線形計画問題

郡　悠希　廣安 知之　日和 悟

2016 年 11 月 9 日

IS Report No. 2017022307

IS Report

Medical Information
System Laboratory

Abstract

本稿では最適化について述べた後，実例を踏まえながら最適化問題の種類や最適解の定義などについて述べる．そして，最適化問題の一つである線形計画問題について単体法を用いて解く方法を実例を用いて述べる．
キーワード: 最適化, 最適化問題, 線形計画問題, スラック変数, 単体法（シンプレックス法）

目 次

第 1 章　最適化問題

1.1　最適化

最適化とは，関数やプログラム，製造物などを最適な状態に近づけることで，意思決定や問題解決のための一つの手段である．現実問題を最適化問題に定式化し，最適解の計算，分析，検証を行う事で解決に導く [1]．そのため，最適化問題への定式化が重要となる．

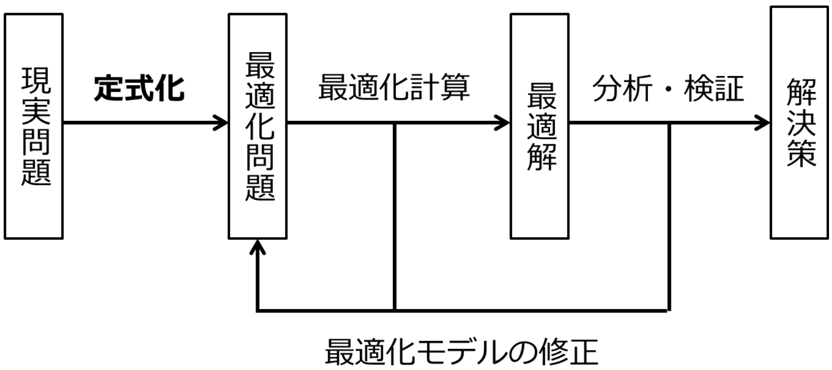


Fig. 1.1　最適化手法による問題解決アプローチ（参考文献 [1] を参考に自作）

1.2　最適化問題

最適化問題とは，制約条件を満たす解の中で目的関数を最小化（最大化）する解を求める問題である [1]．以下に具体例を挙げる [2]．

[例 **1**]

縦横の辺の長さの和が 4 となる長方形の中で，面積が最大となるのはどの様な長方形か？

この問題は次のように定式化することができる．縦横の辺の長さをそれぞれ x と y とすると，面積は xy となる．辺の長さの合計が 4 となる長方形を考えるため，この問題は以下の様に定式化される．この様な問題を（制約付き）最適化問題と呼ぶ．

$$\begin{aligned} &\text{最大化} \quad f(x,y) := xy \\ &\text{制　約} \quad x+y=4, x \geq 0, y \geq 0 \end{aligned} \tag{1.1}$$

[例 **2**]

平面に 4 点（1，3）,（2，5）,（3，5）,（4，7）が与えられた時，これらの点の最も近くを通る直線は？

上記の問題は以下の最適化問題を用いて解くことが可能である．直線は一般に $y = ax + b$ と表すことができ，点（1,3）と直線との誤差は

$$3 - (1 * a + b) \tag{1.2}$$

となる．この誤差の二乗の和を最小にする直線を求める方法を最小二乗法と呼ぶ．この問題は以下のような最小化問題として定式化することが出来る．この様な問題を制約なし最適化問題と呼ぶ．

$$\begin{aligned} &\textbf{最小化} \quad f(a,b) = \{3-(a+b)\}^2 + \{5-(2a+b)\}^2 \\ &\qquad\qquad\qquad + \{5-(3a+b)\}^2 + \{7-(4a+b)\}^2 \\ &\textbf{制　約} \quad \text{なし} \end{aligned} \tag{1.3}$$

1.3 制約なし最適化問題

一般に最適化問題は，例 1 の様に x を選ぶ範囲に何らかの制約があるものが多い．この制約がない問題を制約なし最適化問題という．

$$\begin{aligned} &\textbf{最小化} \quad f(x) \\ &\textbf{制　約} \quad \text{なし} \end{aligned} \tag{1.4}$$

ここで，最小化する関数 $f(x)$ を目的関数と呼ぶ．関数を最大化する問題を最大化問題と呼び，逆に最小化する問題を最小化問題と呼ぶ．また，これら二つをまとめて最適化問題と呼ぶ [2].

1.3.1 最適解の定義

最適化問題の解である最適解にはいくつかの種類が存在する．制約なし最小化問題に対しては，次のように定義される [2].

[定義 1] $\overline{x}$ が全ての $x \in \mathbb{R}^n$ に対して

$$f(x) \geq f(\overline{x}) \tag{1.5}$$

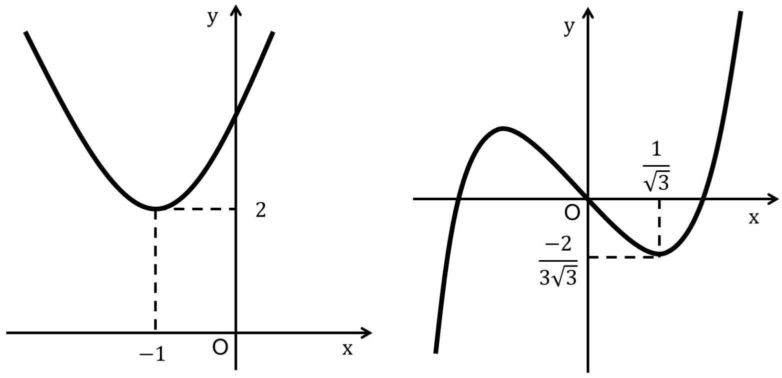
の時 $f(\overline{x})$ を大域最小値，$\overline{x}$ を大域最小解と呼ぶ.

大域最小解を見つけられれば最もよいが，一般的に難しい．そこで，より見つけやすい以下の解を探索することがある.

[定義 2] $\overline{x}$ に十分近い全ての $x \in \mathbb{R}^n$ に対して

$$f(x) \geq f(\overline{x}) \tag{1.6}$$

の時 $f(\overline{x})$ を局所最小値，$\overline{x}$ を局所最小解と呼ぶ．最大化問題の場合は，不等号が逆向きになり，それぞれ「最小」を「最大」に置き換える．最小化と最大化をひとまとめにして，局所最適値，局所最適解などと呼ぶ [2].



(a) 大域最小解を持つ関数　　(b) 大域最小解は持たないが，局所最小解をもつ関数

Fig. 1.2　大域最小解と局所最小解の例（参考文献 [2] を参考に自作）

1.3.2　極値問題

微分積分で扱う極値問題も最適化問題である．ただし，最適解の中でも特別なものを求めることを目的としている [2].

[定義 3]

- $x \neq \overline{x}$ を満たし，$\overline{x}$ に十分に近いすべての x に対して，$f(x) > f(\overline{x})$ が成り立つとき，f は $\overline{x}$ で極小値をとるという.
- $x \neq \overline{x}$ を満たし，$\overline{x}$ に十分に近いすべての x に対して，$f(x) < f(\overline{x})$ が成り立つとき，f は $\overline{x}$ で極大値をとるという.

極小値と極大値をまとめて極値と呼ぶ.

局所最小値は等号も含んだ「$f(x) \geq f(\overline{x})$」で定義されるため，極小値は局所最小値となる．よって，極値をとる点は局所的に一意の最適解となる．包含関係は Fig. 1.3 の様になる [2].

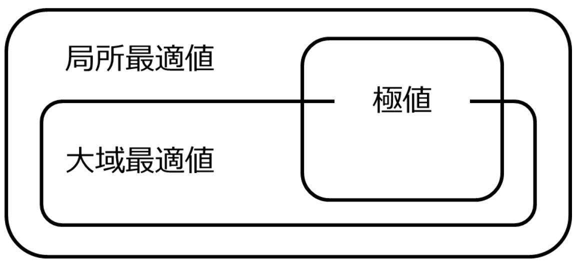


Fig. 1.3　最適値と極値の関係図（参考文献 [2] を参考に自作）

1.4　制約付き最適化問題

集合 C を数ベクトル空間 $\mathbb{R}^n$ の部分集合として，

$$\begin{aligned} &\text{最小化} \quad f(x) \\ &\text{制　約} \quad x \in C \end{aligned} \tag{1.7}$$

を制約付き最小化問題と呼ぶ．ここで，集合 C を実行可能領域，C の点を実行可能解と呼ぶ．ただし,「$x \in C$」とは「x が C に含まれる」ことを意味する．

例えば，例 1 の制約は

$$x + y = 4, x \geq 0, y \geq 0 \tag{1.8}$$

のように式を用いて表される．このような式を制約式と呼ぶ．

[定義 4]

$\overline{x} \in C$ が $\overline{x}$ に十分近い C 上のすべての x に対して $f(x) \geq f(\overline{x})$ の時，$\overline{x}$ を式 (1.7) の局所最小解，$f(\overline{x})$ を局所最小値と呼ぶ．

[定義 5]

$\overline{x} \in C$ が C 上のすべての x に対して $f(x) \leq f(\overline{x})$ の時，$\overline{x}$ を式 (1.7) の大域最小解，$f(\overline{x})$ を大域最小値と呼ぶ．

最小解と最大解を合わせて最適解と呼ぶ．最適値も同様である [2]．

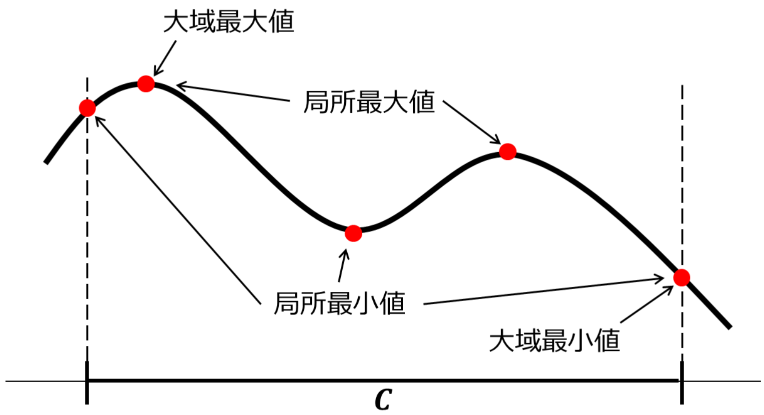


Fig. 1.4　大域最大値，局所最大値の関係（参考文献 [2] を参考に自作）

1.5　最適化問題の分類

最適化問題は目的関数・制約条件が線形か非線形かによって分類することが出来る．目的関数も制約条件も線形の場合を線形最適化問題と呼び，目的関数・制約条件のどちらかが線形でない場合を非線形最適化問題と呼ぶ．これらの包含関係は Fig. 1.5 の様になる．線形最適化問題は比較的簡単であるが，非線形最適化問題は難しいとされる．

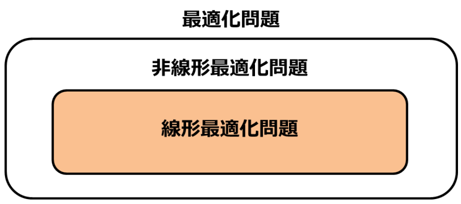


Fig. 1.5　最適化問題の分類（自作）

また，最適化問題は設計変数が連続値か離散値かでも分類することが出来る．連続値の場合を連続最適化問題，離散値の場合を組み合わせ最適化問題と呼び，実問題の多くは組み合わせ最適化へ分類

される．

この組み合わせ最適化問題には以下の様にいくつか名前の付いた標準問題がある [3]．

- グラフ・ネットワーク問題
  最小全域木問題，最大安定集合問題，最大カット問題，最小頂点被覆問題，最短路問題，最大流問題，最小費用流問題
- 経路問題
  運搬経路問題，巡回セールス問題
- 集合被覆・分割問題
  集合被覆問題，集合分割問題
- スケジューリング問題
  ジョブショップ問題，勤務スケジューリング問題
- 切り出し・詰め込み問題
  ナップサック問題，ビンパッキング問題，n 次元詰め込み問題
- 配置問題
  施設配置問題，容量制約なし施設配置問題
- 割り当て・マッチング問題
  2 次割合問題，一般化割当問題，最大マッチング問題，重みマッチング問題，安定マッチング問題

第 2 章　線形計画問題

最適化問題において，目的関数も制約式も 1 次式であるものを線形計画問題と呼ぶ [2].

$$
\begin{aligned}
\text{最小化}\quad & f(x) \\
\text{制　約}\quad & g(x) \leq 0
\end{aligned}
\tag{2.1}
$$

線形計画問題は一見特殊な問題に見られるが，大規模な問題でも高速に解くことが出来るため，工学から経済まで非常に幅広く使用されている．以下に線形計画問題の例を挙げる [4].

[例]

あるレストランで手持ちの材料からハンバーグとオムレツを作り，販売する．ハンバーグ 1 個に必要な材料，オムレツ 1 個に必要な材料，またそれぞれの販売価格や手持ちの材料は Table. 2.1 に示した通りである．この時，ハンバーグとオムレツをいくつずつ作れば利益は最大になるか？

Table. 2.1　各商品の材料

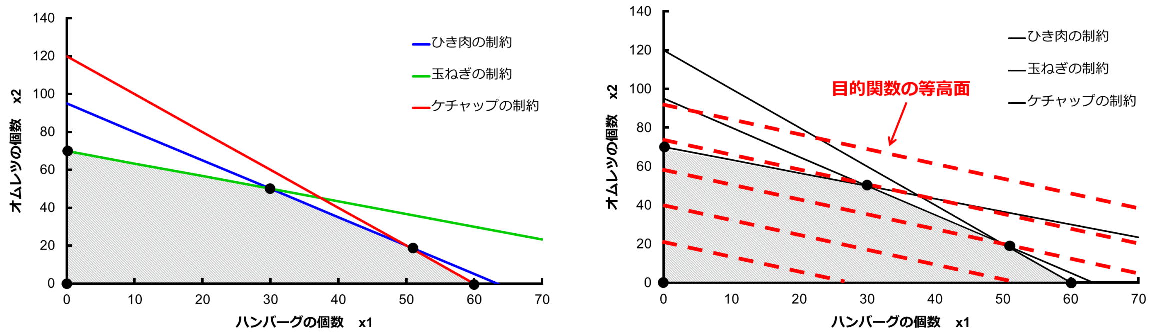
	ハンバーグ	オムレツ	在庫
ひき肉	60 g	40 g	3800 g
玉ねぎ	20 g	30 g	2100 g
ケチャップ	20 g	10 g	1200 g
価格	400 円	300 円	

この時，ハンバーグの個数を x_1，オムレツの個数を x_2 とし，この問題を最適化問題で定式化すると，以下の様に表すことが出来るため線形計画問題となる．

$$
\begin{aligned}
\text{目的関数}\quad & f(x) = 400x_1 + 300x_2 \\
\text{制約条件}\quad & \begin{cases} 60x_1 + 40x_2 \leq 3800 \\ 20x_1 + 30x_2 \leq 2100 \\ 20x_1 + 10x_2 \leq 1200 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}
\end{aligned}
\tag{2.2}
$$

この問題の実行可能領域は，Fig. 2.1(a) の様な凸多面体とその頂点から成る．設計変数の数によっては立体となることもある．また，実行可能領域は問題によって空であったり，無限の広がりを持つ場合もある．

この問題の目的関数は 1 次関数であるため，その等高面は Fig. 2.1(b) の様な直線となる．したがって，実行可能領域の頂点を全て求めれば，頂点における目的関数値を比べて，最も小さいものが最小解となる．



(a) 実行可能領域 (b) 目的関数の等高面

Fig. 2.1 実行可能領域と等高面（参考文献 [2] [4] を参考に自作）

2.1 単体法

前節で述べたように線形計画問題の解は求めることが出来る．しかし，全ての頂点を求めることや実行可能解の頂点が未知の場合は解を求めることが困難である．そこで主に用いられるのは単体法（シンプレックス法）と呼ばれるアルゴリズムである．単体法は線形計画問題の解法の一つであり，以下のステップから成る．いくつか例外的な場合も存在するが，本節では前節の線形計画問題を例に例外的な現象が起こらない場合の単体法について述べる [2] [4].

Step.1 スラック変数を導入

問題をスラック変数と呼ばれる s_1,s_2,s_3 を導入して式 (2.3) に変形する．スラックとは「余裕」という意味であり，もとの不等式において統合が成立する条件に対して，どれだけ余裕があるかを表している．ここでのスラックス変数の物理的な意味は s_1：ひき肉の残量，s_2：玉ねぎの残量，s_3：ケチャップの残量である．

$$
\begin{aligned}
&\text{最小化} \quad f(x) = -400x_1 - 300x_2 \\
&\text{制約条件} \begin{cases}
60x_1 + 40x_2 + s_1 = 3800 \\
20x_1 + 30x_2 + s_2 = 2100 \\
20x_1 + 10x_2 + s_3 = 1200 \\
x_1 \geq 0, x_2 \geq 0, s_1 \geq 0, s_2 \geq 0, s_3 \geq 0
\end{cases}
\end{aligned}
\tag{2.3}
$$

Step.2 辞書を作成

スラック変数を導入した後，式 (2.3) の制約式でスラック変数 s_1，s_2，s_3 を左辺に残し，残

りを右辺へ移行する．

$$
\text{（辞書 1）}\quad \text{最小化}\quad f(x) = -400x_1 - 300x_2
$$

$$
\text{制約条件}\begin{cases} s_1 = 3800 - 60x_1 - 40x_2 \\ s_2 = 2100 - 20x_1 - 30x_2 \\ s_3 = 1200 - 20x_1 - 10x_2 \\ x_1 \geq 0, x_2 \geq 0, s_1 \geq 0, s_2 \geq 0, s_3 \geq 0 \end{cases} \tag{2.4}
$$

左辺に現れる変数 s_1，s_2，s_3 を基底変数，右辺に現れる変数 x_1，x_2 を非基底変数と呼ぶ．また，この形を線形計画問題の辞書と呼ぶ．

Step.3 実行可能基底解を求める

ここで，辞書の非基底変数 x_1，x_2 をすべて 0 として得られる実行可能解を辞書の実行可能基底解と呼ぶ．

$$
\begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 3800 \\ 2100 \\ 1200 \end{bmatrix} \tag{2.5}
$$

Step.4 解の更新

次に，この実行可能基底解を次のようなルールで変更する．

解の更新ルール

(i) 非基底変数の中から，目的関数 n において係数が負であるものを 1 つ選び，その値を t，その他の非基底変数の値を 0 とおく．

(ii) すべての基底変数が負にならない範囲で，(i) で選んだ非基底変数の値 t を最大まで増やす．

まず，更新ルール (i) より非基底変数を選ぶ．辞書 1 では非基底変数は x_1，x_2 である．ここで，x_1 と x_2 のどちらを増やせば $f(x)$ をより小さく出来るか考える．つまり，まずハンバーグとオムレツのどちらか一方を作る事にし，どちらを作るとより利益が上がるかを考える．

この問題では，ハンバーグを作る事にする．よって，$x_1 = t$，$x_2 = 0$ とおくと，

$$
\begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} t \\ 0 \\ 3800 - 60t \\ 2100 - 20t \\ 1200 - 20t \end{bmatrix} \tag{2.6}
$$

となる．次に更新ルール (ii) を満たす t を求めると，

$$
\begin{array}{lcccccccc}
s_1 = 3800 - 60t\ , & s_1 & \geq & 0 & \Rightarrow & 3800/60 \geq & s_1 & \geq & 0 \\
s_2 = 2100 - 20t\ , & s_2 & \geq & 0 & \Rightarrow & 2100/20 \geq & s_2 & \geq & 0 \\
s_3 = 1200 - 20t\ , & s_3 & \geq & 0 & \Rightarrow & 1200/20 \geq & s_3 & \geq & 0
\end{array}
\tag{2.7}
$$

なので，どの s_1，s_2，s_3 も負でない最大の t は $t = 1200/20 = 60$ となる．

代入すると

$$
\begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} t \\ 0 \\ 3800 - 60t \\ 2100 - 20t \\ 1200 - 20t \end{bmatrix} \rightarrow \begin{bmatrix} 60 \\ 0 \\ 200 \\ 900 \\ 0 \end{bmatrix} \tag{2.8}
$$

となり，新しい実行可能解が得られる．この時の目的関数値は

$$
f(x) = -400x_1 - 300x_2 = (-400) * t + (-300) * 0 = -24000 \tag{2.9}
$$

となり，目的関数値は減少している．

Step.5 辞書の更新

更新ルールの適用後，元の実行可能基底解と新しい実行可能解を比べると

$$
(\text{旧}) \begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 3800 \\ 2100 \\ 1200 \end{bmatrix} \qquad (\text{新}) \begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 60 \\ 0 \\ 200 \\ 900 \\ 0 \end{bmatrix} \tag{2.10}
$$

[非基底変数] x_1 →正，[基底変数] $s_3 \rightarrow 0$

となっている．

次に，非基底変数 x_1 を基底変数に，基底変数 s_3 を非基底変数に入れ換えた新しい辞書を作成する．まず，辞書 1 で s_3 の関係する制約式を

$$
\begin{aligned}
20x_1 &= 1200 - 10x_2 - s_3 \\
x_1 &= 60 - 1/2x_2 - 1/20s_3
\end{aligned}
\tag{2.11}
$$

と変形する．これを他の式に代入し，全ての式（目的関数も含む）の右辺から x_1 を消去すると，新しい辞書 2 が得られる．

$$
\begin{array}{lll}
\textbf{（辞書 2）} & \text{最小化} & f(x) = -24000 + 200x_2 + 20s_3 \\
 & \text{制約条件} & \begin{cases} x_1 = 60 - 1/2x_2 - 1/20s_3 \\ s_1 = 200 - 10x_2 + 3s_3 \\ s_2 = 900 - 20x_2 + s_3 \\ x_1 \geq 0, x_2 \geq 0, s_1 \geq 0, s_2 \geq 0, s_3 \geq 0 \end{cases}
\end{array}
\tag{2.12}
$$

Step.6 Step.4，5 の反復

辞書 2 に同様の操作を繰り返す．辞書 2 では非基底変数は x_2，s_3 である．次に $x_2 = t$，その他の比基底変数 $s_3 = 0$ とし，辞書を更新する．この時，t の増加幅は更新ルール（ii）より，$t = 20$ となる．

$$\begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 60 - 1/2t \\ t \\ 200 - 10t \\ 900 - 20t \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 50 \\ 20 \\ 0 \\ 500 \\ 0 \end{bmatrix} \tag{2.13}$$

[非基底変数] x_2 →正，[基底変数] $s_1 \rightarrow 0$

より，x_2 が基底変数，s_1 が非基底変数になるように辞書を更新する．辞書 2 の s_1 の関係式より得られる

$$x_2 = 20 - 1/10s_1 + 3/10s_3 \tag{2.14}$$

を用いて辞書を更新し，辞書 3 を求める．

$$\text{（辞書 3）}\quad \text{最小化} \quad f(x) = -26000 + 10s_1 - 10s_3$$

$$\text{制約条件} \begin{cases} x_1 = 50 + 1/20s_1 - 1/5s_3 \\ x_2 = 20 - 1/10s_1 + 3/10s_3 \\ s_2 = 500 + 2s_1 - 5s_3 \\ x_1 \geq 0, x_2 \geq 0, s_1 \geq 0, s_2 \geq 0, s_3 \geq 0 \end{cases} \tag{2.15}$$

反復 2 回目

辞書 3 では s_1，s_3 が非基底変数となる．目的関数の係数より，$s_3 = t$，$s_1 = 0$ とし，辞書を更新する．この時の t の増加幅は $t = 100$ となる．

新しい実行可能解は

$$\begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 50 - 1/5t \\ 20 + 3/10t \\ 0 \\ 500 - 5t \\ t \end{bmatrix} \rightarrow \begin{bmatrix} 30 \\ 50 \\ 0 \\ 0 \\ 100 \end{bmatrix} \tag{2.16}$$

[非基底変数] s_3 →正，[基底変数] $s_2 \rightarrow 0$

となり，新しい辞書は以下のようになる．

$$
\text{（辞書 4）}\quad \text{最小化}\quad f(x) = -27000 + 6s_1 + 2s_2
$$
$$
\text{制約条件}\begin{cases} x_1 = 30 + 1/100s_1 + 1/25s_2 \\ x_2 = 50 + 1/50s_1 + 3/50s_2 \\ s_3 = 100 + 2/5s_1 - 1/5s_2 \\ x_1 \geq 0, x_2 \geq 0, s_1 \geq 0, s_2 \geq 0, s_3 \geq 0 \end{cases} \tag{2.17}
$$

この時，目的関数の変数 s_1，s_2 の係数は全て 0 以上になっているため，更新ルール (i) が適用できない．そのため，ここで解の更新を終了する．

Step.7 単体法の終了

すべての変数が 0 以上であるという制約があるので，辞書 4 の最適値はの時-27000 である．制約式より，辞書 3 の最適解は

$$
\begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 30 \\ 50 \\ 0 \\ 0 \\ 100 \end{bmatrix} \tag{2.18}
$$

で，最適値は-27000 となる．従って，元の問題の最適解は

$$
\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 30 \\ 50 \end{bmatrix} \tag{2.19}
$$

であり，最適値は-27000 となる．

この様に，単体法は Fig. 2.2 に示すように目的関数値が減る方向にある頂点を求めることで，最適解を求めるアルゴリズムである．

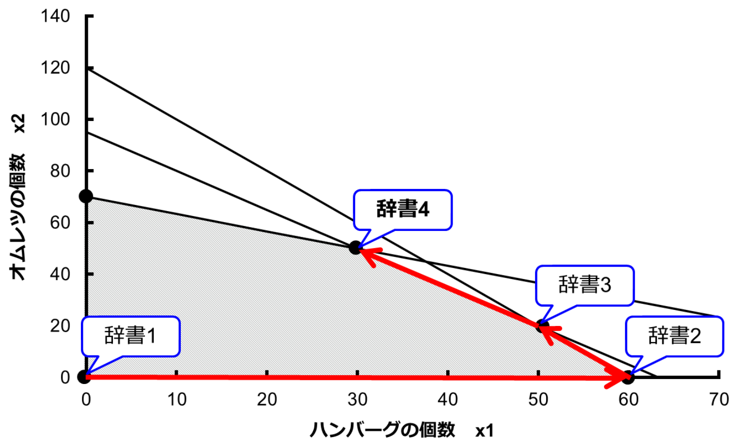


Fig. 2.2　凸多面体上での解の動き（参考文献 [4] を参考に自作）

参考文献

[1] 梅谷俊治, 組合せ最適化入門　線形計画から整数計画まで, 言語処理学会第 19 回年次大会（NLP2013）, 2013.

[2] 関口良行, はじめての最適化, 初版第 2 版, 近代科学社, 2015.

[3] 穴井宏和, 斉藤努, 今日から使える!組合せ最適化 離散問題ガイドブック (KS 理工学専門書), 講談社, 2015.

[4] 藤本康孝, "システム工学講義資料　線形計画法の例題," http://www.fujilab.dnj.ynu.ac.jp/lecture/system2.pdf, 閲覧日：2017 年 2 月 22 日.