第 2 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

1　欲張り法

前回のゼミでも触れたが，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．ここでは次の例題を用いて説明していく．

$$\begin{aligned} &Maximize && 2x_1 + 3x_2 + 2x_3 + 8x_4 \\ &Subject && 3x_1 + 2x_2 + x_3 + 2x_4 \leq 4 \\ & && x_1, x_2, x_3, x_4 \in \{1, 0\} \end{aligned}$$

この問題は，$x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1$ の時, 最適解 11 となる．これは前回のゼミにおいて学んだ分枝限定法などで解くことが可能であるが，この問題を欲張り法を用いて解くことにする (分枝限定での解法は省略する)．この問題における「その時点で最善であると思われるもの」とは効率良く目的関数を大きくすることとする．つまり制約条件をなるべく大きくせずかつ目的関数をより大きくすることである．

順番に見ていくことにする．

$x_1 = 1$ にすると，制約条件は 3，目的関数は 2 それぞれ増える．同様に $x_2 = 1$ にすると，制約条件は 2，目的関数は 3 増える．$x_3 = 1$ にすると，制約条件は 1，目的関数は 2，同様に $x_4 = 1$ にすると，制約条件は 2，目的関数は 8 増える．

これらより，ここにおけるもっとも最善であると思われる解は $x_4 = 1$ である．この時点で目的関数は 8 となる．

次にこの時点でもっとも最善であると思われる解は $x_3 = 1$ である．よって目的関数は 10 となる．

ここで次に最善な解である $x_2 = 1$ にすると制約条件を満たさないのでここで終了となる．つまり大域的最適解の 11 にはならず，局所的最適解の 10 になったことがわかった．

このように単純な方法では問題の大域的最適解が構築できる保証はないが，ある種の問題に対しては実際に最適解を求めることができるのである．

2　線形計画問題

2.1　シンプレックス法

線形計画問題（Linear Programming Probrem：LP）の代表的な解法にシンプレックス法がある．シンプレックス法とは，最適解が境界線上に現れる最も単純な線形最適化問題である．この章では,「シンプレックス法」を具体的に計算で行うにはどのようにしたらよいかを説明する．以下のような例を用意した．

＜簡単なシンプレックス法の例＞

製品 X を１ kg 生産するには，原料 A を４ kg，原料 B を２ kg，原料 C を１ kg 必要とし，製品 Y を１ kg 生産するには，原料 A １ kg，原料 B を２ kg，原料 C を３ kg 必要とする．原料の在庫量は，A は７２ kg，B は４８ kg，C は４８ kg ある．製品 X の売価は３万円／kg，製品 Y の売価を２万円とするとき，利益（＝売上高．原料や生産の費用は考えないことにする）を最大にするには，製品 X と製品 Y をどれだけ生産すればよいだろうか．

Table 1　材料と製品の関係

	製品 X	製品 Y	原料在庫量
原料 A	4	1	72
原料 B	2	2	48
原料 C	1	3	48
単価	3 万円	2 万円	

今回は，次章のシンプレックスタブローを用いた説明が分かりやすくなるように，その流れに沿った解法を行ってみる．

$a)$ 目的関数の設定

売上高は 1 つにつき，X が 3 万円，Y が 2 万円なので，X を x_1 個，Y を x_2 個売ったとすると，売上の合計は $3x_1 + 2x_2$ となる．今回の目的はこの合計金額をできる限り大きくすることなので，$Max(3x_1 + 2x_2)$ が目的関数となる．

b) **制約条件の設定**
原料 A は製品 X を作るのに 4kg，製品 Y を作るのに 1kg 必要だが，その合計量は 72kg しかないので，

$$4x_1 + x_2 \leq 72$$

となる．同様に原料 B，原料 C についても考えると，

$$2x_1 + 2x_2 \leq 48$$
$$x_1 + 3x_2 \leq 48$$

c) **非負条件の設定**
今回のように,「ものの個数」などを扱う最適化問題では，数が負になることはありえない．そのことから，以下のように条件が与えられる．

$$x_1 \geq 0,\ x_2 \geq 0$$

d) **シンプレックス法による解法**
ここまで得られた条件をまとめると，以下のようになる．

$$Max(3x_1 + 2x_2)$$
$$4x_1 + x_2 \leq 72$$
$$2x_1 + 2x_2 \leq 48$$
$$x_1 + 3x_2 \leq 48$$
$$x_1 \leq 0,\ x_2 \leq 0$$

このとき，目的関数，制約条件ともにシンプレックス法を適応できる形 (最小化問題) になっていないので，目的関数・制約条件の修正が必要となる．

$$Min(Linear)$$
$$(\text{制約条件}) = 0$$

ここで制約条件を等式で示すために新たに変数 λ_1，λ_2，λ_3 を導入する．

・目的関数の変形
今回の目的関数 $Max(3x_1 + 2x_2)$ は，条件を負にしてやることで最小化問題に書き換えることが可能である．書き換えられた目的関数は以下のようになる．

$$Min(-3x_1 - 2x_2)$$

・制約条件の変形
次に，制約条件をシンプレックス法に当てはめるために，次のように変形させる．

$$4x_1 + x_2 + \lambda_1 = 72 \tag{1}$$

$$2x_1 + 2x_2 + \lambda_1 = 48 \tag{2}$$

$$x_1 + 3x_2 + \lambda_1 = 48 \tag{3}$$

この問題は最小化問題となっているので，$\lambda_1 \geq 0$，$\lambda_2 \geq 0$，$\lambda_3 \geq 0$ となっている．

このとき，目的関数を見ると，$Min(-3x_1 - 2x_2)$ であるので，係数比較より x_1 が大きい方が全体として小さくなっていくことがわかる．あとは，いかにこの条件を満たす制約条件を探すかが重要となってくる．

ここで，非負条件を満たしつつある最小となる x_2 である，$x_2 = 0$ を (1)，(2)，(3) に入れて代入し，変形すると以下のようになる．

$$\lambda_1 = 72 - 4x_1 \tag{4}$$

$$\lambda_2 = 48 - 2x_1 \tag{5}$$

$$\lambda_3 = 48 - x_1 \tag{6}$$

ここで，$\lambda_1 \geq 0$，$\lambda_2 \geq 0$，$\lambda_3 \geq 0$ の条件により，(4)，(5)，(6) よりそれぞれ

$$x_1 \leq 18,\ x_1 \leq 24,\ x_1 \leq 48$$

という値を導くことができるので，$x_1 \leq 18$ を導くことができる．

ここでもっとよい解はないかを確認する．$x_1 = 6$ を導くのに利用した (3) を変形して，

$$x_1 = 18 - \frac{1}{4}x_2 - \frac{1}{4}\lambda_1 \tag{7}$$

よってこれを (1)，(2) に代入すると，

$$\lambda_2 = 12 - \frac{3}{2}x_2 + \frac{1}{2}\lambda_1 \tag{8}$$

$$\lambda_3 = 30 - \frac{11}{4}x_2 + \frac{1}{4}\lambda_1 \tag{9}$$

より大きい x_2 を求めるために，(7)，(8)，(9) を変形し，

$$x_2 = 72 - 4x_1 - \lambda_1 \tag{10}$$

$$x_2 = 8 + \frac{1}{3}\lambda_1 - \frac{2}{3}\lambda_2 \tag{11}$$

$$x_2 = \frac{120}{11} - \frac{1}{11}\lambda_1 + \frac{4}{11}\lambda_3 \tag{12}$$

x_2 の範囲は (10)，(11)，(12) からそれぞれ，$x_2 \leq 72$，$x_2 \leq 8$，$x_2 \leq \frac{120}{11}$ が得られる．すべての条件を満たすものは $x_2 \leq 8$ なので，その最大値 $x_2 = 8$ を選ぶ．すると $x_1 = 16$，$x_2 = 8$，$\lambda_1 = 0$，$\lambda_2 = 0$，$\lambda_3 = 8$，このとき目的関数は $min(-64)$ となる．ここで，λ_4，λ_5 は，

後に加えた任意の変数なので，目的関数の最小値は-64であると決定できる．この目的関数は初めに負をかけることで最大化問題を変形したものなので，もう一度正条件に戻すと，この問題の真の目的関数 $max(3x_1+2x_2)$ は x_1= 16，x_2= 8 のとき，満たすといえる．よって最適解は x_1= 16，x_2= 8 のとき最適解 64 を得ることができる．

2.2　シンプレックスタブロー

シンプレックスタブローとは，別名シンプレックス表とも呼ばれ，前述のシンプレックス法をより簡単に，より効率的行うために利用されるものである．まず，シンプレックスタブローを利用するために，Table 2 の表を製作する．Table 2 は最小化問題に変換した目的関数と (1)〜(3) に示された初期の制約条件を忠実に示しており，最適解に近づけながら更新していく．そのアルゴリズムは Fig. 1 のようになる．

STEP1

目的関数の行を各要素のうち，最小の要素を持つ列をL列とする．
その要素が0か正であれば，最適解に達したことになり終了する．

STEP2

L列の制約条件の正のものについて，定数項をその要素で割った値が最小になる行をK行とする．

STEP3

K行L列の要素をピボットとして掃出計算をする．
STEP1に戻る．

掃出計算について

表でi行i列の要素をA(i，j)ということにする．ピボットの要素A(K，L)になるが，その値をＰとする．

・K行以外の行（目的関数も含む）の全要素について，
A(i，j)＝A(i，j)－A(K，j)×A(i，L)／Ｐ

・K行の全要素（定数項も含む）をピボットPで割る．

Fig. 1　シンプレックス法のアルゴリズム

Table 2　シンプレックスタブローの初期状態

	x_1	x_2	λ$_1$	λ$_2$	λ$_3$	定数項
目的関数	-3	-2	0	0	0	0
λ$_1$	4	1	1	0	0	72
λ$_2$	2	2	0	1	0	48
λ$_3$	1	3	0	0	1	48

実際に Table 2 を Fig. 1 に示したアルゴリズムを用いて変形していく．

STEP1 目的関数の行では x_1 での-3が最小なので，L＝1 列となる．-3 は 0 より小さく，まだ最適解にはなっていないので STEP2 へ進む．

STEP2 72/4 = 18，48/2 = 24，48/1 = 48 なので，λ$_1$ の行が K になる．ピボットの値 P は 4 になる．

STEP3　掃出法を行っていく．目的関数の行については，次のようになる．

x_1 の要素：$A(i,j)=-3$, $A(K,j)=4$, $A(i,L)=-3$, $P=4$ なので，

$(-3)-4\times(-3)/4=0$

x_2 の要素：$(-2)-1\times(-3)/4=-5/4$

x_1 の要素：$0-1\times(-3)/4=3/4$

λ$_2$ の要素：$0-0\times(-3)/4=0$

λ$_3$ の要素：$0-0\times(-3)/4=0$

また λ$_2$，λ$_3$ についても同様に操作をすることで Table 3 の結果を得ることができる．

Table 3　シンプレックスタブローの第二段階

	x_1	x_2	λ$_1$	λ$_2$	λ$_3$	定数項
目的関数	0	-5/4	3/4	0	0	54
x_1	1	1/4	1/4	0	0	18
λ$_2$	0	3/2	-1/2	1	0	12
λ$_3$	0	11/4	-1/4	0	1	30

次に目的関数の行での最小値は x_2 の列の-1.25 である．よって L は 2 列となる．Table 3 を求めたときと同様に求めていくと，Table 4 の結果を得ることができる．

Table 4　シンプレックスタブローの第三段階

	x_1	x_2	λ$_1$	λ$_2$	λ$_3$	定数項
目的関数	0	0	1/3	5/6	0	64
x_1	1	0	1/3	-1/6	0	16
x_2	0	1	-1/3	2/3	0	8
λ$_3$	0	0	2/3	-11/6	1	8

よって解として x_1= 16，x_2= 8，最適解 64 を得ることができる．

3　キューンタッカー条件

【キューンタッカー条件】

x^* が制約つき問題の局所的最適解であり，制約条件 c_m のうち，活性な制約条件[1] c_i（$i=1,2,\cdots,l, 0\leq l\leq m$）の勾配ベクトル $f(x^*)$ と c_i（x^*）が，その点において 1

[1]最適解において，等式が成り立つ制約条件

次独立ならば，式 13 が成り立つようなベクトル $u^*{=}(u_1^*, u_2^*,, u_m^*)$ が存在することがいえる．

$$\begin{cases} \triangledown f(x^*) + \sum_{i=1}^{m} u_i^* \triangledown c_i(x^*) = 0 & \\ c_i(x^*) = 0 & (i = 1, 2, , l)) \\ c_i(x^*) \leq 0, u_i^* \geq 0 & (i = l+1, 2, , m) \\ c_i(x^*) > 0, \Rightarrow u_i^* = 0 & (i = l+1, 2, , m) \end{cases} \tag{13}$$

この条件をキューン・タッカー条件という．式 13 は，活性な制約条件 c_i（$i = 1, 2, \cdots, l, 0 \leq l \leq m$）の解 x^* における勾配ベクトル $\triangledown c_i(x^*)$ が 1 次独立の場合，その点における目的関数 $f(x)$ の勾配ベクトル $\triangledown f(x^*)$ と $\triangledown c_i(x^*)$ の和の最適解において，等式が成り立つ制約条件ベクトルをスカラー倍したベクトルが釣り合うことを意味している．式だけでは分かりにくいので，例を用いて説明する．

以下のような目的関数，制約条件を考えます．

$$\begin{cases} \text{目的関数：} & f(x) = (x_1 - 1)^2 + (x_2 - 2)^2 \Rightarrow \text{最小} \\ \text{制約関数：} & c_1(x) = x_1^2 + x_2^2 - 2 \leq 0 \\ & c_2(x) = -x_1 + x_2 \leq 0 \\ & c_3(x) = -x_2 \leq 0 \end{cases} \tag{14}$$

この問題の最適解は目的関数の等高線と実行可能領域が交わった点 $x^* = (1, 1)^\top$ である.（Fig. 2 参照）x^* における目的関数の勾配ベクトルと活性な制約条件 $\triangledown f(x^*)$, $\triangledown c_1(x^*)$，$\triangledown c_2(x^*)$ が釣り合っているような形になっている．

最適解 $(x_1, x_2) = (1, 1)$ を式 14 に代入すると，三つの不等式制約条件はそれぞれ $c_1(x^*) = 0, c_2(x^*) = 0, c_3(x^*) = -1$ となる．このとき，１番目と２番目の制約条件は等式が成り立っていることから，最適解における有効制約条件と呼ぶ．ここで，最適解における目的関数と有効制約の勾配ベクトルに注目すると，Fig. 2 からわかるように三つのベクトル $\triangledown f(x^*), \triangledown c_1(x^*), \triangledown c_2(x^*)$ がつりあい状態になっていることがわかる．このことは，式 15 を満たすような実数 $u_1^* \leq 0, u_2^* \leq 0$ が存在することを示している．

$$\triangledown f(x^*) + u_1^* \triangledown c_1(x^*) + u_2^* \triangledown c_2(x^*) = 0 \tag{15}$$

式 15 は，活性な制約条件のみを含んだ式となっているが，次のように書き換えることができる．

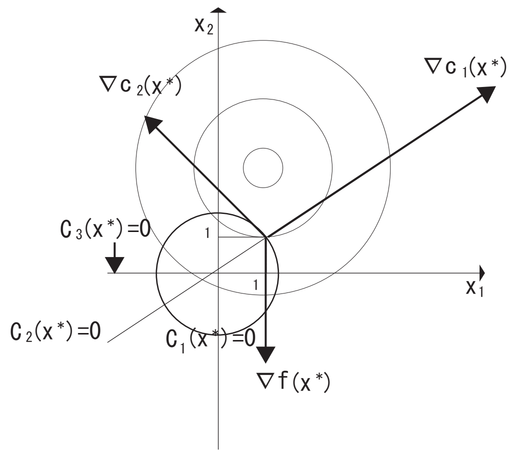

Fig. 2　目的関数と制約関数の勾配

$$\begin{cases} \triangledown f(x^*) + u_1^* \triangledown c_1(x^*) + u_2^* \triangledown c_2(x^*) + u_3^* \triangledown c_3(x^*) = 0 \\ c_i(x^*) = 0 \quad (i = 1, 2) \\ c_i(x^*) \leq 0, u_i^* \geq 0 \quad (i = 3) \\ c_i(x^*) > 0, \Rightarrow u_i^* = 0 \quad (i = 3) \end{cases} \tag{16}$$

式 16 の第 1 式は見かけ上すべての制約関数を含んでいるが，第 16 式より，活性な制約ではない 3 番目の制約条件の係数 u_3^* は 0 になるので，式 (3) と式 (4) は実質的に同じものを表している．そして，式 16 を満たす局所解 $x^* = (1, 1)^\top$ は式 13 を満たしいる．

局所解はキューン・タッカー条件を満たすが，最終的に見つかった解がキューンタッカー条件を満たすからといって，その解が局所解であるかどうかは分からない．ただし局所解は必ずキューンタッカー条件を満たすので，最終的に求まった解が条件を満たしていない場合は確実に局所解ではないことが分かる．この条件により，解候補をある程度挙げることができる．