最適化ゼミ

指導 渡邉 (M2)　　佐野 (M1)
担当　チーフ：花田　サブチーフ中村，下神納木

1　最適化問題概要

1.1　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 problem) という．目的とするものを数式化したものを目的関数 (objective 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) という．これらの言葉を用いて最適化の定義を言い換えると以下のようになる．

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

$$Minimize \quad f(x) \quad (x \in R)$$
$$Subject \quad to \quad g(x) \leq 0 \quad or \quad Subject \quad to \quad g(x) < 0$$

求まった $f(x)$ の値を（最適解,optimal solution）という．
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せばよいので，この式では，目的関数の最小値を求める記述になってる．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．

1.2　現実世界の現象の定式化（モデル化）の必要性

実世界には複雑な問題がたくさん存在します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 Fig1 に示す．

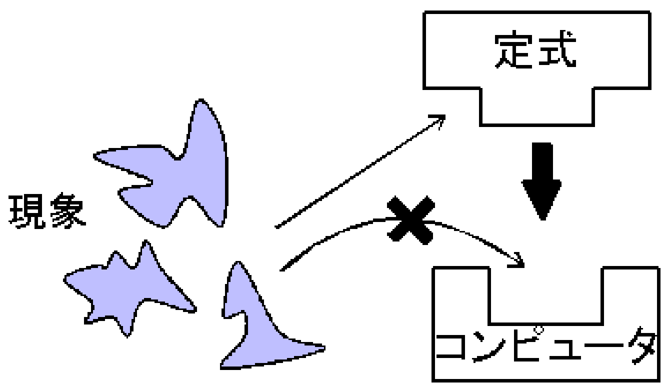

Fig. 1　現象の定式化

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と，コンピュータで解けるようになる．
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である．

1.3　単峰性と多峰性

目的関数には様々な関数が使われます．使われる関数にはそれぞれ特徴があるが，単峰性，多峰性という，コンピュータで最適化問題を扱うときに重要な特徴がある．Fig2 と Fig3 はその一例である．この 2 つのグラフを比べると違いがあるのがわかる．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，その個数が違う．Fig2 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よぶ．一方，Fig2 は山あるいは谷が複数個ある．このような関数を多峰性の関数とよぶ．多峰性の関数の場合，極小値（極大値）が複数個ある．定義域全体での最小値を大域的最適解とよび，その他の極小値を局所的最適解とよぶ．Fig2,Fig3 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．

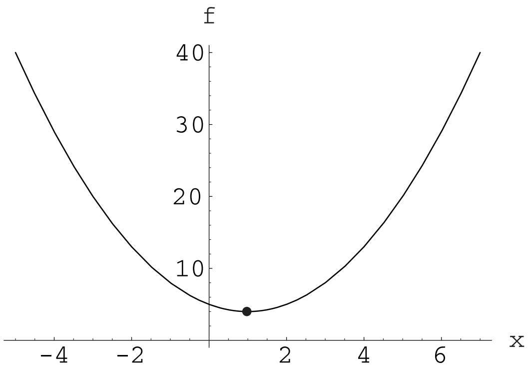

Fig. 2　単峰性の関数

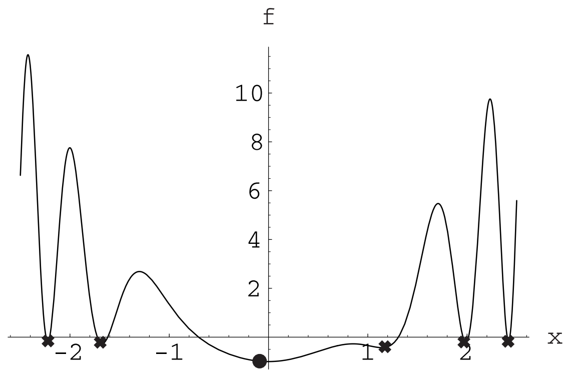

Fig. 3　多峰性の関数

2　最適化問題の例

最適化問題の例として，食品の摂取量のついて考えてみる．

食品 1, 食品 2 について，食費をできるだけ抑えたいということを最大の目的とする．そして，そのときに考慮するものは,Table 1 に示すように，栄養素 A, 栄養素 B, 栄養素 C の最低摂取量を越えることとする．そして，各食品の 100g あたりの価格はそれぞれ,200 円,300 円だとする．

Table 1　食品 100g と必要な栄養素の関係

	食品 1	食品 2	最低必要量
栄養素 A	4	2	20
栄養素 B	1	8	10
栄養素 C	6	4	16

最適化すべき目的関数の設計変数は，食品 1，食品 2 の摂取量である．これらを，それぞれ x_1, x_2 とする．そして，Table 1 から，目的関数 f を考えると以下のようになる．

$$f = 200x_1 + 300x_2 \quad \rightarrow \quad min \tag{1}$$

考えやすいように，この式の両辺を 100 で割り,$\frac{f}{100}$ を新たに f とおく．

$$f = 2x_1 + 3x_2 \quad \rightarrow \quad min \tag{2}$$

そして，式 (2) を，x_2 についての式に変換する．そして，$\frac{f}{3}$ を新たに F とおく．

$$x_2 = -\frac{2}{3}x + F \tag{3}$$

この式は，目的関数 f の等高線を表している．これから F を最小化することを考える．まず，Fig. 6 に，目的関数式 (2) の外形を示す．縦軸は目的関数 f，横軸はそれぞれ設計変数 x_1, x_2 である．

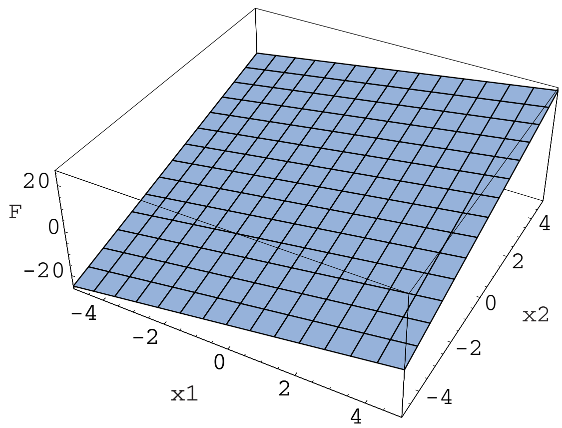

Fig. 4　目的関数のグラフ

この平面上に食品 1 と食品 2 の合計の金額がくる．それに，制約条件を加えることによって，平面のどの点で最低摂取量を満たしつつ，一番値段が安くなるのかということが分かる．それを探すために実際解くにあたっては，2 次元平面のグラフで見ると分かりやすいので，それを Fig. 5 に示す．

Fig. 5 の直線群は全て，制約条件の境界線である．これらは Table 1 から，導かれる．第一に，栄養素 A は，最低で

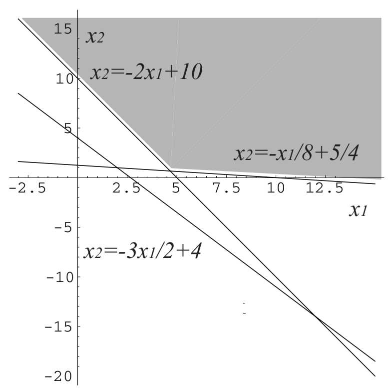

Fig. 5　目的関数のグラフと制約条件の関係

も 20 必要で，食品 1 からは 100g あたり 4，食品 2 からは 100g あたり 2 だけ摂取することができる．以下の式のように制約条件を求めることができる．

$$g_1 : 4x_1 + 2x_2 \geq 20 \tag{4}$$

同様に，栄養素 B,C に関しても，

$$g_2 : x_1 + 8x_2 \geq 10 \tag{5}$$

$$g_3 : 6x_1 + 4x_2 \geq 16 \tag{6}$$

のように，制約条件を求めることができる．制約条件から x_1, x_2 の満たす範囲は絞られるが，それは，Fig. 5 の色づけされた部分となる．

この図から，目的関数を考える．目的は，合計金額にあたる f を最小にすることで，これは，式を変形させたあとの，F を最小にするということになる．これは等高線の x_2 切片が F となるわけだが，この F を最小にするには，制約条件を満たしつつ，しかも，すなわち原点に近くなるようにしなければならないのが分かる．これを求めるには，原点をとおる等高線 $x_2 = -\frac{2}{3}x$ を平行移動させて，制約条件を満たす領域[1]に初めて交わる点を探せばよい．その様子を Fig. ??に示す．Fig. ??のグラフ上で，目的関数の直線を平行移動させて，x_1 と x_2 の範囲を満たし，かつ F が最小

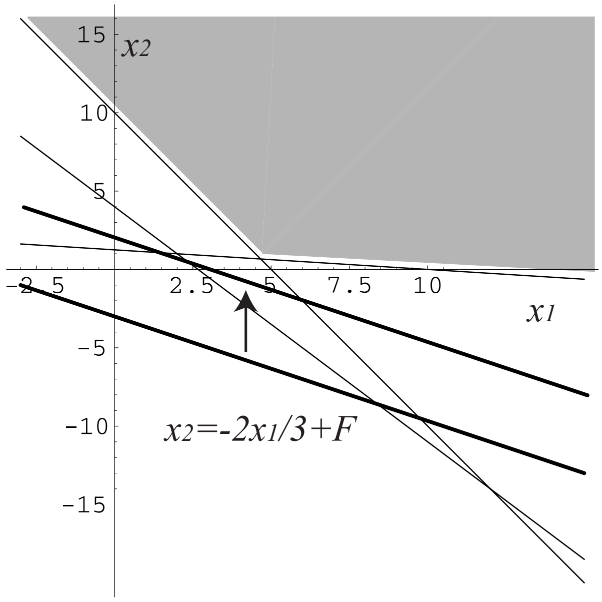

Fig. 6　目的関数が表す直線の平行移動

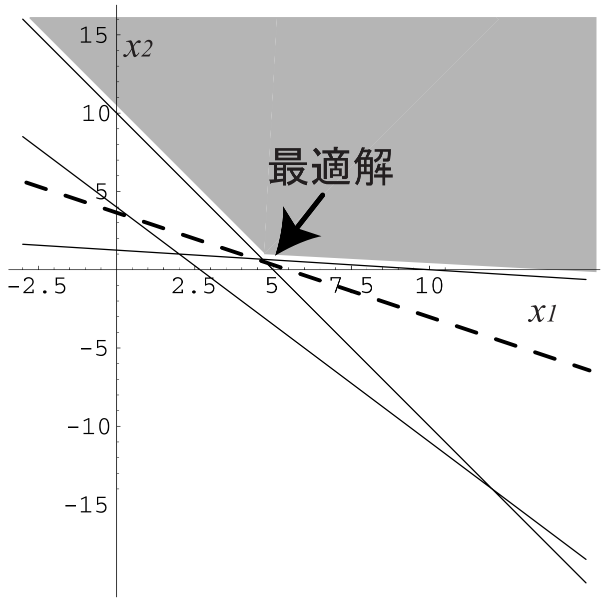

Fig. 7　求めらる最適解

となるところを探す．それによって，最適解が求められる．この問題の場合，最適解は

$$(x_1, x_2) = (\tfrac{14}{3}, \tfrac{2}{3})$$

で，Fig. 7 で表される点である．

3　最適化問題の分類とアルゴリズムの例

最適化問題は，その数学的性質から，まず，連続最適化問題と離散最適化問題に分類される．また，連続最適化問題は，さらに線形最適化問題と非線形計画問題に分類されることになる．ここでは，それぞれの最適化問題について述べていくことにする．まず，Table 2 に，最適化問題を分類した表を示す．

Table 2　最適化問題分類とアルゴ リズム

連続最適化問題	線形計画問題	探す値が連続的に分布している	シンプレックス法
	非線形計画問題		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		探す値が離散的に分布している	欲張り法，分枝限定法，SA，GA

[1] これを実行可能領域とよぶ

3.1 連続最適化問題（線形・非線形）

3.1.1 線形計画問題

線形計画問題（Linear programing，LP）とは，モデルの目的関数と制約条件が，すべて線形である（すべての変数について 1 次式であらわすことができる）ような問題のことです．前章で取り上げげた，例題も線形問題である．線形問題は，一般的な表現形式として，式 8，式 9 のようにあらわすことができる．

$$\text{目的関数}\ f = c_1x_1 + c_2x_2 + \cdot + c_nx_n \rightarrow \max(\text{or min})$$

$$\begin{aligned}\text{制約条件}\ a_{11}x_1 + a_{12}x2 + \ldots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x2 + \ldots + a_{2n}x_n &\leq b_2 \\ &\vdots\vdots \\ a_{m1}x_1 + a_{m2}x2 + \ldots + a_{mn}x_n &\leq b_m\end{aligned}$$

これを行列・ベクトル表現を用いて簡潔に表すことができる．
A, b, c, x をそれぞれ

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}, c = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

と置くと線形問題は，であるといえる．ここで，$^\top$ がでてくるが，これは転置行列を示す記号である．

制約条件 $Ax \leq b$ のもとで，目的関数 $f(x) = c^\top x$ を最小化するような変数ベクトル
$x = (x_1, x_2, \ldots, x_n)^\top$ を見いだす問題

線形計画問題に対するアルゴ リズムでもっとも一般的なものは，**シンプレックス法（*simplex method*）**である．前章の問題をみてもらうと，2 変数の線形計画問題においては，実行可能領域[2]は，凸多角形[3]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一つが最適解になっていることがわかる．さらに，これと同様の性質は一般の n 変数の線形計画問題に対しても成り立つ．

3.1.2 非線形計画問題

次に，非線形計画問題（Nonlinear programing）について説明する．非線形計画問題とは，目的関数または制約条件の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を指します．非線形計画問題の例を Fig. 8 に示しておく．縦軸は目的関数，横軸は設計変数を表す．非線形計画問題を解くアルゴ リズムには，最急降下方，ニュートン法，そして，準ニュートン法があるが，このうち最急降下法とニュートン法については，次回に説明を行う．これらのアルゴ リズムは，みな非線形計画問題の特性である，微分値をとることができるという性質を利用している．

3.1.3 離散最適化問題

離散最適化問題（discrete programming problem）とは，目的関数や制約条件が離散集合である最適化問題のことを指す．離散的なものとしては，整数の集合，グラフにおけるノード やアークの集合などが挙げられる．離散集合では，連続性，微分，そして積分などの概念直接適用することができないという性質と，パターンを数えることができ，総当りですべてのパターンを調べれば，必ず最適解を見つけることができるという性質を持っている．しかし，現実には組合せ爆発などが起こり，すべてのパターンを調べられる問題というのは，実質的には少なく，実際に問題になるようなケースでは，幅広い探索が必要となり，時間がかかりすぎるので，効率よく探索の幅を小さくする必要が出

[2]制約条件が示す領域の積集合
[3]内角がすべて 2 直角より小さい多角形

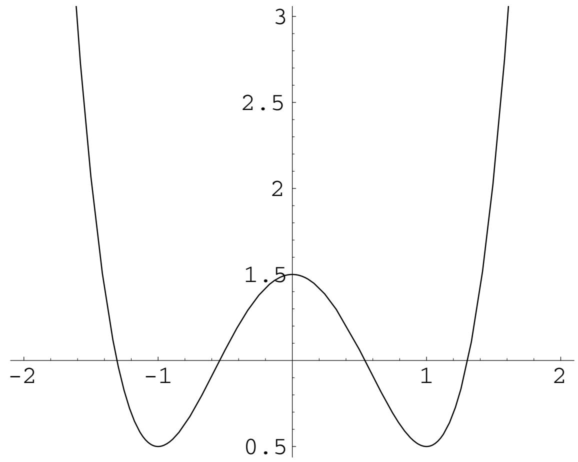

Fig. 8　非線形計画問題の例

てくる．離散最適化問題に用いられるアルゴリズムというのは，いかにして効率よく探索し，かかる時間を少なくするかという目的によるものである．Fig. 9 に離散的最適化問題の簡単な例を示す．縦軸は目的関数，横軸は設計変数を表す．Fig. 9 のように目的関数や制約条件をグラフで表した場合に，連続問題とは正反対で座標点だけがグラフ上

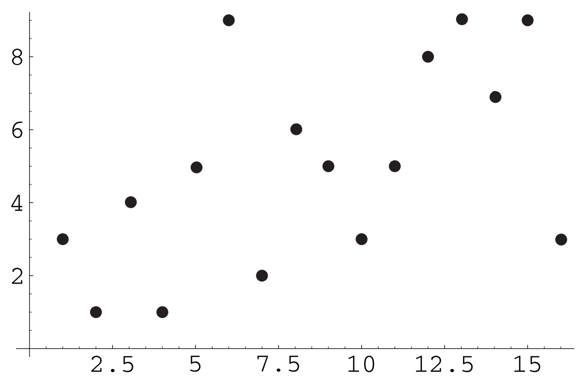

Fig. 9　離散的最適化問題の例

に散らばった状態で表されるものが離散最適化問題です．ここで，離散最適化問題を解くアルゴリズムである欲張り法，分枝限定法，SA と GA について簡単に紹介する．

- 欲張り法
  欲張り法とは，有限である実行可能解に対する目的関数から値を計算して，でてきた値がその段階で最適であればその値を解として取り入れるという方法を繰り返し行い，最適解を求めていくという方法である．この方法では，解となり得る点についてすべて調べるわけではないので，問題の（大域的）最適解が得られる保証はないが，組み合わせが多い問題に対しては実際に最適解を効率的に得ることができる．

- 分枝限定法
  実行可能解が有限個である離散最適化問題は，原理的にはそれらすべて列挙することにより厳密解を求めることができる．この前提をふまえた上で，分枝限定法とは，実行可能解を列挙するために場合分けを行っていく過程で，最適解が得られる見込みのない不必要な場合分けをできるだけ省略して，探索する範囲を絞り込むことにより，計算時間を短縮しようとする方法である．ただし，欲張り法と同様に，大域的最適解が得られる保証はない．

これらの方法は，ある時点において，とりあえず最小となる点を探すといった単純な方法であるので，最後に歪が生じる可能性がある．

- SA
  シミュレーテッドアニーリング（Simulated Annealing）は，高温で加熱した金属を徐々に少しずつ温度を下げて冷やすことによって，元の金属より欠陥の少ない優れた結晶構造を作る物理プロセスに着想を得て，これを計算機上で模擬することにより最適化問題を解こうとするアルゴリズムである．
- GA
  遺伝的アルゴリズム（Genetic Algorithms）は，適用範囲の非常に広い，生物の進化を工学的にモデル化したり，参考にした学習アルゴリズムである．

離散最適化問題は工学のほとんどすべての分野に登場する非常に重要な問題である．しかしながら，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．また，各分野で生じる問題は多様であるため，それらに共通する構造を見抜き，一般化することも困難であるといえる．

宿題

身の回りの最適化問題を探して，その問題の目的関数および設計変数，制約条件をそれぞれ挙げてください．