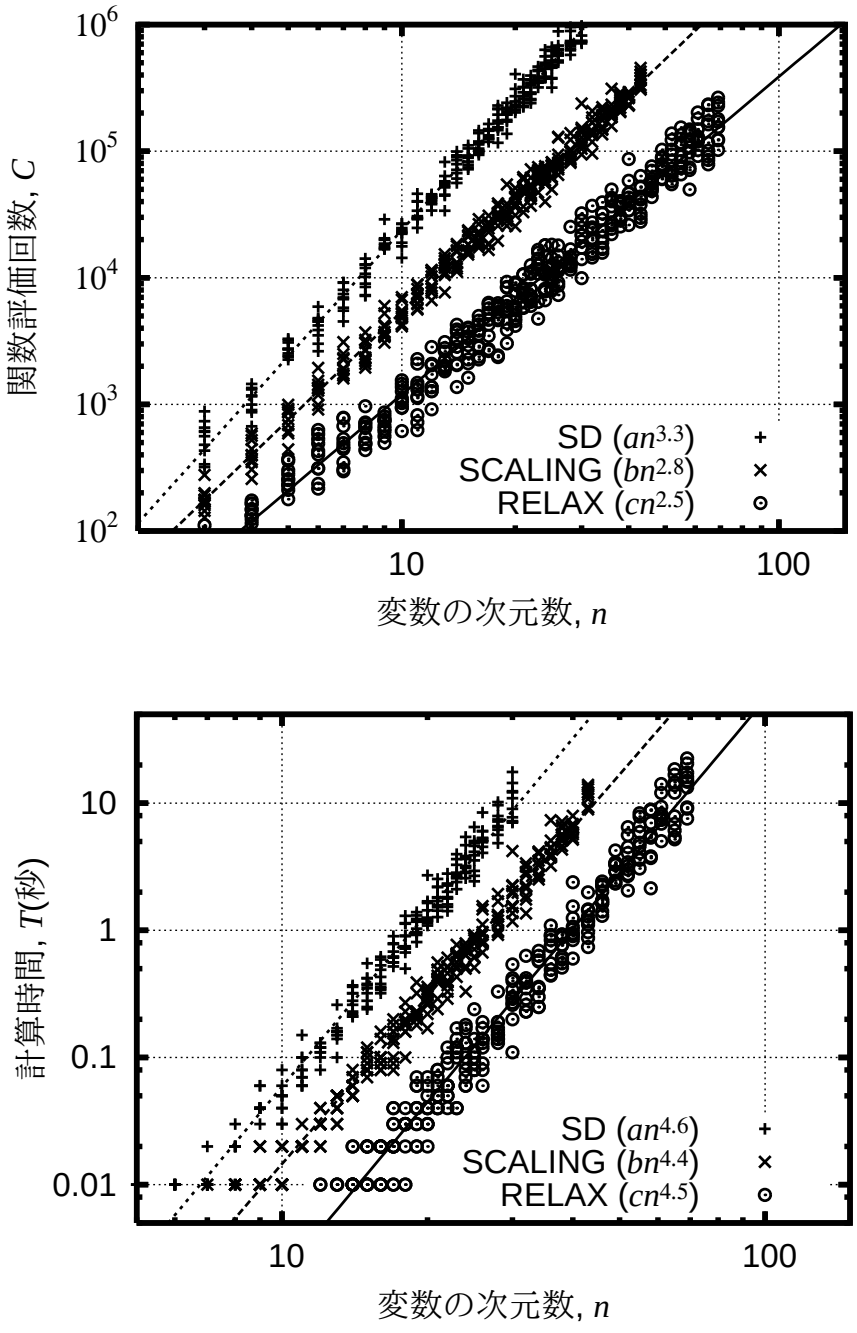


図 3.3. $\mathrm{L}^\natural$ 凸関数最小化に必要な目的関数の評価回数と計算時間

表 3.2. $\mathrm{L}^\natural$ 凸関数最小化にかかった計算量（測定値）

アルゴリズム	SD	SCALING	RELAX
関数評価回数 C	$n^{3.3}$	$n^{2.8}$	$n^{2.5}$
計算時間 T	$n^{4.6}$	$n^{4.4}$	$n^{4.5}$

第 4 章

M 凸関数の連続緩和と最小化

4.1 概要

既に述べたように、離散凸解析においては、L$^\natural$ 凸/L 凸関数と M$^\natural$ 凸/M 凸関数という 2 種類の離散凸性が定義され、互いに共役な関係にあることが明らかにされている。このうち M$^\natural$ 凸/M 凸関数はマトロイドと密接な関係にあり、M$^\natural$ 凸/M 凸関数の最小化問題は、最小木問題や最小凸費用流問題、資源配分問題などを含む実用上も重要な問題である。この章では、離散 M$^\natural$ 凸/M 凸関数の最小化問題に対して、連続緩和手法を適用することを考える。

離散 M$^\natural$ 凸関数とは、関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ のうち、交換公理（式 (2.56)）

> (M$^\natural$-EXC[$\mathbb{Z}$]) 任意の $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^n$ と任意の $i \in \mathrm{supp}^+(\boldsymbol{x} - \boldsymbol{y})$ に対して，ある $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y}) \cup \{0\}$ が存在して
>
> $$f(\boldsymbol{x}) + f(\boldsymbol{y}) \geq f(\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j) + f(\boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_j) \tag{4.1}$$

を満たすものである。

離散 M 凸関数とは、関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ のうち、交換公理（式 (2.58)）

> (M-EXC[$\mathbb{Z}$]) 任意の $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^n$ と任意の $i \in \mathrm{supp}^+(\boldsymbol{x} - \boldsymbol{y})$ に対して，ある $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y})$ が存在して
>
> $$f(\boldsymbol{x}) + f(\boldsymbol{y}) \geq f(\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j) + f(\boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_j) \tag{4.2}$$

を満たすものである。M$^\natural$ 凸関数と M 凸関数は等価に書き換えができるので（第 2.3.2 節参照）、最小化アルゴリズムとしては離散 M 凸関数について記述する。

離散 M 凸関数最小化（あるいは類似する最小化問題）のアルゴリズムとしては、大きく分けて、(a) 素朴な降下法、(b) スケーリング法、(c) 連続緩和法の 3 種類が考えられる。（また、それぞれに領域縮小法を組み合わせることもある。）3 種類のアルゴリズムは、それぞれ (a) 最小性規準、(b) 格子間隔を粗くした場合の近接定理、(c) 格子間隔を無限に細かくした場合の近接定理に基づいたものである。

本章では、離散 M 凸関数の枠組みとして、(c) 連続緩和法を提案する。そのために必要な、格子間隔を無限に細かくした場合の近接定理も証明する。そして、M 凸関数最小化問題の特

殊ケースである資源配分問題のいくつかの問題クラスについて、個別の近接定理を証明して計算量を詳しく解析する。

資源配分問題の特殊ケースのいくつかの問題クラスに対しては、例えば Hochbaum–Hong [30] によって連続緩和法が提案されており、目的関数が 2 次の場合の計算量が解析されている。本章では, この手法の適用対象を M 凸関数最小化に一般化するものである。一般化した結果、対象全体の最悪ケースの計算量を改善できることが証明されているわけではないが、いくつか重要な問題クラスに対しては計算量を改善する。なお、計算量の解析においては、M 凸関数の評価回数を規準にするだけでなく、資源配分問題の特殊ケースに対しては、目的関数の構造を規定した上で関数値の算出にかかる計算量を削減する工夫を行い、先行研究と比較する。

4.1.1　M 凸関数最小化問題

M 凸関数最小化問題の定式化を行う．M 凸関数最小化問題は

$$\text{(MC)} \qquad \text{Minimize} \quad f(\boldsymbol{x}) \qquad \text{subject to} \quad \boldsymbol{x} \in \mathrm{dom}_{\mathbb{Z}} f$$

と定式化される．ここで $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は M 凸関数である．次の性質は (MC) に対する連続緩和の定義に有用である．

定理 4.1 ([62, 第 6.11 節])**.**
(1) 任意の離散 M$^\natural$ 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ に対して，ある連続変数の閉真 M$^\natural$ 凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が存在して，任意の $\boldsymbol{x} \in \mathbb{Z}^n$ に対して $F(\boldsymbol{x}) = f(\boldsymbol{x})$ を満たし，また $\mathrm{dom}_{\mathbb{R}} F$ が $\mathrm{dom}_{\mathbb{Z}} f$ の閉凸包となる．
(2) 任意の離散 M 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ に対して，ある連続変数の閉真 M 凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が存在して，任意の $\boldsymbol{x} \in \mathbb{Z}^n$ に対して $F(\boldsymbol{x}) = f(\boldsymbol{x})$ を満たし，また $\mathrm{dom}_{\mathbb{R}} F$ が $\mathrm{dom}_{\mathbb{Z}} f$ の閉凸包となる． ■

この定理の関数 F の例としては、式 (3.10) で定義した f の連続化（凸閉包）が挙げられる。
この性質に基づいて，(MC) の連続緩和 ($\overline{\text{MC}}$) を

$$(\overline{\text{MC}}) \qquad \text{Minimize} \quad F(\boldsymbol{x}) \qquad \text{subject to} \quad \boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F$$

と定義する．ただし $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ は閉真 M 凸関数であり，すべての $\boldsymbol{x} \in \mathbb{Z}^n$ に対して $F(\boldsymbol{x}) = f(\boldsymbol{x})$ を満たし，また $\mathrm{dom}_{\mathbb{R}} F$ が $\mathrm{dom}_{\mathbb{Z}} f$ の閉凸包とする．

4.2　最小化アルゴリズムの基本形

4.2.1　離散変数関数の場合

離散 M$^\natural$ 凸/M 凸関数には最小性規準があることを、定理 2.25 と定理 2.27 で次のように述べた。

定理 4.2.

(1) 関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が $\mathrm{M}^\natural$ 凸関数のとき，$\boldsymbol{x} \in \mathrm{dom}_{\mathbb{Z}} f$ が f の最小点であるためには，任意の $i, j \in \{0, 1, \ldots, n\}$ に対して

$$f(\boldsymbol{x}) \leq f(\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j)$$

となることが必要十分である．

(2) 関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が M 凸関数のとき，$\boldsymbol{x} \in \mathrm{dom}_{\mathbb{Z}} f$ が f の最小点であるためには，任意の $i, j \in \{1, \ldots, n\}$ に対して

$$f(\boldsymbol{x}) \leq f(\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j)$$

となることが必要十分である． ■

離散 M 凸関数を最小化するアルゴリズムとしては，この定理 4.2(2) による大域最小性の局所的な特徴付けから，自然に最急降下法 [62, 第 10.1.1 節] を導くことができる．

M 凸関数の最小化アルゴリズム（最急降下法）：$\mathrm{SD}(f, \boldsymbol{x})$
Input: 離散 M 凸関数 f と初期解 $\boldsymbol{x} \in \mathrm{dom}_{\mathbb{Z}} f$
Output: f の一つの最小解
Step 1: $f(\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j)$ を最小化する $i, j \in \{1, \ldots, n\} (i \neq j)$ を探す．
Step 2: もし $f(\boldsymbol{x}) \leq f(\boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j)$ であれば，$\boldsymbol{x}$ を返す（$\boldsymbol{x}$ は f の最小解の一つ）．
Step 3: $\boldsymbol{x} := \boldsymbol{x} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j$ として Step 1 に戻る．

アルゴリズムの計算量解析のために、関数 f の実効定義域 $\mathrm{dom}_{\mathbb{Z}} f$ は有界であると仮定し、

$$K_1 = \max\{\|\boldsymbol{x} - \boldsymbol{y}\|_1 \mid \boldsymbol{x}, \boldsymbol{y} \in \mathrm{dom}_{\mathbb{Z}} f\} \tag{4.3}$$

とおく．また f の関数値は、定数時間 T_{func} で得られるものと仮定する。

手順 Step 1，つまり定理 4.2(2) の検証は，$\mathrm{O}(n^2 T_{\mathrm{func}})$ の計算時間で終了する。手順 Step 1 から Step 3 までの反復回数は、実効定義域の大きさに比例して $\mathrm{O}(K_1)$ である。したがって、最急降下法では f の最小解は $\mathrm{O}(n^2 T_{\mathrm{func}} K_1)$ の計算時間で求まる．つまり，最急降下法は（多項式時間ではなく）擬多項式時間アルゴリズムである．

4.2.2 連続変数関数の場合

連続変数の $\mathrm{M}^\natural$ 凸/M 凸関数にも、離散関数と同様の最小性規準があることを、定理 2.37 で次のように述べた。

定理 4.3.

(1) 関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が $\mathrm{M}^\natural$ 凸関数のとき，$\boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F$ が F の最小点であるためには，任意の $i, j \in \{0, 1, \ldots, n\}$ に対して

$$F'(\boldsymbol{x}; -\boldsymbol{\chi}_i + \boldsymbol{\chi}_j) \geq 0$$

となることが必要十分である．

(2) 関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が M 凸関数のとき，$\boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F$ が F の最小点であるためには，任意の $i, j \in \{1, \ldots, n\}$ に対して

$$F'(\boldsymbol{x}; -\boldsymbol{\chi}_i + \boldsymbol{\chi}_j) \geq 0$$

となることが必要十分である． ■

最小化における連続 $\mathrm{M}^\natural$ 凸/M 凸関数の状況は、連続 $\mathrm{L}^\natural$ 凸/L 凸関数とまったく同じである。つまり、この最小性規準の存在にもかかわらず、連続 $\mathrm{M}^\natural$ 凸/M 凸関数には特徴的な最小化アルゴリズムが確立されているわけではない。連続 $\mathrm{M}^\natural$ 凸/M 凸関数の最小化においても、一般の凸関数の最小化で用いられる準ニュートン法のようなアルゴリズムが用いられるが、離散 $\mathrm{M}^\natural$ 凸/M 凸関数に比べて最小化にかかる計算量は少ないと言える。

4.2.3　M 凸関数最小化問題の新しい貪欲アルゴリズム

M 凸関数最小化問題 (MC) に対する新しい貪欲アルゴリズムを提案する [53]．この貪欲アルゴリズムは，第 4.4.3 節で述べる連続緩和法で用いる．文献 [52, 88] で提案された既存の貪欲アルゴリズムと比べると，類似点は多いが，より高速に動作する．

このアルゴリズムは，定理 4.2(2) で述べた問題 (MC) の最小性規準を利用している．この局所的な性質から、M 凸関数の最小解を含む領域に関する次の有益な定理が得られる．なお、第 2.1.1 節で述べたように $N = \{1, 2, \ldots, n\}$ とする。

定理 4.4 ([87, 定理 2.2])**.** $\mathrm{argmin}_{\mathbb{Z}} f \neq \emptyset$ と仮定する．$\boldsymbol{y} \in \mathrm{dom}_{\mathbb{Z}} f$ と $h \in N$ について，次が成り立つ．

(i) 成分 $i \in N$ が

$$f(\boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_h) = \min_{i' \in N} f(\boldsymbol{y} + \boldsymbol{\chi}_{i'} - \boldsymbol{\chi}_h)$$

の条件を満たすとき，ある $\boldsymbol{y}_* \in \mathrm{argmin}_{\mathbb{Z}} f$ が存在して

$$y_*(i) \geq (\boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_h)(i) = y(i) + 1 - \chi_h(i)$$

を満たす [*1]．

(ii) 成分 $j \in N$ が

$$f(\boldsymbol{y} + \boldsymbol{\chi}_h - \boldsymbol{\chi}_j) = \min_{j' \in N} f(\boldsymbol{y} + \boldsymbol{\chi}_h - \boldsymbol{\chi}_{j'})$$

の条件を満たすとき，ある $\boldsymbol{y}_* \in \mathrm{argmin}_{\mathbb{Z}} f$ が存在して

$$y_*(j) \leq (\boldsymbol{y} + \boldsymbol{\chi}_h - \boldsymbol{\chi}_j)(j) = y(j) - 1 + \chi_h(j)$$

を満たす． ■

[*1] $\chi_h(i)$ の値は、$i = h$ のとき 1、$i \neq h$ のとき 0 であることに注意。

次に (MC) に対するアルゴリズムを述べる．関数の定義域に含まれる初期ベクトル $\boldsymbol{y}^\circ \in \mathrm{dom}_{\mathbb{Z}} f$ が与えられているものとする．

M 凸関数の最小化アルゴリズム: NewGreedy$(f, \boldsymbol{y}^\circ)$
Input: 離散 M 凸関数 f と初期解 $\boldsymbol{y}^\circ \in \mathrm{dom}_{\mathbb{Z}} f$
Output: f の一つの最小解

> **Step 0:** $\boldsymbol{y} := \boldsymbol{y}^\circ$ とし，すべての要素 $k \in N$ について $\ell(k) := -\infty$, $u(k) := +\infty$ とする．
> **Step 1:** $\ell(h) < u(h)$ を満たす要素 $h \in N$ を任意に選ぶ．
> **Step 2:** $\boldsymbol{\ell} \le \boldsymbol{y} + \boldsymbol{\chi}_{i_1} - \boldsymbol{\chi}_h \le \boldsymbol{u}$ の制約の下で $f(\boldsymbol{y} + \boldsymbol{\chi}_{i_1} - \boldsymbol{\chi}_h)$ を最小化する要素 $i_1 \in N$ を求める．
> **Step 3:** $\boldsymbol{\ell} \le \boldsymbol{y} - \boldsymbol{\chi}_{i_2} + \boldsymbol{\chi}_h \le \boldsymbol{u}$ の制約の下で $f(\boldsymbol{y} - \boldsymbol{\chi}_{i_2} + \boldsymbol{\chi}_h)$ を最小化する要素 $i_2 \in N$ を求める．
> **Step 4:** もし $h = i_1 = i_2$ であれば Step 5 に進む．もし $h \neq i_1$ であれば Step 6 に進む．どちらでもなければ（つまり $h \neq i_2$ であれば）Step 7 に進む．
> **Step 5:** $\ell(h) := y(h)$, $u(h) := y(h)$ と更新して Step 8 に進む．
> **Step 6:** $\boldsymbol{\ell} \le \boldsymbol{y} + \boldsymbol{\chi}_{i_1} - \boldsymbol{\chi}_{j_1} \le \boldsymbol{u}$ の制約の下で $f(\boldsymbol{y} + \boldsymbol{\chi}_{i_1} - \boldsymbol{\chi}_{j_1})$ を最小化する要素 $j_1 \in N \setminus \{i_1\}$ を求める．$\ell(i_1) := y(i_1) + 1$, $u(j_1) := y(j_1) - 1$, $\boldsymbol{y} := \boldsymbol{y} + \boldsymbol{\chi}_{i_1} - \boldsymbol{\chi}_{j_1}$ と更新して Step 8 に進む．
> **Step 7:** $\boldsymbol{\ell} \le \boldsymbol{y} - \boldsymbol{\chi}_{i_2} + \boldsymbol{\chi}_{j_2} \le \boldsymbol{u}$ の制約の下で $f(\boldsymbol{y} - \boldsymbol{\chi}_{i_2} + \boldsymbol{\chi}_{j_2})$ を最小化する要素 $j_2 \in N \setminus \{i_2\}$ を求める．$u(i_2) := y(i_2) - 1$, $\ell(j_2) := y(j_2) + 1$, $\boldsymbol{y} := \boldsymbol{y} - \boldsymbol{\chi}_{i_2} + \boldsymbol{\chi}_{j_2}$ と更新して Step 8 に進む．
> **Step 8:** もしすべての要素 $k \in N$ について $\ell(k) = u(k)$ を満たせば，f の最小解の 1 つとして $\boldsymbol{y}$ を出力して終了する．そうでなければ Step 1 に戻る．

以下では、まずこのアルゴリズムの正当性を述べ、次に計算量を解析する。

正当性

補題 4.5. $\boldsymbol{y} \in \mathrm{dom}_{\mathbb{Z}} f$ と $h \in N$ について，次が成り立つ．
(i) ある $\boldsymbol{y}_* \in \mathrm{argmin}_{\mathbb{Z}} f$ が $y_*(h) \le y(h) - 1$ を満たすとする．このとき，$i \in N$ が存在して $i \neq h$ かつ $f(\boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_h) = \min_{i' \in N} f(\boldsymbol{y} + \boldsymbol{\chi}_{i'} - \boldsymbol{\chi}_h)$ を満たす．
(ii) ある $\boldsymbol{y}_* \in \mathrm{argmin}_{\mathbb{Z}} f$ が $y_*(h) \ge y(h) + 1$ を満たすとする．このとき，$j \in N$ が存在して $j \neq h$ かつ $f(\boldsymbol{y} + \boldsymbol{\chi}_h - \boldsymbol{\chi}_j) = \min_{j' \in N} f(\boldsymbol{y} + \boldsymbol{\chi}_h - \boldsymbol{\chi}_{j'})$ を満たす． ■

証明. (i) だけを証明する．(ii) は同様に示せる．$f(\boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_h) \le f(\boldsymbol{y})$ を満たす $i \in N \setminus \{h\}$ の存在を示せば十分である．$h \in \mathrm{supp}^+(\boldsymbol{y} - \boldsymbol{y}_*)$ であるので，f の交換公理 (M-EXC[$\mathbb{Z}$]) によって，

$$f(\boldsymbol{y}) + f(\boldsymbol{y}_*) \ge f(\boldsymbol{y} - \boldsymbol{\chi}_h + \boldsymbol{\chi}_i) + f(\boldsymbol{y}_* + \boldsymbol{\chi}_h - \boldsymbol{\chi}_i) \tag{4.4}$$

を満たす $i \in \mathrm{supp}^-(\boldsymbol{y} - \boldsymbol{y}_*)$ の存在が保証される．また $\boldsymbol{y}_* \in \mathrm{argmin}_{\mathbb{Z}} f$ であるので，

$f(\boldsymbol{y}_* + \boldsymbol{\chi}_h - \boldsymbol{\chi}_i) \geq f(\boldsymbol{y}_*)$ となり，これと式 (4.4) より $f(\boldsymbol{y}) \geq f(\boldsymbol{y} - \boldsymbol{\chi}_h + \boldsymbol{\chi}_i)$ となる．このとき $i \neq h$ であることに注意する． ■

補題 4.6. $i, j \in N$ と $\alpha, \beta \in \mathbb{Z}$ について，次が成り立つ．ただし，ある $\boldsymbol{y}', \boldsymbol{y}'' \in \operatorname{argmin}_{\mathbb{Z}} f$ が存在して $y'(i) \geq \alpha$ かつ $y''(j) \leq \beta$ とする．
(i) $j \neq i$ であれば，ある $\boldsymbol{y}_* \in \operatorname{argmin}_{\mathbb{Z}} f$ が存在して，$y_*(i) \geq \alpha$ かつ $y_*(j) \leq \beta$ を満たす．
(ii) $j = i$ かつ $\alpha \leq \beta$ であれば，ある $\boldsymbol{y}_* \in \operatorname{argmin}_{\mathbb{Z}} f$ が存在して，$\alpha \leq y_*(i) \leq \beta$ を満たす． ■

証明. $\boldsymbol{y}'$ は $\boldsymbol{y}'(i) \geq \alpha$ を満たす f の最小解であり，$\boldsymbol{y}''$ は $\boldsymbol{y}''(j) \leq \beta$ を満たす f の最小解であり，$\boldsymbol{y}''$ が $\boldsymbol{y}''(j) \leq \beta$ となるすべての最小解の中で $\boldsymbol{y}''(i)$ の値を最大にするとする．ここで $\boldsymbol{y}''(i) < \alpha$ と仮定し，矛盾を導く．$i \in \operatorname{supp}^+(\boldsymbol{y}' - \boldsymbol{y}'')$ であるので，f の交換公理 (M-EXC[$\mathbb{Z}$]) によって

$$f(\boldsymbol{y}') + f(\boldsymbol{y}'') \geq f(\boldsymbol{y}' - \boldsymbol{\chi}_i + \boldsymbol{\chi}_h) + f(\boldsymbol{y}'' + \boldsymbol{\chi}_i - \boldsymbol{\chi}_h)$$

を満たす $h \in \operatorname{supp}^-(\boldsymbol{y}' - \boldsymbol{y}'')$ が存在することが保証される．$\boldsymbol{y}', \boldsymbol{y}'' \in \operatorname{argmin}_{\mathbb{Z}} f$ であるので，この不等式から $\boldsymbol{y}' - \boldsymbol{\chi}_i + \boldsymbol{\chi}_h, \boldsymbol{y}'' + \boldsymbol{\chi}_i - \boldsymbol{\chi}_h \in \operatorname{argmin}_{\mathbb{Z}} f$ であることがわかる．$\boldsymbol{y}_* = \boldsymbol{y}'' + \boldsymbol{\chi}_i - \boldsymbol{\chi}_h$ とおく．$j \neq i$ のとき，$\boldsymbol{y}_* \in \operatorname{argmin}_{\mathbb{Z}} f$ となり，$y_*(j) \leq y''(j) \leq \beta$，従って $y_*(i) > y''(i)$ となり，$\boldsymbol{y}''$ の仮定に矛盾する．$j = i$ かつ $\alpha \leq \beta$ のとき，$\boldsymbol{y}_* \in \operatorname{argmin}_{\mathbb{Z}} f$ となり，$y''(i) < y_*(i) \leq \alpha \leq \beta$ となり，$\boldsymbol{y}''$ の仮定に矛盾する．従って，いずれの場合も $\boldsymbol{y}''(i) \geq \alpha$ が成り立つ． ■

ベクトル $\boldsymbol{\ell} \in (\mathbb{Z} \cup \{-\infty\})^n$ と $\boldsymbol{u} \in (\mathbb{Z} \cup \{+\infty\})^n$ に対して，f を区間 $[\boldsymbol{\ell}, \boldsymbol{u}]$ に制限した関数 $f_{\boldsymbol{\ell}}^{\boldsymbol{u}} : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ を

$$f_{\boldsymbol{\ell}}^{\boldsymbol{u}}(\boldsymbol{y}) = \begin{cases} f(\boldsymbol{y}) & (\boldsymbol{\ell} \leq \boldsymbol{y} \leq \boldsymbol{u}), \\ +\infty & (\text{その他}) \end{cases} \qquad (\boldsymbol{y} \in \mathbb{Z}^n) \tag{4.5}$$

と定義する．関数の M 凸性は，この制限操作によっても保存される．

補題 4.7 ([57, 補題 2.5])**.** ベクトル $\boldsymbol{\ell} \in (\mathbb{Z} \cup \{-\infty\})^n$ と $\boldsymbol{u} \in (\mathbb{Z} \cup \{+\infty\})^n$ について，式 (4.5) によって定義された関数 $f_{\boldsymbol{\ell}}^{\boldsymbol{u}} : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は、

$$\operatorname{dom}_{\mathbb{Z}} f_{\boldsymbol{\ell}}^{\boldsymbol{u}} = \{\boldsymbol{y} \in \mathbb{Z}^n \mid \boldsymbol{y} \in \operatorname{dom}_{\mathbb{Z}} f,\ \boldsymbol{\ell} \leq \boldsymbol{y} \leq \boldsymbol{u}\} \neq \emptyset$$

である限り、M 凸関数になる。 ■

アルゴリズムの正当性を証明する．

補題 4.8. アルゴリズム NewGreedy において、区間 $[\boldsymbol{\ell}, \boldsymbol{u}]$ は常に f の最小解の 1 つを含む． ■

証明. 反復回数についての数学的帰納法を用いて証明する．アルゴリズムの最初の段階で，$[\boldsymbol{\ell}, \boldsymbol{u}]$ が f の最小解を含むことは明らかである．

まず，Step 4 で $h = i_1 = i_2$ が成り立つ場合を考える．関数 $f_{\boldsymbol{\ell}}^{\boldsymbol{u}}$ が式 (4.5) で定義されているとすると，関数 $f_{\boldsymbol{\ell}}^{\boldsymbol{u}}$ は補題 4.7 より M 凸であることがわかる．帰納法の仮定により，$\operatorname{argmin}_{\mathbb{Z}} f_{\boldsymbol{\ell}}^{\boldsymbol{u}} \subseteq \operatorname{argmin}_{\mathbb{Z}} f$ が成り立つ．定理 4.4 を $f_{\boldsymbol{\ell}}^{\boldsymbol{u}}$ に適用すると，ある $\boldsymbol{y}', \boldsymbol{y}'' \in \operatorname{argmin}_{\mathbb{Z}} f_{\boldsymbol{\ell}}^{\boldsymbol{u}}$ が存在して，$y'(h) \geq y(h)$ かつ $y''(h) \leq y(h)$ を満たすことがわかる．そして，補題 4.6 (ii) より，$y(h) \leq y_*(h) \leq y(h)$ を満たす $\boldsymbol{y}_* \in \operatorname{argmin}_{\mathbb{Z}} f_{\boldsymbol{\ell}}^{\boldsymbol{u}}$ の存在することがわかる．従って，Step 5 の $\ell(h)$ と $u(h)$ の更新の後に，$[\boldsymbol{\ell}, \boldsymbol{u}]$ は f 最小解を含む．

次に，Step 4 で $h \neq i_1$ となる場合を考える．定理 4.4 (i) を $f_{\boldsymbol{\ell}}^{\boldsymbol{u}}$ に適用すると，ある $\boldsymbol{y}' \in \operatorname{argmin}_{\mathbb{Z}} f_{\boldsymbol{\ell}}^{\boldsymbol{u}}$ が存在して，$y'(i_1) \geq y(i_1) + 1$ を満たすことがわかる．定理 4.4 (ii) と補題 4.5 (ii) より，ある $\boldsymbol{y}'' \in \operatorname{argmin}_{\mathbb{Z}} f_{\boldsymbol{\ell}}^{\boldsymbol{u}}$ が存在して，$y''(j_1) \leq y(j_1) - 1$ を満たすことがわかる．そして補題 4.6 (i) は，ある $\boldsymbol{y}_* \in \operatorname{argmin}_{\mathbb{Z}} f_{\boldsymbol{\ell}}^{\boldsymbol{u}}$ が存在して，$y_*(i_1) \geq y(i_1) + 1$ と $y_*(j_1) \leq y(j_1) - 1$ を満たすことを保証する．従って，Step 6 の $\ell(i_1)$ と $u(j_1)$ の更新の後，$[\boldsymbol{\ell}, \boldsymbol{u}]$ は f の最小解を含むことがわかる．

Step 4 で $h \neq i_2$ となる場合も同様である．よって補題の主張は証明された． ■

アルゴリズムの出力を $\boldsymbol{y}_{\mathrm{out}}$ と表す．アルゴリズムが終了するときには，ベクトル $\boldsymbol{y}_{\mathrm{out}}$ は $\boldsymbol{y}_{\mathrm{out}} = \boldsymbol{\ell} = \boldsymbol{u}$ となる，つまり $\boldsymbol{y}_{\mathrm{out}}$ は $[\boldsymbol{\ell}, \boldsymbol{u}]$ に含まれる唯一のベクトルとなる．

ゆえに，補題 4.8 より $\boldsymbol{y}_{\mathrm{out}}$ は f の最小解となる．

反復回数

ここからは，反復回数を解析する．

補題 4.9. Step 6 あるいは Step 7 が実行されると，$\|\boldsymbol{y} - \boldsymbol{y}_{\mathrm{out}}\|_1$ は 2 だけ減少する． ■

証明. アルゴリズムのどの反復においても，ベクトル $\boldsymbol{\ell}$ は非減少であり，$\boldsymbol{u}$ は非増加であるので，ベクトル $\boldsymbol{y}_{\mathrm{out}}$ は常に区間 $[\boldsymbol{\ell}, \boldsymbol{u}]$ に含まれている．Step 6 が実行されるときを考える．(Step 7 についても同じように扱える．) Step 6 で，更新前のベクトル $\boldsymbol{y}$ を $\boldsymbol{y}_{\mathrm{old}}$，更新後を $\boldsymbol{y}_{\mathrm{new}}$ で表す．同様に更新後のベクトル $\boldsymbol{\ell}$ と $\boldsymbol{u}$ を $\boldsymbol{\ell}_{\mathrm{new}}$ と $\boldsymbol{u}_{\mathrm{new}}$ で表すと，

$$\begin{aligned} y_{\mathrm{out}}(i_1) \geq \ell_{\mathrm{new}}(i_1) = y_{\mathrm{old}}(i_1) + 1 = y_{\mathrm{new}}(i_1) > y_{\mathrm{old}}(i_1), \\ y_{\mathrm{out}}(j_1) \leq u_{\mathrm{new}}(j_1) = y_{\mathrm{old}}(j_1) - 1 = y_{\mathrm{new}}(j_1) < y_{\mathrm{old}}(j_1) \end{aligned}$$

となる．ゆえに，$\|\boldsymbol{y}_{\mathrm{new}} - \boldsymbol{y}_{\mathrm{out}}\|_1 = \|\boldsymbol{y}_{\mathrm{old}} - \boldsymbol{y}_{\mathrm{out}}\|_1 - 2$ が成り立つ． ■

補題 4.10. アルゴリズム NewGreedy は，$\mathrm{O}(n + \|\boldsymbol{y}^\circ - \boldsymbol{y}_{\mathrm{out}}\|_1)$ 回の反復で終了する． ■

証明. アルゴリズムのどの反復においても，ベクトル $\boldsymbol{\ell}$ は非減少であり，$\boldsymbol{u}$ は非増加であるので，ある $h \in N$ について一度 $\ell(h) = u(h)$ が成り立つと，その後の反復においても常に成立ち続ける．ゆえに，Step 5 は高々 n 回実行される．加えて，$\|\boldsymbol{y} - \boldsymbol{y}_{\mathrm{out}}\|_1$ の値は Step 5 を実行しても変化しない．補題 4.9 により，Step 6 と Step 7 のどちらを実行しても $\|\boldsymbol{y} - \boldsymbol{y}_{\mathrm{out}}\|_1$ の値は 2 だけ減る．$\|\boldsymbol{y} - \boldsymbol{y}_{\mathrm{out}}\|_1$ の初期値は $\|\boldsymbol{y}^\circ - \boldsymbol{y}_{\mathrm{out}}\|_1$ であるので，Step 6 と Step 7 は $\|\boldsymbol{y}^\circ - \boldsymbol{y}_{\mathrm{out}}\|_1/2$ 回実行される．ゆえに，アルゴリズム NewGreedy は $\mathrm{O}(n + \|\boldsymbol{y}^\circ - \boldsymbol{y}_{\mathrm{out}}\|_1)$ 回の反復で終了する． ■

アルゴリズムの各反復は $\mathrm{O}(nT_{\mathrm{func}})$ の計算時間で実行できることが容易にわかる．ただし T_{func} は，与えられた M 凸関数 f の，与えられたベクトル $\boldsymbol{y} \in \mathbb{Z}^n$ における関数値 $f(\boldsymbol{y})$ を評価するのにかかる計算時間を表す．ゆえに，次の結果が得られる．

定理 4.11. アルゴリズム NewGreedy は $\mathrm{O}(nT_{\mathrm{func}}(n + \|\boldsymbol{y}^\circ - \boldsymbol{y}_{\mathrm{out}}\|_1))$ の計算時間で M 凸関数 f の最小解の 1 つを出力する．■

摂動

定理 4.11 より，アルゴリズム NewGreedy にかかる計算時間は，初期ベクトル $\boldsymbol{y}^\circ$ と，アルゴリズムで求められる M 凸関数 f の最小解 $\boldsymbol{y}_{\mathrm{out}}$ の間の距離に依存することがわかる．このため初期ベクトルに近い最小解を求めることに意義がある．文献 [63] で提案された修正と同様の工夫をすると，このアルゴリズムの求める f の最小解が，常に $\boldsymbol{y}^\circ$ からの L_1 距離を最小にするものになる．

その工夫とは，元の関数 f の代わりに，摂動関数

$$f_\varepsilon(\boldsymbol{y}) = f(\boldsymbol{y}) + \varepsilon\|\boldsymbol{y}^\circ - \boldsymbol{y}\|_1 \qquad (\boldsymbol{y} \in \mathbb{Z}^n)$$

を用いることである．ただし ε は十分に小さい正の数である．$\varepsilon\|\boldsymbol{y}^\circ - \boldsymbol{y}\|_1$ は $\boldsymbol{y}$ についての分離凸関数であり，M 凸関数と分離凸関数の和は M 凸関数であるので，f_ε は M 凸関数である．ε を十分小さい正数に選んでおけば，$\boldsymbol{y}_* \in \operatorname{argmin}_{\mathbb{Z}} f_\varepsilon$ であるためには，$\boldsymbol{y}_* \in \operatorname{argmin}_{\mathbb{Z}} f$ かつ $\|\boldsymbol{y}^\circ - \boldsymbol{y}_*\|_1 = \min\{\|\boldsymbol{y}^\circ - \boldsymbol{y}\|_1 \mid \boldsymbol{y} \in \operatorname{argmin}_{\mathbb{Z}} f\}$ の成り立つことが必要十分である．ゆえに，f_ε の最小解を求めれば十分である．

アルゴリズム NewGreedy を摂動関数 f_ε に適用する．これは，ε を陽に導入しなくても、手順 Step 2, Step 3, Step 6, Step 7 に次のルールを加えることで実現できる．

> **(Rule 1)** $\boldsymbol{\ell} \le \boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_h \le \boldsymbol{u}$ の制約の下で，$f(\boldsymbol{y} + \boldsymbol{\chi}_i - \boldsymbol{\chi}_h)$ を最小化する要素 i を求めるときには、もし $y(i) < y^\circ(i)$ を満たす i がみつかればそれを採用し，なければ任意の最小化する要素を採用する．
>
> **(Rule 2)** $\boldsymbol{\ell} \le \boldsymbol{y} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_h \le \boldsymbol{u}$ の制約の下で，$f(\boldsymbol{y} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_h)$ を最小化する要素 i を求めるときには、もし $y(i) > y^\circ(i)$ を満たす i がみつかればそれを採用し，なければ任意の最小化する要素を採用する．

定理 4.12. アルゴリズム NewGreedy に (Rule 1) と (Rule 2) のルールを加えると，M 凸関数 f の 1 つの最小解 $\boldsymbol{y}_*$ のうち、

$$\|\boldsymbol{y}^\circ - \boldsymbol{y}_*\|_1 = \min\{\|\boldsymbol{y}^\circ - \boldsymbol{y}\|_1 \mid \boldsymbol{y} \in \operatorname{argmin}_{\mathbb{Z}} f\}$$

を満たすものが得られる．必要な計算時間は

$$\mathrm{O}(nT_{\mathrm{func}}(n + \min\{\|\boldsymbol{y}^\circ - \boldsymbol{y}\|_1 \mid \boldsymbol{y} \in \operatorname{argmin}_{\mathbb{Z}} f\}))$$

である．■

最急降下法 SD では $O(n^2 T_{\mathrm{func}} K_1)$ の計算時間がかかるのに比べて、摂動を加えたアルゴリズム NewGreedy では，n の次数が 1 だけ小さく、反復回数に当たる項が $n + \min\{\|\boldsymbol{y}^\circ - \boldsymbol{y}\|_1 \mid \boldsymbol{y} \in \operatorname{argmin}_{\mathbb{Z}} f\}$ と最小限に抑えられている点で高速化されている。

4.3 スケーリング

4.3.1 M 凸関数に対するスケーリング

$L^\natural$ 凸/L 凸関数の場合は、スケーリングを行っても $L^\natural$ 凸/L 凸性が保持された。しかしながら $M^\natural$ 凸/M 凸関数では、スケーリングを行うと $M^\natural$ 凸/M 凸性は一般には保持されない。このことを強調するために、次のように命題の形で示す。記号 f^α は、(3.6) で定義した f のスケーリングを表す。

命題 4.13.
(1) 関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が $M^\natural$ 凸でも，f^α は $M^\natural$ 凸とは限らない.
(2) 関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が M 凸でも，f^α は M 凸とは限らない. ■

スケーリングによって $M^\natural$ 凸/M 凸性が保持される例もある。2 次関数や層凸関数 (2.83) がそのような例である。

命題 4.14.
(1) 関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が 2 次 $M^\natural$ 凸関数ならば，f^α も 2 次 $M^\natural$ 凸関数である.
(2) 関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が 2 次 M 凸関数ならば，f^α も 2 次 M 凸関数である.
(3) 関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が層凸関数ならば，f^α も層凸関数である. ■

スケーリングによって $M^\natural$ 凸/M 凸性が保持されない具体例を挙げる。

例 4.1 ([65, 例 7.13])**.** $\mathbb{Z}^4$ の部分集合

$$S = \{c_1(1,0,-1,0) + c_2(1,0,0,-1) + c_3(0,1,-1,0) + c_4(0,1,0,-1) \mid c_i \in \{0,1\}\}$$

は M 凸集合である．しかし，$2\boldsymbol{x}$ が S に含まれる整数点 $\boldsymbol{x}$ の全体

$$S^2 = \{\boldsymbol{x} \in \mathbb{Z}^4 \mid 2\boldsymbol{x} \in S\} = \{(0,0,0,0), (1,1,-1,-1)\}$$

は M 凸集合でない．これを標示関数で解釈すると、次のように述べられる。集合 S の標示関数 $f = \delta_S$ は M 凸関数である．S^2 は M 凸集合でないので，集合 S^2 の標示関数 f^2 は M 凸関数でない. ■

4.3.2 スケーリングの近接定理

$M^\natural$ 凸/M 凸関数がスケーリングを行ったときに $M^\natural$ 凸/M 凸性を保持するかどうかにかかわらず、次の近接定理が成り立つ。この定理は、$M^\natural$ 凸/M 凸関数に関して、スケーリングさ

れた関数 f^α の極小点から (3.7) のように計算した点 $\boldsymbol{x}^\alpha$ の近くに，f の最小点 $\boldsymbol{x}^*$ が存在することを保証している。なお，命題 4.13 により，f^α の極小点は f^α の最小点とは限らない。

定理 4.15 ([62, 定理 6.37][79])**.** α を正整数とする.

(1) $f : \mathbb{Z}^n \to \mathbb{R}\cup\{+\infty\}$ を $\mathrm{M}^\natural$ 凸関数とし, $\boldsymbol{x}^\alpha \in \mathrm{dom}_{\mathbb{Z}}\, f$ とする．任意の $i, j \in \{0, 1, \ldots, n\}$ に対して

$$f(\boldsymbol{x}^\alpha) \leq f(\boldsymbol{x}^\alpha + \alpha(\boldsymbol{\chi}_j - \boldsymbol{\chi}_i))$$

ならば，$\mathrm{argmin}_{\mathbb{Z}}\, f \neq \emptyset$ であって,

$$\boldsymbol{x}^\alpha - n(\alpha-1)\mathbf{1} \leq \boldsymbol{x}^* \leq \boldsymbol{x}^\alpha + n(\alpha-1)\mathbf{1}$$

を満たす $\boldsymbol{x}^* \in \mathrm{argmin}_{\mathbb{Z}}\, f$ が存在する.

(2) $f : \mathbb{Z}^n \to \mathbb{R}\cup\{+\infty\}$ を M 凸関数とし，$\boldsymbol{x}^\alpha \in \mathrm{dom}_{\mathbb{Z}}\, f$ とする．任意の $i, j \in \{1, \ldots, n\}$ に対して

$$f(\boldsymbol{x}^\alpha) \leq f(\boldsymbol{x}^\alpha + \alpha(\boldsymbol{\chi}_j - \boldsymbol{\chi}_i))$$

ならば，$\mathrm{argmin}_{\mathbb{Z}}\, f \neq \emptyset$ であって,

$$\boldsymbol{x}^\alpha - (n-1)(\alpha-1)\mathbf{1} \leq \boldsymbol{x}^* \leq \boldsymbol{x}^\alpha + (n-1)(\alpha-1)\mathbf{1}$$

を満たす $\boldsymbol{x}^* \in \mathrm{argmin}_{\mathbb{Z}}\, f$ が存在する. ■

4.3.3　スケーリング法

$\mathrm{L}^\natural$ 凸関数の場合と同じように、上の近接定理を利用して M 凸関数 f を最小化するスケーリング法のアルゴリズムを書き下すと、次のようになる。

M 凸関数の最小化アルゴリズム（スケーリング法）：SCALING$(f, \boldsymbol{x})$
Input: 離散 M 凸関数 f と初期解 $\boldsymbol{x} \in \mathrm{dom}_{\mathbb{Z}}\, f$
Output: f の一つの最小解
Step 0: $k := \lceil \log_2 K_\infty \rceil$, $\boldsymbol{y}_{k+1} := \mathbf{0}$ とおく.
Step 1: $\alpha_k := 2^k$ とおく。
$f_k(\boldsymbol{y}) := f(\boldsymbol{x} + \alpha_k \boldsymbol{y})$, $\mathrm{dom}_{\mathbb{Z}}\, f_k := \{\boldsymbol{y} \in \mathbb{Z}^n \mid 2\boldsymbol{y}_{k+1} - n\mathbf{1} \leq \boldsymbol{y} \leq 2\boldsymbol{y}_{k+1} + n\mathbf{1}\}$
として最急降下法 SD$(f_k, 2\boldsymbol{y}_{k+1})$ を呼び出し、得られた最小解を $\boldsymbol{y}_k$ とする。
Step 2: $k = 0$ ならば $\boldsymbol{x} + \boldsymbol{y}_0$ を返す（$\boldsymbol{x} + \boldsymbol{y}_0$ は f の最小解の一つ）.
Step 3: $k := k - 1$ として Step 1 に戻る.

ただし、このアルゴリズムが多項式時間で終了する保証があるのは、スケーリングした f_k も M 凸関数である場合に限られる。つまり、f が 2 次 M 凸関数や層凸関数などであれば高速に動作するが、一般の M 凸関数に対しては、正しい結果を返すものの、多項式時間で終了する保証がない [52]。

スケーリングした f_k も M 凸関数である場合の計算量を解析すると、次のようになる。手順 Step 1 で呼び出す最急降下法 SD(f_k, $2\boldsymbol{y}_{k+1}$) の計算量は $O(n^3)$ である。手順 Step 1 から Step 3 の反復は、ちょうど $(\lceil \log_2 K_\infty \rceil + 1)$ 回であるので、全体の計算量は $O(n^3 \lceil \log_2 K_\infty \rceil)$ となり、スケーリング法は多項式時間で終了することがわかる [88].

一般の M 凸関数でも多項式時間で終了するスケーリング法としては、Shioura [88] と Tamura [91] のアルゴリズムが知られている。このうち Shioura のアルゴリズムは動作が単純で、前述のアルゴリズムの動作を一部修正するだけでよい。つまり、探索のステップサイズを 1 で行い、実際に移動するステップサイズは α_k とする。もう一方の Tamura のアルゴリズムのアイデアは、軸ごとのスケーリングサイズを独立して変化させることにある。

4.4 連続緩和

4.4.1 M 凸関数に対する離散化と連続化

$M^\natural$ 凸/M 凸関数が凸拡張可能であることを、定理 2.24 と定理 2.26 で次のように述べた。したがって任意の $M^\natural$ 凸/M 凸関数に対して、その連続化が存在する。

定理 4.16.
(1) $M^\natural$ 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は凸拡張可能である.
(2) M 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ は凸拡張可能である. ■

ここでは $M^\natural$ 凸/M 凸関数と $M^\natural$ 凸/M 凸集合について，離散化と連続化が $M^\natural$ 凸/M 凸性を保つかどうかを考える．結論としては、連続化については $M^\natural$ 凸/M 凸性は保たれるものの、離散化については、一般には $M^\natural$ 凸/M 凸性は保たれない。このため、連続緩和法が $L^\natural$ 凸/L 凸関数よりも設計しにくい。

まず離散化について詳しく述べる。連続変数の $M^\natural$ 凸/M 凸関数が、交換公理 M-EXC[$\mathbb{R}$], $M^\natural$-EXC[$\mathbb{R}$] を $\alpha = 1$ の場合も含めて満たせば、離散化しても $M^\natural$ 凸/M 凸関数となるが、一般にはそうはならない。

命題 4.17.
(1) 連続変数の $M^\natural$ 凸関数の離散化 (3.9) は，離散変数の $M^\natural$ 凸関数とは限らない.
(2) 連続変数の M 凸関数の離散化 (3.9) は，離散変数の M 凸関数とは限らない. ■

離散化すると $M^\natural$ 凸/M 凸性を失う関数の例としては，例 2.12 に示した 2 次関数がある．逆に，条件 (2.97), (2.98) を満たす行列で定義される 2 次関数は離散化しても $M^\natural$ 凸関数であり (定理 2.33)，条件 (2.99) を満たす行列で定義される 2 次関数は離散化しても M 凸関数である (定理 2.35).

次に，連続化ついて述べる．次の定理が成り立つ。

定理 4.18.
(1) 離散変数の $M^\natural$ 凸関数の連続化 (3.10) は，連続変数の $M^\natural$ 凸関数である.

(2) 離散変数の M 凸関数の連続化 (3.10) は，連続変数の M 凸関数である． ■

これらの命題と定理の特殊ケースとして、関数が標示関数である場合を考えると、集合の離散化に関する命題と、集合の連続化（凸包）に関する定理が導かれる．

命題 4.19.
(1) $\mathrm{M}^\natural$ 凸多面体 S に含まれる整数ベクトルの全体 $S \cap \mathbb{Z}^n$ は，離散の $\mathrm{M}^\natural$ 凸集合とは限らない．しかし，S が有界でその頂点がすべて整数ベクトルの場合には，$S \cap \mathbb{Z}^n$ は離散の $\mathrm{M}^\natural$ 凸集合である [*2]．
(2) M 凸多面体 S に含まれる整数ベクトルの全体 $S \cap \mathbb{Z}^n$ は，離散の M 凸集合とは限らない．しかし，S が有界でその頂点がすべて整数ベクトルの場合には，$S \cap \mathbb{Z}^n$ は離散の M 凸集合である． ■

定理 4.20.
(1) 離散の $\mathrm{M}^\natural$ 凸集合の凸包は，$\mathrm{M}^\natural$ 凸多面体である．
(2) 離散の M 凸集合の凸包は，M 凸多面体である． ■

4.4.2　連続緩和の近接定理

連続緩和に基づくアルゴリズムの効率性は，元の問題の最適解と連続緩和の最適解との距離に依存するため，この 2 つの最適解の近さを理論的に保証する「近接定理」が重要になる。本章における主な成果は，問題 (MC) に対する近接定理であり，元の (MC) とその連続緩和問題 ($\overline{\mathrm{MC}}$) の最適解の L_∞ 距離が $n-1$ 以下であると主張するものである [53].

定理 4.21 ((MC) に対する L_∞ 距離の近接定理)**.** $n \geq 2$ とする。
(i) (MC) の任意の最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ に対して，($\overline{\mathrm{MC}}$) のある最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ が存在して $\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_\infty < n-1$ を満たす.
(ii) ($\overline{\mathrm{MC}}$) の任意の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ に対して, (MC) のある最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して $\|\boldsymbol{y}_* - \boldsymbol{x}_*\|_\infty < n-1$ を満たす. ■

この定理の証明は第 4.7.1 節で与える．この定理は，($\overline{\mathrm{MC}}$) の最適解が存在するときに限り (MC) の最適解が存在することも意味している．定理 4.21 に現れる $n-1$ という限界値は、後に示す第 4.6.3 節の例 4.3 により、これより改善できないことがわかる。

次に L_1 距離に関する近接定理に関する結果を述べる．定理 4.21 から，(MC) に関する次の系が直ちに得られる．

系 4.22 ((MC) に対する L_1 距離の近接定理)**.** $n \geq 2$ とする。
(i) (MC) の任意の最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ に対して，($\overline{\mathrm{MC}}$) のある最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ が存在して

[*2] S が有界でなくても，任意の整数区間 $[\boldsymbol{a}, \boldsymbol{b}]_\mathbb{Z}$ との共通部分 $S \cap [\boldsymbol{a}, \boldsymbol{b}]_\mathbb{Z}$ の頂点がすべて整数ベクトルであればよい．(2) においても同様.

$\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1 < n(n-1)$ を満たす.
(ii) ($\overline{\mathsf{MC}}$) の任意の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ に対して，(MC) のある最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して $\|\boldsymbol{y}_* - \boldsymbol{x}_*\|_1 < n(n-1)$ を満たす. ■

4.4.3 連続緩和法

連続緩和手法に基づく M 凸関数最小化問題 (MC) の新しいアルゴリズムを提案する．第 4.2.3 節で提案した新しい貪欲法を用いる。

$f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が M 凸関数であるとする．$F(\boldsymbol{y}) = f(\boldsymbol{y})$ $(\forall \boldsymbol{y} \in \mathbb{Z}^n)$ を満たす閉真 M 凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が与えられていて，$\mathrm{dom}_{\mathbb{R}} F$ が $\mathrm{dom}_{\mathbb{Z}} f$ の閉凸包であると仮定する．f を最小化するアルゴリズムは次のようになる．

M 凸関数の最小化アルゴリズム（連続緩和法）：RELAX(f, F)
Input: 閉真 M 凸関数 f と連続緩和の M 凸関数 F
Output: f の一つの最小解
Step 1: 閉真 M 凸関数 F の最小解の 1 つを求め，$\boldsymbol{x}_* \in \mathrm{dom}_{\mathbb{R}} F$ とする．
Step 2: $\|\boldsymbol{y}^\circ - \boldsymbol{x}_*\|_1 < n$ を満たす整数ベクトルを求め，$\boldsymbol{y}^\circ \in \mathrm{dom}_{\mathbb{Z}} f$ とする．
Step 3: $\boldsymbol{y}^\circ$ を初期解として，アルゴリズム NewGreedy を M 凸関数 f に対して適用する．

$\mathrm{dom}_{\mathbb{R}} F$ は，M 凸集合 $\mathrm{dom}_{\mathbb{Z}} f$ の閉凸包であるので，整数格子点上の基多面体となる（定理 4.20）．ゆえに，任意の $\boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F$ に対して，ある整数ベクトル $\boldsymbol{y} \in \mathrm{dom}_{\mathbb{R}} F \cap \mathbb{Z}^n = \mathrm{dom}_{\mathbb{Z}} f$ が存在して，$\|\boldsymbol{y} - \boldsymbol{x}\|_\infty < 1$ を満たし，このベクトル $\boldsymbol{y}$ は $\|\boldsymbol{y} - \boldsymbol{x}_*\|_1 < n$ も満たす．

第 4.4.2 節で示した近接定理を用いると，提案手法の計算時間は次の定理のようになる．ここで T_{relax} は ($\overline{\mathsf{MC}}$) の連続緩和解を求めるのに必要な計算時間を表し，T_{round} は ($\overline{\mathsf{MC}}$) の得られた実行可能解 $\boldsymbol{x} \in \mathbb{R}^n$ を (MC) の実行可能解 $\boldsymbol{y} \in \mathbb{Z}^n$ に条件 $\|\boldsymbol{y} - \boldsymbol{x}\|_1 < n$ を満たしながら丸めるのに必要な計算時間を表すとする．K_∞ を式 (3.5) で定義したものとするとき，$T_{\mathrm{round}} = \mathrm{O}(n^2 \log K_\infty)$ となる [87].

定理 4.23. 連続緩和による (MC) の最適解を求めるアルゴリズム RELAX の計算時間は $\mathrm{O}(T_{\mathrm{relax}} + T_{\mathrm{round}} + n^3 T_{\mathrm{func}})$ である. ■

証明. 系 4.22 より，ある $\boldsymbol{y}_* \in \mathrm{argmin}_{\mathbb{Z}} f$ が存在して，$\|\boldsymbol{y}_* - \boldsymbol{x}_*\|_1 < n(n-1)$ を満たす．ゆえに，Step 2 で求めるベクトル $\boldsymbol{y}^\circ$ は

$$\|\boldsymbol{y}_* - \boldsymbol{y}^\circ\|_1 \le \|\boldsymbol{y}_* - \boldsymbol{x}_*\|_1 + \|\boldsymbol{x}_* - \boldsymbol{y}^\circ\|_1 < n(n-1) + n = n^2$$

を満たす．このことと，定理 4.12 をあわせると，Step 3 のアルゴリズム NewGreedy は $\mathrm{O}(n^3 T_{\mathrm{func}})$ の計算時間で終了する．

Step 1 にかかる時間，つまり連続緩和問題 ($\overline{\mathsf{MC}}$) を解くのにかかる時間は T_{relax} であり，Step 2 にかかる時間は T_{round} であるので、アルゴリズム RELAX で (MC) の最適解を求めるには、$\mathrm{O}(T_{\mathrm{relax}} + T_{\mathrm{round}} + n^3 T_{\mathrm{func}})$ だけの時間で済む。 ■

表 4.1. 実装した $\mathrm{M}^\natural$ 凸関数最小化アルゴリズム

表記	アルゴリズム
SD	最急降下法 (第 4.2.1 節)
SD2	修正最急降下法 [52]
SCALING	スケーリング法 (第 4.3.3 節)
RELAX	提案する連続緩和法 (第 4.4.3 節)

注意 4.1. 実際には，($\overline{\mathsf{MC}}$) の連続緩和の厳密な最適解を求める必要はなく，($\overline{\mathsf{MC}}$) の最適解の「近似値」$\boldsymbol{x}_a \in \mathbb{R}^n$ が求まれば十分である．「近似値」とは，($\overline{\mathsf{MC}}$) のある最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ に対して $\|\boldsymbol{x}_a - \boldsymbol{x}_*\|_\infty \leq n$ を満たすことを意味する．このような解を求めるのに必要な計算時間を $T_{\text{relax-apx}}$ で表すと，通常の場合 $T_{\text{relax-apx}}$ は T_{relax} よりも小さい．また、定理 4.21 より (MC) の最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ の中に，$\|\boldsymbol{y}_* - \boldsymbol{x}_a\|_\infty \leq \|\boldsymbol{y}_* - \boldsymbol{x}_*\|_\infty + \|\boldsymbol{x}_* - \boldsymbol{x}_a\|_\infty \leq 2n$ を満たすものがあることがわかる．この限界値を用いると, 定理 4.23 の計算量を $\mathrm{O}(T_{\text{relax-apx}} + T_{\text{round}} + n^3 T_{\text{func}})$ に改善できる。 ■

($\overline{\mathsf{MC}}$) の計算は、第 4.2.2 節で述べたように、実用上は準ニュートン法のようなアルゴリズムで高速に行えるが、理論的な計算量を解析すると次のようになる。($\overline{\mathsf{MC}}$) の最適解の近似値 $\boldsymbol{x}_a$ は，(MC) と同様の手法で効率的に求められる．例えば，[88] の (MC) に対するスケーリング手法を ($\overline{\mathsf{MC}}$) に適用することで,

$$\mathrm{O}((n^3 + n^2 \log(K_\infty/n))(\log(K_\infty/n)/\log n) T_{\text{func}})$$

の計算時間のアルゴリズムが得られる。ただし，目的関数 f の方向微分の計算に $\mathrm{O}(T_{\text{func}})$ かかるものとする．この計算時間が $T_{\text{relax-apx}}$ に対応するので、我々の連続緩和法の計算時間は

$$\mathrm{O}((n^3 + n^2 \log(K_\infty/n))(\log(K_\infty/n)/\log n) T_{\text{func}})$$

となる．これは，先行研究における (MC) に対する結果 [52, 87, 88, 91] の最良のものと同じである．従って，最悪計算量の理論的上界という意味においては、連続緩和手法は一般の (MC) に対する計算時間を改善するわけではない．しかし、第 4.5 節で述べるように、実際上の計算時間を大きく改善する。

4.5　実験的評価

提案する連続緩和法の性能を、既存のアルゴリズムと比較した。この数値実験から、既存のアルゴリズムよりも提案手法がはるかに高速であることがわかった.

表 4.1 に示す 4 通りの離散 $\mathrm{M}^\natural$ 凸関数最小化アルゴリズムを C 言語で実装し，性能を比較した．以下のライブラリを利用した.

- Nocedal による 'L-BFGS' [*3] と，工藤による C++ 言語用インタフェース [*4] は準ニュートン法の実装で，連続関数最適化を行う [44]．このルーチンには目的関数の勾配が必要であるので，差分で近似した．1 つの差分を得るのに $n+1$ 回の関数評価が必要である．このライブラリは RELAX（提案する連続緩和法）でのみ用いた．
- 斎藤・松本による 'SIMD-oriented Fast Mersenne Twister' [*5] は擬似乱数を生成する．問題例を生成するのに利用した．

問題例に用いた離散 $\mathrm{M}^\natural$ 凸関数は

$$f(\boldsymbol{x}) = \sum_{X \in \mathcal{T}} \{a_X x(X)^2 + b_X x(X) + c_X\} \quad (\boldsymbol{x} \in \mathbb{Z}^n)$$

と表されるものである．ただし $\mathcal{T}$ は層族である。連続緩和法では，連続緩和の $\mathrm{M}^\natural$ 凸関数として

$$F(\boldsymbol{x}) = \sum_{X \in \mathcal{T}} \{a_X x(X)^2 + b_X x(X) + c_X\} \quad (\boldsymbol{x} \in \mathbb{R}^n)$$

を用いた．1 つの n につき 10 問の問題例を生成した．整数の係数は、$X \in \mathcal{T}$ について

$$\begin{aligned} 0 < a_X &\leq 1000, \\ -1000 \leq b_X &\leq 1000, \\ -1000 \leq c_X &\leq 1000 \end{aligned}$$

の範囲から一様ランダムに選んだ．初期解は，各問題例ごとに

$$-10n \leq p_0(i) \leq 10n$$

を満たす整数格子点からランダムに選んだ．

計算機環境は HP dx5150 SF/CT, AMD Athlon 64 3200+ processor (2.0GHz, 512KB L2 cache), 4GB memory, Vine Linux 4.1 (kernel 2.6.16), gcc 3.3.6 である．

ここで実装した 4 種類のアルゴリズム（SD, SD2, SCALING, RELAX）では，どれも最小化したい $\mathrm{M}^\natural$ 凸関数の値のみを用いており、関数の内部構造は利用していない．そして問題ごとに，最小化に必要となった関数評価回数と計算時間を測定した．数値計算の結果は図 4.1 にまとめた．図 4.1 の上のグラフは両対数グラフであり，縦軸は関数評価回数 C，横軸は関数の次元 n である．図 4.1 の下のグラフも両対数グラフであり，縦軸は計算時間 T，横軸は関数の次元 n である．これより，関数評価回数 C について、すべてのアルゴリズムで $\log C$ と $\log n$ が比例していることが読み取れる．つまり，ある l を用いて $C = \mathrm{O}(n^l)$ と表せることになる．計算時間 T についても同様に $T = \mathrm{O}(n^l)$ と表せる．l を最小二乗法であてはめて求めた値を、表 4.2 にまとめた．RELAX がオーダーとしてもっとも小さいことがわかる．

以上のように，計算機実験によって，ランダムなテスト問題に対して，連続緩和法は先行手法よりも高速に $\mathrm{M}^\natural$ 凸関数を最小化できることが確かめられた．

[*3] http://www.ece.northwestern.edu/~nocedal/lbfgs.html

[*4] http://chasen.org/~taku/software/misc/lbfgs/

[*5] http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/SFMT/

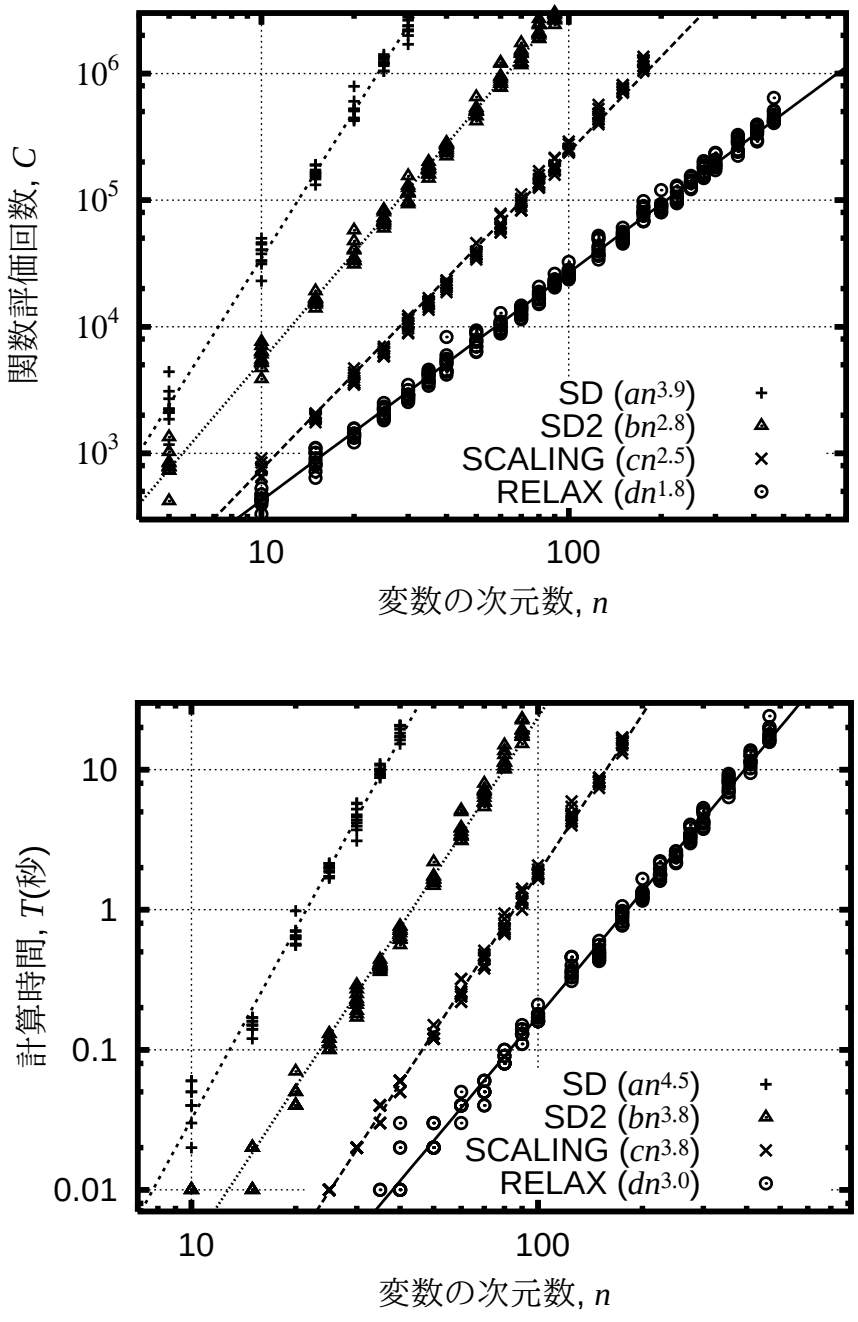


図 4.1. $\mathrm{M}^\natural$ 凸関数最小化に必要な目的関数の評価回数と計算時間

表 4.2. $\mathrm{M}^\natural$ 凸関数最小化にかかった計算量（測定値）

アルゴリズム	SD	SD2	SCALING	RELAX
関数評価回数 C	$n^{3.9}$	$n^{2.8}$	$n^{2.5}$	$n^{1.8}$
計算時間 T	$n^{4.5}$	$n^{3.8}$	$n^{3.8}$	$n^{3.0}$

4.6 資源配分問題と連続緩和手法

ここからは、M 凸関数最小化問題 (MC) の特殊ケースである資源配分問題を対象とする。特に目的関数が 2 次の場合に、理論的な計算量を解析し、先行研究と比較する。

4.6.1 資源配分問題

資源配分問題とは、資源量と資源の配分先が与えられ、配分先ごとに資源量に従った目的関数が定められたときに、目的関数の和が最大（または最小）になるように資源を配分する最適化問題である。

資源の量は、離散値の場合も連続値の場合もある。例えば、プロジェクトに人員を割り当てる場合は、資源の量＝人員の人数は離散値になる。ここでは、主として離散値の場合を考える。ただし資源の種類は 1 種類に限る。

各配分先の目的関数（利益などを表す評価関数）にもバリエーションが考えられるが、ここでは 1 変数の凸関数として、各目的関数の和の最小化を考えることにする。

このように状況設定した資源配分問題は、M 凸関数最小化問題 (MC) の特殊ケースととらえることができる。配分先の数が、M 凸関数のベクトルの次元数 n に対応する。さらに資源配分問題は、配分の仕方にさまざまな制約を加えた形の問題も定式化されている [33]。問題のクラスを強く制限すれば、定式化の柔軟性は低下するがアルゴリズムの性能は向上し、逆に複雑な状況を表現できるような柔軟性をもつ定式化に対しては、アルゴリズムの設計は難しくなる。

ここでは問題 (MC) の特殊ケースとなる資源配分問題を説明する。まず、層族制約つき資源配分問題 (Laminar) と劣モジュラ制約つき資源配分問題 (SC) を説明する。これらは複雑な制約を表現できる柔軟性をもちながらも、比較的高性能なアルゴリズムが設計できるようにうまく切り出された問題クラスであり、先行研究でも取り上げられる機会の多いものである。そして、(SC) の特殊ケースとなる問題クラスをいくつか説明する。

層族制約つき資源配分問題

層族制約つき資源配分問題 (Laminar) は、

$$
\text{(Laminar)} \quad \left| \begin{array}{ll} \text{Minimize} & \displaystyle\sum_{Y \in \mathcal{F}} F_Y(x(Y)) \\ \text{subject to} & x(N) = K, \\ & \ell_Y \le x(Y) \le u_Y \quad (Y \in \mathcal{F}), \\ & \boldsymbol{x} \ge \mathbf{0}, \quad \boldsymbol{x} \in \mathbb{Z}^n \end{array} \right.
$$

と定式化される ([52],[62, 第 6.3 節]). ただし, $\mathcal{F} \subseteq 2^N$ は層族であり, $F_Y : \mathbb{R} \to \mathbb{R}$ $(Y \in \mathcal{F})$ は 1 変数凸関数であり, $K \in \mathbb{Z}_+$, $\ell_Y, u_Y \in \mathbb{Z}_+$ $(Y \in \mathcal{F})$ であるとする. 一般性を失うこと

なく，

$$\emptyset \notin \mathcal{F}, \quad N \in \mathcal{F}, \quad \{i\} \in \mathcal{F} \ (\forall i \in N) \tag{4.6}$$

という仮定を置くことができる。$|\mathcal{F}| = \mathrm{O}(n)$ であることに注意する．

ここでは問題 (Laminar) が実行可能解をもつことを仮定する．(Laminar) の連続緩和問題 ($\overline{\text{Laminar}}$) は (Laminar) から整数制約「$\boldsymbol{x} \in \mathbb{Z}^n$」を除くことで得られる．

層族制約つき資源配分問題 (Laminar) は M 凸関数最小化問題 (MC) の特殊ケースである．なぜなら，関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ を

$$f(\boldsymbol{x}) = \begin{cases} \displaystyle\sum_{Y \in \mathcal{F}} F_Y(x(Y)) & (\boldsymbol{x} \in \mathbb{Z}^n \text{が (Laminar) の実行可能解である場合}), \\ +\infty & (\text{その他}) \end{cases}$$

と定義すると，(M-EXC[$\mathbb{Z}$]) を満たすからである ([52, 例 2.3],[62, 第 6.3 節])．(Laminar) は (MC) の特殊ケースとして重要である．なぜなら，目的関数が非分離形でありながらも，提案する連続緩和手法が自然に適用できる例だからである．

問題 (Laminar) は，次に示すように，ツリーネットワーク上の凸費用流問題としても定式化されることが知られている．

$$V = \{v_Y \mid Y \in \mathcal{F}\}, \qquad A = \{(v_X, v_Y) \mid Y \in \mathcal{F} \setminus \{N\},\ X = p(Y)\}$$

と定義される（有向）木 $T = (V, A)$ を考える．なお，頂点 v_N は T の根ノードであり，単集合に対応する頂点は T の葉ノードである．ここで，$p(Y) \in \mathcal{F}$ は $Y \in \mathcal{F} \setminus \{N\}$ の**親 (parent)** と呼ばれ，Y を真に含む（つまり $p(Y) \supset Y$ かつ $p(Y) \neq Y$ である）$\mathcal{F}$ の要素の中の（一意的に定まる）極小な集合である．また、$|X| \geq 2$ となる $X \in \mathcal{F}$ について，$X = p(Y)$ であれば $Y \in \mathcal{F}$ を X の**子 (child)** と呼ぶ．条件 (4.6) は，$|X| \geq 2$ となる任意の $X \in \mathcal{F}$ が，少なくとも 1 つの子をもつことも意味している．

凸費用流の定式化では，根ノード v_N から葉ノードへと流れるものとする．つまり，v_N がソースで葉ノードがシンクである．弧 $(i, j) \in A$ の流量を $\varphi(i, j)$ で表し，流量に関して、整数制約

$$\varphi(i, j) \in \mathbb{Z} \qquad (\forall (i, j) \in A)$$

と流量のバランス制約

$$\begin{aligned} &\sum\{\varphi(v_N, v_Y) \mid v_Y \in V,\ (v_N, v_Y) \in A\} = K, \\ &\sum\{\varphi(v_X, v_Y) \mid v_Y \in V, (v_X, v_Y) \in A\} = \varphi(v_{p(X)}, v_X) \qquad (\forall v_X \in V,\ v_X \neq v_N) \end{aligned}$$

を考える．枝 $(v_{p(Y)}, v_Y) \in A$ のそれぞれに対して，凸費用関数 F_Y，枝の容量の上限 u_Y と下限 ℓ_Y が与えられている．問題 (Laminar) が，上で定義した凸費用流問題と等価であること，そして ($\overline{\text{Laminar}}$) が，流量変数から整数制約を取り除いた，凸費用流問題の連続緩和と等価であることは容易にわかる．

劣モジュラ制約つき資源配分問題

劣モジュラ制約つき資源配分問題 (SC) は、

$$\text{(SC)} \left| \begin{array}{ll} \text{Minimize} & \displaystyle\sum_{i=1}^{n} F_i(x(i)) \\ \text{subject to} & x(N) = \rho(N), \\ & x(Y) \leq \rho(Y) \quad (Y \in 2^N), \\ & \boldsymbol{x} \geq \mathbf{0}, \quad \boldsymbol{x} \in \mathbb{Z}^n \end{array} \right.$$

と定式化される [17, 29, 33, 38]．ただし $F_i : \mathbb{R} \to \mathbb{R}$ $(i \in N)$ は 1 変数凸関数であり，$\rho : 2^N \to \mathbb{Z}_+ \cup \{+\infty\}$ は非負値をとる劣モジュラ関数で，$\rho(\emptyset) = 0$ かつ $\rho(N) < +\infty$ であるとする．

劣モジュラ制約つき資源配分問題 (SC) は広く議論され多くの研究成果がある．サーベイ論文として [17, 33, 38] があり，効率的なアルゴリズムは [22, 25, 29, 30] に述べられている．

劣モジュラ制約つき資源配分問題 (SC) は M 凸関数最小化問題 (MC) の特殊ケースである．なぜなら，関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ が

$$f(\boldsymbol{x}) = \begin{cases} \displaystyle\sum_{i=1}^{n} F_i(x(i)) & (\boldsymbol{x} \in \mathbb{Z}^n \text{が (SC) の実行可能解である場合}), \\ +\infty & (\text{その他}) \end{cases}$$

の形のとき，(M-EXC[$\mathbb{Z}$]) を満たすからである ([58, 例 2.2],[62, 第 6.3 節])．

その他の資源配分問題

劣モジュラ制約つき資源配分問題 (SC) の特殊ケースである資源配分問題のうち、文献 [29, 30] で議論されたものを説明する。1 つめは**単純な資源配分問題** (Simple) であり，

$$\text{(Simple)} \left| \begin{array}{ll} \text{Minimize} & \displaystyle\sum_{i=1}^{n} F_i(x(i)) \\ \text{subject to} & x(N) = K, \\ & \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{u}, \quad \boldsymbol{x} \in \mathbb{Z}^n \end{array} \right.$$

と定式化される．ただし，$F_i : \mathbb{R} \to \mathbb{R}$ $(i \in N)$ は 1 変数凸関数であり，$K \in \mathbb{Z}_+$ かつ $\boldsymbol{u} \in \mathbb{Z}_+^n$ である．

文献 [29, 30] で議論された残りの (SC) の特殊ケースは、問題 (Simple) に制約を付け加えたものである．

- **一般上界制約つき資源配分問題** (GUB) は，問題 (Simple) に

  $$x(S_t) \leq u_{S_t} \ (t = 1, 2, \ldots, m)$$

  という制約を加えたものである．ただし，$\{S_1, S_2, \ldots, S_m\}$ は N の分割であり，$u_{S_t} \in \mathbb{Z}_+$ $(t = 1, 2, \ldots, m)$ とする．

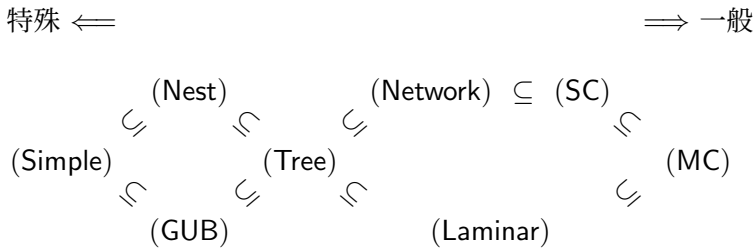


図 4.2. 離散凸最適化問題の包含関係

- **入れ子制約つき資源配分問題** (Nest) は，問題 (Simple) に

$$x(S_t) \leq u_{S_t} \ (t = 1, 2, \ldots, m)$$

という制約を加えたものである．ただし，$\{S_1, S_2, \ldots, S_m\}$ は N の部分集合の入れ子になった族, つまり $\emptyset \neq S_1 \subset S_2 \subset \cdots \subset S_m \subseteq N$ であり、$u_{S_t} \in \mathbb{Z}_+ \ (t = 1, 2, \ldots, m)$ とする．

- **ツリー制約つき資源配分問題** (Tree) は，問題 (Simple) に

$$x(Y) \leq u_Y \ (Y \in \mathcal{F})$$

という制約を加えたものである．ただし，$\mathcal{F}(\subseteq 2^N)$ は層族で，$u_Y \in \mathbb{Z}_+ \ (Y \in \mathcal{F})$ とする．したがって，(Tree) は (Laminar) の特殊ケースでもある．

- **ネットワーク制約つき資源配分問題** (Network) は，1 ソース多シンクのネットワークを使って定義される．頂点集合 V，弧集合 A とする有向グラフ $G = (V, A)$ が与えられたとき，$s \in V$ を唯一のソースノードとし，$N = \{1, 2, \ldots, n\} \ (\subseteq V)$ を複数あるシンクノードの集合とする．ソースの供給量を $K \in \mathbb{Z}_+$ であらわし，各弧 $(i, j) \in A$ の容量を $c(i, j) \in \mathbb{Z}_+$ で表す．変数 $\{x(i) \mid i \in N\}$ に加えて流量変数 $\{\varphi(i, j) \mid (i, j) \in A\}$ を用いることで，問題 (Network) の制約は以下のように記述される．この制約に $x(N) = K$ の条件が含まれていることは容易にわかる。

$$\begin{aligned}
&\textstyle\sum\{\varphi(s, j) \mid j \in V, \ (s, j) \in A\} - \sum\{\varphi(j, s) \mid j \in V, \ (j, s) \in A\} = K, \\
&\textstyle\sum\{\varphi(i, j) \mid j \in V, \ (i, j) \in A\} - \sum\{\varphi(j, i) \mid j \in V, \ (j, i) \in A\} = -x(i) \quad (\forall i \in N), \\
&\textstyle\sum\{\varphi(i, j) \mid j \in V, \ (i, j) \in A\} - \sum\{\varphi(j, i) \mid j \in V, \ (j, i) \in A\} = 0 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\forall i \in V \setminus (\{s\} \cup N)), \\
&0 \leq \varphi(i, j) \leq c(i, j) \ (\forall (i, j) \in A), \ \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{u}, \ \boldsymbol{x} \in \mathbb{Z}^n.
\end{aligned}$$

これまでに説明した離散凸最適化問題の包含関係をまとめると，図 4.2 のようになる．また、(SC), (Simple), (GUB), (Nest), (Tree), (Network) のそれぞれについて，「$\boldsymbol{x} \in \mathbb{Z}^n$」の条件を取り払った連続緩和問題を ($\overline{\text{SC}}$), ($\overline{\text{Simple}}$), ($\overline{\text{GUB}}$), ($\overline{\text{Nest}}$), ($\overline{\text{Tree}}$), ($\overline{\text{Network}}$) と表す．

4.6.2 資源配分問題に対する連続緩和手法の先行研究

この節では，資源配分問題や関連する問題に対する連続緩和手法の既存の成果をまとめる．

単純な資源配分問題 (Simple)

Weinsten–Yu [98] は単純な資源配分問題 (Simple) に対する連続緩和に基づくアルゴリズムを提案した．このアルゴリズムの計算量は，n 変数の連続緩和問題 ($\overline{\text{Simple}}$) を解くのにかかる計算量を $T_{\overline{\text{Simple}}}(n)$ で表すとき，$\mathrm{O}(T_{\overline{\text{Simple}}}(n) + n\log n)$ となる．(Simple) に関する近接定理は文献 [98] に暗に示されていて，その表現は，

> ($\overline{\text{Simple}}$) の任意の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ に対して，(Simple) のある最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して，$\|\boldsymbol{y}_* - \boldsymbol{x}_*\|_1 = \mathrm{O}(n)$ を満たす。

となっている．後に Ibaraki–Katoh [33, 第 4.6 節] によって Weinsten–Yu のアルゴリズムが改良され，計算量が $\mathrm{O}(T_{\overline{\text{Simple}}}(n) + n)$ となった．目的関数が 2 次であれば，連続緩和問題 ($\overline{\text{Simple}}$) は Brucker のアルゴリズム [7] を用いると線形時間で解けるので，2 次の (Simple) は Ibaraki–Katoh のアルゴリズムによって $\mathrm{O}(n)$ の計算量で解けることになる．

一般上界制約つき資源配分問題 (GUB)

Hochbaum [29] は一般上界制約つき資源配分問題 (GUB) が単純な資源配分問題 (Simple) に帰着できることを示した．これにより、(GUB) が 2 次の目的関数をもつときには，$\mathrm{O}(n)$ の計算量で解けることがわかる．

問題 (GUB) を問題 (Simple) に帰着するには、次のようにする。第 4.6.1 節で述べたように、(Simple) は，

$$\text{(Simple)} \quad \left| \begin{array}{ll} \text{Minimize} & \displaystyle\sum_{i=1}^{n} F_i(x(i)) \\ \text{subject to} & x(N) = K, \\ & \boldsymbol{0} \le \boldsymbol{x} \le \boldsymbol{u}, \quad \boldsymbol{x} \in \mathbb{Z}^n \end{array} \right.$$

と定式化され、(GUB) は、

$$\text{(GUB)} \quad \left| \begin{array}{ll} \text{Minimize} & \displaystyle\sum_{i=1}^{n} F_i(x(i)) \\ \text{subject to} & x(N) = K, \\ & x(S_t) \le u_{S_t} \ (t = 1, 2, \ldots, m), \\ & \boldsymbol{x} \ge \boldsymbol{0}, \quad \boldsymbol{x} \in \mathbb{Z}^n \end{array} \right.$$

と定式化される。ただし，$\{S_1, S_2, \ldots, S_m\}$ は N の分割であり，$u_{S_t} \in \mathbb{Z}_+$ $(t = 1, 2, \ldots, m)$ である．この問題 (GUB) を直接解く代わりに、(Simple) を何度か解いて同じ結果を得ることを考える。

問題 (Simple) では $\boldsymbol{x}$ の $i\,(\in N)$ 成分に対して上界 $u(i)$ が与えられているが、(GUB) では N の分割 S_t ごとの和 $x(S_t)$ に対して上界 u_{S_t} が与えられている点が異なる。これを、成分ごとの上限を与えるように変形すれば (Simple) に帰着することができる。そのためには、N の分割 S_t ごとに (Simple) を 1 回解く。つまり、

$$\begin{array}{l|ll} (\mathsf{Simple}_{(t)}) & \text{Minimize} & \displaystyle\sum_{i\in S_t} F_i(x(i)) \\ & \text{subject to} & x(S_t) = u_{S_t}, \\ & & \boldsymbol{x} \geq \boldsymbol{0}, \quad \boldsymbol{x} \in \mathbb{Z}^n \end{array}$$

と定式化される部分問題 ($\mathsf{Simple}_{(t)}$) を解く。部分問題 ($\mathsf{Simple}_{(t)}$) は通常の (Simple) に比べると、$\boldsymbol{x}$ に上界が設定されていない点で異なるが、同様に解くことができる。各 S_t について ($\mathsf{Simple}_{(t)}$) を解き、得られた解 $\boldsymbol{x}^{*t}$ から S_t 成分を取り出して $u^*(i) = \boldsymbol{x}^{*t}(i) \quad (i \in S_t)$ とすると、u^* のすべての要素が決定される。これを次に解く (Simple) の上界として用いる。

元の問題 (GUB) は、制約 $\sum_{i=1}^n F_i(x(i))$ を取り払い、代わりに得られた上界 $\boldsymbol{u}^*$ を用いた制約 $\boldsymbol{x} \leq \boldsymbol{u}^*$ を付け加えることで、最適解を変化させずに (Simple) と同じ問題クラスに変形することができる。この問題を解くことで、求めたい (GUB) の解が得られる。

得られた解が (GUB) の制約を満たすことは明らかであるが、最適性も満たしていることは明らかではなく、証明を要することである [29, 補題 6.2.1]。

線形制約つき分離凸関数最小化問題 (IP)

Hochbaum–Shanthikumar [31] は，線形制約と整数制約のある分離凸関数最小化問題

$$(\mathsf{IP}) \quad \text{Minimize} \quad \sum_{i=1}^n F_i(x(i)) \quad \text{subject to} \quad A\boldsymbol{x} \geq \boldsymbol{b}, \quad \boldsymbol{x} \in \mathbb{Z}^n$$

に連続緩和手法を適用した．ただし，$F_i : \mathbb{R} \to \mathbb{R}$ $(i \in N)$ は 1 変数凸関数であり，A は m 行 n 列の整数行列であり，$\boldsymbol{b} \in \mathbb{Z}^m$ である．行列 A の小行列式の絶対値の最大値を $\Delta \in \mathbb{Z}_+$ で表すことにする．問題 (IP) の連続緩和問題 ($\overline{\mathsf{IP}}$) は，(IP) から整数制約「$\boldsymbol{x} \in \mathbb{Z}^n$」を取り除くことで容易に導かれる．

文献 [31] では、線形制約つき分離凸関数最小化問題 (IP) に対して次の近接定理が示され、この近接定理を用いて，(IP) に対する効率的なアルゴリズムが考案された．

定理 4.24 ([31, 定理 3.3])**.**
(i) (IP) の任意の最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ に対して，($\overline{\mathsf{IP}}$) のある最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ が存在して $\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_\infty \leq n\Delta$ を満たす．
(ii) ($\overline{\mathsf{IP}}$) の任意の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ に対して，(IP) のある最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して $\|\boldsymbol{y}_* - \boldsymbol{x}_*\|_\infty \leq n\Delta$ を満たす． ■

なお、(SC) は (IP) の $\Delta = \mathrm{O}(2^n)$ の場合として定式化されるのに対して，(Simple), (GUB), (Nest), (Tree), (Network), (Laminar) は，それぞれ (IP) の $\Delta = 1$ の場合として定式化され

ることが容易にわかる．また、Granot–Skorin-Kapov [24] が，目的関数が分離 2 次の場合の (IP) について議論し，類似の近接定理を得ていたことにも触れておく．

劣モジュラ制約つき資源配分問題 (SC)

Hochbaum [29] は、劣モジュラ制約つき資源配分問題 (SC) に連続緩和手法を適用した。そのときに次の「近接定理」が主張された．

主張 A ([29] の系 4.3)
(i) (SC) の任意の最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ に対して，($\overline{\text{SC}}$) のある最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ が存在して，
$\boldsymbol{y}_* - \mathbf{1} < \boldsymbol{x}_* < \boldsymbol{y}_* + n\mathbf{1}$ を満たす．
(ii) ($\overline{\text{SC}}$) の任意の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ に対して，(SC) のある最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して，
$\boldsymbol{y}_* - \mathbf{1} < \boldsymbol{x}_* < \boldsymbol{y}_* + n\mathbf{1}$ を満たす． ■

主張 A の (ii) によると，ベクトル $\boldsymbol{u} = \boldsymbol{x}_* + \mathbf{1}$ を上界とする (SC) の最適解が存在することになる．しかしながら，この主張は誤りであり、次に述べる例 4.2 は、上の主張に対する反例を与える。この例は目的関数が 2 次の (Simple) であり、従って (SC) の特殊ケースでもあるが，$\boldsymbol{u}$ よりも遠いところに唯一の最適解が存在する。

例 4.2. 十分に小さい正の数 $\delta < 1$ を考える．(Simple) について，$K = n-1$, $u(i) = +\infty$ $(i \in N)$, $\alpha \in \mathbb{R}$ として，目的関数を

$$\begin{aligned} F_1(\alpha) &= \delta\alpha, \\ F_i(\alpha) &= (\alpha - 0.5 + \delta)^2 \qquad (i = 2, 3, \ldots, n) \end{aligned}$$

と定義する．このとき，$F_i(1) - F_i(0) = 2\delta > \delta$ $(i = 2, 3, \ldots, n)$ が成り立つ．このことから，(Simple) の最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ はただ 1 つで，$\boldsymbol{y}_* = (n-1, 0, \ldots, 0)$ に等しい．一方で，関数 F_i の傾きは，$\alpha = (1-\delta)/2$ のときに δ に等しく，$\alpha > (1-\delta)/2$ のときに δ よりも大きい $(i = 2, 3, \ldots, n)$．従って，連続緩和問題 ($\overline{\text{Simple}}$) の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ は，ただ 1 つで、

$$\boldsymbol{x}_* = \left(\frac{(n-1)(1+\delta)}{2}, \frac{1-\delta}{2}, \frac{1-\delta}{2}, \ldots, \frac{1-\delta}{2} \right)$$

によって与えられる．δ は十分に小さい正の数であることから，

$$y_*(1) - x_*(1) = (n-1) - \frac{(n-1)(1+\delta)}{2} = \frac{(n-1)(1-\delta)}{2}$$

であることがわかる．これより，$n \geq 4$ であれば $y_*(1) - x_*(1) > 1$ となり，したがって不等式 $\boldsymbol{y}_* - \mathbf{1} < \boldsymbol{x}_*$ が成り立たない． ■

Hochbaum [29] の連続緩和手法は，Hochbaum–Hong [30] によって目的関数が 2 次の (SC) の特殊ケースにも適用された．効率的な連続緩和法を開発するために，彼らは目的関数が 2 次の連続緩和問題 ($\overline{\text{Nest}}$)，($\overline{\text{Tree}}$)，($\overline{\text{Network}}$) に対する高速なアルゴリズムを考案して次の結果を得た。

定理 4.25 ([30])**.**
(i) ($\overline{\text{Nest}}$) と ($\overline{\text{Tree}}$) は，目的関数が 2 次であれば $\mathrm{O}(n \log n)$ の計算時間で解くことができる．
(ii) ($\overline{\text{Network}}$) は，目的関数が 2 次であれば $\mathrm{O}(|V||A| \log(|V|^2/|A|))$ の計算時間で解くことができる．■

この結果に基づき、先に述べた（誤った）主張 A を用いて，目的関数が 2 次の場合の (Nest), (Tree), (Network) に対するアルゴリズムが提案された [30]。その時間計算量は，(Nest) では $\mathrm{O}(n \log n)$，(Tree) では $\mathrm{O}(n \log n)$，(Network) では $\mathrm{O}(|V||A| \log(|V|^2/|A|))$ であると主張された．しかし主張 A は誤りであるから、この議論は正しくないことになる。

主張 A の代わりに，定理 4.24 (ii) からすぐに得られる系を用いて修正すると、以下のようになる．(Nest), (Tree), (Network) では，定理 4.24 の Δ が 1 に等しいことに注意する．

系 4.26. $\boldsymbol{x}_* \in \mathbb{R}^n$ を ($\overline{\text{Nest}}$)（あるいは ($\overline{\text{Tree}}$) や ($\overline{\text{Network}}$)）の最適解とする．このとき，(Nest) (あるいは (Tree) や (Network)) のある最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して，$\boldsymbol{y}_* \geq \boldsymbol{x}_* - n\mathbf{1}$ かつ $\|\boldsymbol{y}_* - (\boldsymbol{x}_* - n\mathbf{1})\|_1 = \mathrm{O}(n^2)$ を満たす．■

この性質は，(Nest), (Tree), (Network) に対して，最適解の下界としてベクトル $\boldsymbol{\ell} = \boldsymbol{x}_* - n\mathbf{1}$ を用いてよいことを示している．この事実と，文献 [30] と同様の手法で，(Nest), (Tree), (Network) はそれぞれ $\mathrm{O}(n^2 \log n)$, $\mathrm{O}(n^2 \log n)$, $\mathrm{O}(n^2|A| + |V||A| \log(|V|^2/|A|))$ の計算時間で解けることがわかる．これらは [30] で示された計算時間よりも n の係数部分が大きい．

本研究では，新たな (SC) の近接定理 (定理 4.27) を示し，(Nest), (Tree), (Network) への連続緩和法を改良し，上に述べた時間計算量を削減する．

4.6.3　連続緩和の近接定理

第 4.4.3 節で提案した連続緩和法を、M 凸関数最小化問題 (MC) の特殊ケースである (Laminar), (Nest), (Tree), (Network) に適用する。系 4.22 で述べたように、(MC) に対する近接定理の L_1 距離の限界値は $n(n-1)$ であるが、(SC) と (Laminar) の場合は以下に述べるように $2(n-1)$ に改良できる．(SC) と (Laminar) の連続緩和問題を，それぞれ ($\overline{\text{SC}}$) と ($\overline{\text{Laminar}}$) で表す．

定理 4.27 ((SC) に対する L_1 距離の近接定理)**.**
(i) (SC) の任意の最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ に対して，($\overline{\text{SC}}$) のある最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ が存在して $\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1 < 2(n-1)$ を満たす．
(ii) ($\overline{\text{SC}}$) の任意の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ に対して，(SC) のある最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して $\|\boldsymbol{y}_* - \boldsymbol{x}_*\|_1 < 2(n-1)$ を満たす．■

定理 4.28 ((Laminar) に対する L_1 距離の近接定理)**.**
(i) (Laminar) の任意の最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ に対して，($\overline{\text{Laminar}}$) のある最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ が存在して $\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1 < 2(n-1)$ を満たす．

(ii) ($\overline{\text{Laminar}}$) の任意の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ に対して，(Laminar) のある最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して $\|\boldsymbol{y}_* - \boldsymbol{x}_*\|_1 < 2(n-1)$ を満たす. ■

定理 4.27 の証明は第 4.7.2 節で、定理 4.28 の証明は第 4.7.3 節で与える.

これらの近接定理がこれ以上改善できないことを示す例を与えよう。(SC) や (Laminar) の特殊ケースである、単純な資源配分問題 (Simple) の例を考える。

例 4.3. δ を任意に選んだ微小な正の数とし，$\eta = 3\delta(1-\delta) - \delta = 2\delta - 3\delta^2$ (> 0) とする. $K = n-1$, $u(i) = +\infty$ $(i \in N)$, $\alpha \in \mathbb{R}$ として，(Simple) の目的関数が

$$\begin{aligned} f_1(\alpha) &= 2\delta\alpha, \\ f_i(\alpha) &= \max\left\{-\frac{\eta}{\delta}(\alpha-\delta), 3\delta(\alpha-\delta)\right\} \quad (i = 2, 3, \ldots, n) \end{aligned}$$

である場合を考える.

このとき，$i = 2, 3, \ldots, n$ について,

$$\begin{aligned} &f_i(\alpha+2) - f_i(\alpha+1) = 3\delta > \delta = f_i(1) - f_i(0) \qquad (\forall \alpha \in \mathbb{Z}_+), \\ &f_i(1) - f_i(0) = \delta < 2\delta = f_1(\alpha+1) - f_1(\alpha) \qquad (\forall \alpha \in \mathbb{Z}_+) \end{aligned}$$

が成り立つ．これらの不等式から，ベクトル $\boldsymbol{y}_* = (0, 1, \ldots, 1)$ が (Simple) の唯一の最適解であることがわかる．一方，関数 f_i $(i = 2, 3, \ldots, n)$ の傾きを 2 つの区間に分けて考えると，区間 $[0, \delta]$ では関数 f_i の傾きは $-\eta/\delta$ であり，関数 f_1 の傾き 2δ よりも真に小さい．区間 $[\delta, +\infty)$ では関数 f_i の傾きは 3δ であり，2δ よりも真に大きい．従って，連続緩和問題 ($\overline{\text{Simple}}$) の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ はただ 1 つに定まり，$\boldsymbol{x}_* = ((n-1)(1-\delta), \delta, \ldots, \delta)$ となる.

このとき,

$$\begin{aligned} \|\boldsymbol{y}_* - \boldsymbol{x}_*\|_\infty &= (n-1)(1-\delta), \\ \|\boldsymbol{y}_* - \boldsymbol{x}_*\|_1 &= 2(n-1)(1-\delta) \end{aligned}$$

が成り立って，L_∞ 距離は $n-1$ に、L_1 距離は $2(n-1)$ に，いくらでも近づけることができる. ■

4.6.4 連続緩和法の計算量

ここでは、資源配分問題に対する連続緩和法の計算量を解析して、先行研究と比較する。連続緩和手法は、第 4.4.3 節で述べたように、一般の M 凸関数最小化問題 (MC) に対しては最悪計算量を改善するわけではないが，以下のように (MC) の特殊ケースに限れば最悪計算量を改善する場合がある。特に、Hochbaum–Hong [30] と類似の手法で、2 次の目的関数の (Laminar) や (SC) の特殊ケースに対して改善できる。一般に、連続変数における凸 2 次の最適化問題のいくつかのクラスは，強多項式時間で解けることが知られている [7, 30, 90]. この事実を用い，またさらに効率的な連続緩和手法の実装を工夫することにより，2 次の目的関数の (Laminar) や (SC) の特殊ケースが強多項式時間で解けることを示す.

(Laminar) に対しては、目的関数が 2 次の場合、Tamir [90] の連続緩和手法による $O(n^3)$ が最良の計算時間である。ここで用いられた近接定理は、定理 4.24 と同様の (IP) を対象とした弱いもの [24] である。第 4.6.5 節では、近接定理を (Laminar) を対象に精密化し (定理 4.28)、アルゴリズム上の工夫を行なって計算時間を改善し、次の定理を得る。

定理 4.29. 層族制約つき資源配分問題 (Laminar) は，目的関数が 2 次であれば，$O(n^2)$ の計算時間で解ける．■

(SC) の特殊ケースであるネットワーク制約つき資源配分問題 (Network) に対しては、目的関数が 2 次の場合、Hochbaum–Hong [30] は、連続緩和問題の効率的なアルゴリズムを開発した上で、Hochbaum [29] の近接定理を用いて強多項式時間で解けると述べた．しかしながら彼らの用いた近接定理は，第 4.6.2 節で指摘したように誤りであるので、その計算時間の解析結果は無効であり、有効なのは連続緩和問題に関する部分にとどまる。第 4.6.6 節では，本論文で示した近接定理 (定理 4.27) を用いて，このネットワーク制約つき資源配分問題 (Network) が [30] に述べられたのと（ほぼ）同じ計算量で解けることを示す．特にこの (Network) に対しては，提案手法が最速のアルゴリズムとなる．

これまでに述べた問題クラスごとの計算量をまとめると、表 4.3 のようになる。

4.6.5　層族制約つき資源配分問題への応用

第 4.4.3 節で提案した連続緩和法 RELAX を，層族制約つき資源配分問題 (Laminar) に適用する．特に，目的関数が 2 次関数の場合の計算時間についての定理 4.29 を証明する．以下では，(Laminar) がツリーネットワーク上の凸費用流問題として定式化されているものとする（凸費用流の定式化については第 4.1 節を参照のこと）．層族 $\mathcal{F}$ を表現するツリーネットワークにおいては，任意の $X \in \mathcal{F}$ に対して X の子を $O(k)$ の計算時間で見つけることができる．ただし k は X の子の数である．さらに，$X \neq N$ であれば X の親を $O(1)$ の計算時間で求められる．(Laminar) の目的関数に用いられる各関数 f_X $(X \in \mathcal{F})$ の評価が定数時間でできることを仮定する．

まず，2 次の目的関数をもつ ($\overline{\text{Laminar}}$) の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ を求める方法を述べる．この問題はツリーネットワーク上の 2 次凸費用実数値フロー問題と等価であるので，Tamir のアルゴリズム [90] で $O(n^2)$ の計算時間で解くことができる．ゆえに，Step 1 のアルゴリズム RELAX は，目的関数が 2 次であれば，$O(n^2)$ の計算時間で行える．

次に，($\overline{\text{Laminar}}$) の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ を，どのようにして (Laminar) の実行可能解に丸めるのかを述べる．ここでは，ネットワークフローの技巧を用いる．不等式 $\ell_Y \leq x(Y) \leq u_Y$ $(Y \in \mathcal{F})$ を定義する係数行列は完全ユニモジュラであるので，ある整数ベクトル $\boldsymbol{y} \in \mathbb{Z}^n$ が存在して，任意の $Y \in \mathcal{F}$ に対して

$$(\ell_Y \leq)\ \lfloor x_*(Y) \rfloor \leq y(Y) \leq \lceil x_*(Y) \rceil\ (\leq u_Y) \tag{4.7}$$

を満たし（[2, 84] などを参照)、従って $\boldsymbol{y}$ が (Laminar) の実行可能解であることがわかる．ま

表 4.3. 離散凸最適化問題の計算量

問題クラス	近接定理の距離		目的関数が 2 次の場合の計算量	
	L_∞	L_1	連続緩和	全体
(MC)	$n-1$ (定理 4.21) —	$n(n-1)$ (系 4.22) —	$*^1$ —	$*^1$ (定理 4.23) $*^1$ [88]
(SC)	↑ $O(n2^n)$ [31]	$2(n-1)$ (定理 4.27) —	↑	↑
(Network)	↑ n [31]	↑	— $*^2$ [30]	$*^2$ (定理 4.31) $*^3$ (系 4.26)
(Laminar)	↑	$2(n-1)$ (定理 4.28) —	— $O(n^2)$ [90]	$O(n^2)$ (定理 4.29) $O(n^3)$ [90]
(Tree)	↑	↑	— $O(n \log n)$ [30]	↑ $*^4$ (系 4.26)
(Nest)	↑	↑	— $O(n \log n)$ [30]	↑ $*^4$ (系 4.26)
(GUB)	↑	↑ ↓ [29]	— ↓ [29]	— ↓ [29]
(Simple)	↑	↑ $O(n)$ [98]	— $O(n)$ [7]	— $O(n)$ [33]

$*^1$: $O((n^5 + n^4 \log(K_\infty/n))(\log(K_\infty/n)/\log n))$
$*^2$: $O(|V||A| \log(|V|^2/|A|))$
$*^3$: $O(n^2|A| + |V||A| \log(|V|^2/|A|))$
$*^4$: $O(n^2 \log n)$

上段：本章による
下段：先行研究による
—：成果なし
↑：上の欄に同じ
↓：下の欄に同じ
網掛：本章による改善

た、任意の $i \in N$ について $\{i\} \in \mathcal{F}$ であるので，条件 (4.7) から $\|\boldsymbol{y} - \boldsymbol{x}_*\|_1 < n$ を満たす。

式 (4.7) を満たす整数ベクトル $\boldsymbol{y} \in \mathbb{Z}^n$ は，以下のようにすると $O(n)$ の計算時間で求まる．各 $Y \in \mathcal{F}$ について $\ell'_Y = \lfloor \boldsymbol{x}_*(Y) \rfloor$ かつ $u'_Y = \lceil \boldsymbol{x}_*(Y) \rceil$ とする．任意の $Y \in \mathcal{F}$ に対して $\ell'_Y \le y(Y) \le u'_Y$ を満たすベクトル $\boldsymbol{y} \in \mathbb{Z}^n$ を求めるために，条件

$$\begin{aligned}
&\ell'_Y \le \varphi_Y \le u'_Y \ (\forall Y \in \mathcal{F}),\\
&\varphi_N = K,\\
&\varphi_X = \sum \{\varphi_Y \mid Y \in \mathcal{F},\ Y \text{ は } X \text{ の子}\} \ (\forall X \in \mathcal{F},\ |X| \ge 2)
\end{aligned}$$

を満たす φ_Y $(Y \in \mathcal{F})$ の値を，以下のような方法で天下り的に求める．

Step 0: $\varphi_N = K$ とおく．
Step 1: φ_Y の値がすべての $Y \in \mathcal{F}$ について求められていれば，$y(i) = \varphi_{\{i\}}$ $(i \in N)$ で定義されるベクトル $\boldsymbol{y} \in \mathbb{Z}^n$ を出力して，手続きを終了する．

Step 2: $X \in \mathcal{F}$ のうち，$|X| \geq 2$ であり，かつ φ_X は既に求められているが，X のすべての子 Y についての φ_Y の値が求められているわけではないものを取り上げる．$Y_1, Y_2, \ldots, Y_k \in \mathcal{F}$ を X のすべての子とする．$t = 1, 2, \ldots, k$ について，$\sum_{t=1}^{k} \varphi_{Y_t} = \varphi_X$ が成り立つように，ℓ'_{Y_t} あるいは u'_{Y_t} の値のどちらかを適切に選んで，φ_{Y_t} の値とする．Step 1 に戻る．

Step 2 は $\mathrm{O}(k)$ の計算時間で実行できる．なぜなら，ℓ'_Y, u'_Y の値は

$$u'_Y = \ell'_Y \text{ または } u'_Y = \ell'_Y + 1 \quad (\forall Y \in \mathcal{F}),$$
$$\sum_{t=1}^{k} \ell'_{Y_t} \leq \ell'_X \leq u'_X \leq \sum_{t=1}^{k} u'_{Y_t}$$
$$(\forall X \in \mathcal{F},\ |X| \geq 2,\ Y_1, Y_2, \ldots, Y_k \in \mathcal{F} \text{ は } X \text{ の子})$$

の性質を満たすからである．$|\mathcal{F}| = \mathrm{O}(n)$ であるので，上のアルゴリズムは $\mathrm{O}(n)$ の計算時間で実行できる．このことから，アルゴリズム RELAX の Step 2 は $\mathrm{O}(n)$ の計算時間で実行されることがわかる．

最後に，アルゴリズム RELAX の Step 3 が $\mathrm{O}(n^2)$ の計算時間で実行されることを示す．Step 2 で得られるベクトル $\boldsymbol{y}^\circ \in \mathbb{Z}^n$ は $\|\boldsymbol{y}^\circ - \boldsymbol{x}_*\|_1 < n$ を満たすので，(Laminar) の近接定理 (定理 4.28) より，(Laminar) のある最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して，

$$\|\boldsymbol{y}_* - \boldsymbol{y}^\circ\|_1 \leq \|\boldsymbol{y}_* - \boldsymbol{x}_*\|_1 + \|\boldsymbol{x}_* - \boldsymbol{y}^\circ\|_1 < 2(n-1) + n < 3n$$

を満たすことがわかる．ゆえに，アルゴリズム RELAX の Step 3 で用いられる NewGreedy は，定理 4.12 により $\mathrm{O}(n)$ 回の反復で終了する．

アルゴリズム NewGreedy の 1 反復あたりの計算時間は $\mathrm{O}(nT_{\mathrm{func}})$ である。$|\mathcal{F}| = \mathrm{O}(n)$ であるので，目的関数の評価にかかる時間 T_{func} は $\mathrm{O}(n)$ である．従って，素朴に実装すると，NewGreedy の各反復には $\mathrm{O}(n^2)$ の計算時間が必要である．この計算量は，ネットワークフローの技巧を用いて $\mathrm{O}(n)$ に削減することができる。その手順は次の通りである．

ここでは (Laminar) をツリーネットワーク上の凸費用流問題として定式化しなおす．凸費用流問題の，いわゆる残余ネットワークを構築する（[2] などを参照）．(Laminar) の与えられた実行可能解 $\boldsymbol{y} \in \mathbb{Z}^n$ について，有向グラフ $G_{\boldsymbol{y}} = (V, A_{\boldsymbol{y}})$ を作る．グラフの頂点集合は $V = \{v_Y \mid Y \in \mathcal{F}\}$ とし，枝集合 $A_{\boldsymbol{y}}$ は

$$A_{\boldsymbol{y}} = \{(v_{p(Y)}, v_Y) \mid Y \in \mathcal{F} \setminus \{N\},\ y(Y) < u_Y\} \cup \{(v_Y, v_{p(Y)}) \mid Y \in \mathcal{F} \setminus \{N\},\ y(Y) > \ell_Y\}$$

とする．枝の長さは次のように定める．

- $(v_{p(Y)}, v_Y)$ の形の枝は，その長さを $f_Y(y(Y)+1) - f_Y(y(Y))$ とする．
- $(v_Y, v_{p(Y)})$ の形の枝は，その長さを $f_Y(y(Y)-1) - f_Y(y(Y))$ とする．

さて，f_Y は凸関数であるので，

$$\big[f_Y(y(Y)+1) - f_Y(y(Y))\big] + \big[f_Y(y(Y)-1) - f_Y(y(Y))\big] \geq 0$$

である．このことから，グラフ $G_{\boldsymbol{y}}$ が負の長さの有向閉路をもたないことがわかる．

$i,j \in N$ について，ベクトル $\boldsymbol{y}-\boldsymbol{\chi}_i+\boldsymbol{\chi}_j$ が (Laminar) の実行可能解であるためには，頂点 $v_{\{i\}}$ から $v_{\{j\}}$ までの有向パスが存在することが必要十分である．ある $i,j \in N$ について $\boldsymbol{y}-\boldsymbol{\chi}_i+\boldsymbol{\chi}_j$ が実行可能解であれば，$f_{\mathrm{sum}}(\boldsymbol{y}-\boldsymbol{\chi}_i+\boldsymbol{\chi}_j)-f_{\mathrm{sum}}(\boldsymbol{y})$ の値は，頂点 $v_{\{i\}}$ から $v_{\{j\}}$ までの最も短い有向パスの長さに等しい．ただし，$\boldsymbol{y}' \in \mathbb{Z}^n$ について $f_{\mathrm{sum}}(\boldsymbol{y}')=\sum_{Y\in\mathcal{F}} f_Y(y'(Y))$ とする．ネットワーク構造を定めている（無向）グラフ $G_{\boldsymbol{y}}$ はツリーであるので，ある頂点から他の頂点への最短路は 1 つに定まる．線形時間グラフ探索アルゴリズムを用いると，定められた $i \in N$ に対して，すべての $j \in N$ に対して $v_{\{i\}}$ から $v_{\{j\}}$ までの有向の最短路の長さを，$\mathrm{O}(n)$ の計算時間で求めることができる．同様に，定められた $i \in N$ に対して，すべての $j \in N$ に対して，逆向きの $v_{\{j\}}$ から $v_{\{i\}}$ までの最も短い有向路の長さを，$\mathrm{O}(n)$ の計算時間で求めることができる．このことから，アルゴリズム NewGreedy の各反復が $\mathrm{O}(n)$ の計算時間で実行できることがわかる．

これまでの議論をまとめると，目的関数が 2 次であれば，(Laminar) を $\mathrm{O}(n^2)$ の計算時間で解くことができる．従って，定理 4.29 が証明された．(Nest) と (Tree) は (Laminar) の特殊ケースであるので，定理 4.29 の系として，次の結果が得られる．

系 4.30. 問題 (Nest) と問題 (Tree) は，目的関数が 2 次であれば，$\mathrm{O}(n^2)$ の計算時間で解くことができる． ■

定理 4.25(i) により，(Nest) と (Tree) のそれぞれの連続緩和問題である ($\overline{\text{Nest}}$) と ($\overline{\text{Tree}}$) は，どちらも ($\overline{\text{Laminar}}$) に必要な $\mathrm{O}(n^2)$ よりも少ない $\mathrm{O}(n\log n)$ の計算時間で解くことができる．しかしながら (Nest) や (Tree) でも，Step 3 に $\mathrm{O}(n^2)$ の計算時間がかかるため，アルゴリズム RELAX 全体の計算量が削減できるわけではない．(Nest) と (Tree) の目的関数が 2 次の場合に，$\mathrm{O}(n\log n)$ の計算時間で解けるアルゴリズムが存在するかどうかは，未解決問題である．

4.6.6 ネットワーク制約つき資源配分問題への応用

第 4.4.3 節で提案した連続緩和法 RELAX を，(SC) の特殊ケースであるネットワーク制約つき資源配分問題 (Network) に適用する．目的関数が 2 次関数で与えられた場合を考える．以下では，(Network) の目的関数に用いられる各関数 f_i $(i \in N)$ は，定数時間で関数値の評価ができるものとする．

まず，アルゴリズム RELAX の Step 1 の連続緩和問題 ($\overline{\text{Network}}$) が解けたとする．定理 4.25 により，($\overline{\text{Network}}$) は，目的関数が 2 次であれば，$\mathrm{O}(|V||A|\log(|V|^2/|A|))$ の計算時間で解くことができる．ただし，V と A はネットワーク構造を与える有向グラフの頂点集合と枝集合を表す．また、$N=\{1,2,\ldots,n\}$ はシンクノードの集合で、V の部分集合である。

次に，連続緩和問題の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ を元問題の実行可能解に丸める方法を説明する．$(\boldsymbol{x}_*,\boldsymbol{\varphi}_*) \in \mathbb{R}^n \times \mathbb{R}^{|A|}$ を ($\overline{\text{Network}}$) の最適解とする．($\overline{\text{Network}}$) の制約を定義する係数行列は完全ユニモジュラであるので，整数ベクトル $\boldsymbol{y} \in \mathbb{Z}^n$ と $\boldsymbol{\psi} \in \mathbb{Z}^{|A|}$ のペアが存在して，$(\boldsymbol{y},\boldsymbol{\psi})$

が (Network) の実行可能解であり，かつ

$$\lfloor x_*(i) \rfloor \leq y(i) \leq \lceil x_*(i) \rceil \ (\forall i \in N), \quad \lfloor \varphi_*(a) \rfloor \leq \psi(a) \leq \lceil \varphi_*(a) \rceil \ (\forall a \in A) \tag{4.8}$$

を満たす ([2, 84] などを参照)．このような $(\boldsymbol{y}, \boldsymbol{\psi})$ は最大流のアルゴリズムを用いて効率的に求めることができる．例えば，Goldberg–Tarjan [23] のアルゴリズムならば $O(|V||A| \log(|V|^2/|A|))$ の計算時間で求めることができる．ゆえに，アルゴリズム RELAX の Step 2 は $O(|V||A| \log(|V|^2/|A|))$ の計算時間で実行できる．なお、条件 (4.8) より，$\|\boldsymbol{y} - \boldsymbol{x}_*\|_1 < n$ であることがわかる．

最後に，アルゴリズム RELAX の Step 3 が $O(n(|V| + |A|))$ の計算時間で実行できることを示す．Step 2 で求めたベクトル $\boldsymbol{y}^\circ \in \mathbb{Z}^n$ が $\|\boldsymbol{y}^\circ - \boldsymbol{x}_*\|_1 < n$ を満たすことと，(SC) の近接定理 (定理 4.27) を合わせて考えると，(Network) の最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して，

$$\|\boldsymbol{y}_* - \boldsymbol{y}^\circ\|_1 \leq \|\boldsymbol{y}_* - \boldsymbol{x}_*\|_1 + \|\boldsymbol{x}_* - \boldsymbol{y}^\circ\|_1 < 2(n-1) + n < 3n$$

を満たすことがわかる．ゆえに，RELAX の Step 3 で用いられるアルゴリズム NewGreedy は，定理 4.12 より $O(n)$ の反復回数で終了する．以下では，Step 3 で用いられる NewGreedy の各反復が，ネットワークフローの技工を用いると，$O(|V| + |A|)$ の計算時間で行えることを示す．

凸費用流問題に対する，いわゆる残余ネットワークを構築する（[2] などを参照）．(Network) の与えられた実行可能解 $(\boldsymbol{y}, \boldsymbol{\psi}) \in \mathbb{Z}^n \times \mathbb{Z}^{|A|}$ について，有向グラフ $G_{\boldsymbol{\psi}} = (V, A_{\boldsymbol{\psi}})$ を作る．枝集合 $A_{\boldsymbol{\psi}}$ は

$$A_{\boldsymbol{\psi}} = \{(h,k) \mid (h,k) \in A,\ \psi(h,k) < c(h,k)\} \cup \{(k,h) \mid (h,k) \in A,\ \psi(h,k) > 0\}$$

とする．ただし $c(i,j) \in \mathbb{Z}_+$ は弧 (i,j) の容量とする。$i, j \in N$ について, ベクトル $\boldsymbol{y} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j$ が (Network) の実行可能解であるためには，$\boldsymbol{0} \leq \boldsymbol{y} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j \leq \boldsymbol{u}$ かつ，$G_{\boldsymbol{\psi}}$ に頂点 i から j までの有向パスが存在することが必要十分である．また，

$$f_{\mathrm{sum}}(\boldsymbol{y} - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j) - f_{\mathrm{sum}}(\boldsymbol{y}) = \big[f_i(y(i) - 1) - f_i(y(i))\big] + \big[f_j(y(j) + 1) - f_j(y(j))\big]$$

が成り立つ．ただし，$\boldsymbol{y}' \in \mathbb{Z}^n$ について $f_{\mathrm{sum}}(\boldsymbol{y}') = \sum_{k \in N} f_k(y'(k))$ とする．線形時間グラフ探索アルゴリズムを用いると，定められた $i \in N$ に対して，頂点 i から $G_{\boldsymbol{\psi}}$ の中の到達可能な頂点集合を $O(|V| + |A|)$ の計算時間で求めることができる．同様に，定められた $i \in N$ に対して，頂点 i に到達できる $G_{\boldsymbol{\psi}}$ の中の頂点集合も，$O(|V| + |A|)$ の計算時間で求めることができる．このことから，アルゴリズム NewGreedy の各反復が $O(|V| + |A|)$ の計算時間で実行できることがわかる．

これまでの議論と定理 4.25 (ii) より，目的関数が 2 次の問題 (Network) について，次の結果が得られたことになる．

定理 4.31. ネットワーク制約つき資源配分問題 (Network) は，目的関数が 2 次であれば，$O(|V||A| \log(|V|^2/|A|))$ の計算時間で解くことができる． ■

4.7 近接定理の証明

連続緩和の 3 つの近接定理の証明を与える [53].

4.7.1 M 凸関数最小化問題の近接定理の証明

M 凸関数最小化問題 (MC) の近接定理である，定理 4.21 を証明する．この定理の証明のためには，(MC) の代わりに，以下のような問題 (GMC)

$$\text{(GMC)} \qquad \text{Minimize} \quad F(\boldsymbol{x}) \qquad \text{subject to} \quad \boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F \cap \mathbb{Z}^n$$

を考えるほうが都合がよい．ただし，$F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ は連続変数の閉真 M 凸関数である．定理 4.1 に述べたように，任意の離散 M 凸関数 $f : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ について，連続変数のある閉真 M 凸関数 $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ が存在して，任意の $\boldsymbol{x} \in \mathbb{Z}^n$ に対して $F(\boldsymbol{x}) = f(\boldsymbol{x})$ を満たすので、問題 (GMC) は (MC) よりも一般化されていることがわかる。

(GMC) の連続緩和問題は，ごく自然に

$$(\overline{\text{GMC}}) \qquad \text{Minimize} \quad F(\boldsymbol{x}) \qquad \text{subject to} \quad \boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F$$

と定義される．問題 (GMC) の近接定理を示す.

定理 4.32.
(i) (GMC) の任意の最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ に対して，($\overline{\text{GMC}}$) のある最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ が存在して，$\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_\infty < n - 1$ を満たす.
(ii) ($\overline{\text{GMC}}$) の任意の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ に対して，(GMC) のある最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して，$\|\boldsymbol{y}_* - \boldsymbol{x}_*\|_\infty < n - 1$ を満たす. ■

問題 (MC) は (GMC) の特殊ケースであるので，定理 4.21 は定理 4.32 から即座に導かれる．定理 4.32 から、閉真 M 凸関数 F が $\mathrm{argmin}_{\mathbb{R}} F \neq \emptyset$ を満たすためには，$\mathrm{argmin}\{F(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathbb{Z}^n\} \neq \emptyset$ となることが必要十分であることがわかる（後述する注意 4.2 も参照されたい）.

定理 4.32 (i) の証明

定理 4.32(i) の証明のために，これから述べる 2 つの性質を用いる．次の補題の主張は，閉真 M 凸関数 F を任意に選んだ座標軸 $i \in N$ に沿って射影すると，優モジュラ関数になるというものである.

補題 4.33 ([72, 命題 3.12])**.** 任意の $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ と任意の $i \in N$ について，$F(\boldsymbol{x}) + F(\boldsymbol{y}) \leq$

$F(\hat{\boldsymbol{x}}) + F(\check{\boldsymbol{y}})$ が成り立つ．ただし，$\hat{\boldsymbol{x}}$ と $\check{\boldsymbol{y}}$ は

$$\hat{x}(j) = \begin{cases} \min\{x(j), y(j)\} & (j \in N \setminus \{i\} \text{ のとき}), \\ x(N) - \sum_{k \in N \setminus \{i\}} \min\{x(k), y(k)\} & (j = i \text{ のとき}), \end{cases}$$

$$\check{y}(j) = \begin{cases} \max\{x(j), y(j)\} & (j \in N \setminus \{i\} \text{ のとき}), \\ y(N) - \sum_{k \in N \setminus \{i\}} \max\{x(k), y(k)\} & (j = i \text{ のとき}) \end{cases}$$

と定義されるベクトルである。 ■

実数 $\gamma \in \mathbb{R}$ に対して，

$$\mathrm{level}(F, \gamma) = \{\boldsymbol{x} \in \mathbb{R}^n \mid F(\boldsymbol{x}) \leq \gamma\}$$

と定義する．このとき，F は閉凸関数であるので，$\mathrm{level}(F, \gamma)$ は閉集合である（[83, 定理 7.1] などを参照）．

補題 4.34. ベクトル $\boldsymbol{y}_* \in \mathrm{dom}_{\mathbb{R}} F$ と，$\mathrm{level}(F, \gamma)$ が空にならない実数値 $\gamma \in \mathbb{R}$ を考える．ベクトル $\tilde{\boldsymbol{x}} \in \mathrm{level}(F, \gamma)$ が $\mathrm{level}(F, \gamma)$ の中で $\|\tilde{\boldsymbol{x}} - \boldsymbol{y}_*\|_1$ の値を最小にするとする．
(i) $k \in N$ が

$$F(\boldsymbol{y}_* - \boldsymbol{\chi}_i + \boldsymbol{\chi}_k) \geq F(\boldsymbol{y}_*) \qquad (\forall i \in N) \tag{4.9}$$

の条件を満たすとき，$\tilde{x}(k) - y_*(k) < n - 1$ となる．
(ii) $k \in N$ が

$$F(\boldsymbol{y}_* - \boldsymbol{\chi}_k + \boldsymbol{\chi}_j) \geq F(\boldsymbol{y}_*) \qquad (\forall j \in N)$$

の条件を満たすとき，$\tilde{x}(k) - y_*(k) > -(n - 1)$ となる． ■

証明. (i) のみを証明する．(ii) も同様に証明できる．$\tilde{x}(k) > y_*(k)$ の場合を考えればよい．このとき，

$$\begin{aligned} &F(\tilde{\boldsymbol{x}} - \varepsilon(\boldsymbol{\chi}_k - \boldsymbol{\chi}_i)) > F(\tilde{\boldsymbol{x}}) \\ &\qquad (\forall i \in \mathrm{supp}^-(\tilde{\boldsymbol{x}} - \boldsymbol{y}_*),\ 0 < \forall \varepsilon \leq \min\{\tilde{x}(k) - y_*(k), y_*(i) - \tilde{x}(i)\}) \end{aligned} \tag{4.10}$$

が成り立つ．なぜなら，そうでないとすれば，あるベクトル $\boldsymbol{x}' \in \mathrm{dom}_{\mathbb{R}} F$ が存在して，$F(\boldsymbol{x}') \leq F(\tilde{\boldsymbol{x}}) \leq \gamma$ かつ $\|\boldsymbol{x}' - \boldsymbol{y}_*\|_1 < \|\tilde{\boldsymbol{x}} - \boldsymbol{y}_*\|_1$ を満たすことになるが，これは $\tilde{\boldsymbol{x}}$ の選び方に矛盾するからである．$\mathrm{supp}^-(\tilde{\boldsymbol{x}} - \boldsymbol{y}_*) = \{i_1, i_2, \ldots, i_t\}$ とする．ただし，$t = |\mathrm{supp}^-(\tilde{\boldsymbol{x}} - \boldsymbol{y}_*)|$ $(\leq n-1)$ である．$\boldsymbol{y}_0 = \boldsymbol{y}_*$ として，$\lambda_h \in \mathbb{R}_+$ と $\boldsymbol{y}_h \in \mathbb{R}^n$ を $h = 1, 2, \ldots, t$ に対して順次

$$\begin{aligned} \lambda_h = \sup\{\lambda \mid\ & \boldsymbol{y}_{h-1} + \lambda(\boldsymbol{\chi}_k - \boldsymbol{\chi}_{i_h}) \in \mathrm{dom}_{\mathbb{R}} F, \\ & \lambda \leq \min\{\tilde{x}(k) - y_{h-1}(k), y_{h-1}(i_h) - \tilde{x}(i_h)\}, \\ & F(\boldsymbol{y}_{h-1} + \lambda'(\boldsymbol{\chi}_k - \boldsymbol{\chi}_{i_h})) \text{ は} \lambda' \in [0, \lambda] \text{ の範囲で真に減少する } \}, \\ \boldsymbol{y}_h = \boldsymbol{y}_{h-1} + \lambda_h & (\boldsymbol{\chi}_k - \boldsymbol{\chi}_{i_h}) \end{aligned}$$

と定義する．ここで、$\lambda_h = 0$ の場合もあり得る。$\boldsymbol{y}_h$ の定義と F の閉凸性から，

$$F(\boldsymbol{y}_h) < F(\boldsymbol{y}_{h-1}) \quad (\lambda_h > 0 \text{ のとき}), \tag{4.11}$$

$$\begin{aligned} &F(\boldsymbol{y}_h + \lambda(\boldsymbol{\chi}_k - \boldsymbol{\chi}_{i_h})) \geq F(\boldsymbol{y}_h)\ (\forall \lambda > 0) \\ &\qquad (\tilde{x}(k) > y_h(k) \text{ かつ } y_h(i_h) > \tilde{x}(i_h) \text{ のとき}) \end{aligned} \tag{4.12}$$

が成り立つ.

Claim 1: $\sum_{h=1}^{t}\lambda_h=\tilde{x}(k)-y_0(k)$.
[Claim 1 の証明] 矛盾を導くために, $\sum_{h=1}^{t}\lambda_h<\tilde{x}(k)-y_0(k)$ と仮定する. $k\in\mathrm{supp}^+(\tilde{\boldsymbol{x}}-\boldsymbol{y}_t)$ であるので, (M-EXC[$\mathbb{R}$]) よりある $i_h\in\mathrm{supp}^-(\tilde{\boldsymbol{x}}-\boldsymbol{y}_t)$ と十分に小さな $\lambda>0$ が存在して,

$$F(\tilde{\boldsymbol{x}})+F(\boldsymbol{y}_t)\geq F(\tilde{\boldsymbol{x}}-\lambda(\boldsymbol{\chi}_k-\boldsymbol{\chi}_{i_h}))+F(\boldsymbol{y}_t+\lambda(\boldsymbol{\chi}_k-\boldsymbol{\chi}_{i_h}))$$

を満たす. 補題 4.33 の $i=k$ の場合を考えると,

$$F(\boldsymbol{y}_h+\lambda(\boldsymbol{\chi}_k-\boldsymbol{\chi}_{i_h}))+F(\boldsymbol{y}_t)\leq F(\boldsymbol{y}_t+\lambda(\boldsymbol{\chi}_k-\boldsymbol{\chi}_{i_h}))+F(\boldsymbol{y}_h)$$

となる. この 2 つの不等式を合わせると,

$$F(\boldsymbol{y}_h+\lambda(\boldsymbol{\chi}_k-\boldsymbol{\chi}_{i_h}))-F(\boldsymbol{y}_h)\leq F(\tilde{\boldsymbol{x}})-F(\tilde{\boldsymbol{x}}-\lambda(\boldsymbol{\chi}_k-\boldsymbol{\chi}_{i_h}))<0 \tag{4.13}$$

となる. 最後の不等式は, $i_h\in\mathrm{supp}^-(\tilde{\boldsymbol{x}}-\boldsymbol{y}_t)\subseteq\mathrm{supp}^-(\tilde{\boldsymbol{x}}-\boldsymbol{y}_*)$ と式 (4.10) による。この式 (4.13) は, 式 (4.12) に矛盾する. [Claim 1 の証明終わり]

Claim 2: $h=1,2,\ldots,t$ について, $\lambda_h>0$ であれば $F(\boldsymbol{y}_*+\lambda_h(\boldsymbol{\chi}_k-\boldsymbol{\chi}_{i_h}))<F(\boldsymbol{y}_*)$ が成り立つ.
[Claim 2 の証明] $h\in\{1,2,\ldots,t\}$, $\lambda_h>0$ とする. 補題 4.33 の $i=k$ の場合を考えると,

$$F(\boldsymbol{y}_*+\lambda_h(\boldsymbol{\chi}_k-\boldsymbol{\chi}_{i_h}))+F(\boldsymbol{y}_{h-1})\leq F(\boldsymbol{y}_h)+F(\boldsymbol{y}_*)$$

となり,

$$F(\boldsymbol{y}_*+\lambda_h(\boldsymbol{\chi}_k-\boldsymbol{\chi}_{i_h}))-F(\boldsymbol{y}_*)\leq F(\boldsymbol{y}_h)-F(\boldsymbol{y}_{h-1})<0$$

が成り立つ. なお、最後の不等式は式 (4.11) による. [Claim 2 の証明終わり]

不等式 (4.9) と F の凸性を用いると,

$$F(\boldsymbol{y}_*+\beta(\boldsymbol{\chi}_k-\boldsymbol{\chi}_i))\geq F(\boldsymbol{y}_*)\qquad(\text{任意の}\beta\geq 1,\ i\in N\text{ に対して})$$

が成り立つ. 従って Claim 2 より任意の $h=1,2,\ldots,t$ に対して $\lambda_h<1$ が言えて, Claim 1 と合わせると, 目的の不等式

$$\tilde{x}(k)-y_*(k)=\tilde{x}(k)-y_0(k)=\sum_{h=1}^{t}\lambda_h<t\leq n-1$$

の成り立つことがわかる. ■

定理 4.32(i) を証明する準備が整った. $\boldsymbol{y}_*$ が (GMC) の最小解である, つまり $F(\boldsymbol{y}_*)=\min\{F(\boldsymbol{y})\mid\boldsymbol{y}\in\mathbb{Z}^n\}$ を満たすとする. 任意の $k\in N$ に対して不等式

$$F(\boldsymbol{y}_*-\boldsymbol{\chi}_i+\boldsymbol{\chi}_k)\geq F(\boldsymbol{y}_*)\qquad(\text{任意の}\ i\in N\text{ に対して}), \tag{4.14}$$
$$F(\boldsymbol{y}_*-\boldsymbol{\chi}_k+\boldsymbol{\chi}_j)\geq F(\boldsymbol{y}_*)\qquad(\text{任意の}\ j\in N\text{ に対して}) \tag{4.15}$$

が成り立つ.

まず $\mathrm{argmin}_{\mathbb{R}} F \neq \emptyset$ を仮定し，$\gamma = \min\{F(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathbb{R}^n\}$ とする．このとき，$\mathrm{level}(F,\gamma) = \mathrm{argmin}_{\mathbb{R}} F$ となる．$\tilde{\boldsymbol{x}} \in \mathbb{R}^n$ を $\mathrm{level}(F,\gamma)$ のすべてのベクトルの中で，$\|\tilde{\boldsymbol{x}} - \boldsymbol{y}_*\|_1$ の値を最小化するものと仮定する．式 (4.14) と補題 4.34(i) より，任意の $k \in N$ に対して $\tilde{x}(k) - y_*(k) < n-1$ となる．同様に，式 (4.15) と補題 4.34(ii) より，任意の $k \in N$ に対して $\tilde{x}(k) - y_*(k) > -(n-1)$ となる．これより，$\tilde{\boldsymbol{x}} \in \mathrm{level}(F,\gamma) = \mathrm{argmin}_{\mathbb{R}} F$ は $\|\tilde{\boldsymbol{x}} - \boldsymbol{y}_*\|_\infty < n-1$ を満たすことがわかる.

$\mathrm{argmin}_{\mathbb{R}} F \neq \emptyset$ を示せば証明が完了する．このために，次の性質

$$\begin{array}{l}\mathrm{level}(F,\gamma) \neq \emptyset \text{ を満たす任意の} \gamma \in \mathbb{R} \text{ に対して,} \\ \text{ある } \boldsymbol{x} \in \mathrm{level}(F,\gamma) \text{ が存在して，} \|\boldsymbol{x} - \boldsymbol{y}_*\|_\infty \leq n-1 \text{ を満たす}\end{array} \tag{4.16}$$

に着目する．実際、$\tilde{\boldsymbol{x}} \in \mathrm{dom}_{\mathbb{R}} F$ を $\mathrm{level}(F,\gamma)$ のベクトルの中で，$\|\tilde{\boldsymbol{x}} - \boldsymbol{y}_*\|_1$ の値を最小化するものとすると、補題 4.34 (i) と式 (4.14) より任意の $k \in N$ に対して $\tilde{x}(k) - y_*(k) < n-1$ となり、同様に，補題 4.34 (ii) と式 (4.15) より任意の $k \in N$ に対して $\tilde{x}(k) - y_*(k) > -(n-1)$ となるので、式 (4.16) が成り立つことがわかる.

式 (4.16) の性質から,

$$\begin{aligned}\inf\{F(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F,\ \|\boldsymbol{x} - \boldsymbol{y}_*\|_\infty \leq n-1\} &= \inf\{F(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F\}, \\ \mathrm{argmin}\,\{F(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F,\ \|\boldsymbol{x} - \boldsymbol{y}_*\|_\infty \leq n-1\} &\subseteq \mathrm{argmin}_{\mathbb{R}} F\end{aligned}$$

が成り立つことがわかる．また、

$$\mathrm{argmin}\,\{F(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F, \|\boldsymbol{x} - \boldsymbol{y}_*\|_\infty \leq n-1\} \neq \emptyset$$

が成り立つ．なぜなら，F は閉真凸関数であり，$\{\boldsymbol{x} \in \mathrm{dom}_{\mathbb{R}} F \mid \|\boldsymbol{x} - \boldsymbol{y}_*\|_\infty \leq n-1\}$ は有界かつ閉集合だからである．ゆえに，$\mathrm{argmin}_{\mathbb{R}} F \neq \emptyset$ となる.

定理 4.32 (ii) の証明

定理 4.32 (i) と (ii) は、近接の方向がいわば逆方向の関係にある。そこで (ii) の証明には，先に証明した (i) に摂動を適用する．$\boldsymbol{x}_* \in \mathbb{R}^n$ を ($\overline{\mathsf{GMC}}$) の最適解とする．$\boldsymbol{y}_* \in \mathbb{Z}^n$ を (GMC) の最適解のうち，$\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1$ の値を最小にするものとする．正の数 δ を用いて，新しい問題 (GMC^δ) を

$$(\mathsf{GMC}^\delta) \qquad \text{Minimize} \quad F(\boldsymbol{y}) + \delta\|\boldsymbol{y} - \boldsymbol{x}_*\|_1 \qquad \text{subject to} \quad \boldsymbol{y} \in \mathbb{Z}^n$$

と定義する．関数 $\delta\|\boldsymbol{y} - \boldsymbol{x}_*\|_1$ は $\boldsymbol{y}$ についての分離凸関数であるので，$F(\boldsymbol{y}) + \delta\|\boldsymbol{y} - \boldsymbol{x}_*\|_1$ は $\boldsymbol{y}$ についての閉真 M 凸関数である．なぜなら、分離凸関数との和は M 凸性を保存するからである [62, 定理 6.49]．$\boldsymbol{x}_*$ が (GMC^δ) の連続緩和問題の唯一の最適解であることは容易にわかる．さらに，δ が十分に小さい正の数であれば，$\boldsymbol{y}_*$ は (GMC^δ) の最適解である．ゆえに，定理 4.32(i) を (GMC^δ) と連続緩和問題に適用すると，$\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_\infty < n-1$ が得られる.

注意 4.2. 定理 4.32 より，閉真 M 凸関数 F が $\mathrm{argmin}_{\mathbb{R}} F \neq \emptyset$ を満たすためには，$\mathrm{argmin}\,\{F(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathbb{Z}^n\} \neq \emptyset$ の成り立つことが必要十分である．しかし，F が M 凸関

数ではない一般の場合，$\mathrm{argmin}_{\mathbb{R}} F \neq \emptyset$ と $\mathrm{argmin}\{F(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathbb{Z}^n\} \neq \emptyset$ の 2 つの条件は，次に示す 2 つの例のように，必要条件でも十分条件でもない．

1 つめの例として、$F : \mathbb{R}^2 \to \mathbb{R} \cup \{+\infty\}$ を

$$F(x_1, x_2) = \begin{cases} \dfrac{(2x_2 - 1)^2}{x_1 + 1} & (x_1 \geq 0 \text{ かつ } 0 \leq x_2 \leq 1 \text{ のとき}), \\ +\infty & (\text{その他}) \end{cases}$$

と定義された閉真凸関数とする．$\mathrm{argmin}_{\mathbb{R}} F = \{(x_1, 0.5) \mid x_1 \in \mathbb{R},\ x_1 \geq 0\} \neq \emptyset$ となる．一方，任意の整数ベクトル $\boldsymbol{y} = (y_1, y_2) \in \mathbb{Z}^2$ について $F(\boldsymbol{y}) > 0$ であり、さらに $\inf\{F(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathbb{Z}^2\} = \inf\{1/(y_1 + 1) \mid y_1 \in \mathbb{Z}_+\} = 0$ である．このことから $\mathrm{argmin}\{F(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathbb{Z}^2\} = \emptyset$ となることがわかる．

2 つめの例として，$G : \mathbb{R}^2 \to \mathbb{R} \cup \{+\infty\}$ を

$$G(x_1, x_2) = \begin{cases} 1/(x_1 + 1) & (x_1 \geq 0 \text{ かつ } x_2 = \sqrt{2} \cdot x_1 \text{のとき}), \\ +\infty & (\text{その他}) \end{cases}$$

と定義された閉真凸関数とする．$(0, 0)$ は関数 G が有限値をとる唯一の整数ベクトルであるので，$\mathrm{argmin}\{G(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathbb{Z}^2\} = \{(0, 0)\}$ となる．一方，任意の $\boldsymbol{x} \in \mathbb{R}^2$ について $G(\boldsymbol{x}) > 0$ であり，$\inf G = \inf\{1/(x_1 + 1) \mid x_1 \geq 0\} = 0$ となる．このことから $\mathrm{argmin}_{\mathbb{R}} G = \emptyset$ となることがわかる． ■

4.7.2 劣モジュラ制約つき資源配分問題の近接定理の証明

劣モジュラ制約つき資源配分問題 (SC) の近接定理である，定理 4.27 を証明する．(SC) の実行可能解集合は，

$$\rho_+(Y) = \min\{\rho(Z) \mid Z \supseteq Y\} \quad (Y \subseteq N)$$

として与えられる劣モジュラ関数 $\rho_+ : 2^N \to \mathbb{Z} \cup \{+\infty\}$ を用いた M 凸集合 $B(\rho_+) \cap \mathbb{Z}^n$ として表現できることが知られている（[17, 第 3.1 節 (b)] を参照）．ただし $B(\rho_+)$ は、劣モジュラ関数 ρ_+ が式 (2.67) のように定める基多面体である。ゆえに，問題 (SC) は，

$$\text{Minimize} \quad \sum_{i=1}^{n} F_i(x(i)) \qquad \text{subject to} \quad \boldsymbol{x} \in B(\rho_+) \cap \mathbb{Z}^n$$

と書き換えることができる．

このことから，(SC) よりも若干一般化された問題である，M 凸集合上の分離凸関数の最小化問題

$$(\mathsf{GSC}) \qquad \text{Minimize} \quad \sum_{i=1}^{n} F_i(x(i)) \qquad \text{subject to} \quad \boldsymbol{x} \in B(\rho) \cap \mathbb{Z}^n,$$

を考える．ただし，$F_i : \mathbb{R} \to \mathbb{R}$ $(i \in N)$ は 1 変数凸関数であり，$\rho : 2^N \to \mathbb{Z} \cup \{+\infty\}$ は整数値の劣モジュラ関数で、$\rho(\emptyset) = 0$ かつ $\rho(N) < +\infty$ を満たすとする．(GSC) の連続緩和は，

自然に

$$(\overline{\mathsf{GSC}}) \qquad \text{Minimize} \quad \sum_{i=1}^{n} F_i(x(i)) \qquad \text{subject to} \quad \boldsymbol{x} \in B(\rho)$$

と定義される．

問題 (GSC) の近接定理を示す．

定理 4.35.
(i) (GSC) の任意の最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ に対して，($\overline{\mathsf{GSC}}$) のある最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ が存在して，$\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1 < 2(n-1)$ を満たす．
(ii) ($\overline{\mathsf{GSC}}$) の任意の最適解 $\boldsymbol{x}_* \in \mathbb{R}^n$ に対して，(GSC) のある最適解 $\boldsymbol{y}_* \in \mathbb{Z}^n$ が存在して，$\|\boldsymbol{y}_* - \boldsymbol{x}_*\|_1 < 2(n-1)$ を満たす． ■

この定理より定理 4.27 は直ちに導かれる．

以下では，問題 (GSC) に関する補題を用いて，定理 4.35 を証明する．$\boldsymbol{x} \in \mathbb{R}^n$ に関して，

$$F_{\mathrm{sum}}(\boldsymbol{x}) = \sum_{i \in N} F_i(x(i))$$

と表すことにする．

補題 4.36. $\varphi : \mathbb{R} \to \mathbb{R}$ を 1 変数凸関数とする．$\alpha, \beta \in \mathbb{R}$ を $\alpha < \beta$，$\varepsilon \in \mathbb{R}$ を $0 \leq \varepsilon \leq \beta - \alpha$ とすると，$\varphi(\alpha) + \varphi(\beta) \geq \varphi(\alpha + \varepsilon) + \varphi(\beta - \varepsilon)$ となる． ■

補題 4.37 ([72])**.** $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $i \in \mathrm{supp}^+(\boldsymbol{x} - \boldsymbol{y})$, $j \in \mathrm{supp}^-(\boldsymbol{x} - \boldsymbol{y})$ に対して

$$F_{\mathrm{sum}}(\boldsymbol{x}) + F_{\mathrm{sum}}(\boldsymbol{y}) \geq F_{\mathrm{sum}}(\boldsymbol{x} - \alpha(\boldsymbol{\chi}_i - \boldsymbol{\chi}_j)) + F_{\mathrm{sum}}(\boldsymbol{y} + \alpha(\boldsymbol{\chi}_i - \boldsymbol{\chi}_j))$$

が $0 \leq \alpha \leq \min\{x(i) - y(i), y(j) - x(j)\}$ を満たす任意の $\alpha \in \mathbb{R}$ について成り立つ．

証明. この不等式は補題 4.36 から導かれる． ■

定理 4.35(i) の証明

$\boldsymbol{y}_* \in B(\rho) \cap \mathbb{Z}^n$ を (GSC) の最適解とする．$\boldsymbol{x}_* \in B(\rho)$ を ($\overline{\mathsf{GSC}}$) の最適解のうち，$\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1$ の値を最小化するものとする．以下では，集合 $\mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*) = \{i \in N \mid x_*(i) \neq y_*(i)\}$ の要素数についての数学的帰納法によって，

$$\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1 < 2(|\mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*)| - 1) \qquad (\boldsymbol{x}_* \neq \boldsymbol{y}_* \text{ のとき}) \tag{4.17}$$

であることを示す．これが言えれば $|\mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*)| \leq n$ から (i) の成り立つことがわかる．なお、$\boldsymbol{x}_*(N) = \boldsymbol{y}_*(N)$ であるので，$\boldsymbol{x}_* \neq \boldsymbol{y}_*$ のとき $|\mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*)| \geq 2$ となる．

$\boldsymbol{x}_* \neq \boldsymbol{y}_*$ であると仮定し，

$$L = \{i \in N \mid |x_*(i) - y_*(i)| \geq 1\}$$

とおく．次の 3 つの場合に分けて考える．

Case 1: $\mathrm{supp}^-(\boldsymbol{x}_* - \boldsymbol{y}_*) \cap L = \emptyset$
Case 2: $\mathrm{supp}^-(\boldsymbol{x}_* - \boldsymbol{y}_*) \cap L \neq \emptyset$ かつ $\mathrm{supp}^+(\boldsymbol{x}_* - \boldsymbol{y}_*) \setminus L = \emptyset$
Case 3: $\mathrm{supp}^-(\boldsymbol{x}_* - \boldsymbol{y}_*) \cap L \neq \emptyset$ かつ $\mathrm{supp}^+(\boldsymbol{x}_* - \boldsymbol{y}_*) \setminus L \neq \emptyset$

(Case 1)　仮定より，

$$|x_*(j) - y_*(j)| < 1 \qquad (\forall j \in \mathrm{supp}^-(\boldsymbol{x}_* - \boldsymbol{y}_*)). \tag{4.18}$$

が成り立つ．$x_*(N) = y_*(N) = \rho(N)$ であるので，

$$\sum_{i \in \mathrm{supp}^+(\boldsymbol{x}_* - \boldsymbol{y}_*)} |x_*(i) - y_*(i)| = \sum_{j \in \mathrm{supp}^-(\boldsymbol{x}_* - \boldsymbol{y}_*)} |x_*(j) - y_*(j)| \tag{4.19}$$

となる．また，

$$\begin{aligned} |\mathrm{supp}^-(\boldsymbol{x}_* - \boldsymbol{y}_*)| &= |\mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*)| - |\mathrm{supp}^+(\boldsymbol{x}_* - \boldsymbol{y}_*)| \\ &\leq |\mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*)| - 1 \end{aligned} \tag{4.20}$$

である．従って

$$\begin{aligned} \|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1 \;=\; 2 \sum_{j \in \mathrm{supp}^-(\boldsymbol{x}_* - \boldsymbol{y}_*)} |x_*(j) - y_*(j)| &< 2|\mathrm{supp}^-(\boldsymbol{x}_* - \boldsymbol{y}_*)| \\ &\leq 2(|\mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*)| - 1) \end{aligned}$$

が成り立つ．ただし，等式は式 (4.19) より，1 つめの不等式は式 (4.18) より，2 つめの不等式は式 (4.20) より導かれる．ゆえに，式 (4.17) が成り立つ．

(Case 2)　$j \in \mathrm{supp}^-(\boldsymbol{x}_* - \boldsymbol{y}_*) \cap L$ とする．$j \in \mathrm{supp}^+(\boldsymbol{y}_* - \boldsymbol{x}_*)$ であるので，定理 2.39 より，ある $i \in \mathrm{supp}^-(\boldsymbol{y}_* - \boldsymbol{x}_*)$ と十分に小さい $\varepsilon > 0$ が存在して，

$$\boldsymbol{y}_* - \varepsilon(\boldsymbol{\chi}_j - \boldsymbol{\chi}_i) \in B(\rho), \qquad \boldsymbol{x}_* + \varepsilon(\boldsymbol{\chi}_j - \boldsymbol{\chi}_i) \in B(\rho) \tag{4.21}$$

を満たす．さらに、$\boldsymbol{y}_* \in B(\rho) \cap \mathbb{Z}^n$ であり，ρ が整数値関数であるので，$\boldsymbol{y}_* - \varepsilon(\boldsymbol{\chi}_j - \boldsymbol{\chi}_i) \in B(\rho)$ から $\boldsymbol{y}_* - (\boldsymbol{\chi}_j - \boldsymbol{\chi}_i) \in B(\rho) \cap \mathbb{Z}^n$ であることがわかる．

Case 2 の仮定より $i \in \mathrm{supp}^-(\boldsymbol{y}_* - \boldsymbol{x}_*) = \mathrm{supp}^+(\boldsymbol{x}_* - \boldsymbol{y}_*) \subseteq L$ が成り立つので、$y_*(i) \leq x_*(i) - 1$ である．$\|(\boldsymbol{x}_* + \varepsilon(\boldsymbol{\chi}_j - \boldsymbol{\chi}_i)) - \boldsymbol{y}_*\|_1 < \|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1$ と $\boldsymbol{x}_*$ の選び方によって，

$$F_{\mathrm{sum}}(\boldsymbol{x}_* + \varepsilon(\boldsymbol{\chi}_j - \boldsymbol{\chi}_i)) > F_{\mathrm{sum}}(\boldsymbol{x}_*) \tag{4.22}$$

となる．F_{sum} の凸性と不等式 (4.22) から

$$F_{\mathrm{sum}}(\boldsymbol{x}_* + (\boldsymbol{\chi}_j - \boldsymbol{\chi}_i)) > F_{\mathrm{sum}}(\boldsymbol{x}_*) \tag{4.23}$$

となる．$y_*(j) \geq x_*(j) + 1$ かつ $y_*(i) \leq x_*(i) - 1$ であるので，補題 4.37 により，

$$F_{\mathrm{sum}}(\boldsymbol{y}_*) + F_{\mathrm{sum}}(\boldsymbol{x}_*) \geq F_{\mathrm{sum}}(\boldsymbol{y}_* - (\boldsymbol{\chi}_j - \boldsymbol{\chi}_i)) + F_{\mathrm{sum}}(\boldsymbol{x}_* + (\boldsymbol{\chi}_j - \boldsymbol{\chi}_i)) \tag{4.24}$$

となる．式 (4.23) と式 (4.24) から $F_{\mathrm{sum}}(\boldsymbol{y}_* - (\boldsymbol{\chi}_j - \boldsymbol{\chi}_i)) < F_{\mathrm{sum}}(\boldsymbol{y}_*)$ となるが，これは $\boldsymbol{y}_*$ が (GSC) の最適解であることに矛盾する．このことから Case 2 が起こりえないことがわかる．

(Case 3)　仮定より，$k \in \mathrm{supp}^+(\boldsymbol{x}_* - \boldsymbol{y}_*) \setminus L$ となる k が存在する．このような k を一つ固定する。

$$B = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \in B(\rho),\ x(i) = y_*(i)\ (\forall i \in N \setminus \mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*)),\ x(k) \leq y_*(k)\}$$

とする．集合 B は整基多面体であり，$B \cap \mathbb{Z}^n$ は M 凸集合である（[17, 第 3.1 節 (b)] などを参照）．このとき $\boldsymbol{y}_* \in B$ かつ $\boldsymbol{x}_* \notin B$ となる．問題 (GSC') とその連続緩和問題 ($\overline{\mathsf{GSC}'}$) を考える:

$$(\mathsf{GSC}') \qquad \text{Minimize} \quad \sum_{i=1}^{n} F_i(x(i)) \qquad \text{subject to} \quad \boldsymbol{x} \in B \cap \mathbb{Z}^n,$$

$$(\overline{\mathsf{GSC}'}) \qquad \text{Minimize} \quad \sum_{i=1}^{n} F_i(x(i)) \qquad \text{subject to} \quad \boldsymbol{x} \in B.$$

$\boldsymbol{y}_* \in B$ かつ $B \subseteq B(\rho)$ であるので，$\boldsymbol{y}_*$ は (GSC') の最適解である．このことと，定理 4.21 より，($\overline{\mathsf{GSC}'}$) の最適解が存在することがわかる．S ($\subseteq B$) を ($\overline{\mathsf{GSC}'}$) の最適解集合とし，S_* を ($\overline{\mathsf{GSC}'}$) の最適解 $\boldsymbol{x}$ のうち，$\|\boldsymbol{x} - \boldsymbol{y}_*\|_1$ の値を最小化するものの集合，つまり

$$S_* = \{\boldsymbol{x} \in S \mid \|\boldsymbol{x} - \boldsymbol{y}_*\|_1 \leq \|\boldsymbol{x}' - \boldsymbol{y}_*\|_1\ (\forall \boldsymbol{x}' \in S)\}$$

とする．後で示すように，ある $\tilde{\boldsymbol{x}} \in S_*$ が存在し，条件

$$\tilde{x}(k) = y_*(k), \tag{4.25}$$

$$\mathrm{supp}^+(\boldsymbol{x}_* - \tilde{\boldsymbol{x}}) = \{k\} \tag{4.26}$$

を満たす．$x_*(N) = \tilde{x}(N) = \rho(N)$ であるので，

$$\sum_{j \in \mathrm{supp}^-(\boldsymbol{x}_* - \tilde{\boldsymbol{x}})} |x_*(j) - \tilde{x}(j)| = \sum_{i \in \mathrm{supp}^+(\boldsymbol{x}_* - \tilde{\boldsymbol{x}})} |x_*(i) - \tilde{x}(i)| \ = \ |x_*(k) - \tilde{x}(k)|$$

が成り立つ．ただし，最後の等式は式 (4.26) による．この等式と式 (4.25) から，

$$\|\boldsymbol{x}_* - \tilde{\boldsymbol{x}}\|_1 = 2|x_*(k) - \tilde{x}(k)| = 2|x_*(k) - y_*(k)| < 2 \tag{4.27}$$

が成り立つことがわかる．ただし，最後の不等式は $k \notin L$ からわかる．$\tilde{\boldsymbol{x}} = \boldsymbol{y}_*$ のときは，不等式 (4.27) より

$$\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1 = \|\boldsymbol{x}_* - \tilde{\boldsymbol{x}}\|_1 < 2 \leq 2(|\mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*)| - 1),$$

が成り立つ，つまり式 (4.17) が成り立つことがわかる．ゆえに，$\tilde{\boldsymbol{x}} \neq \boldsymbol{y}_*$ と仮定する．$\tilde{\boldsymbol{x}} \in B$ より $\mathrm{supp}(\tilde{\boldsymbol{x}} - \boldsymbol{y}_*) \subseteq \mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*)$ となることがわかり，式 (4.25) と合わせると，$\mathrm{supp}(\tilde{\boldsymbol{x}} - \boldsymbol{y}_*) \subseteq \mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*) \setminus \{k\}$ となる．ゆえに，帰納法の仮定を問題 (GSC') とベクトル $\tilde{\boldsymbol{x}}, \boldsymbol{y}_*$ にあてはめて，不等式

$$\|\tilde{\boldsymbol{x}} - \boldsymbol{y}_*\|_1 < 2(|\mathrm{supp}(\tilde{\boldsymbol{x}} - \boldsymbol{y}_*)| - 1) \tag{4.28}$$

を得る．式 (4.27) と式 (4.28) を用いると，

$$\begin{aligned}\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1 &\leq \|\boldsymbol{x}_* - \tilde{\boldsymbol{x}}\|_1 + \|\tilde{\boldsymbol{x}} - \boldsymbol{y}_*\|_1 \\ &< 2 + 2(|\mathrm{supp}(\tilde{\boldsymbol{x}} - \boldsymbol{y}_*)| - 1) \ \leq \ 2(|\mathrm{supp}(\boldsymbol{x}_* - \boldsymbol{y}_*)| - 1)\end{aligned}$$

として式 (4.17) を導くことができた．

式 (4.25) と式 (4.26) を満たす $\tilde{\boldsymbol{x}} \in S_*$ が存在することを示せば，証明が完了する．

$\tilde{\boldsymbol{x}} \in S_*$ を，S_* のベクトルの中で，$\tilde{x}(k)$ の値を最大化するものとする．そのとき，$\tilde{x}(k) < y_*(k)$ であると仮定して，矛盾を導く．$x_*(k) > y_*(k)$ であるので，$k \in \mathrm{supp}^+(\boldsymbol{x}_* - \tilde{\boldsymbol{x}})$ である．従って，定理 2.39 と補題 4.37 より，ある $j \in \mathrm{supp}^-(\boldsymbol{x}_* - \tilde{\boldsymbol{x}})$ と十分に小さい正の数 ε が存在して，$\boldsymbol{x}_* - \varepsilon(\boldsymbol{\chi}_k - \boldsymbol{\chi}_j) \in B(\rho)$ かつ $\tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_k - \boldsymbol{\chi}_j) \in B(\rho)$ を満たし，さらに

$$F_{\mathrm{sum}}(\boldsymbol{x}_*) + F_{\mathrm{sum}}(\tilde{\boldsymbol{x}}) \geq F_{\mathrm{sum}}(\boldsymbol{x}_* - \varepsilon(\boldsymbol{\chi}_k - \boldsymbol{\chi}_j)) + F_{\mathrm{sum}}(\tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_k - \boldsymbol{\chi}_j)) \tag{4.29}$$

を満たす．$\boldsymbol{x}_*$ は ($\overline{\mathsf{GSC}}$) の最適解であるので，

$$F_{\mathrm{sum}}(\boldsymbol{x}_*) \leq F_{\mathrm{sum}}(\boldsymbol{x}_* - \varepsilon(\boldsymbol{\chi}_k - \boldsymbol{\chi}_j)) \tag{4.30}$$

が成り立つ．$\tilde{x}(k) < y_*(k)$ であり，ε は十分に小さいので，$\tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_k - \boldsymbol{\chi}_j) \in B$ が成り立つ．これより，$\tilde{\boldsymbol{x}}$ が ($\overline{\mathsf{GSC}'}$) の最適解であることに注意すると，

$$F_{\mathrm{sum}}(\tilde{\boldsymbol{x}}) \leq F_{\mathrm{sum}}(\tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_k - \boldsymbol{\chi}_j)) \tag{4.31}$$

であることがわかる．式 (4.29), (4.30), (4.31) より，(4.31) の不等式は等号で成り立ち，これより $\tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_k - \boldsymbol{\chi}_j)$ もまた ($\overline{\mathsf{GSC}'}$) の最適解になることがわかる．$\tilde{x}(k) < \tilde{x}(k) + \varepsilon \leq y_*(k)$ であるので，$\|(\tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_k - \boldsymbol{\chi}_j)) - \boldsymbol{y}_*\|_1 \leq \|\tilde{\boldsymbol{x}} - \boldsymbol{y}_*\|_1$ となる．$\tilde{\boldsymbol{x}} \in S_*$ より $\tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_k - \boldsymbol{\chi}_j) \in S_*$ である．しかしながら，これは $\tilde{\boldsymbol{x}}$ が S_* のベクトルの中で $\tilde{x}(k)$ の値を最大化することに矛盾する．ゆえに，$\tilde{\boldsymbol{x}}$ は式 (4.25) を満たす．

$\tilde{\boldsymbol{x}}$ を式 (4.25) を満たす S_* のベクトルの中で，

$$\sum_{i \in N} \max\{x_*(i) - \tilde{x}(i), 0\}$$

の値を最小化するものとする．このように選んだベクトル $\tilde{\boldsymbol{x}}$ は式 (4.26) を満たすことを示す．

条件 (4.26) の否定、すなわち $\mathrm{supp}^+(\boldsymbol{x}_* - \tilde{\boldsymbol{x}}) \setminus \{k\} \neq \emptyset$ を仮定して矛盾を導く．$h \in \mathrm{supp}^+(\boldsymbol{x}_* - \tilde{\boldsymbol{x}}) \setminus \{k\}$ とおく．このとき，定理 2.39 と補題 4.37 より，ある $j \in \mathrm{supp}^-(\boldsymbol{x}_* - \tilde{\boldsymbol{x}})$ と十分に小さい正の数 ε が存在して，$\boldsymbol{x}_* - \varepsilon(\boldsymbol{\chi}_h - \boldsymbol{\chi}_j) \in B(\rho)$ かつ $\tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_h - \boldsymbol{\chi}_j) \in B(\rho)$ を満たし，さらに

$$F_{\mathrm{sum}}(\boldsymbol{x}_*) + F_{\mathrm{sum}}(\tilde{\boldsymbol{x}}) \geq F_{\mathrm{sum}}(\boldsymbol{x}_* - \varepsilon(\boldsymbol{\chi}_h - \boldsymbol{\chi}_j)) + F_{\mathrm{sum}}(\tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_h - \boldsymbol{\chi}_j)) \tag{4.32}$$

を満たす．$\boldsymbol{x}_*$ は ($\overline{\mathsf{GSC}}$) の最適解であるので，

$$F_{\mathrm{sum}}(\boldsymbol{x}_*) \leq F_{\mathrm{sum}}(\boldsymbol{x}_* - \varepsilon(\boldsymbol{\chi}_h - \boldsymbol{\chi}_j)) \tag{4.33}$$

を満たす．$\tilde{\boldsymbol{x}}$ は ($\overline{\mathsf{GSC}'}$) の最適解であり，$\tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_h - \boldsymbol{\chi}_j) \in B$ であるので，

$$F_{\mathrm{sum}}(\tilde{\boldsymbol{x}}) \leq F_{\mathrm{sum}}(\tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_h - \boldsymbol{\chi}_j)) \tag{4.34}$$

を満たす．式 (4.32), (4.33), (4.34) から，(4.33) と (4.34) の不等式は等号で成り立つことになる．これより，$\boldsymbol{x}'_* = \boldsymbol{x}_* - \varepsilon(\boldsymbol{\chi}_h - \boldsymbol{\chi}_j)$ は ($\overline{\mathsf{GSC}}$) の最適解であり，$\tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{x}} + \varepsilon(\boldsymbol{\chi}_h - \boldsymbol{\chi}_j)$ は ($\overline{\mathsf{GSC}'}$) の最適解であることがわかる．$\boldsymbol{x}_*$ の選び方から $\|\boldsymbol{x}'_* - \boldsymbol{y}_*\|_1 \geq \|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1$ であるから，(i) $x_*(h) \leq y_*(h)$ あるいは (ii) $x_*(j) \geq y_*(j)$ が成り立つ。

以下では，$\tilde{\boldsymbol{x}}'$ が式 (4.25) および

$$\sum_{i\in N} \max\{x_*(i) - \tilde{x}'(i), 0\} < \sum_{i\in N} \max\{x_*(i) - \tilde{x}(i), 0\} \tag{4.35}$$

を満たす S_* のベクトルであり，$\tilde{\boldsymbol{x}}$ の選び方に矛盾していることを示す．まず、式 (4.25) の $\tilde{x}'(k) = \tilde{x}(k) = y_*(k)$ は明らかである．次に、

$$\begin{aligned} &|\tilde{x}'(i) - y_*(i)| = |\tilde{x}(i) - y_*(i)| \quad (\forall i \in N \setminus \{h, j\}), \\ &|\tilde{x}'(i) - y_*(i)| \leq |\tilde{x}(i) - y_*(i)| + \varepsilon \quad (\forall i \in \{h, j\}) \end{aligned}$$

が成り立つ．(i) $x_*(h) \leq y_*(h)$ のとき，$\tilde{x}(h) < \tilde{x}(h) + \varepsilon = \tilde{x}'(h) < x_*(h) \leq y_*(h)$ であるので，$|\tilde{x}'(h) - y_*(h)| = |\tilde{x}(h) - y_*(h)| - \varepsilon$ となる．(ii) $x_*(j) \geq y_*(j)$ のとき，$\tilde{x}(j) > \tilde{x}(j) - \varepsilon = \tilde{x}'(j) > x_*(j) \geq y_*(j)$ であるので，$|\tilde{x}'(j) - y_*(j)| = |\tilde{x}(j) - y_*(j)| - \varepsilon$ となる．ゆえに，$\|\tilde{\boldsymbol{x}}' - \boldsymbol{y}_*\|_1 \leq \|\tilde{\boldsymbol{x}} - \boldsymbol{y}_*\|_1$，つまり $\tilde{\boldsymbol{x}}' \in S_*$ が成り立つ．最後に，不等式 (4.35) が

$$\begin{aligned} &\sum_{i\in N} \max\{x_*(i) - \tilde{x}'(i), 0\} - \sum_{i\in N} \max\{x_*(i) - \tilde{x}(i), 0\} \\ &= \big[\max\{x_*(h) - \tilde{x}'(h), 0\} - \max\{x_*(h) - \tilde{x}(h), 0\}\big] \\ &\qquad + \big[\max\{x_*(j) - \tilde{x}'(j), 0\} - \max\{x_*(j) - \tilde{x}(j), 0\}\big] \\ &= \big[x_*(h) - \tilde{x}(h) - \varepsilon\big] - \big[x_*(h) - \tilde{x}(h)\big] \;=\; -\varepsilon \;<\; 0 \end{aligned}$$

として得られる．ただし，2 番目の等式は $h \in \mathrm{supp}^+(\boldsymbol{x}_* - \tilde{\boldsymbol{x}})$, $j \in \mathrm{supp}^-(\boldsymbol{x}_* - \tilde{\boldsymbol{x}})$ と ε が十分に小さい正の数であることから導かれた．

ゆえに，$\tilde{\boldsymbol{x}}$ は式 (4.25) と式 (4.26) を満たす．よって定理 4.35 (i) が証明された． ■

定理 4.35 (ii) の証明

定理 4.35 (i) と (ii) は、近接の方向がいわば逆方向の関係にある。そこで (ii) の証明には，先に証明した (i) に摂動を適用する．$\boldsymbol{x}_* \in B(\rho)$ を ($\overline{\mathsf{GSC}}$) の最適解とする．$\boldsymbol{y}_* \in B(\rho) \cap \mathbb{Z}^n$ を (GSC) の最適解のうち $\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1$ の値を最小にするものとする．正の数 δ を用いて，新たな問題

$$(\mathsf{GSC}^\delta) \qquad \text{Minimize} \quad \sum_{i=1}^{n} \big(F_i(x(i)) + \delta|x(i) - x_*(i)|\big) \qquad \text{subject to} \quad \boldsymbol{x} \in B(\rho) \cap \mathbb{Z}^n$$

を定義する．

この問題も M 凸集合上の分離凸関数を最小化する問題である．$\boldsymbol{x}_*$ が (GSC^δ) の連続緩和の唯一の最適解であることは容易にわかる．更に，δ が十分に小さい正の数のとき，$\boldsymbol{y}_*$ は (GSC^δ) の最適解である．よって，定理 4.35 (i) を問題 (GSC^δ) とその連続緩和問題に適用して，$\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1 < 2(n-1)$ が得られる． ■

4.7.3 層族制約つき資源配分問題の近接定理の証明

層族制約つき資源配分問題 (Laminar) の近接定理である，定理 4.28 を証明する．最初に，有用な補題を示す．$\mathcal{F} \subseteq 2^N$ は条件 (4.6) を満たす層族とする．第 4.1 節で述べた $X \in \mathcal{F}$ の親と子の定義を思い出すと，相異なる 2 つの要素 $i, j \in N$ について，i, j の両方を含む $\mathcal{F}$ の最小の集合は唯一に定まる．これを i と j の「直近の共通祖先」と呼ぶことにする．任意の相異なる $i, j \in N$ について，「i から j への（$\mathcal{F}$ 中の）（有向）パス」は，$\mathcal{F}$ に含まれる集合列 $S_0, S_1, \ldots, S_t$ $(t \geq 2)$ として定義され，以下の条件

(i) $S_0 = \{i\}$, $S_t = \{j\}$
(ii) ある $k (1 \leq k \leq t-1)$ が存在して，S_k が i と j の「直近の共通祖先」である
(iii) $h = 0, 1, \ldots, k-1$ について，集合 S_{h+1} が S_h の親である
(iv) $h = k+1, k+2, \ldots, t$ について，集合 S_{h-1} が S_h の親である

を満たす．i から j へのパスは唯一に定まることに注意する．

$\boldsymbol{z} \in \mathbb{R}^n$ とし，$i, j \in N$ を相異なる要素とする．$i, j, \mathcal{F}, \boldsymbol{z}$ に関して，「容量」$\mathrm{cap}(i, j, \mathcal{F}, \boldsymbol{z})$ を

$$\mathrm{cap}(i, j, \mathcal{F}, \boldsymbol{z}) = \min \left[\min_{0 \leq h \leq k-1} \max(z(S_h), 0), \min_{k+1 \leq h \leq t} \max(-z(S_h), 0) \right],$$

と定める．ただし，$S_0, S_1, \ldots, S_t$ は i から j へのパスであり，S_k は i と j の直近の共通祖先である．$\mathrm{cap}(i, j, \mathcal{F}, \boldsymbol{z}) > 0$ が成り立つためには，

$$z(S_h) > 0 \ (h = 0, 1, \ldots, k-1), \qquad z(S_h) < 0 \ (h = k+1, k+2, \ldots, t)$$

が成り立つことが必要十分であることを注意しておく．

ここでは鍵となる補題を述べる．

補題 4.38. $\mathcal{F} \subseteq 2^N$ を条件 (4.6) を満たす層族とし、$\boldsymbol{z} \in \mathbb{R}^n$ は $z(N) = 0$ を満たすベクトルとする．任意の相異なる要素 $i, j \in N$ に対して $\mathrm{cap}(i, j, \mathcal{F}, \boldsymbol{z}) < 1$ であるとする．このとき，$\|\boldsymbol{z}\|_1 \leq 2(n-1)$ が成り立ち，$n \geq 2$ であれば狭義の不等式 $\|\boldsymbol{z}\|_1 < 2(n-1)$ が成り立つ．

証明. この補題を，集合 N の元の個数 n についての数学的帰納法で示す．$n = 1$ のとき，$z(1) = z(N) = 0$ より $\|z\|_1 = 0 = 2(n-1)$ である．ゆえに，$n \geq 2$ を考えればよい．

Claim 1: 任意の $i \in \mathrm{supp}^+(\boldsymbol{z})$ について, ある $j \in \mathrm{supp}^-(\boldsymbol{z})$ が存在して, $\mathrm{cap}(i, j, \mathcal{F}, \boldsymbol{z}) > 0$ を満たす.

[Claim 1 の証明] $X \subseteq N$ を $z(X) \leq 0$ かつ $i \in X$ を満たす $\mathcal{F}$ の（一意的に定まる）最小の集合とする．$z(N) = 0$ かつ $N \in \mathcal{F}$ であるので，このような X は常に存在する．$z(i) > 0$ より $|X| \geq 2$ である．$S_0, S_1, \ldots, S_k$ $(k \geq 1)$ を $S_0 = \{i\}$, $S_k = X$ かつ S_h が S_{h-1} $(h = 1, 2, \ldots, k)$ の親になっている集合列とする．このとき，X の選び方から $z(S_h) > 0$ $(h = 0, 1, \ldots, k-1)$ となる．$z(S_k) = z(X) \leq 0$ かつ $z(S_{k-1}) > 0$ であるので，X には別の

子 S_{k+1} が存在して $z(S_{k+1}) < 0$ を満たす．$|S_{k+1}| \geq 2$ のとき，S_{k+1} の子 $S_{k+2} \in \mathcal{F}$ が存在して，$z(S_{k+2}) < 0$ を満たす．これを繰り返すと集合列 $S_{k+1}, S_{k+2}, \ldots, S_{t-1}, S_t$ が得られ，$|S_t| = 1$, $z(S_h) < 0$ $(h = k+1, k+2, \ldots, t)$ かつ S_h が S_{h-1} $(h = k+1, k+2, \ldots, t)$ の子になっている．$j \in N$ を S_t のただひとつの要素とする．このとき，上の結果から $j \in \mathrm{supp}^-(\boldsymbol{z})$ かつ $\mathrm{cap}(i, j, \mathcal{F}, \boldsymbol{z}) > 0$ であることがわかる．　[Claim 1 の証明終わり]

$\boldsymbol{z} \neq \boldsymbol{0}$ と仮定してよい．$i_* \in \mathrm{supp}^+(\boldsymbol{z})$ とおく．Claim 1 より，ある $j_* \in \mathrm{supp}^-(\boldsymbol{z})$ が存在して，$\mathrm{cap}(i_*, j_*, \mathcal{F}, \boldsymbol{z}) > 0$ を満たす．i_* から j_* までのパスを $\tilde{S}_0, \tilde{S}_1, \ldots, \tilde{S}_t$ とし，$\tilde{S}_k$ $(1 \leq k \leq t-1)$ を i_* と j_* の直近の共通祖先とする．

$\tilde{\boldsymbol{z}} = \boldsymbol{z} - \delta(\boldsymbol{\chi}_{i_*} - \boldsymbol{\chi}_{j_*})$ と定義する．ただし $\delta = \mathrm{cap}(i_*, j_*, \mathcal{F}, \boldsymbol{z})$ (> 0) とする．このとき，$\|\tilde{\boldsymbol{z}}\|_1 = \|\boldsymbol{z}\|_1 - 2\delta$ となる．

Claim 2:　任意の相異なる $i, j \in N$ に対して，$\mathrm{cap}(i, j, \mathcal{F}, \tilde{\boldsymbol{z}}) \leq \mathrm{cap}(i, j, \mathcal{F}, \boldsymbol{z})$ (< 1) が成り立つ．

[Claim 2 の証明]　任意の $X \in \mathcal{F}$ に対して $z(X) \geq 0$ のときに $0 \leq \tilde{z}(X) \leq z(X)$ が成り立ち，$z(X) < 0$ のときに $0 \geq \tilde{z}(X) \geq z(X)$ が成り立つことを示せば十分である．パスの定義より，$\tilde{S}_0, \ldots, \tilde{S}_{k-1}$ は $\mathcal{F}$ の中のパスになっていて，i_* を含み j_* を含まない．従って，$X \in \{\tilde{S}_0, \ldots, \tilde{S}_{k-1}\}$ に対して、$z(X) \geq \delta$ より $\tilde{z}(X) = z(X) - \delta \geq 0$ となる．同様に，集合 $X \in \{\tilde{S}_{k+1}, \ldots, \tilde{S}_t\}$ はそれぞれ j_* を含み i_* を含まないので，$z(X) \leq -\delta$ より $\tilde{z}(X) = z(X) + \delta \leq 0$ となる．最後に，$\{\tilde{S}_0, \ldots, \tilde{S}_{k-1}\} \cup \{\tilde{S}_{k+1}, \ldots, \tilde{S}_t\}$ に含まれない $X \in \mathcal{F}$ は，i_* と j_* を両方含むか，両方含まないかのどちらかであるので，$\tilde{z}(X) = z(X)$ が成り立つ．　[Claim 2 の証明終わり]

$\delta = \mathrm{cap}(i_*, j_*, \mathcal{F}, \boldsymbol{z})$ の定義により，ある $v \in \{0, \ldots, k-1\} \cup \{k+1, \ldots, t\}$ が存在して，$|\boldsymbol{z}(\tilde{S}_v)| = \delta$ を満たす．これより，$\tilde{\boldsymbol{z}}(\tilde{S}_v) = 0$ となる．$N' = \tilde{S}_v$ とし，$N'' = N \setminus \tilde{S}_v$ とする．$|\tilde{S}_v \cap \{i_*, j_*\}| = 1$ であるので，$1 \leq |N'| < n$ かつ $1 \leq |N''| < n$ が成り立つ．新しい集合族 $\mathcal{F}' \subseteq 2^{N'}$ と $\mathcal{F}'' \subseteq 2^{N''}$ を

$$\mathcal{F}' = \{Y \mid Y \in \mathcal{F},\ Y \subseteq \tilde{S}_v\},$$
$$\mathcal{F}'' = \{Y \mid Y \in \mathcal{F},\ Y \cap \tilde{S}_v = \emptyset\} \cup \{Y \setminus \tilde{S}_v \mid Y \in \mathcal{F},\ Y \supset \tilde{S}_v\}$$

と定義する．このとき $\mathcal{F}'$ と $\mathcal{F}''$ は条件 (4.6) を満たすので層族である．ベクトル $\boldsymbol{z}' \in \mathbb{R}^{N'}$ を $z'(i) = \tilde{z}(i)$ $(i \in N')$ と定義し，ベクトル $z'' \in \mathbb{R}^{N''}$ を $z''(i) = \tilde{z}(i)$ $(i \in N'')$ と定義する．次の Claim 3 は，任意の相異なる $i, j \in N'$（あるいは $i, j \in N''$）に対して，$\mathcal{F}'$（あるいは $\mathcal{F}''$）の i から j へのパスが $\mathcal{F}$ の i から j へのパスに一致することから容易に導かれる．

Claim 3:　任意の相異なる $i, j \in N'$ に対して，$\mathrm{cap}(i, j, \mathcal{F}', \boldsymbol{z}') = \mathrm{cap}(i, j, \mathcal{F}, \tilde{\boldsymbol{z}})$ が成り立つ．また、任意の相異なる $i, j \in N''$ に対して，$\mathrm{cap}(i, j, \mathcal{F}'', \boldsymbol{z}'') = \mathrm{cap}(i, j, \mathcal{F}, \tilde{\boldsymbol{z}})$ が成り立つ．

Claim 2 と Claim 3 より，任意の相異なる $i, j \in N'$ に対して $\mathrm{cap}(i, j, \mathcal{F}', \boldsymbol{z}') < 1$ が成り立ち，任意の相異なる $i, j \in N''$ に対して $\mathrm{cap}(i, j, \mathcal{F}'', \boldsymbol{z}'') < 1$ が成り立つ．ゆえに，帰納法

の仮定を適用して，$\|\boldsymbol{z}'\|_1 \le 2(|N'|-1)$ と $\|\boldsymbol{z}''\|_1 \le 2(|N''|-1)$ が得られる．$\delta < 1$ であるので，

$$\begin{aligned}\|\boldsymbol{z}\|_1 &= 2\delta + \|\tilde{\boldsymbol{z}}\|_1 = 2\delta + \|\boldsymbol{z}'\|_1 + \|\boldsymbol{z}''\|_1 \\ &< 2 + 2(|N'|-1) + 2(|N''|-1) = 2(|N'|+|N''|-1) = 2(n-1)\end{aligned}$$

が成り立つ． ■

定理 4.28 (i) の証明

問題 (Laminar) においては，$n \ge 2$ が前提であった．$\boldsymbol{y}_* \in \mathbb{Z}^n$ を (Laminar) の最適解とし，$\boldsymbol{x}_* \in \mathbb{R}^n$ を ($\overline{\text{Laminar}}$) の最適解の中で，$\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1$ の値を最小化するものとする．$\boldsymbol{x}_* = \boldsymbol{y}_*$ のときは，直ちに (i) が成り立つことがわかるので，$\boldsymbol{x}_* \ne \boldsymbol{y}_*$ と仮定してよい．$\boldsymbol{z} = \boldsymbol{x}_* - \boldsymbol{y}_*$ とする．このとき，$z(N) = x_*(N) - y_*(N) = K - K = 0$ が成り立つ．任意の相異なる要素 $i, j \in N$ に対して，$\mathrm{cap}(i, j, \mathcal{F}, \boldsymbol{z}) < 1$ であることを示そう．これが示せると，補題 4.38 より不等式 $\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1 < 2(n-1)$ の成り立つことがわかる．

$i, j \in N$ を固定する。$S_0, S_1, S_2, \ldots, S_t$ を i から j までのパスとし，S_k $(1 \le k \le t-1)$ を i と j の直近の共通祖先とする．ここで $\mathrm{cap}(i, j, \mathcal{F}, \boldsymbol{z}) \ge 1$ と仮定すると、$h = 0, 1, \ldots, k-1$ に対して $z(S_h) \ge 1$ となり，$h = k+1, k+2, \ldots, t$ に対して $z(S_h) \le -1$ となる．以下では、ここから矛盾を導く。

ベクトル $\boldsymbol{x}'$ と $\boldsymbol{y}'$ を，それぞれ $\boldsymbol{x}' = \boldsymbol{x}_* - \boldsymbol{\chi}_i + \boldsymbol{\chi}_j$，$\boldsymbol{y}' = \boldsymbol{y}_* + \boldsymbol{\chi}_i - \boldsymbol{\chi}_j$ と定義する．このとき，

$$\begin{aligned}
&u_X \ge x_*(X) > x'(X) = x_*(X) - 1 \ge y_*(X) \ge \ell_X \quad (X \in \{S_0, \ldots, S_{k-1}\} \text{ のとき}),\\
&\ell_X \le y_*(X) < y'(X) = y_*(X) + 1 \le x_*(X) \le u_X \quad (X \in \{S_0, \ldots, S_{k-1}\} \text{ のとき}),\\
&\ell_X \le x_*(X) < x'(X) = x_*(X) + 1 \le y_*(X) \le u_X \quad (X \in \{S_{k+1}, \ldots, S_t\} \text{ のとき}),\\
&u_X \ge y_*(X) > y'(X) = y_*(X) - 1 \ge x_*(X) \ge \ell_X \quad (X \in \{S_{k+1}, \ldots, S_t\} \text{ のとき}),\\
&x'(X) = x_*(X),\ y'(X) = y_*(X) \quad (X \in \mathcal{F},\ X \notin \{S_0, \ldots, S_{k-1}\} \cup \{S_{k+1}, \ldots, S_t\} \text{ のとき})
\end{aligned}$$

となる．ゆえに，$\boldsymbol{y}'$ と $\boldsymbol{x}'$ は，それぞれ (Laminar) と ($\overline{\text{Laminar}}$) の実行可能解である．上記の不等式と補題 4.36 より，

$$F_X(x_*(X)) + F_X(y_*(X)) \ge F_X(x'(X)) + F_X(y'(X)) \qquad (\text{任意の } X \in \mathcal{F} \text{ に対して}),$$

が成り立つことがわかり，これより

$$\sum_{X \in \mathcal{F}} F_X(x_*(X)) + \sum_{X \in \mathcal{F}} F_X(y_*(X)) \ge \sum_{X \in \mathcal{F}} F_X(x'(X)) + \sum_{X \in \mathcal{F}} F_X(y'(X)) \tag{4.36}$$

となる．$\|\boldsymbol{x}' - \boldsymbol{y}_*\| < \|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1$ であるので，$\boldsymbol{x}_*$ の定義により、

$$\sum_{X \in \mathcal{F}} F_X(x_*(X)) < \sum_{X \in \mathcal{F}} F_X(x'(X))$$

となる．更に式 (4.36) を用いると

$$\sum_{X\in\mathcal{F}} F_X(y_*(X)) > \sum_{X\in\mathcal{F}} F_X(y'(X))$$

となり，$\boldsymbol{y}_*$ が (Laminar) の最適解であることに矛盾する． ■

定理 4.28 (ii) の証明

定理 4.32 (ii) や定理 4.35 (ii) の証明と同様に摂動によって証明することができるが、ここでは摂動によらない直接的な証明を記す。

$\boldsymbol{x}_* \in \mathbb{R}^n$ を ($\overline{\text{Laminar}}$) の最適解とし，$\boldsymbol{y}_* \in \mathbb{Z}^n$ を (Laminar) の最適解の中で $\|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1$ の値を最小にするものとする．$\boldsymbol{x}_* \neq \boldsymbol{y}_*$ であると仮定し，$\boldsymbol{z} = \boldsymbol{x}_* - \boldsymbol{y}_*$ とおく．このとき，定理 4.28 (i) の証明と同様に，任意の相異なる要素 $i, j \in N$ に対して $\mathrm{cap}(i, j, \mathcal{F}, \boldsymbol{z}) < 1$ となることがわかる．これには $\|\boldsymbol{x}_* - \boldsymbol{y}'\|_1 < \|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1$ を満たす任意の $\boldsymbol{y}' \in \mathbb{Z}^n$ に対して，$\sum_{X\in\mathcal{F}} F_X(y_*(X)) < \sum_{X\in\mathcal{F}} F_X(y'(X))$ が成り立つことを用いた．ゆえに，補題 4.38 から不等式 $\|\boldsymbol{z}\|_1 = \|\boldsymbol{x}_* - \boldsymbol{y}_*\|_1 < 2(n-1)$ が導かれる． ■

第 5 章

離散凸最適化ソルバの開発

5.1 概要

連続最適化の分野では、線形最適化問題・非線形最適化問題を問わず、定式化された問題を数値的に解くソルバソフトウェアが数多く開発されている。線形計画問題に対しては、問題を構成する変数の数が数万以上であっても、現実的な計算時間で数値解を求めることができるソルバが多数ある。非線形最適化問題においても、凸計画問題や半正定値計画問題など、いくつかの問題クラスに対しては線形計画問題に準じる状態にあり、商用・非商用を問わず、高性能なソルバが活発に開発され、性能が競われている。

一方、離散最適化の分野に目を向けると、整数計画問題や混合整数計画問題に対するソルバソフトウェアが開発されている。対象となる問題は、通常の（連続変数の）線形計画問題に整数変数の制約を加えて表現されるが、この整数の制約が求解を困難にすることが知られている。そのため、この分野のソルバは近年性能が著しく向上しているにもかかわらず、連続変数の問題に対するような性能を実現することができず、問題規模が制限される、あるいは最適性の保証のない近似解法が用意されるなどの影響がある。また非線形計画問題に整数変数の制約を加えた問題は、さらに困難なことが知られている。このような問題は現実問題に数多く見られ、実用面からも求解が期待されているが、対応する汎用のソルバソフトウェアは少なく、実用的な性能を持つものはほとんどないと言ってよい状況にある。特定の問題クラスに対象を絞った専用のアルゴリズムが個別に実装されているのが現状である。

このように整数変数の非線形計画問題は困難ではあるが、中には比較的容易に解ける問題クラスも存在する。このような問題の解きやすさ／難しさを整理する手掛かりを求めるために，様々な離散凸関数の概念が考案され，理論研究が展開されてきた．その中でも，マトロイド・劣モジュラ関数の研究の流れを汲んだ離散凸解析の枠組みが 1990 年代後半以降注目され，今や多くの成果が報告されている [17, 61, 62, 65, 92]．離散凸解析理論では，L 凸関数と M 凸関数という互いに共役な二種類の離散凸性が定義され，それらの最小化問題に対して効率的なアルゴリズムが開発されている．

本論文では、離散凸解析理論の普及のために、離散凸最適化ソルバ ODICON (Optimization algorithms for DIscrete CONvex functions) を開発し，公開した [94]．対象とする問題は、

通常の（連続変数の）凸計画問題の特殊例に整数変数の制約がついたものとみなせる。離散凸解析理論の成果を応用分野の研究者・実務家が利用できるように，関連するアルゴリズムとそれを紹介するデモンストレーションソフトウェア，およびアプリケーションソフトウェアを開発して Web 上に公開している．アルゴリズムの詳細を理解しなくても離散凸解析関連の研究や応用事例研究が行えるような環境を整備することを目指している．現段階で公開しているソフトウェアの例としては，離散凸関数の最小化ソルバ，インタラクティブに離散 2 次凸関数を最小化できるオンラインソルバ，在庫管理アプリケーション，コールセンターのシフトスケジューリングアプリケーションなどがある．本章では，これらについて報告する．

5.2　離散凸関数最小化アルゴリズム

$\mathrm{L}^\natural$ 凸関数や $\mathrm{M}^\natural$ 凸関数の最小化のためのアルゴリズムには，さまざまなものが提案されている．ここでは，特に ODICON に関係の深いものについて簡単に述べる．なお，$\mathrm{L}^\natural$ 凸関数と $\mathrm{M}^\natural$ 凸関数について述べるが，L 凸関数と M 凸関数についても，それぞれ $\mathrm{L}^\natural$ 凸関数と $\mathrm{M}^\natural$ 凸関数と等価に変換ができるので，アルゴリズムの存在や計算量についての状況は同じである．

5.2.1　$\mathrm{L}^\natural$ 凸関数の最急降下法

離散凸関数の最小解を局所的に特徴づける定理から，最小化のための最急降下法が自然に導かれる．$\mathrm{L}^\natural$ 凸関数に対しては，第 3.2.1 節で述べたように、定理 2.5 から次の形となる [60, 61, 62, 39].

$\mathrm{L}^\natural$ 凸関数に対する最急降下法

手順 0: $\boldsymbol{x}$ を $\mathrm{dom}_{\mathbb{Z}} f$ に含まれる任意のベクトルとする．

手順 1: $\varepsilon \in \{1, -1\}$ と $X \subseteq \{1, 2, \ldots, n\}$ を次のように決定する．

　手順 1-1: X^+ を $\rho^+_{\boldsymbol{x}}(X) = f(\boldsymbol{x} + \boldsymbol{\chi}_X) - f(\boldsymbol{x})$ の任意の最小解とする．

　手順 1-2: X^- を $\rho^-_{\boldsymbol{x}}(X) = f(\boldsymbol{x} - \boldsymbol{\chi}_X) - f(\boldsymbol{x})$ の任意の最小解とする．

　手順 1-3:(ε, X) を以下のように定める：　$(\varepsilon, X) = \begin{cases} (1, X^+) & \text{if } X^+ \leq X^- \\ (-1, X^-) & \text{if } X^+ > X^- \end{cases}$

手順 2: もし $f(\boldsymbol{x}) \leq f(\boldsymbol{x} + \varepsilon\boldsymbol{\chi}_X)$ ならば終了．このとき $\boldsymbol{x}$ は f の最小解.

手順 3: $\boldsymbol{x} := \boldsymbol{x} + \varepsilon\boldsymbol{\chi}_X$ とおく．手順 1 へ戻る．　□

手順 1 の局所探索には，$\rho^+_{\boldsymbol{x}}(X)$, $\rho^-_{\boldsymbol{x}}(X)$ に対して劣モジュラ集合関数最小化アルゴリズム（[17, 34, 62] など参照）を適用すればよい．本章では，Iwata–Fleischer–Fujishige [35] の組合せ論的な多項式時間アルゴリズム (IFF と略記）と最小ノルム基を利用する Fujishige–Wolfe アルゴリズム (FW と略記）を用いている（第 5.3 節参照）.

5.2.2　$\mathrm{M}^\natural$ 凸関数の最急降下法

$\mathrm{M}^\natural$ 凸関数 f の最小化に対しても，第 4.2.1 節で述べたように、同様の最急降下法が定理

2.25 から導かれる．近傍探索の範囲は $\mathrm{L}^\natural$ 凸関数とは異なり、定理 2.25 に従って修正する必要がある [61, 62, 65, 88]．また、$\mathrm{M}^\natural$ 凸関数に対する最急降下法にはいくつかの変種が提案されており，

(i) $f(\boldsymbol{x}-\boldsymbol{\chi}_i+\boldsymbol{\chi}_j)$ を最小化する (i,j) を見出す基本形（[61, 227 頁] の「降下法」，[62, 281〜283 頁] の ‘Steepest descent algorithm’），
(ii) 任意に選んだ i に対して $f(\boldsymbol{x}-\boldsymbol{\chi}_i+\boldsymbol{\chi}_j)$ を最小化する j を見出す修正形（[52] の ‘Modified_Steepest_Descent’），
(iii) 修正形 (ii) に領域縮小を組み込んだもの（[88, 306 頁] の ‘Greedy’）

のようなバリエーションがある．

5.2.3 $\mathrm{L}^\natural$ 凸/$\mathrm{M}^\natural$ 凸関数のスケーリング法

最急降下法は，そのままでは多項式時間アルゴリズムではないが，これにスケーリング技法を適用した効率的な多項式時間アルゴリズムが提案されている [61, 62, 65]．第 3.3.1 で述べたように、スケーリングとは，与えられた関数 $f:\mathbb{Z}^n\to\mathbb{R}\cup\{+\infty\}$ に対して，正整数 α を用いて

$$f_\alpha(\boldsymbol{x})=f(\alpha\boldsymbol{x}) \qquad (\boldsymbol{x}\in\mathbb{Z}^n) \tag{5.1}$$

により定義される関数 $f_\alpha:\mathbb{Z}^n\to\mathbb{R}\cup\{+\infty\}$ を考えることをいう．関数 f_α は整数格子点上の点を α 刻みで取って f を近似したものであり，f_α の最小解は f の最小解の近くに位置する可能性が高い．一方，関数 f_α は f よりも容易に最小化できるので，まず f_α の最小解 $\boldsymbol{x}_\alpha$ を求め，それを初期解として f の最小解を計算すると効率が良い．さらに，f_α の最小解を求める際には同じ考え方を再帰的に適用できる．このようなアルゴリズムをスケーリング法とよぶ．

第 3.3.3 節の近接定理によって、離散 $\mathrm{L}^\natural$ 凸関数 f_α の最小解 $\boldsymbol{x}_\alpha$ と元の関数 f の最小解との距離の近さが評価され、第 3.3.4 節のスケーリング法の計算量も理論的に保証される．

離散 $\mathrm{M}^\natural$ 凸関数に対しても、離散 $\mathrm{L}^\natural$ 凸関数と同様に、スケーリングアルゴリズムを考えることができる。第 4.3.2 節の近接定理によって、離散 $\mathrm{M}^\natural$ 凸関数 f_α の最小解 $\boldsymbol{x}_\alpha$ と元の関数 f の最小解との距離の近さが評価される。しかし、計算量が保証されるアルゴリズムを設計するには、第 4.3.3 節で述べたように、工夫が必要である．

5.2.4 $\mathrm{L}^\natural$ 凸/$\mathrm{M}^\natural$ 凸関数の連続緩和法

離散凸関数 $f:\mathbb{Z}^n\to\mathbb{R}\cup\{+\infty\}$ に対して，

$$f(\boldsymbol{x})=F(\boldsymbol{x}) \qquad (\forall\boldsymbol{x}\in\mathbb{Z}^n) \tag{5.2}$$

を満たす連続変数の凸関数 $F:\mathbb{R}^n\to\mathbb{R}\cup\{+\infty\}$ が与えられているとき，第 3.4 節で述べたように、連続最適化の手法により F の最小解 $\boldsymbol{y}\in\mathbb{R}^n$ を求め，$\boldsymbol{y}$ の各成分を適当に整数に丸めたベクトル $\boldsymbol{x}\in\mathbb{Z}^n$ を初期点として最急降下法などを f に適用する手法を，連続緩和法とよ

ぶ [56, 53]．連続緩和解 $\boldsymbol{y}$ と元の関数 f の最小解との距離を評価する定理（近接定理）により，連続緩和法の計算量が理論的に保証される．

$L^\natural$ 凸関数については定理 3.6、$M^\natural$ 凸関数については定理 4.16 により，離散凸関数 f に対して式 (5.2) を満たす連続変数凸関数 F の存在は保証されている．さらに，応用の問題においては，滑らかな連続変数関数 F が与えられて，関数 f が式 (5.2) のように定義される場合も多い．なお，与えられた f から F を計算することについては，$L^\natural$ 凸関数の場合には計算方法が知られており，$M^\natural$ 凸関数の場合には一般に容易ではない．

5.3　離散凸最適化ソルバ：ODICON

本節では、開発した離散凸最適化ソルバ ODICON (Optimization algorithms for DIscrete CONvex functions) [94, 95, 96] の機能を記述する．実装した離散凸関数最小化アルゴリズムは、L / $L^\natural$ / M / $M^\natural$ 凸関数のそれぞれに対する最急降下法（第 5.2.1 節と第 5.2.2 節），スケーリング法（第 5.2.3 節），連続緩和法（第 5.2.4 節）である．L 凸関数と $L^\natural$ 凸関数は，理論的には等価であって，互いに変換可能ではあるが，利用者の利便性を考えて，どちらの最小化ルーチンも準備してある．M 凸と $M^\natural$ 凸についても同じである．

ODICON は C 言語によるオープンソースソフトウェアであり，単体での活用はもちろん，別のプログラムに組み込まれることも想定している．最小化したい離散関数をもつ利用者は，その関数を C 言語上の関数として記述して，このソルバのルーチンを呼び出せばよい．

一連のルーチンは，アルゴリズムの素直な実装を目指し，入出力インタフェースも自然なものになるように留意した．特に，C 言語において配列の要素数を，コンパイル時ではなく，実行時の状況に応じて決定しようとすると，その扱いに標準手法が確立されておらず，どのように実現するかは自明ではない．この点についても十分に検討して，標準となりうる手法を選ぶようにした．このような工夫により，他人のソフトウェアを組み込む時にありがちな，どのように結合すればよいのかわからない，という問題を最小限にとどめている．

5.3.1　実装ルーチンの構成

ODICON には，表5.1 のような最小化ルーチンを実装した．以下はそのうちの 1 つである [*1]．

```
double mgconv_minimize(int dim, double f(int dim, int x[]),
                       int init[], int lower[], int upper[]);
```

これは，`dim` 次元の $M^\natural$ 凸関数 `f()` を最小化するルーチンである．利用者は，最小化しようとする離散関数が L / $L^\natural$ / M / $M^\natural$ 凸性のうちどの離散凸性を有するかに従って，該当するルーチンを呼び出すことになる．上記の例では，ルーチンは初期解 `init[]` から探索を行い，最急降下法で最小解にたどりついて，その最小値を返す．そして初期解 `init[]` を上書きして，た

[*1] 関数名の接頭辞の `mg` の g は general の意味で，M 凸の一般化である $M^\natural$ 凸を表している．

表 5.1. ODICON に実装した離散凸関数最小化ルーチン

劣モジュラ関数最小化	(局所探索の方法)
`iff_call`	IFF
`wolfe_call`	FW
L 凸関数最小化	(局所探索の方法)
`lconv_minimize`	最急降下法 (第 3.2.1 節) （全列挙）
`lconv_minimize_IFF`	最急降下法 (第 3.2.1 節) （IFF）
`lconv_minimize_FW`	最急降下法 (第 3.2.1 節) （FW）
`lconv_minimize_scaling`	スケーリング法 (第 3.3.4 節) （全列挙）
`lconv_minimize_scaling_IFF`	スケーリング法 (第 3.3.4 節) （IFF）
`lconv_minimize_scaling_FW`	スケーリング法 (第 3.3.4 節) （FW）
`lconv_minimize_relax`	連続緩和法 (第 3.4.5 節) （IFF）
$L^\natural$ 凸関数最小化	(局所探索の方法)
`lgconv_minimize`	最急降下法 (第 3.2.1 節) （全列挙）
`lgconv_minimize_IFF`	最急降下法 (第 3.2.1 節) （IFF）
`lgconv_minimize_FW`	最急降下法 (第 3.2.1 節) （FW）
`lgconv_minimize_scaling`	スケーリング法 (第 3.3.4 節) （全列挙）
`lgconv_minimize_scaling_IFF`	スケーリング法 (第 3.3.4 節) （IFF）
`lgconv_minimize_scaling_FW`	スケーリング法 (第 3.3.4 節) （FW）
`lgconv_minimize_relax`	連続緩和法 (第 3.4.5 節) （IFF）
M 凸関数最小化	(局所探索の方法)
`mconv_minimize`	最急降下法 (i) (第 4.2.1 節)
`mconv_minimize2`	最急降下法 (ii) [52]
`mconv_minimize3`	最急降下法 (iii) [88]
`mconv_minimize_scaling`	スケーリング法 (第 4.3.3 節) [52]
`mconv_minimize_relax`	連続緩和法 (第 4.4.3 節)
$M^\natural$ 凸関数最小化	(局所探索の方法)
`mgconv_minimize`	最急降下法 (i) (第 4.2.1 節)
`mgconv_minimize2`	最急降下法 (ii) [52]
`mgconv_minimize3`	最急降下法 (iii) [88]
`mgconv_minimize_scaling`	スケーリング法 (第 4.3.3 節) [52]
`mgconv_minimize_relax`	連続緩和法 (第 4.4.3 節)

どり着いた最小解を書き込む．最小解が複数あっても，得られるのはそのうちの一つだけである．探索範囲は `lower`[i]≤`init`[i]≤`upper`[i] ($0 \leq i <$`dim`) に限定される．

上記のルーチンの第 2 引数には，最小化する離散 $M^\natural$ 凸関数を与える．この離散関数は次の型（出力が `double`，入力が `int` と `int` の配列）で宣言しておく．

```
double f(int dim, int x[]);
```

この関数を「関数（ルーチン）の引数」として渡すのだが，通常の C 言語プログラムではあまり使われない手法のため，一般の利用者にとっては難しく思えるかもしれない．しかし記述は

簡単で，引数に関数名をそのまま書くだけでよい．なお，最小化したい離散関数としては，いずれのルーチンでもこの型を受け取るよう統一している．

例えば，次のような 3 次元の離散 $\mathrm{M}^\natural$ 凸関数

$$f(\boldsymbol{x}) = x_0^4 + (x_1 - 3)^2 + 5(x_2 - 7)^2$$

を C 言語で実装すると，次のようになる．

```
double f(int dim, int x[]) {
    double r = 0;

    assert(dim == 3);  /* check dim */
    r += x[0] * x[0] * x[0] * x[0];
    r += (x[1] - 3) * (x[1] - 3);
    r += 5 * (x[2] - 7) * (x[2] - 7);
    return r;
}
```

この関数を，原点から探索して最小化するには，別の関数で次のように処理する．探索範囲は $-100 \leq x_i \leq 100 \quad (0 \leq i <$`dim`$)$ とする．

```
int main() {
    int x[3]     = { 0, 0, 0 };
    int lower[3] = { -100, -100, -100 };
    int upper[3] = {  100,  100,  100 };
    double min = mconv_minimize(3, f, x, lower, upper);
    ...
```

このように呼び出しを行うと，`min` には最小値 `0`，`x[]` にはそれを実現する最小解 {`0,3,7`} が代入される．

このようなルーチンを，離散 L / $\mathrm{L}^\natural$ / M / $\mathrm{M}^\natural$ 凸関数の 4 種類の関数クラスの最小化に対して用意し，それぞれの関数クラスについて，複数の最小化アルゴリズム (最急降下法，スケーリング法，連続緩和法) を実装し利用者がアルゴリズムを指定できるようにした．ルーチンの引数はほぼ共通しており，呼び出すルーチン名を変更するだけで異なるアルゴリズムを試すことができる．

ただし，ソルバの正しい動作が保証されるのは，離散関数の性質とアルゴリズムの組み合わせが正しい時，つまり最小化しようとする関数に適した最小化アルゴリズムを採用した時である．そうでなかった場合は，最小値でないものを出力するなど，結果が不正となるが，不正であるかどうかの判定は行っていない．なぜなら，入力された最小化しようとする関数が，アルゴリズムの要求する離散凸性（L / $\mathrm{L}^\natural$ 凸性あるいは M / $\mathrm{M}^\natural$ 凸性）を有するかどうかを判定することは（探索の開始前も開始後も）一般には簡単ではないからである．なお後述するよう

に、ODICON には限られた条件の下で離散凸性を判定するルーチンを用意した。

離散凸関数最小化アルゴリズムの実装にあたっては，劣モジュラ集合関数最小化や，連続凸関数最小化のルーチンも必要になる．これらには以下のプログラムを利用したが，ルーチンの呼び出し方を統一するように変換するインタフェースも整備した．

- 劣モジュラ関数最小化に，Iwata–Fleischer–Fujishige (IFF) アルゴリズム [35] の岩田による実装.
- 劣モジュラ関数最小化に，Fujishige–Wolfe (FW) アルゴリズム [18]（最小ノルム基法）の藤重・磯谷による実装.
- 連続関数最小化に，quasi-Newton 法の J. Nocedal による FORTRAN 言語での実装である ‘L-BFGS’[*2] と，工藤による C++ 言語へのラッパー [*3].
- 擬似乱数の生成に，斎藤・松本による ‘SIMD-oriented Fast Mersenne Twister’[*4].

表5.1 に示したルーチンについて説明する．

- L / L$^\natural$ 凸関数に対する最急降下法には，局所探索の劣モジュラ集合関数最小化に (i) 全列挙，(ii) IFF，(iii) FW を用いた 3 種類のルーチンを用意している．IFF や FW は（関数値評価の他に）実数計算を含むため丸め誤差の影響を受けるが，全列挙は（多項式時間アルゴリズムではないものの）より安定している．劣モジュラ集合関数最小化において，FW は初期点と最小解が近いときに早く終了するが，IFF の実行時間は初期点と最小解の距離にあまり依存しないことが観察されている．
- L / L$^\natural$ 凸関数に対するスケーリング法においても，局所探索の劣モジュラ集合関数最小化に (i) 全列挙，(ii) IFF，(iii) FW を用いた 3 種類のルーチンを用意している．
- L / L$^\natural$ 凸関数に対する連続緩和法において，仕上げ段階に用いる最急降下法には FW に基づく最急降下法を採用した．これは，連続緩和解を整数に丸めた初期点が真の最小解に十分近いことが多く，そのような場合には FW は非常に早く終了するという経験的事実を考慮した結果である．
- M / M$^\natural$ 凸関数に対する最急降下法としては，第 5.2.2 節に述べた 3 種類のアルゴリズム：
  (i) $f(\boldsymbol{x}-\boldsymbol{\chi}_i+\boldsymbol{\chi}_j)$ を最小化する (i,j) を見出す基本形 [61, 62],
  (ii) 任意に選んだ i に対して $f(\boldsymbol{x}-\boldsymbol{\chi}_i+\boldsymbol{\chi}_j)$ を最小化する j を見出す修正形 [52],
  (iii) 修正形 (ii) に領域縮小を組み込んだもの [88],
  のそれぞれに対応するルーチンを用意している．
- M / M$^\natural$ 凸関数に対するスケーリング法は，Moriguchi–Murota–Shioura [52] のアルゴリズムである．これはスケーリング後も M / M$^\natural$ 凸性が保たれるような M / M$^\natural$ 凸関数についてのみ計算量を保証するアルゴリズムとして提案されたものであるが，一般の

[*2] http://www.ece.northwestern.edu/∼nocedal/lbfgs.html
[*3] http://chasen.org/∼taku/software/misc/lbfgs/
[*4] http://www.math.sci.hiroshima-u.ac.jp/∼m-mat/MT/SFMT/

表 5.2. ODICON に実装した離散凸性判定ルーチン

一般の関数の離散凸性判定	
`is_lconv_gene`	L 凸性判定
`is_lgconv_gene`	$L^\natural$ 凸性判定
`is_mconv_gene`	M 凸性判定
`is_mgconv_gene`	$M^\natural$ 凸性判定
2 次関数の離散凸性判定	
`is_lconv_quadratic`	L 凸性判定
`is_lgconv_quadratic`	$L^\natural$ 凸性判定
`is_mconv_quadratic`	M 凸性判定
`is_mgconv_quadratic`	$M^\natural$ 凸性判定
関数からのヘッセ行列の生成	
`make_hesse_l`	L 凸関数
`make_hesse_lg`	$L^\natural$ 凸関数
`make_hesse_m`	M 凸関数
`make_hesse_mg`	$M^\natural$ 凸関数

M / $M^\natural$ 凸関数に対しても（計算量の理論保証はないものの）真の最小解を出力する．

- M / $M^\natural$ 凸関数に対する連続緩和法において，仕上げ段階に用いる最急降下法には，M / $M^\natural$ 凸関数の最急降下法の (ii) のルーチンを用いている．他の 2 つの最急降下法を使っても大きな差はない．

利用したライブラリについて補足する．

- 劣モジュラ関数最小化の IFF と FW の実装は，どちらも実数演算を行うために，劣モジュラ関数の値が極端に大きい、あるいは小さいときに，誤差の影響を受けることがある．そこで，結果が正しくないと思われる時には，動的に劣モジュラ関数の値を定数倍（スケール）して，誤差の影響を排除するよう試行錯誤する工夫を付け加えた．
- 連続関数最小化をする ‘L-BFGS’ は，連続緩和法の初期解を求めるのに用いた．このライブラリには，最小化したい凸関数だけでなく，その導関数も C 言語上の関数として与える必要があるが，片側差分で近似するしくみを追加したので，表面上は導関数が必要ない．また，通常の最小解を求めるのとは異なり，丸めて整数解として用いるだけの精度があればよいので，収束判定を甘くして，早い段階で解の更新を打ち切るようにした．
- 乱数生成をする ‘SIMD-oriented Fast Mersenne Twister’ は，問題例（インスタンス）を作る際に利用した．なお，C 言語の標準関数にも `rand()` があるが，生成される乱数系列がコンパイラ環境によって変化するため，利用しなかった．

また ODICON には，最小化ルーチンに加えて、表 5.2 のような離散凸性判定ルーチンを実装した．2 次関数の場合には，関数を定義する係数行列の性質を調べることにより，容易に判定が可能である（定理 2.14、定理 2.15、定理 2.33、定理 2.35 参照）．それに対して、一般の

図 5.1. 公開した ODICON ソルバ

関数の場合は、離散凸性の判定は困難である。ここでは、指定した実効定義域内の各点上で差分により離散ヘッセ行列を生成し、すべての点で離散凸性を満たせば、関数全体として離散凸性を満たすと判断をする実装を行ったため、実行には膨大な時間がかかる。なお、離散ヘッセ行列の作り方は自明ではない [5, 11, 28, 51]。

5.3.2 使用法

図 5.1 のように公開した ODICON ソルバの、基本的な使い方を説明する。

動作環境

ODICON は C 言語ソースの形で配布しているので，利用するには C コンパイラが必要である．開発には ANSI C (C89) 規格に準じたものを用いた [*5]。

コンパイル

ダウンロードした `odicon-20YYMMDD.tar.gz` を適当なディレクトリで展開する．20YYMMDD はバージョン番号を表す．（公開した日付でもある．）

```
% gzip -cd odicon-20YYMMDD.tar.gz | tar xvf -
```

[*5] gcc 4.1.2 に`-std=c89` のオプションを付けた．

環境に合わせて C コンパイラのコマンド名などを Makefile に書き込み，'make' コマンドを実行する．

```
% cd odicon-20YYMMDD
% make
```

この作業で，サンプルである odicon という実行ファイルが生成される．これはテスト用の関数を最小化するプログラムである．使い方は次の通りである．

```
% ./odicon L dim [seed]
% ./odicon M dim [seed]
```

引数 1　'L' か 'M' を指定する．
引数 2　インスタンスの次元数を指定する．
引数 3　（省略可）インスタンス生成に使う乱数の種を指定する．

このサンプルプログラムは指定された引数に従ってインスタンスを乱数で生成し，いくつかのアルゴリズムで最小化を行う．

自作プログラムの作り方

上記の使い方を踏襲して，自作プログラムの作り方を説明する．ODICON の配布時点では main() 関数が odicon.c に含まれているが，それを使わずに，自前のプログラムで main() を用意する．その例として，sample.c というプログラムを収録している．このプログラムを元に，hogehoge という名前のプログラムを作るには、まず Makefile 中の

```
PACKAGE = odicon
```

という行を

```
PACKAGE = hogehoge
```

と書き換える．そして sample.c を hogehoge.c という名前でコピーする．

これだけで，もっともシンプルな使い方，つまり，次の 2 つの使い方ができる．

- "make" を実行すると、実行ファイル hogehoge が生成される．
- "make tar" を実行すると，hogehoge-20YYMMDD.tar.gz が生成される。このファイルは、コンパイルに必要なソースコードをすべて含んでいるので、バージョン管理や、ソフトウェアを公開する場合に有益である。

この後，hogehoge.c に含まれる main() 関数を自由に書き換えればよい．最小化したい関数も hogehoge.c に書き加えてよい。

5.4 Web アプリケーション

前述の離散凸性（L 凸性と M 凸性）を有する関数の最小化アルゴリズムや関連するアルゴリズムを実装し，離散凸関数最適化ソルバを始めとするアプリケーションソフトウェアを公開すると同時に，ソルバの動作を手軽に試せるように，いくつかは Web 上のデモンストレーションソフトウェアとしても公開している [68]．ここでは，その中から，離散凸関数最小化ソルバ，在庫管理アプリケーション，コールセンターのシフトスケジューリングアプリケーションを取り上げる．

5.4.1 離散凸関数最小化ソルバのデモアプリケーション

離散凸関数最小化ソルバのデモンストレーションを目的として，2 次の $\mathrm{L}^\natural$ / $\mathrm{M}^\natural$ 凸関数，擬分離 $\mathrm{L}^\natural$ 凸関数，層型 $\mathrm{M}^\natural$ 凸関数などの最小化を扱う Web アプリケーションを提供した．

2 次関数

$\mathrm{L}^\natural$ 凸 2 次関数最小化のデモンストレーションにおいては，利用者が定義する 2 次の $\mathrm{L}^\natural$ 凸関数

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top A\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x}$$

の最小化問題を最急降下法で解く．利用者は，n 次対称行列 $A = (a_{ij})$ と n 次元ベクトル $\boldsymbol{b} = (b_i)$ を入力することによって関数 f を定義し，初期解も指定することができる．アプリケーションは，定理 2.14 に従って利用者の入力が $\mathrm{L}^\natural$ 凸関数を与えるかを判定する（図 5.2 の上）．入力が正しい場合（あるいは利用者が正しいものに修正した場合）には，「Minimize」ボタンを押すことが可能になる．利用者がこのボタンを押すと，アプリケーションは（全列挙に基づく）最急降下法を適用し，計算の途中経過とともに出力する（図 5.2 の下）．$\mathrm{M}^\natural$ 凸 2 次関数に関しても，同様である．

擬分離 $\mathrm{L}^\natural$ 凸関数

擬分離 $\mathrm{L}^\natural$ 凸関数最小化のデモンストレーションにおいては，

$$f(\boldsymbol{x}) = \sum_{i \neq j} f_{ij}(x_i - x_j) + \sum_{i=1}^{n} f_i(x_i)$$

の形の離散凸関数（擬分離 $\mathrm{L}^\natural$ 凸関数）の最小化問題を最急降下法で解く．各 f_{ij}, f_i は 1 変数の凸関数であり，2 次関数，4 次関数，指数関数，絶対値関数などを提供している．利用者は，パラメータを入力することによって関数 f を定義し，初期解も指定することができる．アプリケーションは，（IFF 法に基づく）最急降下法を適用し，計算の途中経過とともに出力する．

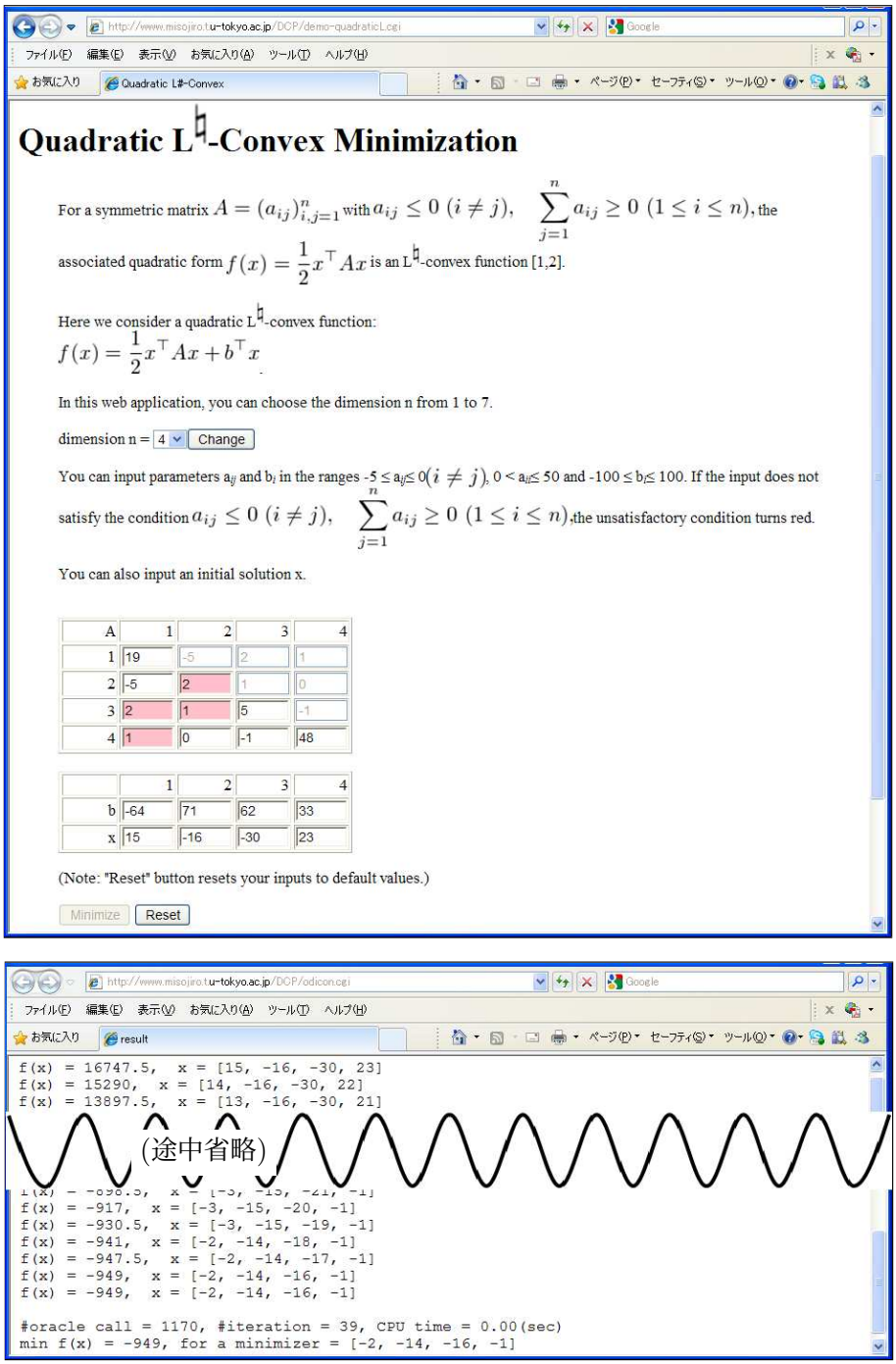


図 5.2. 開発したオンラインソルバ (L♮ 凸 2 次関数最小化問題の入力と出力)

層型 M♮ 凸関数

層型 M♮ 凸関数最小化のデモンストレーションにおいては,

$$f(\boldsymbol{x}) = \sum_{Y \in \mathcal{T}} f_Y(x(Y)), \qquad x(Y) = \sum_{i \in Y} x_i$$

の形の離散凸関数の最小化問題を最急降下法で解く．ここで，$\mathcal{T}$ は層族（任意の $X, Y \in \mathcal{T}$ に対して $X \cap Y$, $X \setminus Y$, $Y \setminus X$ のどれかは空集合である集合族）で，各 f_Y は 1 変数の凸関数である．2 次関数，4 次関数，指数関数，絶対値関数などを提供している．利用者は，パラメータを入力することによって関数 f を定義し，初期解も指定することができる．アプリケーションは，最急降下法 (i) を適用し，計算の途中経過とともに出力する．

5.4.2 在庫管理アプリケーション

離散凸解析は，オペレーションズ・リサーチ (OR) における問題解決にも利用され，在庫管理問題やスケジューリング問題などへの応用が報告されている [6, 41, 49, 99, 100]．在庫管理の理論は長い歴史をもつが，現代の SCM (Supply Chain Management) においてもその基礎として重要である．在庫管理理論と離散凸関数の関わりは古く，1970 年代に Miller [47, 48] は予備品在庫管理問題の研究において，現在「Miller の離散凸関数」と呼ばれる概念を定義している．その後の離散凸解析の研究により，Miller の予備品在庫管理問題に現れる関数は $\mathrm{L}^\natural$ 凸関数であり，したがって，この問題は多項式時間で求解が可能であることが明らかになっている．

本論文では、予備品在庫管理問題に関する在庫コスト最小化アプリケーションを開発して，オンラインソルバとして公開した．最適化エンジンとして、離散凸関数最小化ソルバ ODICON を組み込んだ。アプリケーションでは，部品数 n をはじめとする各種パラメータ $p, c_j, \lambda_j (j = 1, 2, \ldots, n)$ を対話的に入力し (図 5.3 の上)，最適化を実行すると，最適な予備品の準備個数と，その際のコストが表示される (図 5.3 の下)．

ここで用いた予備品在庫管理モデルは Miller [47] によって提案されたもので，需要量，発注量が離散値をとるもの（バックオーダーを考慮した多品種モデル）であり，航空機整備における部品管理等に用いられる．文献 [47] では，在庫費用関数の数学的な取扱いのために「Miller の離散凸関数」の概念を定義し，（計算量が多項式時間ではない）最小化アルゴリズムを提案している．この在庫費用関数は，後に，$\mathrm{L}^\natural$ 凸関数であることが判明している [50, 65]．

Miller のモデルは，部品（品種）数が n からなる製品の製造において，各部品の不足個数（バックオーダー）の最大数によって定まる罰金と，あらかじめ予備品（スペア）を購入しておくための代金の和として定義されるコストをできるだけ減らそうとする在庫モデルである．$c_j > 0$ を品種 j の単価とし，$x_j \in \mathbb{Z}$ を品種 j の予備品の発注量とすると，予備品の購入代金は $\sum_{j=1}^{n} c_j x_j$ となる．一方，品種 j の需要（を表す非負整数値確率変数）が m である確率を $\varphi_j(m)$ とし，累積分布関数を

$$F_j(k) = \sum_{m=0}^{k} \varphi_j(m) \qquad (k \in \mathbb{Z}_+)$$

とすると，n 品種の中の最大バックオーダー量の（定常状態における）期待値は

$$\sum_{k=0}^{\infty} \left(1 - \prod_{j=1}^{n} F_j(x_j + k) \right)$$

となる（詳しくは [65] の 14.7 節を参照のこと）．発注量 $(x_1, x_2, \ldots, x_n)$ をうまく定めて

$$f(\boldsymbol{x}) = p \sum_{k=0}^{\infty} \left(1 - \prod_{j=1}^{n} F_j(x_j + k) \right) + \sum_{j=1}^{n} c_j x_j \tag{5.3}$$

を最小化することが目的である．ただし，$p > 0$ は罰金を表す．ここでは，品種 j の需要の分布として，パラメータ $\lambda_j > 0$ のポアソン分布

$$\varphi_j(m) = \exp(-\lambda_j) \, \frac{\lambda_j^m}{m!} \qquad (m \in \mathbb{Z}_+)$$

を用いる．

すでに述べたように，(5.3) 式の f は $\mathrm{L}^\natural$ 凸関数である．このことから, $\mathrm{L}^\natural$ 凸関数の最小化アルゴリズムにより，$f(\boldsymbol{x})$ の最小化を効率的に行うことができる．

本節で開発したオンラインソルバは，利用者が対話的にパラメータ入力・最適化を行って瞬時に結果を得られる範囲の問題サイズに対して提供しているが，さらに大規模な在庫管理（$n = 50$ 程度まで）を必要とする場合は，ソルバ ODICON をダウンロードして利用者のローカル環境で最適化を実行することができる．

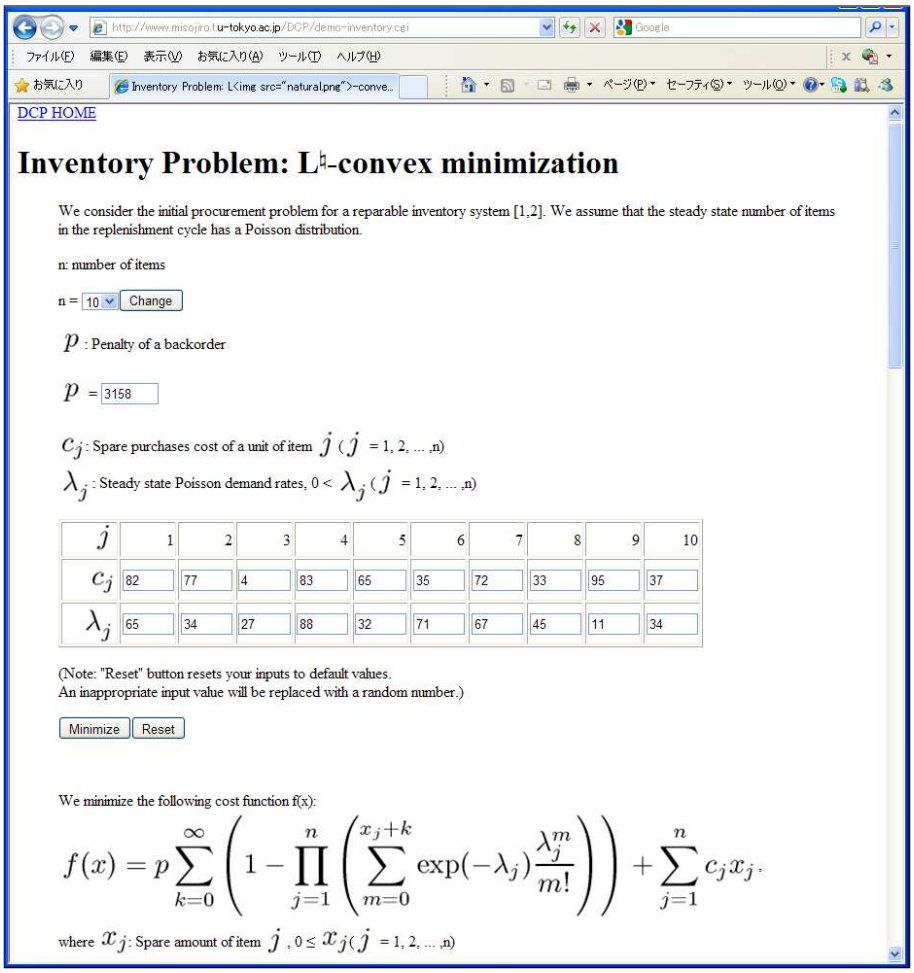
5.4.3　コールセンターにおけるシフトスケジューリングアプリケーション

コールセンターは，企業と顧客の接点として近年，増々重要となってきている．電話を受けることを主とするインバウンドの業務形態において，サービスレベルを保ちつつ各時刻に配置するオペレータの人数を決定するシフトスケジューリング問題が Koole–Sluis [41] で扱われ，そこではマルチモジュラ性 [3, 4, 27] が重要な役割を果たしている．このマルチモジュラ性は（変数変換を通じて）離散凸解析における $\mathrm{L}^\natural$ 凸性と等価な概念である．

本論文では、コールセンターにおけるシフトスケジューリングのアプリケーションを開発して，オンラインソルバとして公開した．最適化エンジンとして、離散凸関数最小化ソルバ ODICON を組み込んだ。アプリケーションでは，各時間区間の客の到着率 λ_i，オペレータのサービス率 μ，客を待たせる時間の限界の基準値 c などのパラメータを対話的に入力し (図 5.4)，最適化を実行すると，各シフトに配置するオペレータの最適な人数と，その際のサービスレベルが表示される (図 5.5)．

ここで用いたモデル（Koole–Sluis [41] のモデル）の概要は以下の通りである．

- コールセンターは，I 個の（連続する）時間区間に渡って稼働する．時間区間を $i = 1, 2, \ldots, I$ で表す．
- 各オペレータは連続した M 個の時間区間に勤務する．
- 勤務シフトは K 種類あり，それぞれの開始時点（と終了時点）は予め指定されている．シフトを $k = 1, 2, \ldots, K$ で表し，第 k シフトの開始時間区間を I_k で表す．ただし，$1 \leq I_1 < I_2 < \cdots < I_K \leq I - M + 1$ とする．図 5.6 にシフトの例を示す（$I = 13$, $M = 5$, $K = 4$, $I_1 = 1$, $I_2 = 3$, $I_3 = 6$, $I_4 = 9$）．



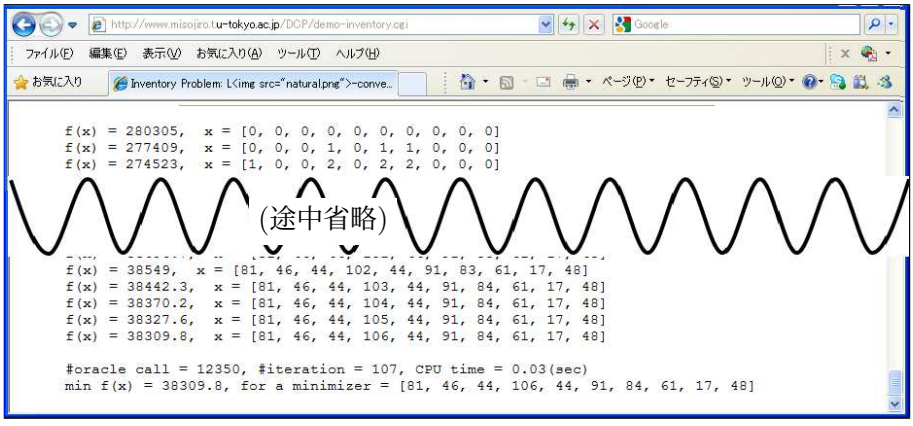


図 5.3. 開発したオンラインソルバ (在庫管理問題の入力と出力)

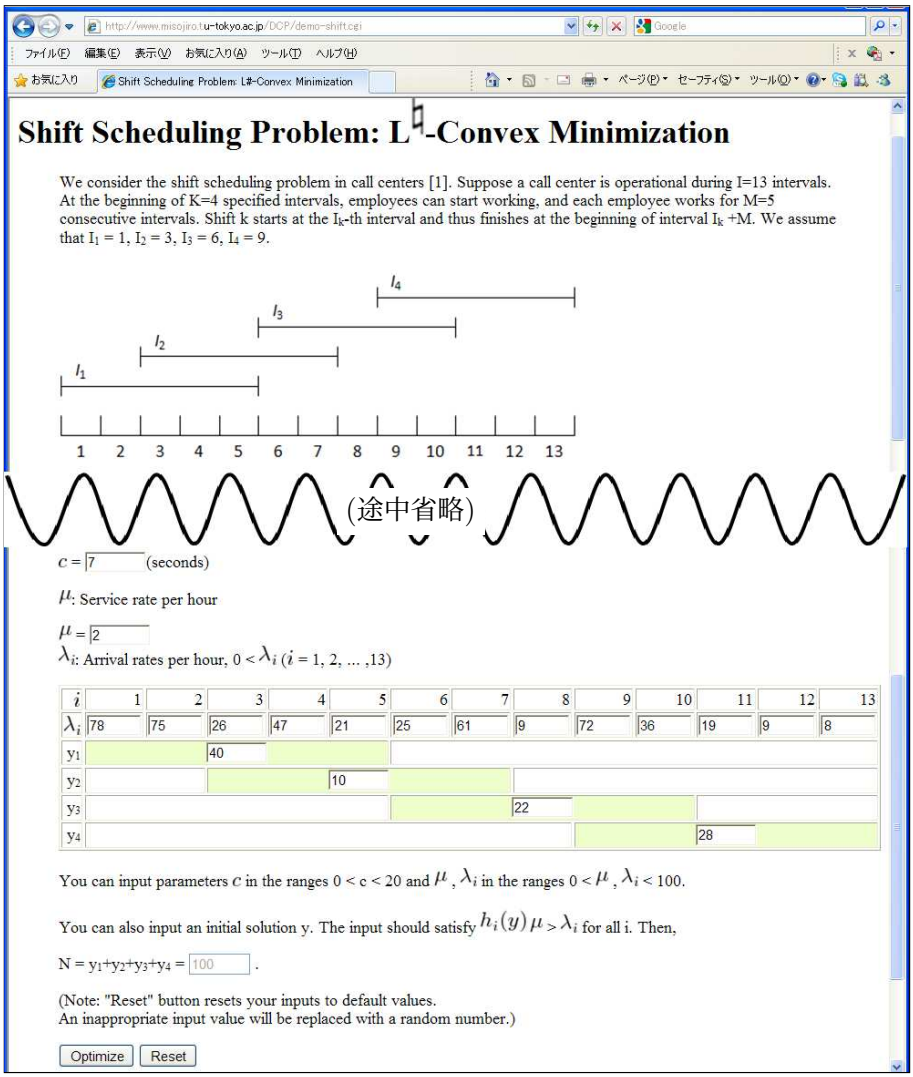


図 5.4. 開発したオンラインソルバ (シフトスケジューリング問題の入力)

- 各時間区間 $i = 1, 2, \ldots, I$ に対して，その区間におけるサービスレベルを表す関数 $g_i(n_i)$ が与えられている．ここで，n_i は時間区間 i に勤務しているオペレータの人数である．関数 g_i は単調増加な凹関数とする．
- 全体のサービスレベル S は $S = \sum_{1 \le i \le I} g_i(n_i)$ で与えられる．

第 k シフトに配置するオペレータの人数を y_k とすると，時間区間 i のオペレータの人数 n_i は

$$h_i(\boldsymbol{y}) = \sum_{k : i - M < I_k \le i} y_k$$

に等しい．したがって，サービスレベル S は $\boldsymbol{y} = (y_1, \ldots, y_K)$ の関数として

$$S(\boldsymbol{y}) = \sum_{1 \le i \le I} g_i(h_i(\boldsymbol{y})) \qquad (\boldsymbol{y} \in \mathbb{Z}^K) \tag{5.4}$$

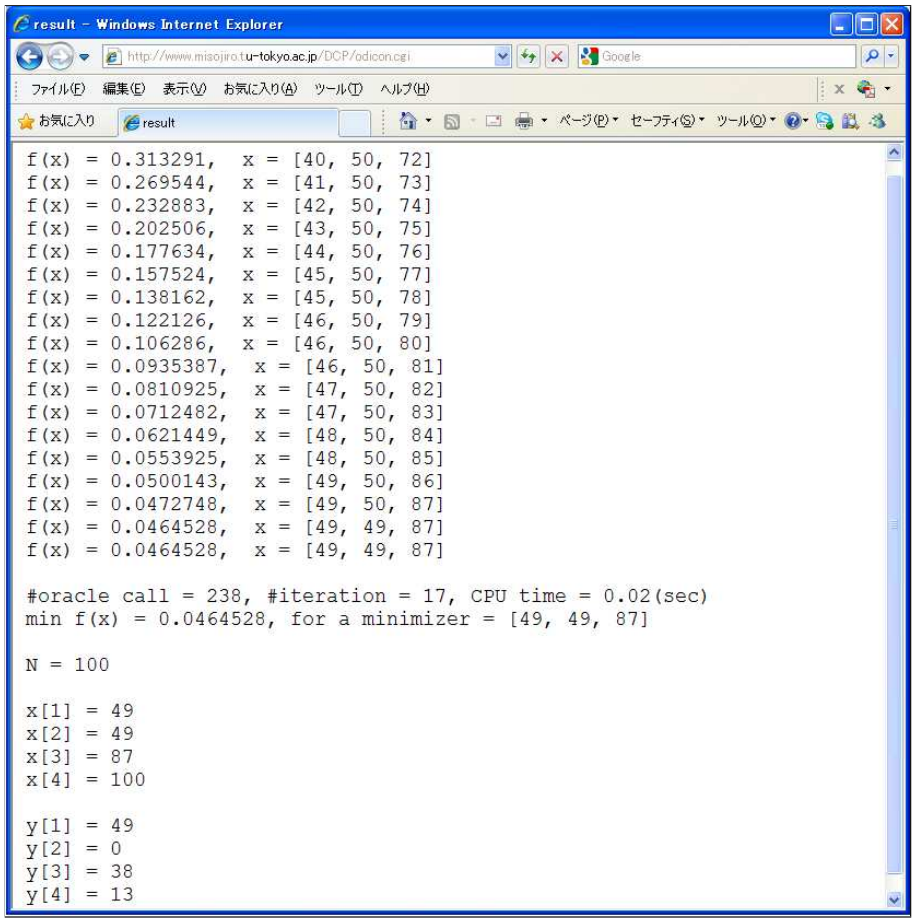


図 5.5. 開発したオンラインソルバ (シフトスケジューリング問題の出力)

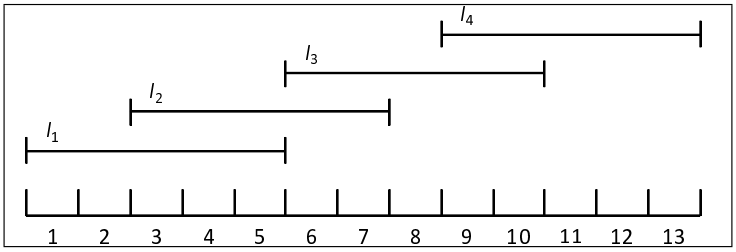


図 5.6. シフトの例

となる．文献 [41] に従い，サービスレベルを一定の水準 s に保ってオペレータの人数を最小化する問題

$$\text{Minimize} \quad \sum_k y_k \quad \text{s.t.} \quad S(\boldsymbol{y}) \geq s, \quad \boldsymbol{y} \in \mathbb{Z}^K \tag{5.5}$$

を考える．このとき，Y をパラメータとする問題

$$\text{Maximize} \quad S(\boldsymbol{y}) \quad \text{s.t.} \quad \sum_k y_k = Y, \quad \boldsymbol{y} \in \mathbb{Z}^K \tag{5.6}$$

を繰り返し解いて，最適な Y を二分探索で見出すことにより，問題 (5.5) の最適解を求めることができる．

サービスレベル $S(\boldsymbol{y})$ の具体形は，待ち行列モデル M/M/n に基づいて，以下のように定める．各時間区間の客の到着率を λ_i $(i = 1, 2, \ldots, I)$, オペレータのサービス率を μ とする．客を待たせる時間の限界の基準値 c（例えば $c = 11$ 秒）を設定し，この時間内にオペレータにつながる客の割合をサービスレベルと定義すると，

$$S(\boldsymbol{y}) = \sum_{1 \le i \le I} \frac{\lambda_i}{\Lambda} \, \mathrm{P}[W_{\lambda_i}(n_i) \le c] = \sum_{1 \le i \le I} \frac{\lambda_i}{\Lambda} \, \mathrm{P}[W_{\lambda_i}(h_i(\boldsymbol{y})) \le c] \tag{5.7}$$

となる．ここで，$n_i = h_i(\boldsymbol{y})$ は時間区間 i のオペレータの人数を表し，$\Lambda = \sum_{1 \le i \le I} \lambda_i$ である．また，$W_\lambda(n)$ は待ち行列モデル M/M/n (到着率 λ, サービス率 μ) における待ち時間 (確率変数) であり，$\mathrm{P}[\cdots]$ は確率を表す．待ち行列理論により，定常状態において

$$\mathrm{P}[W_\lambda(n) \le c] = 1 - \Pi_n \exp[-n\mu(1 - \rho/n)c] \tag{5.8}$$

となることが知られている．ただし，$\rho = \lambda/\mu$, $\rho/n < 1$,

$$\Pi_n = \frac{\dfrac{\rho^n}{n!(1-\rho/n)}}{\displaystyle\sum_{k=0}^{n-1} \frac{\rho^k}{k!} + \frac{\rho^n}{n!(1-\rho/n)}}$$

である．このとき，各 i に対して，

$$g_i(n) = \frac{\lambda_i}{\Lambda} \, \mathrm{P}[W_{\lambda_i}(n) \le c]$$

は n に関して単調増加な凹関数 [*6] となり，モデルの仮定を満たす．

式 (5.4) の関数 $S(\boldsymbol{y})$ は離散凹性をもつ．すなわち，g_i $(i = 1, 2, \ldots, I)$ が単調増加な凹関数であるという仮定の下で，$-S(\boldsymbol{y})$ はマルチモジュラ (multimodular) 関数と呼ばれる離散凸関数になる [41]．マルチモジュラ性は $\mathrm{L}^\natural$ 性と等価な概念であり，マルチモジュラ関数と $\mathrm{L}^\natural$ 凸関数の間には，変数変換を通じて次のような 1 対 1 対応がある [62, 65].

定理 5.1. 関数 $F : \mathbb{Z}^K \to \mathbb{R} \cup \{+\infty\}$ がマルチモジュラ関数であるための必要十分条件は，

$$f(\boldsymbol{x}) = F(x_1, x_2 - x_1, x_3 - x_2, \ldots, x_K - x_{K-1}) \qquad (\boldsymbol{x} \in \mathbb{Z}^K)$$

で定義される関数 $f : \mathbb{Z}^K \to \mathbb{R} \cup \{+\infty\}$ が $\mathrm{L}^\natural$ 凸関数であることであり，このとき

$$F(\boldsymbol{y}) = f(y_1, y_1 + y_2, y_1 + y_2 + y_3, \ldots, y_1 + \cdots + y_K) \qquad (\boldsymbol{y} \in \mathbb{Z}^K)$$

が成り立つ．　■

[*6] 関数 g_i の実効定義域は $\{n \in \mathbb{Z} \mid n > \lambda_i/\mu\}$ であり，$n \le \lambda_i/\mu$ に対しては $g_i(n) = -\infty$ とする．

この事実により，問題 (5.6) は $K-1$ 変数の $\mathrm{L}^\natural$ 凸関数

$$\tilde{f}(\boldsymbol{x}) = -S(x_1, x_2 - x_1, x_3 - x_2, \ldots, x_{K-1} - x_{K-2}, Y - x_{K-1}) \qquad (\boldsymbol{x} \in \mathbb{Z}^{K-1})$$

の制約なしの最小化問題と等価であることが分かる．したがって，問題 (5.6) の厳密解を効率よく求めることができる．

前節の在庫管理アプリケーションと同じく、本節で開発したシフトスケジューリングのオンラインアプリケーションも，利用者が対話的にパラメータ入力・最適化を行って瞬時に結果を得られる範囲の問題サイズに対して提供している。ダウンロードした ODICON ソルバを利用者のローカル環境で実行することで、独自のシフト構成やさらに大規模なスケジューリング（$n = 50$ 程度まで）を最適化することができる。

5.5 成果と考察

ODICON の開発を行うことによって得られた成果を述べる。

ODICON には、図 5.1 のように 3 通りの $\mathrm{L}^\natural$ 凸/L 凸関数最小化アルゴリズムと 5 通りの $\mathrm{M}^\natural$ 凸/M 凸関数最小化アルゴリズムを含めた。これらのアルゴリズムの計算量（関数評価回数）の測定は容易に行えるので、統一された条件の下でアルゴリズムの性能比較が行えるようになった。例えば、第 3 章の図 3.3 や第 4 章の図 4.1 は、ODICON の出力をまとめたものである。

また ODICON には、2 種類の劣モジュラ集合関数最小化ルーチン（IFF と FW）を含めた。2 つのルーチンは、異なる作者による、異なるアルゴリズムに基づくものである。この 2 つのルーチンを ODICON に内蔵することで、同じ条件の下に計算量比較を行うことができるようになった。図 5.7 のグラフは、第 3 章の図 3.3 の上のグラフと同じ条件で行った計算実験の結果であるが、3 通りの $\mathrm{L}^\natural$ 凸関数最小化アルゴリズムについて、内部で用いる劣モジュラ集合関数最小化ルーチンとして IFF と FW を用いた場合のそれぞれについて測定した。この図から、3 通りのいずれのアルゴリズムについても、IFF よりも FW の計算量が少ないことが読み取れる。その差は、最急降下法で小さく、連続緩和法で大きい。

このように、ODICON が統一的なソフトウェア基盤となり、アルゴリズムの性能測定が容易に行えるようになった。

ODICON には、図 5.2 のように離散凸性の判定ルーチンを含めた。このルーチンは、一般の離散関数について高速に判定を行えるわけではないが、2 次関数については即座に判定が終了する。第 2 章の例 2.12 は、この判定ルーチンを利用して、2 次関数を表す行列が $\mathrm{M}^\natural$ 凸性を満たさず、かつ逆行列が $\mathrm{L}^\natural$ 性を満たすものを探索して、容易に作り出すことができた。

ODICON を利用して Web デモを制作した。このデモを通じて、離散凸解析を利用したいと考える実務家にとって、離散凸関数がどのようなものかを理解する手助けとなることを期待する。

ODICON の開発を通じて得た知見を述べる。

劣モジュラ関数最小化のルーチンによって、L / $\mathrm{L}^\natural$ 凸関数最小化アルゴリズムの性能に大き

な違いがあることをあらためて確認した。理論的に解析される計算量は、最悪ケースの計算量が一般的であり、平均的な問題例に対する挙動とは異なる場合もあるが、L / L$^\natural$ 凸関数最小化問題の小問題である劣モジュラ関数最小化問題に対しては、理論解析通りの性能差がみられた。したがって、指数時間アルゴリズムと，多項式時間アルゴリズム（IFF あるいは FW）の違いは、通常の規模の問題が解けるかどうかの違いにほぼ等しい。

劣モジュラ関数最小化の多項式時間アルゴリズム同士での性能の違いについては、前述のように、理論解析上の計算量の少ない IFF よりも、理論的な裏付けのない FW のほうが高速な場面がみられた。速度の違いは、最急降下法で小さく、スケーリング法や連続緩和法で大きいことが観測されていることから、L / L$^\natural$ 凸関数最小化問題の解の探索の収束間近な場合、つまり劣モジュラ集合関数の関数値がこれ以上小さくできない場面で速度差が開く傾向にあると言える。これは、理論解析からは知ることのできない結果である。

M / M$^\natural$ 凸関数は、L / L$^\natural$ 凸関数に比べて容易に最小化できることを確認した。現実的に解ける問題のサイズは、M 凸関数では 1000 次元に対して、L 凸関数では 100 次元と、10 倍ほどの違いがある．M / M$^\natural$ 凸関数と L / L$^\natural$ 凸関数は双対の関係にあるから，実用上は似た難しさにあることも期待されるが、実際には理論解析通りに解きやすさがまったく違うことがわかった。

L / L$^\natural$ 凸関数の最小化アルゴリズムを素朴に実装した場合は、10 次元くらいまでの問題しか解くことができないため、実用的に役立つには程遠い状況にある。M / M$^\natural$ 凸関数の最小化アルゴリズムであれば、素朴な実装でも 100 次元くらいまでの問題が解けるので、状況によっては実用的に用いることもできるであろう。これに対して ODICON では、どちらも 10 倍ほど大きな規模の問題が解けるので、特に L / L$^\natural$ 凸関数最小化問題では実用面で大きなアドバンテージがあると言える。今後、さらなる性能向上が期待される。

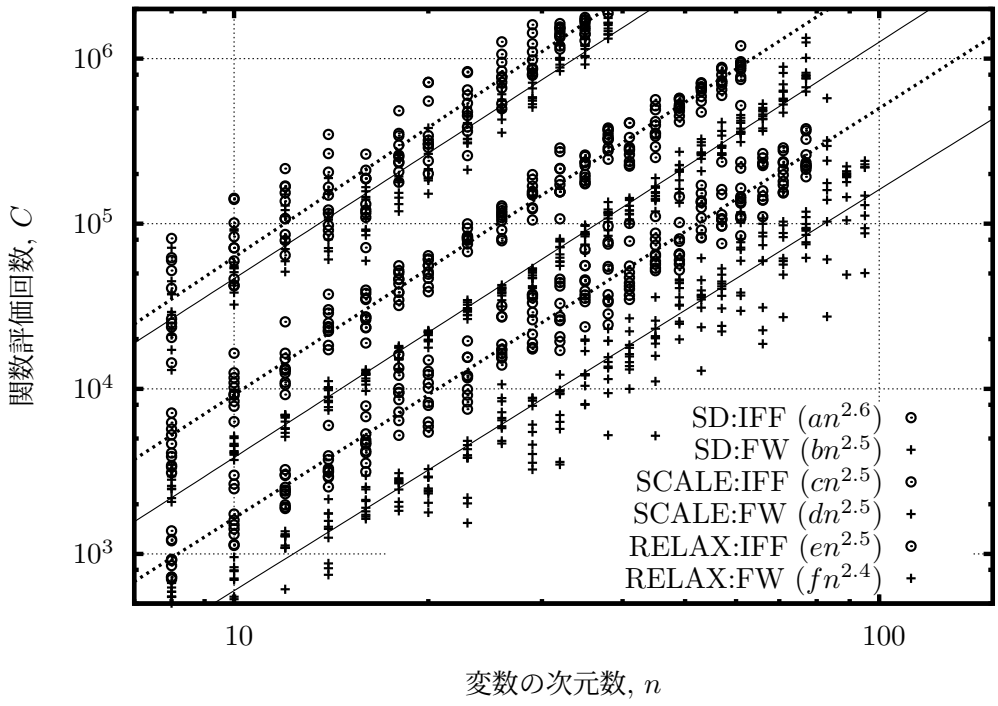


図 5.7. $L^\natural$ 凸関数最小化に必要な目的関数の評価回数（劣モジュラ集合関数最小化ルーチンの違いによる性能比較）

第 6 章

結論

6.1 総括

離散凸解析の分野では、連続変数の凸関数に対応する概念として、離散変数の関数では L 凸関数と M 凸関数という離散凸性が提案され、この 2 つの離散凸性を基盤として美しい理論が構築され、L 凸関数と M 凸関数が Legendre 変換によって互いに移り合うことや、L 凸関数と M 凸関数のそれぞれについて局所最小性と大域最小性が一致することを示す最小性規準の存在が明らかにされていた。そして L 凸関数と M 凸関数のそれぞれの関数最小化問題に対して、降下法に基づく効率的なアルゴリズムが提案され、さらにスケーリング技法を組み合わせることで多項式時間アルゴリズムが確立されていた。

本論文では、現実的な実用性を目指し、L 凸関数と M 凸関数のそれぞれの関数最小化問題に対して、連続緩和手法に基づくアルゴリズムを提案した。連続緩和手法は、しばしば離散最適化問題に用いられる手法であるが、必ずしも容易に適用できるわけではなく、切除平面法などと組み合わせるなど、深い洞察と共に用いられるものである。本論文の提案するアルゴリズムは、連続緩和した滑らかな凸関数が入手できるという前提の下に、まず緩和問題を解き、得られた緩和問題の解を整数ベクトルに丸めて、離散凸関数に対する降下法の初期解とするものである。緩和問題の数値解は、凸解析分野の成果により少ない計算量で得ることができるので、提案手法は高速に計算できることが期待される。そして実際に計算実験を行うことにより、提案手法が従来手法よりも高速に動作することを確かめた。

本論文では、計算実験を行うだけでなく、理論解析を行うことで、提案手法が理論的にも効率的なアルゴリズムであることを確かめた。すなわち、緩和解と求めたい離散解の関係を解析し、その距離の上限が十分に小さいことを保証する近接定理を、L 凸関数の場合と M 凸関数の場合のそれぞれについて証明した。証明した近接定理は、L 凸関数と M 凸関数のどちらの場合でも、形式的にはほぼ同じ不等式で定式化されるが、その背後にある理論的な性質は大きく異なり、証明の方法も異なる。L 凸関数にはスケーリング操作に対しても L 凸性を保持し続けるという特性があるため、スケーリングの近接定理の結果を利用することで、連続緩和の近接定理は容易に証明された。一方、M 凸関数はスケーリング操作に対して M 凸性を保持しないため、スケーリングの近接定理の結果を利用することができず、連続緩和の近接定理の証明

には手数のかかるものとなった。

近接定理を証明した離散 M 凸関数最小化問題は、資源配分問題のバリエーションのいくつかを包含する、包括的な概念である。資源配分問題は、先行研究により問題設定ごとに連続緩和が個別に適用されてきた。そこで本論文では、離散 M 凸関数最小化問題に適用して得られた連続緩和法の結果を、資源配分問題の個別の条件設定にしたがってチューニングを行なった上で、計算量を厳密に解析した。これにより、先行研究と同等の計算量を達成する、あるいは、限られた問題クラスによっては、計算量を改善することを明らかにした。このように、M 凸関数における包括的な成果を、個別の問題に還元することができた。

本論文では、離散凸解析理論成果の普及のために、離散凸最小化アルゴリズムを実装し、ソルバソフトウェアとして公開した。連続最適化の分野では、高性能なソルバが活発に開発されている。離散最適化の分野でも、整数変数の線形計画問題である整数計画問題など、いくつかの問題クラスに対して近年のソルバの性能向上が著しい。しかしながら整数変数の非線形計画問題に対しては、汎用アルゴリズムのソルバは困難な状況にあり、定式化を適切に行なったとしても数値解が容易に求められるとは言いがたい。本論文によるソルバはこの非線形計画問題に対するソルバの一翼を担うものであり、汎用とは言えないまでも、解きやすい凸性を持つ問題を幅広く対象とするものであり、離散凸解析分野における初めてのソルバであると言える。ソルバの性能は、単純な実装に比べて速度で 100〜1000 倍速く、解くことのできる問題規模で 10〜100 倍の大きさを実現する。現実規模の問題も十分に求解の対象となる。このソルバは、周辺分野の研究者や実務家の助けとなることを目指し、他のソフトウェアに組み込んで用いられることを念頭に開発し、再配布が可能となるようライセンスを調整した。またデモンストレーションソフトウェアを開発し、Web 上に公開し、初学者が手軽に問題例を解くことを試せるように配慮した。

6.2　今後の課題

以下のような事柄が今後の課題として残されている.

離散 L 凸性と離散 M 凸性の数値的な判定の効率化　与えられた離散関数に対して、離散凸性を持つことを確かめるには、実効定義域内のすべての点について離散凸性の条件を満たしていることを確かめねばならず、関数最小化問題を解くよりもはるかに多くの計算量が必要であり、これを大幅に削減することは難しい。しかしながら、離散凸性を持たないことを示すには、離散凸性の条件に対する反例を一つ挙げればよい。そこで、メタ解法を利用して反例を少ない計算量で求める工夫が期待される。

離散凸関数最小化ルーチンの高速化　昨今の計算機環境の進歩により、並列計算が容易に行えるようになってきた。（あるいは、半導体製造技術の進歩が曲がり角に来ており、単一プロセッサでの性能向上が望めなくなり、代わりに並列計算による速度向上に頼らざるを得なくなってきたととらえることもできる。）この能力を活かして、ODICON の最小化ルーチンを並列処理によって高速化することが期待される。

連続変数の L 凸/M 凸関数最小化アルゴリズムの効率化 現状では、連続変数の L 凸/M 凸関数を最小化するにあたっては、L 凸/M 凸性を利用する手法は提案されていない。したがって、より弱い条件である凸関数としての性質を利用する準ニュートン法などを用いることになる。2 次関数に限るなどして L 凸/M 凸性を活かすことが一つのアイデアである。実用上と理論上の両方の側面での成果が期待される。

関数を記述するモデリング言語の設計 離散最適化ソルバ ODICON は C 言語で実装しており、最適化したい離散関数も C 言語上で実現する必要がある。つまり、問題例が C 言語プログラムの一部となる必要があり、問題例が変化した場合はプログラムの再コンパイルが必要になる。一方で、通常のソルバと呼ばれるソフトウェアでは、解きたい問題をモデリング言語と呼ばれる枠組みで記述する場合が多い。つまり、問題例がデータとして記述できるので再コンパイルは不要である。現在の ODICON には、このようなモデリング言語による問題記述の枠組みが存在していないが、利用者の利便性を考えると、必要な機能だと考えられる。

連続緩和手法を適用する問題クラスの拡大 新たに連続緩和手法を適用する問題クラスとしては、離散準 L 凸関数最小化問題と離散準 M 凸関数最小化問題が有望である。ただし、離散準 L 凸関数の場合は適用できることがわかっているが [55]、離散 L 凸関数と異なり、連続緩和解を求めた後の手続きに、劣モジュラ関数最小化アルゴリズムが利用できないという困難があり、実用的ではない。また、離散準 M 凸関数の場合は、近接定理が成立するかどうかの調査が必要である。

このような課題を解決すると、ODICON を通じて離散凸解析の裾野を広げることに貢献できると考えられる。

謝辞

関西学院大学の西関隆夫教授には、お忙しい中にもかかわらず、研究活動をサポートしていただき、執筆活動にご指導とお力添えをいただきました。同じく関西学院大学の巳波弘佳教授には、常に新しい話題を提供していただき、研究と執筆活動はもちろんのこと、生活面でも親身に助言をいただきました．東京大学の室田一雄教授には、長年にわたり懇切丁寧に離散凸解析の手ほどきをいただいたことはもちろん、公私にわたってお世話になりました。首都大学東京の森口聡子准教授、東北大学の塩浦昭義准教授にはいくつもの共同研究にお付き合いいただいたことに、感謝の言葉もありません。岩田覚教授（東京大学)、藤重悟教授（京都大学）には、高性能なプログラムをご提供いただき、大きく研究活動が進みました。この場を借りて厚く御礼を申し上げます。

東京大学では数理情報学専攻の大石泰章講師（現 南山大学教授)、牧野和久准教授（現 京都大学准教授)、松浦史郎助手、杉原厚吉教授（現 明治大学特任教授)、松井知己助教授（現 東京工業大学教授)、宮代隆平氏（現 東京農工大学准教授)、平井広志氏（現 東京大学准教授)、来嶋秀治氏（現 九州大学准教授）には、日常生活からお世話になり、研究面でも数々のアドバイスをいただきました。博士課程の垣村尚徳さん（現 東京大学特任講師）や小林佑輔さん（現 東京大学助教)、高松瑞代さん（現 中央大学准教授）を始めとする研究室の皆さんとも楽しい研究室生活が送れました。

京都大学の学部生時代には、茨木俊秀教授（現 京都情報大学院大学)、永持仁助教授（現 教授)、柳浦睦憲助手（現 名古屋大学教授）に丁寧にご指導いただきました。野々部宏司氏（現 法政大学教授）には組合せ最適化問題に対するアプローチの方法を一から手ほどきしていただきました。小野廣隆氏（現 九州大学准教授）を始めとする研究室の皆様には、よく相談に乗っていただきました。そして何よりも、茨木研究室という歴史に裏打ちされた環境の中で、貴重な体験させていただいたことに感謝いたします。久保幹雄教授（東京海洋大学）には、ソフトウェアの作り方から社会の中での立ち振舞について多くの示唆をいただき、貴重な体験をさせていただきました。藤沢克樹助手（現 九州大学教授）にはクラスタコンピュータの管理やソフトウェア開発の生の姿を見せていただき、自らが開発する場面で役立ちました。

関西学院大学 理工学部 情報科学科・人間システム工学科の学生実験準備室の皆様には，日常業務の合間に暖かく見守って執筆活動をサポートしていただきました。心より感謝いたします.

参考文献

[1] A. Ageev, M. Sviridenko: Pipage rounding: A new method of constructing algorithms with proven performance guarantee, *Journal of Combinatorial Optimization*, **8** (2004), 307–328.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin: *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Upper Saddle River, NJ, 1993.

[3] E. Altman, B. Gaujal and A. Hordijk: Multimodularity, convexity, and optimization properties, *Mathematics of Operations Research*, **25** (2000), 324–347.

[4] E. Altman, B. Gaujal and A. Hordijk: *Discrete-Event Control of Stochastic Networks: Multimodularity and Regularity*, Lecture Notes in Mathematics, **1829**, Springer-Verlag, Heidelberg, 2003.

[5] H. Bandelt: Recognition of tree metrics, *SIAM Journal on Discrete Mathematics*, **3** (1990), 1–6.

[6] M. Begen and M. Queyranne: Appointment scheduling with discrete random durations, *Mathematics of Operations Research*, **36** (2011), 240–257.

[7] P. Brucker: An $O(n)$ algorithm for quadratic knapsack problems, *Operations Research Letters*, **3** (1984), 163–166.

[8] N. Buchbinder, M. Feldman, J. Naor, R. Schwartz: A tight linear time (1/2)-approximation for unconstrained submodular maximization, *Proceedings of the 53th Annual IEEE Symposium on Foundation of Computer Science*, 649–658, New Brunswick, New Jersey, 2012.

[9] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák: Maximizing a submodular set function subject to a matroid constraint, *SIAM Journal on Computing*, **40** (2011), 1740–1766.

[10] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver: *Combinatorial Optimization*, John Wiley and Sons, New York, 1998.

[11] E. Dahlhaus: Fast parallel recognition of ultrametrics and tree metrics, *SIAM Journal on Discrete Mathematics*, **6** (1993), 523–532.

[12] A. W. M. Dress and W. Wenzel: Valuated matroids, *Advances in Mathematics*, **93** (1992), 214–250.

[13] M. L. Fisher, G. L. Nemhauser, and L.A. Wolsey: An analysis of approximations for maximizing submodular set functions II, *Mathematical Programming Study*, **8** (1978), 73–87.

[14] A. Frank: An algorithm for submodular functions on graphs, *Annals of Discrete Mathematics*, **16** (1982), 97–120.

[15] S. Fujishige: Theory of submodular programs: A Fenchel-type min-max theorem and subgradients of submodular functions, *Mathematical Programming*, **29** (1984), 142–155.

[16] 藤重悟：グラフ・ネットワーク・組合せ論，共立出版，2002.

[17] S. Fujishige: *Submodular Functions and Optimization*, 2nd ed., Annals of Discrete Mathematics, **58**, Elsevier, 2005.

[18] S. Fujishige and S. Isotani: A submodular function minimization algorithm based on the minimum-norm base, *Pacific Journal of Optimization*, **7** (2011), 3–17.

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

[20] S. Fujishige, K. Murota and A. Shioura: Monotonicity in steepest ascent algorithms for polyhedral L-concave functions, *METR 2014-16* (2014), Department of Mathematical Informatics, University of Tokyo.

[21] 福島雅夫：非線形最適化の基礎，朝倉書店， 2001.

[22] E. Girlich, M. Kovalev, and A. Zaporozhets: A polynomial algorithm for resource allocation problems with polymatroid constraints, *Optimization*, **37** (1996), 73–86.

[23] A. V. Goldberg and R. E. Tarjan: A new approach to the maximum-flow problem, *Journal of the ACM*, **35** (1988), 921–940.

[24] F. Granot and J. Skorin-Kapov: Some proximity and sensitivity results in quadratic integer programming, *Mathematical Programming*, **47** (1990), 259–268.

[25] H. Groenevelt: Two algorithms for maximizing a separable concave function over a polymatroid feasible region, *European Journal of Operational Research*, **54** (1991), 227–236.

[26] M. Grötschel, L. Lovász, and A. Schrijver: The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica*, **1** (1981), 169–197.

[27] B. Hajek: Extremal splittings of point processes, *Mathematics of Operations Research*, **10** (1985), 543–556.

[28] H. Hirai and K. Murota: M-convex functions and tree metrics, *Japan Journal of Industrial and Applied Mathematics*, **21** (2004), 391–403.

[29] D. S. Hochbaum: Lower and upper bounds for the allocation problem and other nonlinear optimization problems, *Mathematics of Operations Research*, **19** (1994), 390–409.

[30] D. S. Hochbaum and S.-P. Hong: About strongly polynomial time algorithms for

quadratic optimization over submodular constraints, *Mathematical Programming*, **69** (1995), 269–309.

[31] D. S. Hochbaum and J. G. Shanthikumar: Convex separable optimization is not much harder than linear optimization, *Journal of the ACM*, **37** (1990), 843–862.

[32] 茨木俊秀，福島雅夫：最適化の手法，共立出版，1993.

[33] T. Ibaraki and N. Katoh: *Resource Allocation Problems: Algorithmic Approaches*, MIT Press, Boston, MA, 1988.

[34] S. Iwata: Submodular function minimization, *Mathematical Programming, Series B*, **112** (2008), 45–64.

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

[36] S. Iwata and M. Shigeno: Conjugate scaling algorithm for Fenchel-type duality in discrete convex optimization, *SIAM Journal on Optimization*, **13** (2003), 204–211.

[37] P. M. Jensen and B. Korte: Complexity of matroid property algorithms, *SIAM Journal on Computing*, **11** (1982), 184–190.

[38] N. Katoh and T. Ibaraki: Resource allocation problems, in: D.-Z. Du and P. M. Pardalos, eds., *Handbook of Combinatorial Optimization, Vol.2*, Kluwer Academic Publishers, Boston, 1998, 159–260.

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

[40] 今野浩，山下浩：非線形計画法，日科技連出版社，1978.

[41] G. Koole and E. van der Sluis: Optimal shift scheduling with a global service level constraint, *IIE Transactions*, **35** (2003), 1049–1055.

[42] B. Korte and J. Vygen: *Combinatorial Optimization: Theory and Algorithms*, 5th ed., Springer-Verlag, Berlin, 2012.

[43] 久保幹雄：組合せ最適化とアルゴリズム，共立出版，2000.

[44] D. C. Liu and J. Nocedal: On the limited memory method for large scale optimization, *Mathematical Programming, Series B*, **45** (1989), 503–528.

[45] L. Lovász: Matroid matching and some applications, *Journal of Combinatorial Theory (B)*, **28** (1980), 208–236.

[46] L. Lovász: Submodular functions and convexity, in: A. Bachem, M. Grötschel and B. Korte, eds., *Mathematical Programming—The State of the Art*, Springer-Verlag, Berlin, 1983, 235–257.

[47] B. L. Miller: On minimizing nonseparable functions defined on the integers with an inventory application, *SIAM Journal on Applied Mathematics*, **21** (1971), 166–185.

[48] B. L. Miller: A multi-item inventory model with joint backorder criterion, *Operations Research*, **19** (1971), 1467–1476.

[49] 水野幸男：在庫管理入門，日科技連出版社，1974.

[50] S. Moriguchi and K. Murota: Discrete Hessian matrix for L-convex functions, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **E88-A** (2005), 1104–1108.

[51] S. Moriguchi and K. Murota: On discrete Hessian matrix and convex extensibility, *Journal of Operations Research Society of Japan*, **55** (2012), 48–62.

[52] S. Moriguchi, K. Murota and A. Shioura: Scaling algorithms for M-convex function minimization, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **E85-A** (2002), 922–929.

[53] S. Moriguchi, A. Shioura and N. Tsuchimura: M-convex function minimization by continuous relaxation approach—Proximity theorem and algorithm, *SIAM Journal on Optimization*, **21** (2011), 633–668.

[54] S. Moriguchi and N. Tsuchimura: Discrete convex functions minimization based on continuous relaxation, Ninth International Conference Approximation and Optimization in the Caribbean - APPOPT ' 2008, San Andres, Colombia, March 4, 2008.

[55] S. Moriguchi and N. Tsuchimura: Minimization of a discrete quasi L-convex function based on continuous relaxation, The 4th Sino-Japanese Optimization Meeting (SJOM2008), National Cheng-Kung University, Tainan, Taiwan, August 27-31, 2008.

[56] S. Moriguchi and N. Tsuchimura: Discrete L-convex function minimization based on continuous relaxation, *Pacific Journal of Optimization*, **5** (2009), 227–236.

[57] K. Murota: Convexity and Steinitz's exchange property, *Advances in Mathematics*, **124** (1996), 272–311.

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

[59] K. Murota: *Matrices and Matroids for Systems Analysis*, Springer-Verlag, Berlin, 2000.

[60] K. Murota: Algorithms in discrete convex analysis, *IEICE Transactions on Systems and Information*, **E83-D** (2000), 344–352.

[61] 室田一雄：離散凸解析，共立出版，2001.

[62] K. Murota: *Discrete Convex Analysis*, SIAM Monographs on Discrete Mathematics and Applications, Vol. 10, Society for Industrial and Applied Mathematics, Philadelphia, 2003.

[63] K. Murota: On steepest descent algorithms for discrete convex functions, *SIAM Journal on Optimization*, **14** (2003), 699–707.

[64] K. Murota: Note on multimodularity and L-convexity, *Mathematics of Operations Research*, **30** (2005), 658–661.

[65] 室田 一雄：離散凸解析の考えかた，共立出版，東京，2007.

[66] K. Murota: Recent developments in discrete convex analysis, in: W. Cook, L. Lovasz and J. Vygen, eds., *Research Trends in Combinatorial Optimization, Bonn 2008*,

Springer-Verlag, Berlin, 2009, Chapter 11, 219–260.

[67] K. Murota: Submodular function minimization and maximization in discrete convex analysis, *RIMS Kokyuroku Bessatsu*, **B23** (2010), 193–211.

[68] K. Murota, S. Iwata, A. Shioura, S. Moriguchi, N. Tsuchimura, and N. Kakimura: DCP (Discrete Convex Paradigm), `http://www.misojiro.t.u-tokyo.ac.jp/DCP/`

[69] K. Murota and A. Shioura: M-convex function on generalized polymatroid, *Mathematics of Operations Research*, **24** (1999), 95–105.

[70] K. Murota and A. Shioura: Extension of M-convexity and L-convexity to polyhedral convex functions, *Advances in Applied Mathematics*, **25** (2000), 352–427.

[71] K. Murota and A. Shioura: Quadratic M-convex and L-convex functions, *Advances in Applied Mathematics*, **33** (2004), 318–341.

[72] K. Murota and A. Shioura: Conjugacy relationship between M-convex and L-convex functions in continuous variables, *Mathematical Programming*, **101** (2004), 415–433.

[73] K. Murota and A. Shioura: Fundamental properties of M-convex and L-convex functions in continuous variables, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **E87-A** (2004), 1042–1052.

[74] K. Murota and A. Shioura: Note on the continuity of M-convex and L-convex functions in continuous variables, *Journal of the Operations Research Society of Japan*, **51** (2008), 265–273.

[75] 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.

[76] K. Murota and A. Shioura: Dijkstra's algorithm and L-concave function maximization, *Mathematical Programming, Series A*, **145** (2014), 163–177.

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

[78] K. Murota and A. Tamura: Application of M-convex submodular flow problem to mathematical economics, *Japan Journal of Industrial and Applied Mathematics*, **20** (2003), 257–277.

[79] K. Murota and A. Tamura: Proximity theorems of discrete convex functions, *Mathematical Programming*, **99** (2004), 539–562.

[80] G. L. Nemhauser, L.A. Wolsey, and M. L. Fisher: An analysis of approximations for maximizing submodular set functions I, *Mathematical Programming*, **14** (1978), 265–294.

[81] J. B. Orlin: A faster strongly polynomial time algorithm for submodular function minimization, *Mathematical Programming*, **118** (2009), 237–251.

[82] A. Recski: *Matroid Theory and Its Applications in Electric Network Theory and in Statics*, Springer-Verlag, Berlin, 1989.

[83] R. T. Rockafellar: *Convex Analysis*, Princeton University Press, Princeton, 1970.

[84] A. Schrijver: *Theory of Linear and Integer Programming*, Wiley, New York, NY,

1986.

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

[86] A. Schrijver: *Combinatorial Optimization—Polyhedra and Efficiency*, Springer-Verlag, Heidelberg, 2003.

[87] A. Shioura: Minimization of an M-convex function, *Discrete Applied Mathematics*, **84** (1998), 215–220.

[88] A. Shioura: Fast scaling algorithms for M-convex function minimization with application to the resource allocation problem, *Discrete Applied Mathematics*, **134** (2003), 303–316.

[89] M. Sviridenko: A note on maximizing a submodular set function subject to a knapsack constraint, *Operations Research Letters*, **32** (2004), 41–43.

[90] A. Tamir: A strongly polynomial algorithm for minimum convex separable quadratic cost flow problems on series-parallel networks, *Mathematical Programming*, **59** (1993), 117–132.

[91] A. Tamura: Coordinatewise domain scaling algorithm for M-convex function minimization, *Mathematical Programming*, **102** (2005), 339–354.

[92] 田村明久：離散凸解析とゲーム理論，朝倉書店，2009.

[93] 田村明久，村松正和：最適化法，共立出版，2002.

[94] 土村展之：離散凸最適化ソルバ ODICON, `http://www.misojiro.t.u-tokyo.ac.jp/~tutimura/odicon/`

[95] 土村 展之：離散凸解析ソルバ ODICON と Web アプリケーション，日本オペレーションズ・リサーチ学会会誌 **56**:6 (2013), 339–345.

[96] 土村 展之，森口 聡子，室田 一雄：離散凸最適化ソルバとデモンストレーションソフトウェア，応用数理学会論文誌 **23**:2 (2013), 233–252.

[97] J. Vondrák: Optimal approximation for the submodular welfare problem in the value oracle model, *Proceedings of the 40th ACM Symposium on Theory of Computing*, 67–74, ACM, New York, 2008.

[98] I. J. Weinsten and O. S. Yu: Comment on an integer maximization problem, *Operations Research*, **21** (1973), 648–650.

[99] P. Zipkin: *Foundations of Inventory Management*, McGraw-Hill, Boston, 2000.

[100] P. Zipkin: On the structure of lost-sales inventory models, *Operations Research*, **56** (2008), 937-944.

発表論文リスト

学術雑誌論文

1. S. Moriguchi, A. Shioura and N. Tsuchimura: M-convex function minimization by continuous relaxation approach—Proximity theorem and algorithm, *SIAM Journal on Optimization*, **21** (2011), 633–668. （第 4.2.3 節, 第 4.4.2 節, 第 4.4.3 節, 第 4.5 節, 第 4.6 節, 第 4.7 節）
2. S. Moriguchi and N. Tsuchimura: Discrete L-convex function minimization based on continuous relaxation, *Pacific Journal of Optimization*, **5** (2009), 227–236. （第 3.4.4 節, 第 3.4.5 節, 第 3.5 節, 第 3.6 節）
3. 土村 展之：離散凸解析ソルバ ODICON と Web アプリケーション，日本オペレーションズ・リサーチ学会会誌 **56**:6 (2013), 339–345. （第 5.3 節, 第 5.5 節）
4. 土村 展之，森口 聡子，室田 一雄：離散凸最適化ソルバとデモンストレーションソフトウェア，応用数理学会論文誌 **23**:2 (2013), 233–252. （第 5.3 節, 第 5.4 節）

口頭発表

1. S. Moriguchi and N. Tsuchimura: Discrete convex functions minimization based on continuous relaxation, Ninth International Conference Approximation and Optimization in the Caribbean - APPOPT ’ 2008, San Andres, Colombia, March 4, 2008.
2. S. Moriguchi and N. Tsuchimura: Minimization of a discrete quasi L-convex function based on continuous relaxation, The 4th Sino-Japanese Optimization Meeting (SJOM2008), National Cheng-Kung University, Tainan, Taiwan, August 27-31, 2008.
3. 土村展之，森口聡子：“離散準 L 凸関数最小化における連続緩和,” 研究集会「最適化：モデリングとアルゴリズム」, 2008 年 3 月 18 日.
4. 土村展之，森口聡子，垣村尚徳，岩田覚，室田一雄：“離散凸最適化ソルバとデモンストレーションソフトウェアの公開,” 情報処理学会第 133 回アルゴリズム研究会, 2011 年 1 月.
5. 土村展之: “離散凸関数の最小化ソルバ ODICON の開発,” 研究集会「数学ソフトウェアの開発と実践」, 2013 年 9 月.