第2章　数理計画法の用語

授業の補足事項：

- 授業中に私語は、授業料を払っている受講者の権利の侵害になります。授業は映画より全然面白くないと思いますが、映画館での私語と同様、授業中の私語も人の迷惑です。もちろん、授業に対する質問等は大歓迎です。
- 授業で配布する資料をダウンロードできるようにしました。万が一、欠席する場合は、適宜ダウンロードしてください。URL は

  http://www-optima.amp.i.kyoto-u.ac.jp/~ nobuo/Ryukoku/

  です。

2.1　等高線

数理計画法では、一般に n 次元の決定変数を扱う。しかし、n 次元の図を書き表すことは不可能であるので、数理計画法のイメージが掴めるよう $n = 1$ または $n = 2$ の場合の図を用いて説明することが多い。しかし、2 変数の問題を表すときでも、2 変数 (x_1, x_2) と目的関数値 $f(x)$ を表示するためには、3 次元のグラフを書く必要がある。そのとき、3 次元の図ではわかりにくいので、関数値は等高線を用いて 2 次元の図に図示することにすることが多い。これは、ちょうど地図において、東西南北の座標と共に、山の高さを等高線で表すことによって、3 次元を 2 次元で表現していることと同じである。ここで、図 2.1 は、ある関数 $f : R^n \to R$ の等高線を表している。

最小化問題を扱っているときには、特に断らない限り、等高線で囲われた中ほどが、関数の値が小さいものとする。そのため、その図における最小化問題では、その等高線の中の点（窪んだ点）を見つけることが目的となる。

また、数理計画法では $\nabla f(x)$ や $\nabla g_i(x)$ などの関数の勾配を用いて議論すること多い。

勾配について　R^n から R への関数 f と点 $x \in R^n$ に対して，次の条件をみたす n 次元ベクトル a が存在するとき，関数 f は x において微分可能といい，$f'(x) = a^T$ と表す.

$$\lim_{d \in R^n, \|d\| \to 0} \frac{f(x+d) - f(x) - a^T d}{\|d\|} = 0$$

このとき, f' の転置したものを、関数 f の勾配と呼び、$\nabla f(x) = f'(x)^T$ と表す。このように定義された勾配は、関数 f の偏微分を用いて、

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}$$

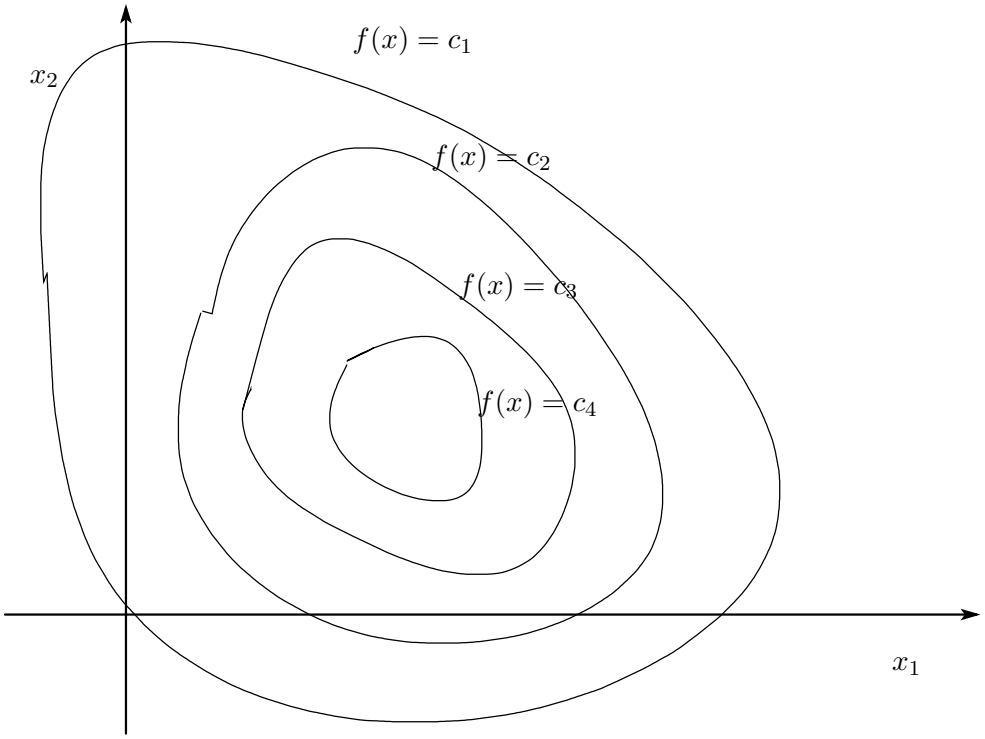

図 2.1: 関数 f の等高線

と表すこともできる。例えば、$R^2 \to R$ の関数 $\hat{f}(x) = x_1^2 + x_1x_2 + 2x_2^2$ の勾配は、

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 + x_2 \\ x_1 + 4x_2 \end{pmatrix}$$

となる。

このとき、これらの勾配を表すベクトルを同じ図の上に書くことによって、議論できたら都合がよい。

たとえば、図 2.1 では、点 $\bar{x}$ と、$\bar{x}$ から $x' := \bar{x} - \nabla f(\bar{x})$ へ、矢印がひかれている。この矢印は、ベクトル $-\nabla f(\bar{x})$ であることに注意しよう。図 2.1 で与えらた最小化問題を解く上で、等高線で与えられた山を降りるとその最小解にいけるわけだが、実は山を降る方向が $-\nabla f(\bar{x})$ である。そのため、このような矢印によって、数理計画法における概念が理解しやすくなる。同様に、制約条件 $g_1(x) \leq 0$ や $h_1(x) = 0$ も、この図の上に表すことができる。ここでは、$g_1(x) \leq 0$ の場合のみ見てみよう。まず、図 2.1 にあるように $g_1(x) = 0$ となる点 x 点の集合が書かれている。図では曲線で書かれている。この曲線によって、領域が $g_1(x) < 0$ となる領域と $g_2(x) > 0$ となる領域の 2 つに分けることができる。このことより、図では、$g_1(x) \leq 0$ の領域が斜線で表されている。よって、この図では、x^* が $g_1(x) \leq 0$ の条件の元で f を最小にする点であることがわかる。

2.2　数理計画法の概念

数理計画問題において、いくつかの重要な概念がある。まず、数理計画問題の最適解の数学的定義を与えよう。数理計画問題において，実行可能集合 $\mathcal{F}$ に所属する点を _____ と呼ぶ。次の不等式を満たす

$$f(x^*) \leq f(x) \quad \text{for all } x \in \mathcal{F}$$

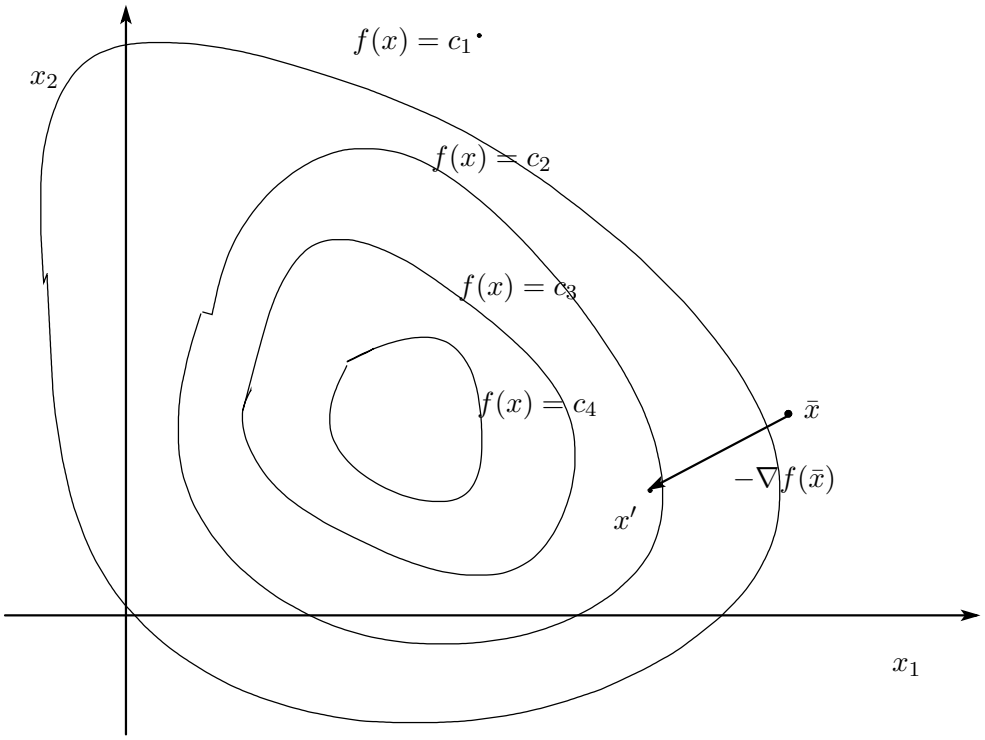

図 2.2: ベクトル $-\nabla(\bar{x})$

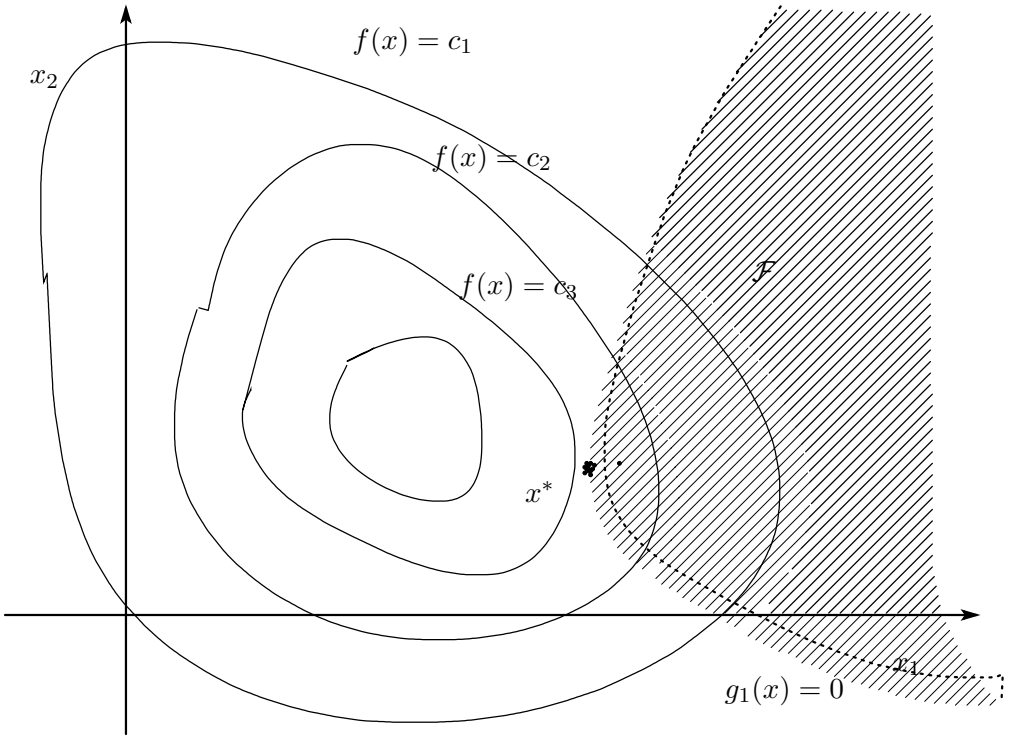

図 2.3: 不等式制約の実行可能集合

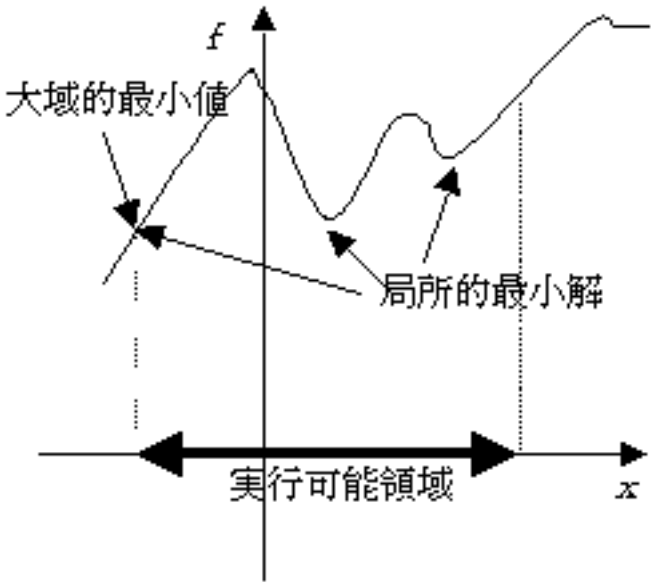

図 2.4: 大域的最小解，局所的最小解

点 $x^* \in \mathcal{F}$ を，その問題の _____ と呼ぶ (図 2.4)．つまり、大域的最小解とは、他の実行可能解の目的関数値よりも大きくならない目的関数値をもつ実行可能解のことである。また、単に最小解、最適解と呼ぶこともある。大域的最小解の目的関数値を _____ と呼ぶ．数理計画法においては、この大域的最小解を求めることが、大きな目的のひとつである。しかしながら，実際に計算機を用いてこのような大域的最小解を求めることは難しい．なぜならば，その点が大域的最小解であるかどうか知るためには，すべての実行可能解の情報を知る必要があるからである．実行可能解が有限である組み合せ最適化問題を除いて、一般の数理計画問題においては、すべての実行可能解を調べることは不可能である。そこで，大域的最適解の代わりに、次に定義する _____ と呼ばれる解を考える．実行可能解 $\hat{x} \in \mathcal{F}$ に対して，$\hat{x}$ の近傍 $N \subseteq R^n$ で

$$f(\hat{x}) \leq f(x) \ \ \forall x \in N \cap \mathcal{F}, x \neq \hat{x}$$

となるものが存在するとき，$\hat{x}$ を局所的最小解と呼ぶ (図 2.4 参照)．さらに、この不等式が狭義に (等号がつかないで) 成り立つとき，$\hat{x}$ を**狭義の局所的最小解**と呼ぶ.

実際には，局所的最小解であるかどうかの判別も困難である．なぜならば、結局、その周辺の無数の点の情報を調べる必要があるからである。そこで，その点の情報だけで、その点が最適解かどうか判別することができる条件があれば便利である。そのような条件は _____ と呼ばれている．次章以降で示すように、目的関数 f および制約を表す関数 h, g が”ある条件”を満たしているとき，最適性の条件を満たした点が扱う問題の大域的最適解となる．そのため，現実問題を数理計画問題に定式化するさいに，f, h, g がそのような条件を満たしているように考慮する必要がある．また、数理計画問題の解法の多くは最適性の条件を満たした点を求める手法である。このような理由のため，モデル化およびアルゴリズムの構築をする以前に，最適性の条件を熟知することが重要である．

数理計画問題の中には、その問題を変形することによって、簡単に解ける問題も少なくない。そのようなに変換される問題のなかに、＿＿＿＿＿＿と呼ばれる問題がある．双対問題は、元の問題を鏡に移したように、いろいろなものが”逆”になった問題である。そのため、ある条件の元では、双対問題の双対問題は元の問題になる。双対問題はもとの問題よりも扱いやすい問題となることがある．また、双対問題の実行可能解における目的関数の値は，もとの問題の目的関数値より必ず小さくなるという性質をもつ。そのため、双対問題の最適解を用いれば、もとの問題の実行可能解が与えられたときに、その解の”良さ”を判定することができる。

2.3 解法の概念

計算機を用いて数理計画問題の解を求めるための計算手順を＿＿＿＿＿＿, または＿＿＿＿＿＿と呼ぶ。この数理計画法の解法を知る上で必要となるいくつかの用語を定義する．これまで実用化されている多くの解法は，最適解 x^* に収束するような点を順次生成する手法である．このように点列を生成していく手法を＿＿＿＿＿＿と呼ぶ．ここで、反復法によって生成さえる点を、$x^0, x^1, \cdots$ と表し、それらをまとめて点列 $\{x^k\}$ と表すことにする。また，一番初めの点 x^0 を特に＿＿＿＿＿＿と呼ぶ．多くの反復法において、x^k が与えられているとき、次の点を求めるために、まず、x^k からどちらの向かったらよいかという方向を決める。このような方向を＿＿＿＿＿＿と呼ぶ。探索方向は d^k と書くことが多い。探索方向や次の反復点を求めるために解く問題を＿＿＿＿＿＿と呼ぶ。もちろん、部分問題は元の数理計画問題より簡単である必要がある。

部分問題について　反復法では、次の点を求めるために、各反復で部分問題を解く必要がある。そのような部分問題は、現在の点 x^k における情報に基づいて構成されることが多い。部分問題が元の問題に近ければ次の反復点が、よりよい点であることが期待できる。しかしながら元の問題に近ければ、元の問題を解く手間とほとんど変わらないことになる。解法を構築する上で重要となるのは、以下に元の問題に近く、一方、元の問題に比べて数段早く解を求めることができる問題を構成するかである。解きやすい問題は限られており、そのような問題は線形方程式であり、2 次計画問題である。2 次計画問題も実際のところは線形方程式を解くことによって求めることとなるため、数理計画法を学ぶ上で、線形方程式の解法を知っておくことは必要不可欠である。本書では線形方程式の解法の詳細を述べないが、LU 分解、コレスキー分解、共役勾配法などを知っておかなければならない

探索方向が決まったとき、その方向にどれくらい進むかを調節することによって、より x^* に近づくことができる。その進む調節するパラメータを＿＿＿＿＿＿と呼ぶ。つまり、探索方向 d^k とステップサイズ t_k が求まれば、次の反復点は

$$x^{k+1} := x^k + t_k d^k$$

で与えられる（図 2.3）。

数理計画法で用いる記号について 本書で用いる記号は、数理計画法を扱った論文や本で標準的に使われているものである。高校などで習うベクトルの表記は、特別な記号を用いるが、本書ではベクトルもスカラー (1 次元変数) も x や y と、同様の記号を用いる。また、変数には x, y, z を使い、問題のパラメータには a, b, c を使う。行列には大文字のアルファベットを用いる。反復回数を表す記号にはアルファベット k を用い、変数がベクトルの場合には下付きの添え字はベクトルの成分の番号を表すので、記号を上に付ける。よって k 回目の反復点は x^k と表される。一方、スカラーの反復回数は、上付きにすると乗数と間違いやすいので、

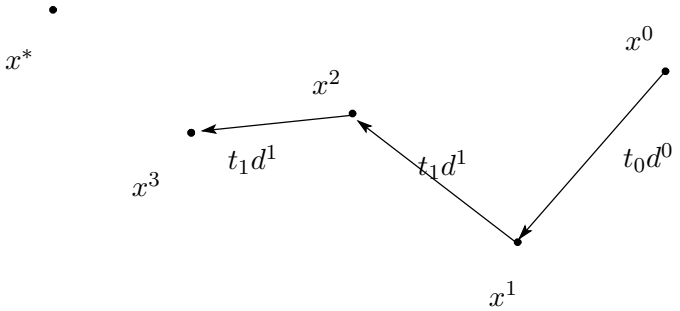

図 2.5: 反復法における探索方向とステップサイズ

下につける。また、$*$ は最適解の意味で用いる。例えば、最適解は x^*, 最適値は $f^* := f(x^*)$ などのように用いる。また、記号の上に横棒 ($\bar{x}$) や山 ($\hat{x}$) をつけることによって、特別な点を表す。

反復法において，任意の初期点から始めて，最小化問題の解に収束するとき，その反復法は _____ するという．

大域的収束について 次のような場合も、「アルゴリズムが大域的収束する」ということがある。

1. 点列 $\{x^k\}$ が解 x^* に収束する．
2. 点列 $\{x^k\}$ が有界で，任意の集積点が解となる．
3. 点列 $\{x^k\}$ の任意の集積点が解となる．

1,2,3 の順番で好ましい「大域的収束」である．点 x^* が点列 $\{x^k\}$ の集積点であるとは，$\{x^k\}$ の中に x^* に収束する無限部分列が存在することを意味する．また，点列 $\{x^k\}$ が有界であれば必ず集積点を持つので，2 の意味で大域的収束するアルゴリズムであれば，解を得るという目的は達成することができる．一方，3 の意味での大域的収束は，点列が発散する場合もあり，その場合は解が得られない．最小化問題に適用するアルゴリズムの大域的収束性を比較するさい，どの意味で「大域的収束する」と言っているか吟味する必要がある．

一方，反復法が，ある特別な初期点から始めたとき，生成された点列が解に収束する場合，反復法は _____ するという．

次に反復法の速さを議論する用語について説明する．有限回の反復で厳密な最適解を得ることができる解法を、有限回反復の解法と呼ぶ。有限回反復の解法の中でも、問題の次元 n や m, r の多項式の反復回数で終了する解法を、多項式オーダーのアルゴリズムと呼ぶ。例えば、mn^4 回以内に厳密な最適解を得ることができる解法は、多項式オーダーのアルゴリズムである。組み合せ最適化問題や線形計画問題、2 次計画問題には、有限回反復の解法が提案されている。

一方、一般の数理計画問題では，有限回の演算で厳密な解を得ることはできない．例えば，$f(x) = x, h(x) = x^2 - 2$ という問題を考えてみよう．この問題の解は $\sqrt{2}$ であるが，$\sqrt{2}$ は有限回の演算で表現することはできない．そのため，数理計画法のアルゴリズムでは，次の _____ を用いてアルゴリズムの優劣を比較することが多い．正の定数 p と $x^* \in R^n$ に対して，点列 $\{x^k\}$ が以下の条件を満たすとき，この点列は x^* に _____ するという．

$$\lim_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^p} < \infty$$

また，$p > 1$ の p 次収束を，_____ という．

ここで，次の列 $\{y^k\}$ と $\{z^k\}$ を考えてみよう．

$$\begin{aligned} y^k &= 0.5^k \\ z^k &= 0.5(z^{k-1})^2. \end{aligned}$$

表 2.1: 1 次収束と 2 次収束

k	y^k	z^k
0	1	1
1	0.5	0.5
2	0.25	0.125
3	0.125	0.0078125
4	0.0625	0.0000305
⋮	⋮	⋮

ここで，$z^0 = 1$ とする．この点列 $\{y^k\}$ は，0 に 1 次収束している．一方, 点列 $\{z^k\}$ は，0 に 2 次収束している．これを表にしてみると，表 2.1 となる．このため，2 次収束するアルゴリズムは極めて速いことがわかる．

多くの解法では無限回に反復することによって解に収束することが保証されている。しかしながら、現実問題として、無限回続けていては、解法として意味をなさない。そこで、厳密な最適解ではなくても、最適解に近いと思われるところで、終了する必要がある。そのように終了する条件を _____ と呼ぶ。どのような終了条件を用いるかは、非常に難しい。最適性の条件を”ほとんど”満たしているとか、ほとんど点列が動かなくなったときなどを終了条件にすることが多い。

次に数理計画問題に対する解法をいくつか簡単に紹介しておこう。解法を選ぶ際には、得られる問題の情報によって、解法を決める必要がある。制約なし最小化問題に対しては、その問題の目的関数値しか計算することができない場合、**直接探索法**を使うとよい。また、目的関数の勾配を計算することができれば、**最急降下法**や**準ニュートン法**を使うことができる。なかでも準ニュートン法は大域的に収束し、なおかつ超一次収束するというよい性質を持っている。目的関数の 2 回微分まで計算できるときは，**ニュートン法**がよい性質を持っている。特に、2 次収束するというよい性質を持っている。しかし、アルゴリズムに工夫を加えない限り、一般には大域的収束しない。大規模な問題には，**共役勾配法**と呼ばれる解法が適している。しかしながら、共益勾配法には、大域的収束性がない。制約付き最小化問題では、線形計画法では**単体法**や**内点法**がよい性質を持っている。比較的、小中規模の問題には単体法が、中大規模な問題には内点法が良いとされている。凸計画問題に対しても、内点法がすぐれている。しかし、凸計画ではない最小化問題には内点法は的用できない。そのようなときには、ニュートン法を制約付き最小化問題に一般化した**逐次 2 次計画法**がよく用いられている．組み合せ最適化問題では、それぞれの問題にあわせて数多くのアルゴリズムが提案されている。問題によっては多項式時間のアルゴリズムが提案されているものもある。多項式時間が提案されていない問題では、その厳密解を求めるために、**分枝限定法**を用いることが多い。分枝限定法は、実行可能解を効率良くしらみつぶしに調べる手法である。しかし、問題によっては、非常に時間がかかることがある。そのようなときには、厳密な最適解を求めることを諦めるかわりに、実用時間で実用的な解を求めることができる**メタヒューリスティックス**と呼ばれる手法を使うことになる。