

# 最適化の基礎

博士(工学) 遠藤 靖典 共著  
工学博士 宮本 定明

コロナ社

# ま え が き

本書は、筑波大学理工学群工学システム学類3年を対象として開講している「システム最適化」において、筆者らが用いてきたテキストを補筆したものである。1章から3章は遠藤が、4章から6章は宮本が担当した。

最適化の分野は長い歴史をもち、多くの研究が行われている。本書では、線形計画法、非線形計画法、組合せ最適化問題の3種類の最適化を扱うが、これらに関しても、理論面・実用面ともに非常に深化しており、関連文献も数多い。そこで本書では、前述のように大学3年への講義を念頭において、数理計画法におけるシンプレックス法、非線形計画法におけるラグランジュの未定乗数法、組合せ最適化におけるメタ戦略といった代表的な手法はもちろんだが、それらを裏付ける数学的定理についても、過不足なく、過度に詳細な記述にならないように配慮しつつ記述した。特に5章は、昨今のビッグデータ解析をはじめとした機械学習における重要性に配慮し、データ解析と最適化の関連を理論面から言及したものとして、本書の大きな特徴となっている。

本書の章末問題には解答がついていない。これは、講義での演習として出題するため、という理由もあるが、わからなければ何回も本書を読み直して、解けるまで考えてほしいからである。解答がわからないと解く気がしない、ということはあるが、本書を読めば解ける問題ばかりなので、ぜひ取り組んでほしい。

本書を執筆するにあたり、適切なお助言をいただいただけでなく、執筆の遅れに対して忍耐強く接してくださったコロナ社に心から感謝する。

2018年1月

遠藤 靖典・宮本 定明

# 目 次

## 1. 最 適 化

1.1 「最適化」の意味 .....	1
1.2 最適化問題の記述 .....	4
1.2.1 記 号 .....	4
1.2.2 最適化問題の記法 .....	5
1.2.3 大域的最適解と局所的最適解 .....	7
1.2.4 数 理 計 画 法 .....	8
1.3 最適化問題の分類と数学との関連 .....	9
章 末 問 題 .....	10

## 2. 線 形 計 画 法

2.1 生産計画問題と栄養問題 .....	12
2.2 図 的 解 法 .....	14
2.3 標準形と行列表現 .....	16
2.3.1 標準形への変換 .....	16
2.3.2 等式標準形の行列形式による表現 .....	20
2.4 線形計画法の基本定理 .....	24
2.4.1 基 本 定 理 .....	24
2.4.2 基本定理の幾何学的考察 .....	27
2.5 シンプレックス法 .....	28

2.6	2 段 階 法	34
2.7	双 対 性	38
2.7.1	双対性の基本定理	38
2.7.2	双対性の解釈	43
2.8	内 点 法	45
	章 末 問 題	49

### 3. 非線形計画法

3.1	凸集合と凸関数	51
3.2	最適化における凸関数の意義	55
3.3	2 次 形 式	57
3.4	微分可能な関数の最適性と凸性の判定条件	58
3.5	制約のある問題における最適性の必要条件	61
3.6	非線形最適化のための計算法	64
3.6.1	非線形方程式の近似解法	65
3.6.2	区間縮小法による 1 次元探索	67
3.6.3	最急降下法	70
	章 末 問 題	71

### 4. 組合せ最適化問題

4.1	重み付きグラフ	75
4.1.1	最 短 路 問 題	77
4.1.2	最 小 木 問 題	82
4.1.3	アルゴリズムの考え方	85
4.2	発見的解法とメタ戦略	86

4.2.1 グリーディアルゴリズム .....	87
4.2.2 メタ戦略 .....	88
4.2.3 メタ戦略の諸手法 .....	89
章末問題 .....	95

## 5. 最適化のデータ解析への応用

5.1 線形回帰 .....	97
5.1.1 回帰直線 .....	97
5.1.2 多次元の場合 .....	99
5.2 主成分分析 .....	100
5.3 自動分類 .....	103
5.3.1 線形分離できない場合 .....	109
5.3.2 カーネル関数を用いた非線形分類境界 .....	111
5.4 クラスタ分析 .....	113
5.4.1 $K$ -means と交互最適化 .....	116
5.4.2 Fuzzy $K$ -means 法 .....	118
章末問題 .....	120

## 6. 補足：NP 完全性について

引用・参考文献 .....	127
索引 .....	129

# 1 | 最適化

## 1.1 「最適化」の意味

本書の本題に入る前に、本書の題目である「最適化」について述べよう。  
まず、「最適 (optimal, optimum)」を辞書で引くと

(名詞・形容動詞) 最も適している・こと (さま)。 (大辞林, 三省堂)

と出る。「化」は

(接尾) 主に漢語の名詞に付いて、そういう物, 事, 状態に変える, または変わるという意を表す。 (大辞林, 三省堂)

なので、それらから考えると「最適化 (optimization)」は

最も適している物, 事, 状態に変える, または変わる。

という意味になりそうだが、「最適化」を改めて辞書で引くと

システム工学などで、ある目的に対し最も適切な計画を立て設計すること。  
また、そのような選択を行うこと。 (大辞林, 三省堂)

と出てくる。つまり「最適化」の対象は、対象をシステム工学をはじめとした工学に限定されており、ここでいう「適切」とは、対象が一般的な工学の場合、

「高効率」の意味で解される。ひるがえって工学的な立場からみると、「最適化」は「システムを最適化すること」の意味で用いられており、「システム最適化」の略としての「最適化」といっても過言ではない。すなわち本書の目的は「システムを最適化する手法」について説明することである。

そこでまず、「最適化」についてもう少し詳細に述べる。

ここで、一般的な工学における「最適化」について、いくつか注意しておこう。

まず第一に、最適化は必ずしも定式化 (formulation) されるとは限らない。例として例 1.1 を挙げよう。

---

### 例 1.1

1. ある工場で、品質管理のためにいくつかの工夫を行ったところ、統計的に明らかな効率改善がみられた。これは、定式化されない最適化の一種である。
2. 熟練したプログラマーは、処理系の性質を考慮して、計算効率がよいようにプログラミングを行う。

---

第二に、最適化の定式化にはさまざまな種類がある、すなわち、いわゆる最適化問題のカテゴリーに属さないものも多い。例として例 1.2 を挙げよう。

---

### 例 1.2

1. プログラムをコンパイル時に最適化 (効率化) することができる。
2. アルゴリズムにおける最適性： $n$  個のデータをソートする際、ヒープソートの計算量は  $O(n \log n)$  である。しかもこれよりオーダーを低くすることはできない。よって、 $O(n \log n)$  は最適なオーダーである。
3. ニューラルネットワークなどの学習では、最適性の概念が explicit (明示的) あるいは implicit に使われている場合がよくある。ホップフィールドネットワークは前者の例であり、バックプロパゲーション

ンによる学習は後者の例である。

第三に、最適化問題には、時間を表す変数が明示的にはならない静的 (static) な最適化問題と時間を表す変数が明示的に含まれる動的 (dynamic) な最適化問題がある。

では「最適化」に出てくる「システム工学」とはなにかというと、やはり辞書によれば

ある目的のための組織体系であるシステムの分析・開発・設計・運用などを合理的に行うための総合的技術。 (大辞泉, 小学館)

である。細かくいえば、分析・開発・設計はシステム工学、運用はオペレーション・リサーチという分野になるが、それらをまとめてシステム工学といって差し支えない。

一般的な工学の枠で最適化を論じるときには、前述の「最適化」の意味で用いられるが、システム工学という限定した枠の中ではより狭義、すなわち「最大化」または「最小化」の意味で用いられることが多い。

本書で扱う最適化も、与えられた問題に対する最大化 (maximization)、または最小化 (minimization) について述べる。すなわち、本書ではまずある問題が与えられ、その問題を与えられた条件下で最大化、または最小化するための方法について説明していく。与えられる問題の最大化、または最小化について述べるので、問題は数学の語法、すなわち関数の形で記述されている必要があり、結果として、最適化で述べられることは

与えられた関数を、与えられた条件下で、最大化または最小化するための数学的方法

となる。「与えられた関数を、与えられた条件下で、最大化または最小化する」問題を最適化問題 (optimization problem) という。ではつぎに、最適化問題を考えていくために必要な数学の記号と記述について説明していこう。



## 1.2 最適化問題の記述

ここでは、本書を読み進めていくのに必要な数学の記号と、最適化問題の記述について説明する。

### 1.2.1 記 号

以下、本書で用いる基本的な記号について列挙しておく。

- $\mathfrak{R}^n$  :  $n$  次元ユークリッド空間を表す。他に  $\mathfrak{R}^m, \mathfrak{R}^p$  などを用いる。
- $x$  :  $\mathfrak{R}^n$  の要素, すなわち  $x \in \mathfrak{R}^n$  である。成分は  $^\top$  を転置記号として, つぎのように表す。

$$\mathbf{x} = (x_1, \dots, x_n)^\top = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

- $f(\mathbf{x}), g(\mathbf{x})$  :  $\mathbf{x}$  の実数値関数, すなわち  $f(\mathbf{x}) \in \mathfrak{R}, g(\mathbf{x}) \in \mathfrak{R}$  である。一般に,  $h$  が空間  $X$  で定義され, 空間  $Y$  に値をとる写像のとき,  $h : X \rightarrow Y$  と書く。したがって,  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}, g : \mathfrak{R}^n \rightarrow \mathfrak{R}$  である。 $f(\mathbf{x})$  は後に述べる目的関数あるいは評価関数,  $g(\mathbf{x})$  は制約関数の意味で用いられることが多い。
- $F(\mathbf{x}), G(\mathbf{x})$  : ベクトルの値あるいは行列の値をとる関数。これに対して前述の  $f(\mathbf{x}), g(\mathbf{x})$  はスカラー値の値をとる関数である。
- $\nabla f$  :  $f$  のグラディエント (gradient) を表す。ここで  $\nabla$  はナブラという。行ベクトルであることに注意する。

$$\nabla f = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)$$

- $\nabla^2 f$  :  $f$  のヘッセ行列 (Hessian matrix) を表す。ラプラシアンではないことに注意する。

$$\nabla^2 f = \left( \frac{\partial^2 f}{\partial x_i \partial x_j} \right) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

- $\|\mathbf{x}\|$  :  $x \in \mathfrak{R}^n$  のユークリッドノルムを表す。すなわち

$$\|\mathbf{x}\| \stackrel{\text{def}}{=} \sqrt{x_1^2 + \cdots + x_n^2} = \sqrt{\sum_{i=1}^n x_i^2}$$

ここで、 $\stackrel{\text{def}}{=}$  は「左辺を右辺と定義する」という意味の記号である。

### 1.2.2 最適化問題の記法

前述のように、最適化問題とは「与えられた関数を、与えられた条件下で、最大化または最小化する」問題である。「与えられた関数」を目的関数 (objective function) といい、「与えられた条件」を制約条件 (constraint), 制約条件を表す関数を制約関数 (constraint function) という。目的関数を  $f$ , 制約関数を  $g_i$  ( $i = 1, \dots, m$ ),  $h_j$  ( $j = 1, \dots, l$ ) としたとき、最適化問題は

制約条件  $g_i(\mathbf{x}) \leq 0$  ( $i = 1, \dots, m$ ) および  $h_j(\mathbf{x}) = 0$  ( $j = 1, \dots, l$ ) のもとで、目的関数  $f(\mathbf{x})$  を最小化 (または最大化) せよ。

となる。ただし、「 $f(\mathbf{x})$  の最大化」は、 $f(\mathbf{x})$  の符号を変えることにより、「 $-f(\mathbf{x})$  の最小化」と変換できるので、一般に最適化問題を考えるときには、最大化と最小化のどちらかを考えればよい。そこで以下、本書では特に明記しない場合、最小化の場合の最適化問題について述べることとし、最適化問題の表現として、以下の3種類を扱う。

(P1) 目的関数の最小化を

$$\begin{aligned} & \text{minimize } f(\mathbf{x}) \\ & f(\mathbf{x}) \rightarrow \min \end{aligned}$$

# 索引

<b>【い】</b>		<b>【く】</b>		サイズ	75
遺伝的アルゴリズム	91	区間縮小法	67	最大化	3
<b>【う】</b>		組合せ最適化問題	9, 73	最短路	85
ウォーク	76	クラス NP	121, 123	最短路問題	77
<b>【え】</b>		クラス P	121	最適	1
栄養問題	13	クラスカルのアルゴリズム	83	最適化	1
エピソード	54	クラスター中心	116	最適解	7
<b>【お】</b>		クラスター分析	113	最適化問題	3, 125
凹関数	53	グラディエント	4	最適実行可能基底解	25
オーダー	75	グリーディアルゴリズム	85, 87	サブグラフ	76
重み付きグラフ	75	<b>【け】</b>		サポートベクトルマシン	104
<b>【か】</b>		計算のモデル	122	<b>【し】</b>	
カーネル関数	111	計算複雑さ	121	実行可能	7, 25
回帰直線	97, 99	計算量	81, 85	実行可能解	7
完全グラフ	76	<b>【こ】</b>		実行可能基底解	25
<b>【き】</b>		交互最適化	116	実行可能集合	6
木	76	交叉	92	実行可能領域	6
基底解	25	高次元写像	111	実行不可能	7
基底変数	25	互換	89	自動分類	103
狭義凹関数	53	コスト	77	充足可能性問題	121
狭義凸関数	53	<b>【さ】</b>		主成分分析	100
教師なし分類	113	最急降下法	70	主問題	38, 108
局所最適解	7	最近隣分類	104	巡回セールスマン問題	74, 89, 93
局所探索	89	サイクル	76	進化計算	95
局所最適解	89	最小化	3	人工変数	35
近傍探索	88	最小木	82, 85, 87	シンプレックスタブロー	28
		最小木問題	82	シンプレックス法	28
		最小二乗法	98	<b>【す】</b>	
		最小スパニングツリー	82	スーパーグラフ	76
				数理計画法	8
				スラック変数	19

<b>【せ】</b>		<b>【つ】</b>		<b>【ふ】</b>	
生産計画問題	12	ツリー	76	不等式標準形	19
正準形	16, 18			部分グラフ	76
静的	3	<b>【て】</b>		分類	103
静的最適化問題	9	データ解析	97	<b>【へ】</b>	
制約関数	5	データマイニング	97	ヘッセ行列	4
制約条件	5			辺	75
制約付き最適化問題	10	<b>【と】</b>		<b>【ほ】</b>	
制約なし最適化問題	10	動的	3	母集団	92
線形回帰	97	動的最適化問題	9	<b>【ま】</b>	
線形境界	105	凸関数	53	マージン	105
線形計画問題	10	凸集合	51	マージン最大化	105
選択	94	突然変異	92	マルチスタート局所探索	90
		凸包	52	<b>【め】</b>	
<b>【そ】</b>		貪欲算法	87	メタ戦略	86, 88
双対性	38	<b>【な】</b>		メタヒューリスティック	88
双対問題	107, 108	内点法	45	<b>【も】</b>	
<b>【た】</b>		ナップザック決定問題	122	模擬焼きなまし法	90
第1段階	35	ナップザック問題	73, 88, 93	目的関数	5
第2段階	35	<b>【に】</b>		<b>【ら】</b>	
大域的最適解	7, 89	ニュートン法	65	ラグランジュ乗数	61
ダイクストラのアルゴリズム	79	<b>【は】</b>		ラグランジュの条件	61
ダイナミックプログラミング	85	パス	76	ラグランジュの未定乗数法	63
多項式オーダー	121	発見的解法	86	<b>【り】</b>	
多項式還元可能性	124	判別	103	離散的最適化問題	9
タブー探索	91	<b>【ひ】</b>		隣接	75
タブーリスト	91	非基底変数	25	<b>【れ】</b>	
多変量解析	99	非決定性機械	123	連結	76
単体表	28	非線形計画問題	10	連結成分	76
単峰性関数	67	非線形分類境界	111	連続的最適化問題	9
<b>【ち】</b>		非退化	25		
頂点	75	ヒューリスティック	86		
		標準形	16		

<b>【C】</b>		<b>【K】</b>		<b>【S】</b>	
Cook の定理	124	<i>K</i> -means	113	SMO	109
$C^p$ 級	58	KKT 条件	61, 107	SVM	104
<b>【F】</b>		<b>【N】</b>		<b>【T】</b>	
Fuzzy <i>c</i> -means	118	NP 完全	86, 121, 124	Turing 機械	122
Fuzzy <i>K</i> -means	118	NP 困難	124	~~~~~	
				2 次形式	57
				2 段階法	35

— 著者略歴 —

遠藤 靖典 (えんどう やすのり)

1990年 早稲田大学理工学部電子通信学科卒業

1994年 早稲田大学助手

1995年 早稲田大学大学院理工学研究科博士

後期課程修了 (電気工学専攻)

博士 (工学)

1997年 東海大学講師

2001年 筑波大学講師

2004年 筑波大学助教授～准教授

2013年 筑波大学教授

現在に至る

宮本 定明 (みやもと さだあき)

1973年 京都大学工学部数理工学科卒業

1975年 京都大学大学院工学研究科修士課程

修了 (数理工学専攻)

1978年 京都大学大学院工学研究科博士課程

修了 (数理工学専攻)

工学博士

1978年 筑波大学研究専任技官 (準研究員)

1981年 筑波大学講師

1987年 筑波大学助教授

1990年 徳島大学教授

1994年 筑波大学教授

2017年 筑波大学名誉教授

## 最適化の基礎

Introduction to Optimization Methods © Yasunori Endo, Miyamoto Sadaaki 2018

2018年3月20日 初版第1刷発行



検印省略

著者 遠藤 靖典

宮本 定明

発行者 株式会社 コロナ社

代表者 牛来 真也

印刷所 三美印刷株式会社

製本所 有限会社 愛千製本所

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844 ・ 電話 (03) 3941-3131 (代)

ホームページ <http://www.coronasha.co.jp>

ISBN 978-4-339-02884-3 C3055 Printed in Japan

(大井)



< 出版者著作権管理機構 委託出版物 >

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、出版者著作権管理機構 (電話 03-3513-6969, FAX 03-3513-6979, e-mail: info@jcopy.or.jp) の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。落丁・乱丁はお取替えいたします。