第 1 回 最適化ゼミ

ゼミ担当者　：　佐藤　史隆, 平井　聡, 林　俊行
指導院生　　：　花田　良子，勝崎　俊樹
開催日　　：　2003 年 4 月 10 日

ゼミ内容: 本ゼミでは，最適化とは何かについて理解を深め，代表的な最適化手法について学習する．
なお，ここで扱うのは，決定的規則を用いた手法である．

1　序論

我々の研究室，知的システムデザイン研究室の最終的な目的は，その名の通り知的なシステムを設計することにある．そして，その研究の柱は，

- 知的化，コラボレーション
- 最適化
- 並列，クラスタ

の 3 つである．これらの 3 つのテーマは，それぞれ密接に関係している．その関係は次の通りである．まず，最適化という手法を用いることで，よりよいシステムの設計を目指すことができる．そのとき，システムの最適化を行うことになるが，それには大きな計算力が必要となるので，クラスタを用いた並列化を行う必要がある．そして，知的・コラボレーション班では，知的なシステムを構築するためのツールとして，最適化・クラスタを利用する．

この最適化ゼミでは，この研究室の 3 つの柱の 1 つ，最適化について学ぶことになる．EC ゼミで説明するように，実際の研究では,（Simulated Annealing：SA），(Genetic Algorithm：GA)，強化学習のような進化的な手法を用いているが，このゼミではもっと基礎的な部分について担当する．

2　最適化とは何か

2.1　最適化の目的

私たちの日々の生活において，費用を最小に抑えて物を作ったり，限られた資源を有効に活用するといったように，ある制約条件の中で何らかの目的を達成しなければならないことはしばしばある．これらは，工学においてもいえることであり,「制約条件を満たした上で，最も適切な計画，設計を作成し選択すること」を最適化という．

2.2　最適化問題の定義

ある条件のもとで目的とするものを最小化（最大化）するような変数を決定する問題を最適化問題 (optimization problems) と呼ぶ．目的とするものを数式化したものを目的関数 (object function)，目的を達成するうえで満足させなければならない事柄を数式化したものを制約条件 (constraint), またこれらに用いられる変数を設計変数（design variable）と呼ぶ．これらの用語を用いると，最適化の定義は以下のようになる．

「与えられた制約条件 $g(x)$ のもとで，ある目的関数 $f(x)$ を最小にするような設計変数を求めること」

また最適化問題は一般的に以下のように表される．

$$
\begin{aligned}
&Minimize \quad f(x) \quad (x \in R)\\
&Subject \quad to \quad g_i(x) = 0 \quad (i = 1, 2, 3 \cdots n)\\
&\qquad\qquad g_i(x) < 0 \quad (i = n+1, \cdots m)
\end{aligned}
$$

$f(x)$ の値を最適値 (optimal solution) と呼ぶ．ここで目的関数 $f(x)$ の最大値を求める問題であったとしても，$-f(x)$ の最小値を求める問題と解釈しなおすことができるので，この式は目的関数の最大値・最小値どちらを求める問題にも対応できる．また，制約条件 $g(x)$ についても，例えば $x^2 + 6x \leq 3$ のように上記の形式に当てはまらない式は $x^2 + 6x - 3 \leq 0$ と変換することができる．

2.3　現実世界の現象への定式化の必要性

実世界には複雑な問題がたくさん存在するが，それらはほとんどの場合，定式で存在しているわけではない．このような問題をコンピュータを用いて解くには定式化する必要がある．現象の特徴を簡潔に記述するのに必要十分なパラメータを抽出し，目的関数および制約条件とよばれる式でモデル化したならば，その現象の最適化問題として扱うことができる．その様子を Fig. 1 に示す．

現在，最適化問題をコンピュータで解くための様々なアルゴリズムが存在する．目的関数，制約条件の数学的特徴により適切なアルゴリズムを選ぶことで，コンピュータで解くことができるようになる．このようにコンピュータで現象を解析するには，モデル化，モデルに合ったアルゴリズムの決定の 2 段階のステップが必要となる．

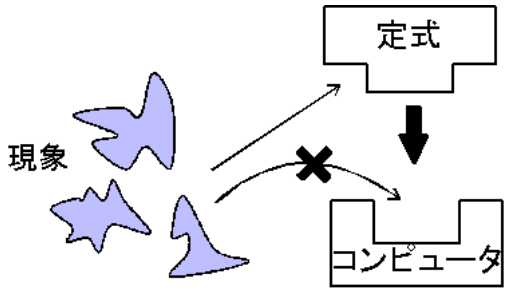

Fig. 1　現象の定式化

2.4　単峰性と多峰性の違い（大域的最適解と局所的最適解）

目的関数には様々な関数が使われる．関数には特徴があるが，その中で特に重要なのは単峰性・多峰性と呼ばれるものである．Fig. 2 と Fig. 3 はその一例である．それぞれの関数には山（凸の部分）あるいは谷（凹の部分）にあたる部分があるが，その 個数が異なる．まず，Fig. 2 には谷が 1 つしかない．このように，山あるいは谷が１つしかない関数を単峰性の関数と呼ぶ．一方，Fig. 3 は山あるいは谷が複数個存在する．このような関数を多峰性の関数と呼ぶ．多峰性の関数の場合，極小値（極大値）が複数個ある．局所的に見たときの，最小あるいは最大となるものを局所的最適解と呼び，定義域全体での最小値を大域的最適解と呼ぶ．Fig. 2,Fig. 3 において，○印は大域的最適解，×印は局所的最適解を示している．

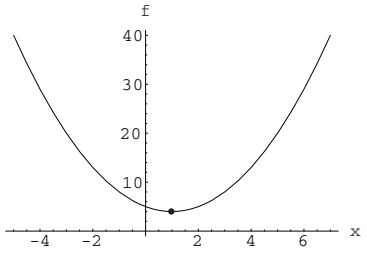

Fig. 2　単峰性の関数

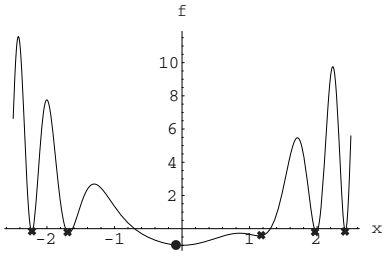

Fig. 3　多峰性の関数

3　最適化問題の例

3.1　最適化問題の例（制約条件と目的関数の違いについて）

最適解の例として，料理の栄養素問題を用いて考えてみる．

2 種類の食品 A と Bがあり，1 キログラムあたりの炭水化物，タンパク質の含有量と価格を Table 1 に示す．

Table 1　食品 1kg と必要な栄養素の関係

	炭水化物	タンパク質	価格
A	400 グラム	100 グラム	1000 円
B	200 グラム	300 グラム	1500 円

ここで炭水化物を 400 グラム以上，タンパク質を 300 グラム以上含む料理を作りたい．料理の価格を最も安くするために A と B をそれぞれ何グラムずつ使えばよいだろうか？

＜問題の定式化＞

・使用する A と B の量 (キログラム) をそれぞれ x,y とすれば，各栄養素の量に対する制約から次の 2 式が得られる．

$400x + 200y \geq 400$　（炭水化物）

$100x + 300y \geq 300$　（タンパク質）

・価格 $F(x, y)$ は，

$F(x, y)$=$1000x$+$1500y$

である．価格の式を簡略化するために 500 で割り，同様に質量に関しても炭水化物は 200, たんぱく質は 100 でそれぞれ割り，$\frac{F(x,y)}{500} = f(x, y)$ とおく．x,y は負でないという条件を付け加えると，次の問題に変換できる．

$$Minimize \qquad f(x, y) = 2x + 3y \tag{1}$$

$$Subject \quad to \quad 2x + y \geq 2 \tag{2}$$

$$x + 3y \geq 3 \tag{3}$$

$$x \geq 0, y \geq 0 \tag{4}$$

式 (1) から式 (4) はすべて線形式であるから，この問題は，線形計画問題 (Linear Programming Problem：LP) である．x,y が設計変数であり，式 (1) が目的関数，変数の範囲を限定する式 (2)〜式 (4) が制約条件である．式 (4) を特に決定変数の非負条件と呼ぶ．

この問題を 平面上で幾何的に考察すると，式 (2)〜式 (4) の条件はそれぞれ半平面を示す．その共通部分が Fig. 4 における色付けされた領域[1]である．

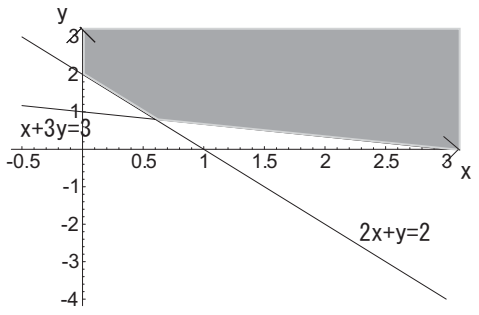

Fig. 4　制約条件のグラフ

ここで，図の上に直線 $2x + 3y = c$(定数) を描くと Fig. 5 のように並行直線群が得られる．

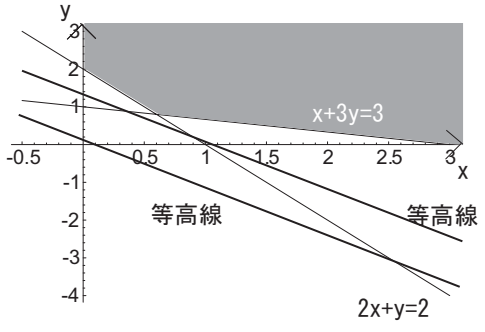

Fig. 5　目的関数が表す直線の平行移動

条件を満たす最小値は，色付け領域と少なくとも 1 つ以上の共通点をもつような直線を与える c の最小値である．この場合，c が最小となるのは Fig. 6 の破線の場合で，このときの交点が求める解となる．

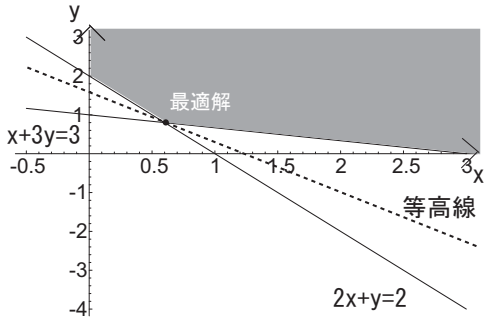

Fig. 6　求められる最適解

計算すると，座標値 $x = \frac{3}{5}, y = \frac{4}{5}$ が解で，そのとき $c = \frac{18}{5}$ となる．

[1]実行可能領域 (feasible region) という

x と y はそれぞれ A と B の量（キログラム）なので，A を 600 グラム，B を 800 グラム使用すれば最も安くすることができ，1800 円で済ますことができる．

4　最適化問題の分類

4.1　連続問題と離散問題

最適化問題は，その数学的性質から，連続最適化問題と離散最適化問題の 2 つに分けることができる．連続最適化問題とは探す値が連続的に分布している問題であり，離散最適化問題とは探す値が離散的に分布している問題である．また，連続最適化問題はさらに線形計画問題と非線形計画問題との 2 つに分けることができる．これらの最適化問題の特徴は以下の通りである．

- 連続最適化問題
  - 線形最適化問題（シンプレックス法など）
  - 非線形最適化問題（最急降下法，ニュートン法，準ニュートン法）
- 離散最適化問題（欲張り法，分枝限定法など）

Fig. 7　最適化問題の分類

a) **線形計画問題**

線形計画問題とは，制約条件と目的関数がともに線形（設計変数に関する 1 次式で表現可）であるような問題のことをいう．この線形計画問題は，一般的な表現形式として，以下のように示すことができる．これは不等式条件で制約条件が構成されているが，等式条件を交えて扱うことも可能である．

$$
\begin{aligned}
\text{目的関数}\quad & f = c_1x_1 + c_2x_2 + \cdots + c_nx_n \rightarrow min(max) \\
\text{制約条件}\quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\
& \cdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m
\end{aligned}
$$

これをベクトル表記を用いて変換すると，以下のようになる．

目的：目的関数 $f = c^T x$ を最大（最小）にすること
条件：$x \in X \equiv \{A_x \leq b\}$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad (5)$$

$$c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (6)$$

このとき，式 (5)，式 (6) に示すように，A は $m \times n$ 定数列ベクトル，b は m 次元定数列ベクトル，c は n 次元定数列ベクトル，x は n 次元変数列ベクトルである．

線形探索問題の例としては，前章の問題が挙げられる．そして, 線形探索問題の中で，もっとも一般的なものがシンプレックス法（単体法）である．前章の問題のように，2 変数の線形計画問題においては，実行可能領域は凸多角形となり，目的関数の等高線は平行な直線となるので，最適解は実行可能領域である凸多角形の境界線上に存在する．さらに，その凸多角形の頂点のうち少なくとも 1 つが最適解になっていることが分かる．これらと同等の性質は一般のｎ変数の線形計画問題に対応して成り立っている．シンプレックス法についての詳しい解説は次回に説明する．

b) **非線形計画問題**

非線形計画問題とは，目的関数または制約条件のうち少なくとも一方が非線形であるような最適化問題のことである．非線形問題の例としては,「均衡価格問題」,「ロケットの最適制御」,「化学平衡問題」などが挙げられる．線形計画法との一番の大きな違いは，微分値を取ることができるという点である．この性質を利用して最急降下法，ニュートン法，準ニュートン法などのアルゴリズムを用いることで，最適解を導くことができる．これらの手法については次回解説する．

4.2 離散最適化問題

離散最適化問題とは，目的関数や制約関数が離散的な関係を持っている場合の最適化問題のことである．離散的なものの例としては整数の集合, グラフにおけるノードなどが挙げられる. 以下が離散的最適化問題の例である.

- 「ナップサック問題」（x 以下という条件の下でより多くのものをナップサックに詰めるためにはどうしたらよいか）
- 「最短経路問題」（A → B までの最短経路を求める）
- 「巡回セールスマン問題（TSP）」（複数の都市を巡回するための最短ルートを見出す問題）

離散集合の性質としては,「連続性，微分，積分などの概念を直接適応できない」ことと,「パターンを数えることができ，総当りでそのパターンを調べれば，必ずその最適解を求めることができる」ことが挙げられる．しかし，現実世界でそれを行おうとすれば,「組み合わせ爆発」が起こるため，すべてのパターンを調べられる問題は少ない．そこで，離散最適化問題で扱うアルゴリズムは,「いかに効率よく探索できるか」,「いかにかかる時間を短縮できるか」という点がもっとも重要である．以下にこれらのアルゴリズムの例として，近似解法である欲張り法と完全列挙である分岐限定法を示す.

欲張り法

欲張り法とは,「解を段階的に構築していく際に，常にその時点で最善であると思われるものを取り入れていく」方法である．つまり「その時点での最善の解の集合体としての最適解」を目的とする方法である．もちろん，単純かつ明快なこの方法では，問題の大域的最適解を求めることは難しい．それでも，ある種の問題に対して能率よく最適解（局所的最適解）を得られる．

欲張り法を用いた例としては最小木問題が挙げられる. 最小木問題とは無向グラフの各枝の長さが与えられているとき，全節点を連結する木で，枝の長さの和が最小のものを見つける問題である. この問題の解法は長さ最小の枝から始め，選ばれた枝の集合が閉路を形成しないという条件の下で，長さの小さいものから順に n-1 本（n は節点数）選ぶ方法である．

分枝限定法

組み合わせ最適化問題の場合，実行可能解は有限個であるため，原理的にはそれらを全て列挙する事により最適解を求める事は可能である．しかし，実行可能解が多くの組み合わせを持っていると多くの計算時間を必要としてしまう．このような組み合わせ爆発を防ぐための方法が分枝限定法である．分枝限定法は実行可能解を列挙するための過程で，最適解になりえない不必要な場合分けをできる限り省略し，探索する範囲を絞り込む事により，計算時間を短縮する手法である．しかも，欲張り法とは違い「完全列挙による解法」であるため，大域的最適解が求められる事ができる．今日は，次の式を用いながら説明していく．

$$\begin{aligned} &Maximize && 3x_1 + 4x_2 + x_3 + 2x_4 \\ &Subject && 2x_1 + 3x_2 + x_3 + x_4 \leq 4 \\ & && x_1, x_2, x_3, x_4 \in \{1, 0\} \end{aligned}$$

Fig. 8 は前式を表したグラフである．Fig. 8 における解が 7 の場合は制約条件を満たしていないため「得られ

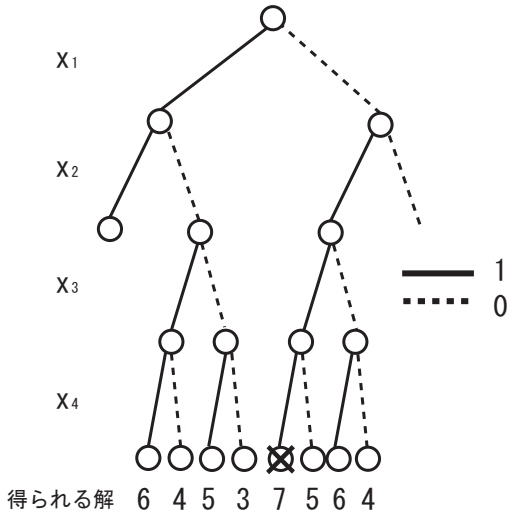

Fig. 8　分枝限定法の例

る解」として扱うことはできない．よって，最適解は 6 であることがわかる．この Fig. 8 においては x_1,x_2 がともに 1 の場合と 0 の場合は探索を行っていない．これには次の 2 つの理由がある．1 つ目の理由は制約条件を満たせなくなる場合で，x_1,x_2 がともに 1 であるならば，x_3,x_4 を探索する必要がないことがわかる．2 つ目は探索を行っても暫定値以上の解になりえない場合である．暫定値とは，探索途中での最適解のことである．x_1,x_2 を探索したときに，解が暫定値以上になる可能性を持たない場合，よりよい解は得られないため探索を終了する．この分枝限定法を使用することにより，計算時間が短縮され，効率よく最適解を求めることができる．

離散最適化問題は，工学のほとんどすべての分野で扱う重要な問題である．しかし，離散最適化問題には連続性や微分といった古典的な手法をとることができない難しさがある．