はじめに

まず，読者の皆さんが本書で扱う題材を厳密な数学的定式化なしに見通せるように，最適化に関する基本的な概念や結果を平易な表現で紹介する．特に，最適化における 3 つの主題である線形最適化，組合せ最適化と非線形最適化の本質的な違いを理解して欲しい．直感的，鳥瞰的に眺めるため，用語の厳密な定義を与えないこともある．

線形性と非線形性

次の写真を見てみよう．

これはマッターホルン近くのスイスアルプスの風景と新しいモンテローザ小屋 (Monte Rosa Hut)[1] である．一見しただけで，自然の風景と人工的な小屋の幾何的な形状の明確な違いに気づくだろう．

小屋の形状は本質的に凸多面体であり，これは立方体，四面体，12 面体といった単純な数学的表現をもつ親しみある形状をしている．これら線形性や凸性をもつ物体は，ある点の高さが局所的に最も高いならば最高点であるという良い性質をもつ．さらに小屋には最高点である角の点（頂点）が常に存在する．これは，もし小屋の最高点を見つけたいならば，頂点でこれより高い頂点が周りに存在しないものを求めるだけでよいことを意味している．このことは，小屋の頂点から辺（あるいは稜）をたどりながら，高い頂点が周りにある限りそれへと移動する（ピボットする）アルゴリズムを自然に示唆する．このアルゴリズムはどんな次元でも機能し，第 1 章から第 5 章の話題である**線形最適化**[2] において最も基本的な成果のひとつである．このアルゴリズムは単体法として知られ，1947 年に George Dantzig により開発されたもので，第 4 章で扱う．単体法は実用面で効率的であることが広く認識されており，最適化の重要性はこの確かな効率性に依存すると主張しても過言ではない．実際に，非線形最適化や組合せ最適化などの難しい最適化問題を解くために，しばしば線形最適化が効率的に実行できるという事実を利用している．

一方自然な疑問として，マッターホルンの頂上は近傍内で，例えば半径 50 km 以内で最高峰なのだろうか．この疑問は，内に潜む数学的な性質について問うているのであって，地球の地理的知識を問うているのではない．この類の数学的な問題については，与えられた位置 $p = (x, y)$ の標高が関数 $f(p)$ で表現されていると仮定する必要がある．コンピュータグラフィックスで非線形な曲面を三角形を貼り合わせた面で表現する技術を用いるように，問題をより具体的にするために，この関数は区分的に線形近似されていると仮定する．このモデルにおいては，局所的な頂上点（三角形の頂点），すなわち，より高い頂点が周り

---

[1] モンテローザ小屋はチューリッヒ工科大学 (ETH Zurich) の建築家 Andrea Deplazes がデザインし，2009 年に落成した．この写真の出典は `http://www.deplazes.arch.ethz.ch/article?id=5ab8cede4258bdoc2013539015` であり，この写真の使用については建築家本人から許可を得たもので，彼に感謝する．

[2] 以前はこの分野を線形計画あるいは線形計画法 (linear programming) とよんでいたが，線形最適化 (linear optimization) とよぶことが世界的にも主流となっている．

に存在しない点を求めるのは単体法のような方法で簡単に求められる．難しい部分は，与えられた局所的な頂上が，最高峰（大域的な頂上）となっているか判定することである．自然な方法は，すべての局所的な頂上を列挙し，その中から最高峰を見つけることである．これが最高峰を見つける唯一の方法だろうか．このような方法は，領域内のすべての頂点 p の標高 $f(p)$ を評価する必要がある．関数についての追加情報がない限り，他の選択肢は望めない．なぜならば，どんな高速なアルゴリズムに対しても，それが評価していない頂点の関数値を変更することでそのアルゴリズムを欺くことができるからである．これは，**非線形最適化**[3] に内在する難しさを示していて，非線形最適化は点 p が高次元空間に存在するとき，極めて難しい（あるいは現実的に解けない）問題となる．

設定を変えて，標高関数 f が滑らかな連続関数で近似されると仮定してみる．このような関数は微分可能であり，導関数が存在する．このとき，異なるタイプのアルゴリズムを適用できる．例えば，与えられた点から最急勾配の方向に移動しながら頂上を目指す最急上昇法がそのひとつである．しかし，滑らかで連続という条件の下では，局所的な頂上はあくまで局所的なものであり，最高峰であるための良い特徴付けはなく，すべての局所的な頂上を比較することは避けられない．非線形最適化において最も重要で特殊な場合は，f が（上に）凸な場合である．この場合には，局所的な頂上は大域的な頂上（最高峰）となる．この事実は，**凸最適化**の重要な理論を導く．

ところで，皆さんは地理的な疑問の答えを知りたいだろうか．驚くことに，有名なマッターホルンの頂上はこの領域においての最高峰ではない．マッターホルンの頂上の標高は 4478 m で，一方，モンテローザの頂上（先ほどの写真では隠れていて，標高 2795 m にある小屋からはるか左上方）の標高は 4634 m で，スイスでの最高峰でもある．

本書では，非線形最適化の困難さのすべてをどのように克服するかということは議論しない．これは，別々に扱うべき多様な課題であり，本書で扱う基本的な最適化の延長上にある．

しかし，本書でも凸最適化のアルゴリズムで重要なクラス，**内点法**について議論をする．線形最適化に適用したこの解法は，ピボット演算を用いたどの方

[3] 以前はこの分野を，非線形計画あるいは非線形計画法 (nonlinear programming) とよんでいたが，非線形最適化 (nonlinear optimization) とよぶことが世界的にも主流となっている．

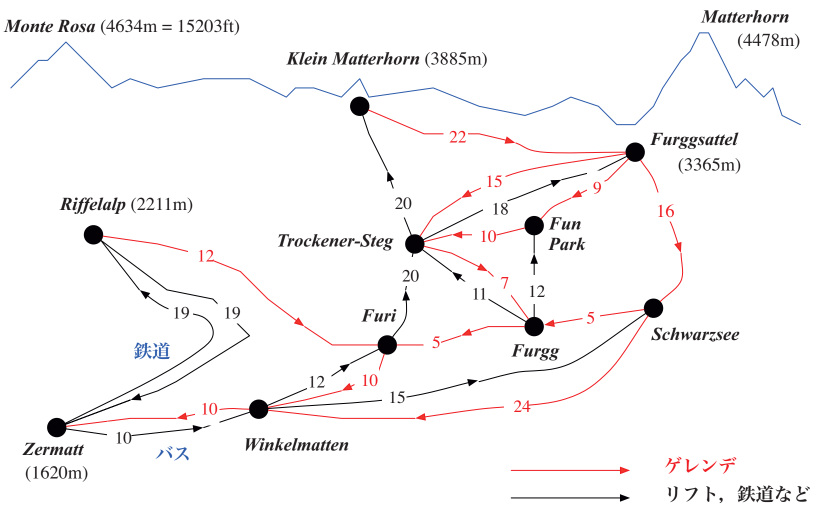

図 0.1　（単純化した）ツェルマットスキーリゾート

法よりも理論的に優れている．内点法については，第 11 章で議論する．

日常生活の中での組合せ的側面

モンテローザ小屋の数百メートル下に，スキーヤーや登山愛好家にとって有名なパラダイスであるツェルマットスキーリゾート (Zermatt Ski Resort) が，広大な範囲を占めている．全エリアは変化に富み，数えきれないほどのゲレンデがある．そのため，リフト，ロープウェイや鉄道といった多くの移動手段が用意されている．図 0.1 は，このリゾートの簡略図である．

実際，本当のリゾートは 5 倍も大きことを想像して欲しい．スキーのためにこのリゾートを訪れた者にとって，良いスキープランを作成するのは難題である．最適なスキープランを見つけるという一般的な問題が自然と導かれる．

最適の意味を明確にするために，図 0.1 は，（仮想スキーヤーの分単位の）滑走時間を表す数値と同様に鉄道などでの移動時間を表す数値を含んでいる．もちろん，これらの数値は天候，雪の状態，混雑度などに依存し変化する．簡単のために，これらの数値は正しいとする．

ここでは，このリゾートに不慣れなスキーヤーが解きたいと考えられる簡単な最適化問題を扱う：

(i) Zermatt をスタートし，すべてのゲレンデを滑って Zermatt に戻る最短時間はどうなるか.

簡略化した問いとして

(ii) 4 時間未満ですべてのゲレンデを滑って Zermatt に戻れるか.

第 1 の問題 (i) は典型的な**最適化問題**，すなわち，いくつかの制約の下である種の目的尺度を最小化（あるいは最大化）する問題である．第 2 の問題 (ii) は答えがイエスかノーという違いがある．このタイプの問題は**判定問題**として知られている.

最適化問題 (i) は，本質的に**組合せ的**であり，その**実行可能**解（スケジュール）は，ゲレンデ，リフト，ロープウェイや鉄道などの列で，すべてのゲレンデを含むものである．実行可能解が**最適**であるとは，すべての実行可能解の中で総時間を最小にするものである．線形最適化や非線形最適化と異なり，すべての実行可能解の集合は，2 つの実行可能解の中点が実行可能解とは限らないという離散集合となる．標高最大化問題では，与えられた領域（半径 50 km の円盤）内の任意の場所が実行可能であり，最適解の候補となる．これが，問題 (i) を**組合せ最適化**問題とよぶ所以である.

与えられた最適化問題，例えば (i) に対して，簡略化した問題 (ii) は対応する判定問題として知られている．すなわち，与えられた定数 k に対して，

(ii′) 目的関数値が k より小さい実行可能解が存在するか

を問うものである．目的関数を最大化する最適化問題の場合には，‘小さい’を‘大きい’に置き換える．この問いの重要性は，与えられた実行可能解の目的関数値が k であるとき，これが最適かどうかを決定できることである.

もし (ii′) に対する答えがイエスであるとき，目的関数値が k より小さい実行可能解はその**証拠**となる．証拠とは，答えの正しさを確認できる情報のことである．証拠が**良い**とは，確認が高速に行える，より正確には，確認に要する時間が問題の**入力長**に関する**多項式関数**で抑えられることである．問題 (ii) の場合には，総時間が 4 時間より短い実行可能スキープランが正に証拠となり，与えられたスキープランがすべてのゲレンデを滑るかと総時間が 4 時間未満であ

るかは簡単に確認できる．判定問題について，答えがイエスである入力例に対して常に良い証拠が存在するとき，この問題はクラス NP に属するといわれる．最適化問題に対応する判定問題がクラス NP に属するとき，元の最適化問題もクラス NP に属するという．スキープラン最適化問題のタイプに対応する判定問題はクラス NP に属する．

もし (ii′) の答えがノーであるとき，何が良い証拠となるだろうか．特殊はスキープラン (ii) の場合には，4 時間未満ですべてのゲレンデを滑るプランが存在しない場合に，何が良い証拠となるだろうか．皆さんも想像できると思うが，一般的に実行可能解（さらに追加条件を満たすもの）が存在しないことを証明するのは簡単ではない．判定問題について，答えがノーである入力例に対して常に良い証拠が存在するとき，この問題はクラス co-NP に属するといわれる．最適化問題に対応する判定問題がクラス co-NP に属するとき，元の最適化問題もクラス co-NP に属するという．NP と co-NP という概念は計算量理論において中心的な主題であり，第 6 章で議論する．様々な組合せ最適化問題を第 7 章から第 10 章で議論する．

スキープラン最適化問題のタイプに対応する判定問題が co-NP に属するか，すなわち NP と co-NP の両方に属するかは自明ではない．これらの事実はスキープラン最適化問題に対する効率的なアルゴリズムを開発するための助けとなり，第 7 章で議論する．スキープラン最適化問題は，実用上重要とは思えないかもしれない．しかし，ネットワーク内のいくつかの固定された部分をできる限り速く訪問するというこのタイプの問題は，郵便配達人の手紙の配達や顧客への弁当の配送など多くの現実的な場面で現れ，これらの問題は本質的に同じ効率的アルゴリズムで解くことができる．

NP ∩ co-NP に含まれる問題とは異なり，このクラスに属さないように思える問題も存在する．NP の中で最も難しい問題のクラスは次の性質をもつ：このクラスの任意の 1 つの問題が co-NP に属することが証明されたら NP に含まれるすべての問題が co-NP に含まれる．この最も難しい問題のクラスは NP 完全とよばれている．NP 完全な最適化問題については，与えられた実行可能解よりも良いものが存在しないことを証明する簡単な方法は知られていない．この状況は一般の非線形最適化と類似していて，例えば与えれらた局所的な頂上が最高峰であることを示すための良い方法がない最高峰問題と同じである．NP

完全な問題を解くためには，分枝限定法（第 8 章），列生成法（第 9 章）や近似アルゴリズム（第 10 章）など多くの異なるアプローチが存在する．

本書の構成

先ほど説明したように，本書は主に 2 つの主題，(1) 線形最適化と (2) 組合せ最適化を扱う．また，(3) 非線形最適化技法に関する限られた考察も与える．(1) の内容の学習は (2) と (3) の両方を理解するために必要であるが，(2) と (3) の話題は独立であり，別々に読むことができる．

以下は本書の主な内容と講義でどのように使えるかを概説する．

(1) **線形最適化**：第 1 章，第 2 章と第 3 章で，線形最適化の基本理論が最小限の数学用語を用いて与えられる．これらの章では，最も重要な理論的結果を証明抜きで与え，学部生あるいは文系を専門とする院生に対して計算機ソフトウエアを用いて教えるために利用できる．

　第 4 章と第 5 章では，線形最適化に対する 2 つのアルゴリズムをその有限終了性の厳密な証明を含めて与える．これらのアルゴリズムは十文字法と単体法である．これらの章を理解するためには，線形代数と証明技法に関する習熟が必要である．

(2) **組合せ最適化**：第 6 章では，計算量理論とともに多くの組合せ最適化問題を紹介する．クラス P, NP, co-NP や NP 完全という概念はそこで議論される．この章と線形最適化の理論の章 (1) は引続く第 7 章から第 10 章では必須である．

　第 7 章は，多項式時間可解な場合，すなわちクラス P に属する問題を扱う．特に割当問題，最小全域木問題とマッチング問題が議論される．この章は，計算量理論の威力を理解するために極めて重要である．

　第 8 章から第 10 章は，NP 完全あるいはさらに難しい問題に対するいくつかのアプローチを与える．第 8 章では分枝限定法を扱うが，この解法は広範な難しい問題に適用可能である．第 9 章では，入力データが陽に与えられていない大規模線形計画問題を実用的に解くための列生成法を議論する．この章では，第 8 章での技法を利用する．

第 10 章では，難しい問題に対して近似解を求めるための様々な技法を紹介する．この章は，第 8 章と第 9 章には依存しない．

(3) **非線形最適化**：ここでの目的は，線形最適化に対する多項式時間解法を構築するための非線形最適化の技法を学ぶことである．第 11 章では，線形最適化に対するいわゆる主双対内点法の主な構成要素を紹介する．この章は，第 7 章から第 10 章とは独立である．

3 つの主題の数学的定式化

3 つの最適化に関する主題の数学的記述を与えよう．以下では，m, n を自然数とし，A を $m \times n$ 有理行列，b を m 次（有理）列ベクトル，c を n 次（有理）列ベクトルとする．

1. 線形最適化
   線形最適化において最も基本的な問題である線形計画問題[4]は以下のように記述される．

$$
\begin{array}{lll}
\text{最大化} & c^\top x & (x \in \mathbb{R}^n) \\
\text{制　約} & Ax \leq b & \\
 & \quad x \geq \mathbf{0}. &
\end{array}
$$

   この問題は，非常に効率的なアルゴリズムを用いて解け，実用上は問題規模の制限はなく，双対定理が中心的な役割を演ずる．第 1 章から第 3 章は証明なしで線形計画問題に関する基本理論を与え，第 4 章と第 5 章は 2 つのアルゴリズム，十文字法と単体法の数学的に正確な記述，厳密な証明と幾何学的解釈を与える．第 12 章では，フリーソフトウエア LP_solve のインストールと使用例を簡単に紹介する．

2. 組合せ最適化
   組合せ最適化問題の多くは以下のように記述できる．

[4] 線形最適化という分野では，線形計画問題以外に線形相補性問題なども扱う．そのため，本書では線形計画問題を‘線形最適化問題’とはよばず，従来通りに線形計画問題とよぶ．

$$\begin{array}{ll} \text{最大化} & c^{\top}x \\ \text{制　約} & x \in \Omega. \end{array}$$

ただし，Ω は離散集合で，例えば

$$\Omega = \{x \in \mathbb{R}^n : Ax \le b,\ x_j \in \{0,1\}\ \forall j\}$$

あるいは与えられたグラフ G に対して

$$\Omega = \{x \in \mathbb{R}^n : x \text{ は } G \text{ のある条件件を満たす辺の集合を表す }\}$$

と記述され，‘ある条件’ は G の固定された 2 頂点を結ぶパスのように適切に定義されなければならない．

上記の第 1 の例は 0/1 **整数線形計画問題**の場合で，第 2 の例は**グラフ最適化問題**あるいは**ネットワーク最適化問題**の典型である．

組合せ最適化は，易しい問題も難しい問題も，すなわちクラス P（多項式時間可解な問題クラス）の問題も NP 完全な問題も両方を含む．0/1 整数線形計画問題は NP 完全であることが知られていて，これは本書の範囲を超えた整数線形計画問題に関する理論の重要な結果である．第 7 章から第 10 章では，多くのグラフ／ネットワーク最適化問題の難しさをどのように判定するかを学び，どのように適切な技法を選択するかを学ぶ．

3. **非線形最適化**

与えられた実数値関数 $f, g_i : \mathbb{R}^n \to \mathbb{R}\ (i = 1, \ldots, m)$ に対して，非線形最適化問題あるいは非線形計画問題は以下のように記述される．

$$\begin{array}{ll} \text{最大化} & f(x) \\ \text{制　約} & g_i(x) \le 0 \quad (i = 1, \ldots, m). \end{array}$$

関数の凸性は重要な役割を演ずる．内点法は，凸最適化問題や線形計画問題を含むうまく定式化された非線形最適化問題を解く．第 11 章では，線形計画問題に適用した多項式時間内点法に注意を集中する．

本書の利用法

本書の理想的な対象は，線形代数と証明技法に習熟した学生である．このグ

ループ（グループ I）は本書から最も多くの恩恵を受けるだろう．しかし他のグループも本書を利用でき，最大限の恩恵を受けるために重要なのは何を教え，何を省略するかを知ることである．線形代数を実際に利用することを目的とし，厳密な証明なしに学んだ学生をグループ II とよび，線形代数をあまり学んでいない，典型的には理系でない学生をグループ III とよぶことにする．それぞれのグループに対して具体的な助言を与えよう．

［グループ I］第 1 章から第 11 章までは章立ての通りに本書を利用し，必要に応じて第 12 章を参考にする．一学期ですべての内容を教えられるだろうが，もし時間的余裕がないならば教員は第 10 章か第 11 章の一方を省略するか，これらの簡単な紹介にとどめるとよいだろう．その他の章とは異なり，第 11 章は解析／微積分に関する概念を利用している．これは，非線形最適化の雰囲気を学生に与え，彼らが発展的内容の授業を後に受けるかどうを判断できるようにするためのものである．よって，解析／微積分の必修科目を受けている前提がないならば，期末試験においては第 11 章を外すとよいだろう．

［グループ II］第 1 章から第 11 章までは章立ての通りに本書を利用し，必要に応じて第 12 章を参考にする．いくつかの証明は省略してもよいが，弱双対定理（定理 2.2）のような簡単な証明は教えることを試みる．証明が難しい定理，例えば強双対定理（定理 2.3）については，教員は証明の概要を与えることができる．オープンソースでも商用でも利用可能な線形計画問題に対するソフトウエアを上手く利用し，これらを必要とする演習問題を与える．教員は第 10 章と第 11 章を証明抜きで学生に教えたいと思うかもしれないが，これらは期末試験から外すことができる．

［グループ III］最初の 3 章（線形最適化の導入と基礎）と第 12 章のみ利用する．最初の 3 章は，行列とベクトル以外の線形代数の知識なしに線形最適化の主結果を理解できるように書かれている．第 12 章で紹介する LP_solve やオープンソースでも商用でも利用可能な線形計画問題に対するソフトウエアを上手く利用し，学生が簡単な線形計画問題を解くことを通して線形最適化の基本概念を理解できるようにする．学生が主最適解と双対最適解を

正しく求める方法を学び，感度分析の意味を説明できるようになることが重要である.

目　次

はじめに iii

第 1 章　線形最適化の紹介 1
- 1.1 線形最適化の重要性 1
- 1.2 例 2
- 1.3 線形計画問題 4
- 1.4 線形計画問題を解く：これは何を意味するのか 6
- 1.5 線形計画法／線形最適化の歴史 8
- 1.6 演習問題 11

第 2 章　線形計画問題の基礎：その 1 13
- 2.1 最適性の判定法 13
- 2.2 双対問題 15
- 2.3 実行不可能性の判定法 19
- 2.4 非有界性の判定法 20
- 2.5 いくつかの形式の双対問題 21
- 2.6 演習問題 23

第 3 章　線形計画問題の基礎：その 2 25
- 3.1 双対問題の解釈 25
- 3.2 演習（感度分析の準備） 27
- 3.3 感度分析 27
- 3.4 演習問題 31

第 4 章　アルゴリズム 33
- 4.1 行列表記法 34

4.2 線形計画問題の辞書形式 36
4.3 ピボット演算 46
4.4 ピボットアルゴリズムと構成的証明 49
4.4.1 十文字法と強双対定理の証明 50
4.4.2 実行可能性と Farkas の補題 55
4.4.3 単体法 56
4.5 ピボット演算の実行例 60
4.5.1 単体法における巡回の例 60
4.5.2 単体法（第 2 段階）のシャトーマキシム生産問題への適用例 62
4.5.3 十文字法のシャトーマキシム生産問題への適用例 62
4.5.4 十文字法の巡回例への適用例 63
4.5.5 Bland の規則を用いた単体法の巡回例への適用例 64
4.6 ピボットアルゴリズムの図解 64
4.6.1 単体法（第 2 段階） 65
4.6.2 十文字法 66
4.7 演習問題 66

第 5 章 線形計画問題：発展 69

5.1 ピボット演算の実装 69
5.2 感度分析の計算法 71
5.3 ピボットアルゴリズムの双対化 73
5.4 単体法のピボット規則 73
5.5 ピボット演算の幾何学的理解 76
5.5.1 単体法の幾何学的観測 76
5.5.2 アルゴリズムの 3 つのパラダイム 80
5.5.3 形式的議論 81
5.6 演習問題 87

第 6 章 組合せ最適化と計算量 89

6.1 例 89

6.2 計算の効率性　94
6.2.1 問題の難しさの評価 95
6.2.2 簡単な歴史 98
6.3 グラフ理論の基本概念　99
6.3.1 グラフと有向グラフ 99
6.4 演習問題　105

第7章 多項式可解問題 —— 109

7.1 最小全域木問題　109
7.2 2 部完全マッチング問題　111
7.3 割当問題　113
7.4 最適マッチング問題　117
7.5 最大流問題　121
7.6 最小費用流問題　122
7.6.1 演習（小さなネットワーク流問題） 123
7.7 演習問題　124

第8章 しらみつぶし探索と分枝限定法 —— 127

8.1 分枝限定法　127
8.2 演習問題　134

第9章 板取り問題と列生成 —— 135

9.1 板取り問題　135
9.2 単体法の復習　137
9.3 列生成　139
9.4 演習問題　140

第10章 近似アルゴリズム —— 141

10.1 集合被覆問題　141
10.2 貪欲法　144
10.3 主双対法　146
10.4 LP 丸め法　148

第 11 章　線形計画問題に対する内点法 ―― 150

11.1　記号の準備　151
11.2　ニュートン法　153
11.3　主双対内点法　156
11.3.1　F に直接的に適用されたニュートン法 157
11.3.2　中心パス 159
11.3.3　多項式計算量 162
11.3.4　実用において多項式計算量はどの程度重要か 164

第 12 章　フリーソフトウエアを使ってみよう ―― 165

12.1　LP_solve のインストール　165
12.2　LP_solve での問題の記述　166
12.3　LP_solve で線形計画問題を解く　167
12.4　LP_solve で整数計画問題を解く　170

付録　最適化に関する有益なリンクとソフトウエアサイト ― 172

演習問題の解答 ―― 175

参考文献 ―― 188

索　引 ―― 192