

離散型資源の公平な配分方法

加藤 直樹

1. はじめに

資源（人，物，機械，お金）を各活動（地域，組織，人）にできるだけ公平に配分したいという状況はしばしば起こる．本稿ではそのための一般的な方法について議論するのであるが，ここでは配分したい資源は1種類であり，その資源量は離散値しかとりえない（たとえば人間をいくつかの場所に配置するとき，配置された人数は整数値でないと意味がない）ものとする．一般的な議論にうつる前に簡単な例題を考えてみることにする．

例1 太郎君（10歳）と次郎君（6歳）がおじいさんから1万円を小遣いとしてもらった．おじいさんは1万円を年齢にできるだけ比例するように分けなさいと言いつ渡した．答えは明らかに，太郎君に6250円，次郎君に3750円である．ところがその場に千円札10枚しかなく，これ以上こまかくすることができなかった．太郎君は自分に7千円，次郎君に3千円といい，次郎君は自分に4千円，太郎君に6千円といって互いにゆずらない．どちらの言い分が妥当なのであろうか．またその根拠は？

このようなもめ事は千円札をもうこれ以上細かくくずすことができないという資源（この場合お金）の不可分性（離散性）から生じている．この

ように，配分したい資源が離散値しかとらない場合，まったく公平に資源配分を行なうことは不可能である．そのような場合，どのような基準で配分を行なえばよいのであろうか．それを考えるために次の例題を考えることにする．

例2 A市には市会議員を選ぶための選挙区が3つある．それを各々1区，2区，3区ということにする．その人口は各々2万4千5百人，10万人，2万人である．市会議員の総定数は10名である．各選挙区の議員定数をその人口にできるだけ比例するよう定めたい．1票の価値を，

選挙区の議員数／選挙区の人口

と定める．市全体の1票の重みは（理想的な1票の重みということにする） $10/144,500 = 6.92 \times 10^{-5}$ である．どの選挙区の1票の重みもちょうどこの値になるようにするには，議員定数を次のようにすればよい．

$$1\text{区}: 24,500 \times 6.92 \times 10^{-5} = 1.70\text{人}$$

$$2\text{区}: 100,000 \times 6.92 \times 10^{-5} = 6.92\text{人}$$

$$3\text{区}: 20,000 \times 6.92 \times 10^{-5} = 1.38\text{人}$$

ところが，議員定数は明らかに正の整数値しかとりえない．したがってどのように配分しても若干の不公平さは残る．そのため次の4つの公平さの基準を設けて議員定数を定めることを考えることにする．第*i*区の議員数を x_i ，人口を p_i とし，理想的な1票の重みを w （ $=1/14,400$ ）と記すことにする．

(i) **分散最小基準**：理想的な1票の重み w と各

選挙区の1票の重みの分散（正確には人口で重み付けをした分散）を最小にする。それは次のように定式化される。

$$(1) \quad \sum_{j=1}^3 p_j (x_j/p_j - w)^2 \longrightarrow \text{最小}$$

$$(2) \quad \text{制約条件: } \sum_{j=1}^3 x_j = 10, x_j: \text{正整数}$$

(ii) **マックスミニ基準**: 選挙区のなかで1票の重みの最小値を最大化する。これはいわゆる弱者救済策で、最も不利な選挙区をできるだけ底上げすることによって全体の公平さのバランスをとろうというものである。これは次のように定式化される。

$$(3) \quad \min_{1 \leq j \leq 3} x_j/p_j \longrightarrow \text{最大}$$

制約条件は(2)と同じ。

(iii) **ミニマックス基準**: 選挙区のなかで1票の重みの最大値を最小化する。最も有利な選挙区の1票の重みをできるだけ引き下げることによって全体の公平さのバランスをとろうというものである。これは次のように定式化される。

$$(4) \quad \max_{1 \leq j \leq 3} x_j/p_j \longrightarrow \text{最小}$$

制約条件は(2)と同じ。

(iv) **最大格差最小基準**: これはいわゆる1票の重みの不均衡の最小化を図るもので、最も有利な選挙区と最も不利な選挙区での1票の重みの比を最小化する。定式化すると次のようになる。

$$(5) \quad (\max_{1 \leq j \leq 3} x_j/p_j) / (\min_{1 \leq j \leq 3} x_j/p_j) \longrightarrow \text{最小}$$

制約条件は(2)と同じ。

以上の4つの基準はいずれも一理あるものである。表1には各基準のもとでの議員定数配分結果を示している。

マックスミニ基準、ミニマックス基準、最大格差最小基準にもとづく議員定数配分案は表1からわかるようにいずれも相異なることに注目してください。

本稿ではここで挙げた4つの基準のうち、どれが最も公平さを表わす基準としてふさわしいかの議論には立ち入らないことにするが、昨今衆議院の定数は正問題が国会や新聞紙上で議論されてい

表1 例2の議員定数配分問題に対する4つの基準による配分結果(カッコ内の数値は1票の重み×10⁴)

	1区 (2万4千5百人)	2区 (10万人)	3区 (2万人)
分散最小基準	2人 (0.816)	7人 (0.7)	1人 (0.5)
マックスミニ基準	2人 (0.816)	6人 (0.6)	2人 (1.0)
ミニマックス基準	1人 (0.408)	8人 (0.8)	1人 (0.5)
最大格差最小基準	2人 (0.816)	7人 (0.7)	1人 (0.5)

るので、そのことについて少し触れたい。国会では8増7減案がすでに国会を通過し、一応最大格差が3倍以内におさまったが、あくまでこれは暫定案であってこのままでは国民の大多数の賛成は到底得られるものではない。現在の選挙区割りを変更することなしに、最大格差がどの程度まで引き下げられるかの検討が[11, 12]によって行なわれ、各選挙区の定数に特に制限をおかない場合、1.79倍まで引き下げられることが明らかになった。越山・一森[12]は法律的観点からいくつかの制約条件を加えた最大格差最小問題を解くことによって、具体的な議員定数配分案を提案している。このあたりの詳しい議論については、機会があれば稿を改めて報告したい。

次節以降では例2で考察した4種の最適化問題を一般化し、それらがどのような方法で効率良く解けるかを順にみていくことにする。アルゴリズムの厳密な正当性の証明はできるだけ避けて、直観的にわかりやすい記述を心がけた。そのため厳密さに欠けるところが目につくかもしれないが、その点はご了承願いたい。

2. 凸型分離形最小化問題

例2の(i)で考察した分散最小基準の最適化問題を次のように一般化して考察する。

$$(6) \quad \text{SR: minimize } \sum_{j=1}^n f_j(x_j)$$

$$(7) \quad \text{sub. to } \sum_{j=1}^n x_j = N,$$

$$(8) \quad x_j: \text{非負整数}, j=1, \dots, n.$$

ここで f_j は凸関数を表わす。 N は総資源量を表

わす正整数定数, n は総活動数を表わす. f_j の増分を

$$(9) \quad d_j(y) = f_j(y) - f_j(y-1), \\ y = 1, 2, \dots, N$$

と定義する. f_j は凸関数であるから,

$$(10) \quad d_j(1) \leq d_j(2) \leq \dots \leq d_j(N)$$

が成り立つ. この性質を利用すると, SR を解く効率の良い解法が構成できる.

その解法は逐次増分法と呼ばれるきわめて簡単な方法である. 次の簡単な例題をもとにその方法をわかりやすく説明する.

例 3 $n=2, N=5, f_1(x_1)=3x_1^2/2, f_2(x_2)=2x_1^3/3$ とする. (9)式より

$$d_1(y) = 3y - 3/2$$

$$d_2(y) = 2y^2 - 2y + 2/3$$

とする. 逐次増分法は初期解 $(x_1, x_2) = (0, 0)$ から出発し, 資源を 1 つずつ追加し, 最終的に最適解を求める. $(x_1, x_2) = (0, 0)$ から x_1 を 1 増やしたときの目標関数値の増分は

$$f_1(1) - f_1(0) = d_1(1) = 3/2$$

であり, x_2 を 1 増やしたときの目標関数値の増分は

$$f_2(1) - f_2(0) = d_2(1) = 2/3$$

である. したがって x_2 を 1 増やした方が目標関数の値は小さくなるので, $(x_1, x_2) = (0, 1)$ とする. さらにここから x_1 を 1 増やした方がいいか, x_2 を 1 増やした方がいいか, 上と同様に考える. すると, $d_1(1) = 3/2 < d_2(2) = 14/3$ であるので, 今度は x_1 を 1 増やし, $(x_1, x_2) = (1, 1)$ を得る. このプロセスを $x_1 + x_2 = 5$ となるまでくりかえすと, $(x_1, x_2) = (3, 2)$ を得て, これが最適解である. 最適解を得るまでのプロセスが図 1 に示されている. アルゴリズムの一般的な記述は次のようになる.

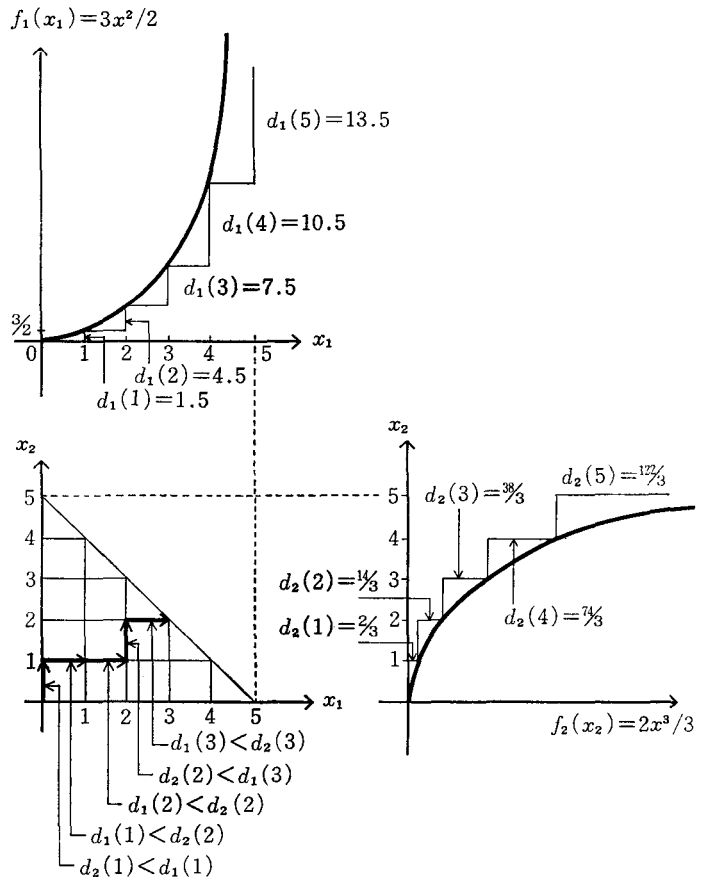


図 1 例 3 を逐次増分法で解くプロセス

Algorithm INCREMENT

Step 1 : $x := (0, 0, \dots, 0)$ とおく.

Step 2 : 次式を満たす j^* を見つける.

$$(15) \quad d_j^*(x_j^* + 1)$$

$$= \min \{ d_j(x_j + 1) \mid x_j < N, j = 1, \dots, n \}$$

$$x_{j^*}^* := x_{j^*}^* + 1 \text{ とおく.}$$

Step 3 : $\sum_{j=1}^n x_j = N$ なら終了. そうでなければ **Step 2** へもどる

このアルゴリズムが常に最適解を与えることの厳密な証明は付録にゆずる. 付録でも証明されているように, その計算手間は $O(N \log n + n)$ であり N が比較的小さい時は, 非常に速い. 計算の複雑さの理論からいうと, このアルゴリズムは正確には多項式アルゴリズムではない. というのは入力データ N は $\log_2 N$ ビットで表現されるので $(N \log n + n)$ は, 入力データ長の指数オーダーとなるから

$j \backslash y$	1	2	3
1	-0.976	-1.284	-0.884
2	-0.160	-1.084	0.116
3	0.656	-0.884	1.116
4	1.472	-0.684	
5		-0.484	
6		-0.284	
7		-0.084	
8		0.116	
9		0.316	
10			

図 2 例 2 (i)における $d_j(y)$ の値 (図の数値は実際の値を 10^4 倍している)

である. 逐次増分法は最初に Gross[7] によって提案されその後 Fox[2], Shih[13]によって再発見されている.[8]には定理 1 の別証明がある.

その後 Katoh, Ibaraki and Mine[9], Galil and Megiddo[5], Frederickson and Johnson[3] によって SR の多項式アルゴリズムが作られた. その中でも[3]によるアルゴリズムは $O(\max\{n, n \log(N/n)\})$ の計算手間をもち, 最も速い.

最後に例 2 (i) では $f_j(x_j) = p_j(x_j/p_j - w)^2$ であるので

$$d_j(y) = (2y-1)/p_j - 2w$$

となり, $d_j(y)$ は図 2 のようになるので, 最適解 $(2, 7, 1)$ を得る.

3. ミニマックスおよびマックスミニ配分問題

例 2 (ii), (iii) で扱った問題はおのおの次の一般形で定式化される.

$$(11) \text{ MAXIMIN: maximize } \min_{1 \leq j \leq n} f_j(x_j)$$

sub. to (7) and (8).

$$(12) \text{ MINIMAX: minimize } \max_{1 \leq j \leq n} f_j(x_j)$$

sub. to (7) and (8).

ここで f_j は非減少関数である.

図 3 には例 3 で扱った問題における $f_1(x_1)$, $f_2(N-x_1)$ をプロットしている. x_1, x_2 が連続量ならば, 明らかに $f_1(x_1) = f_2(N-x_1)$ となる点 $x_1 =$

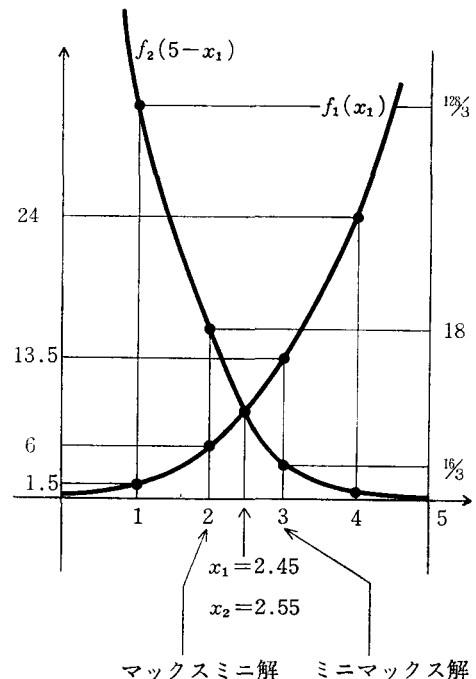


図 3 例 3 におけるミニマックス, マックスミニ解

2.55, $x_2 = 2.45$ が MAXIMIN および MINIMAX の最適解であるが, 残念ながら x_1, x_2 は離散量であるので, MAXIMIN と MINIMAX の最適解は異なり, $(x_1, x_2) = (2, 3)$ が MAXIMIN の最適解, $(x_1, x_2) = (3, 2)$ が MINIMAX の最適解となる.

一般的には MAXIMIN と MINIMAX の最適解は 2 節で扱った問題 SR と同様, 逐次増分法によって求められる. 例 3 に沿ってその方法を具体的に説明しよう.

MAXIMIN では $(x_1, x_2) = (0, 0)$ から出発し, 現在の $f_1(x_1) (=0), f_2(x_2) (=0)$ のうち小さいほうの値を与える x_j を 1 増やす. $f_1(0) = f_2(0) = 0$ であるので, x_1, x_2 のどちらでもいいから 1 増やす. たとえば x_1 を 1 増やすと $(x_1, x_2) = (1, 0)$ を得る. 再びここで $f_1(x_1) (=3/2), f_2(x_2) (=0)$ を求め, その値の小さい方 x_2 を 1 増やす. これを $x_1 + x_2 = 5$ となるまでくりかえすと確かに $(x_1, x_2) = (2, 3)$ を得ることがわかる.

MINIMAX も同様に $(x_1, x_2) = (0, 0)$ から出発

するが、今度は x_j を 1 増やしたときの f_j の値、 $f_1(x_1+1)=f_1(1)=\frac{3}{2}$, $f_2(x_2+1)=f_2(1)=\frac{2}{3}$ を計算し、その値の小さい方 x_2 を 1 増やして $(x_1, x_2)=(0, 1)$ を得る。ここで再び $f_1(x_1+1)=3/2$, $f_2(x_2+1)=16/3$ を計算し、その値の小さい方 x_1 を増 1 やす。これを $x_1+x_2=5$ となるまでくりかえすと、 $(x_1, x_2)=(2, 3)$ を得る。

一般的なアルゴリズムは MAXIMIN (MINI-MAX) に対しては、2 節で述べたアルゴリズム INCREMENT において $d_j(x_j+1)$ をすべて $f_j(x_j)(f_j(x_j+1))$ におきかえるだけでよい。したがって MAXIMIN, MINIMAX とも SR と同じ計算手間で解けると考えてよい。この方法の正当性の証明はここでは省略する (たとえば [11] を参照)。

例 2 (ii) のマックスミニ問題、例 2 (iii) のミニマックス問題もこの逐次増分法を用いて解くと、たしかにマックスミニ解 $(2, 6, 2)$ 、ミニマックス解 $(1, 8, 1)$ を得ることがわかりいただけるであろう。

4. 最大格差最小配分問題

例 2 (iv) で扱った問題は次のように一般化される。

(13) FR:

$$\begin{aligned} & \text{minimize } g(\max_{1 \leq j \leq n} f_j(x_j), \min_{1 \leq j \leq n} f_j(x_j)) \\ & \text{sub. to } (7) \text{ and } (8). \end{aligned}$$

ここで f_j は非減少関数、 $g(u, v)$ は u に関して非減少、 v に関して非増加である。 $g(u, v)$ としてはたとえば $g(u, v)=u/v$ または $u-v$ などが考えられる。例 2 (iv) では $g(u, v)=u/v$ である。

この問題は最初マックスミニ、ミニマックス問題を解き、そこで得られた情報をもとにして、最適解を構成する。MAXIMIN, MINIMAX の最適解に対する目標関数値を v_a, v_b とする。明らかに

$$(14) \quad v_a \leq v_b$$

が成り立つ。また、

$$(15) \quad l_j = \min\{y \mid y=0, 1, \dots, N, f_j(y) \geq v_a\}$$

(16) $u_j = \max\{y \mid y=0, 1, \dots, N, f_j(y) \leq v_b\}$ を $1 \leq j \leq n$ に対して定義する。次の不等式は明らかであろう。

$$(17) \quad \sum_{j=1}^n l_j \leq N \leq \sum_{j=1}^n u_j.$$

補題 1 (15), (16) で定まる l_j, u_j に対し、 $l_j \leq u_j$ ($1 \leq j \leq n$) なら $\sum_{j=1}^n x_j = N$, $l_j \leq x_j \leq u_j$ を満たす任意のベクトル $x=(x_1, \dots, x_n)$ は FR の最適解である。

(証明) (17) より補題の条件を満たすベクトル x は必ず存在する。 x の最適性は $g(u, v)$ の単調性と v_a, v_b の最適性よりしたがう。□

補題の条件は常に満たされるように思いがちであるが資源の離散性ゆえ必ずしもそうとは限らない。離散変数のいやらしいところである。例 3 では図 2 より

$$v_a = f_1(2) = 6, \quad v_b = f_1(3) = 13.5$$

であるので、

$$l_1 = 2, \quad l_2 = 3, \quad u_1 = 3, \quad u_2 = 2$$

となるので、 $l_2 > u_2$ となる。こんな簡単な例でもたしかに補題の条件は満たされていない。以下そのような場合を扱うことにする、話を簡単にするために

(18) $f_j(x_j) \neq f_{j'}(x_{j'})$ ($j \neq j'$ または $x_j \neq x_{j'}$) を仮定する。選挙の問題の場合はこれは満たされていると考えてさしつかえなかろう。

アルゴリズムの正確な記述は後回しにして例 2 (iv) がどのようにして解けるかをみてみよう。 $f_j(x_j) = x_j/P_j$ であり、図 4 にはミニマックス解、マックスミニ解、 v_a, v_b の値、 l_j, u_j ($1 \leq j \leq 3$) の値を示した。図からわかるように $l_1 = 2 > u_1 = 1$, $l_3 = 2 > u_3 = 1$ となり、これもまた補題 1 の条件は残念ながら満たされていない。ここで選挙区を次の 2 つのタイプに分類する。

タイプ 1: $v_a \leq x_j/p_j \leq v_b$ となる正整数 x_j が存在する。すなわち $l_j \leq u_j$ である。

タイプ 2: タイプ 1 以外の選挙区。すなわち $l_j > u_j$ である。

第 2 区はタイプ 1、第 1、3 区はタイプ 2 であ

$$(f_j(x_j) = x_j/r_j)$$

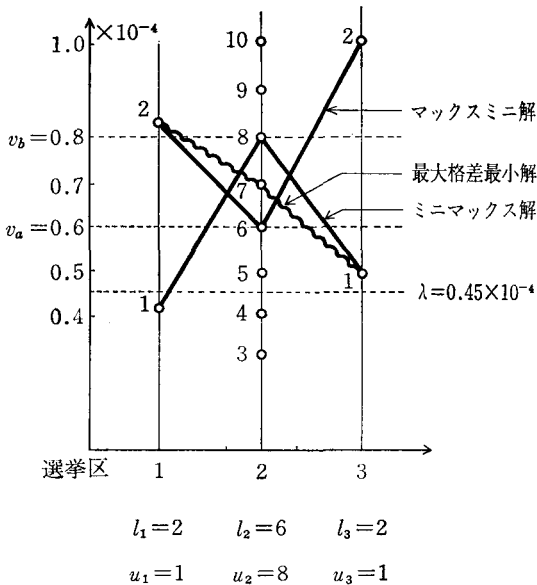


図 4 例 2 (iv) の最大格差最小化問題を解くアルゴリズムの説明図

る。さてここで、次のようにしてミニマックス配分とマックスミニ配分の中間に位置するような定数配分をいくつか生成してやり、ミニマックス配分、マックスミニ配分と合わせた中で最大格差が最小の配分が求める最大格差最小配分となる（厳密な証明は[4]にゆずる）。

(1) 基準となる配分としてミニマックス配分を考える。

(2) タイプ 2 の選挙区の中で基準となる配分における 1 票の重み最小の選挙区を選ぶ。その選挙区の定数を 1 増やし、代りにタイプ 1 の選挙区の中から 1 つ選びその定数を 1 減らす。その選び方は 1 減らした後の 1 票の重みが v_a を下回らないような選挙区を選ぶようにする。このようにして新しく得られた配分を記憶しておく。

(3) (2) で得られた配分がマックスミニ配分であればそれを基準配分とし (2) へゆく。

例 2 (iv) の問題では最初の基準配分は (1, 8, 1) であり、そこでタイプ 2 の選挙区で 1 票の重み最小の選挙 1 区は第 1 区でその値は 4.08×10^{-5} である。したがって第 1 区の定数を 1 増やし、代りにタイ

プ 1 の第 2 区を 1 減らし、(2, 7, 1) という新しい配分を得、これを記憶しておく。この配分を基準配分とし、再び (2) をくりかえすと、第 3 区に定数が 1 追加され、第 2 区の定数が 1 減少し、(2, 6, 2) の配分を得るが、これはマックスミニ配分であるので上の手続きは終了する。このようにして得られた 3 つの定数配分 (1, 8, 1), (2, 7, 1), (2, 6, 2) のなかで最大格差を最小にする配分 (2, 7, 1) が、求める最大格差最小配分となる。

次にアルゴリズムを一般的に書くことにする。

$$(19) \quad I = \{j \mid 1 \leq j \leq n, l_j > u_j\}$$

とする。これはタイプ 2 の選挙区の集合に対応している。(14), (15), (16) より、

$$(20) \quad \begin{cases} l_j - u_j = 1 \\ f_j(u_j) < v_a \leq v_b < f_j(l_j) \end{cases} \quad j \in I$$

$$(21) \quad v_a \leq f_j(l_j) \leq f_j(u_j) \leq v_b \quad j \notin I$$

が成り立つ。(18) の仮定より、マックスミニ解

x^a (ミニマックス解 x^b) において

$$(22) \quad x_j^a \geq l_j, \quad x_j^b \leq u_j \quad j \in I$$

となっていることに注意されたい。というのは、もし $x_j^a < l_j (j \in I)$ なら (20) より、 $f_j(x_j^a) < v_a$ となり、 x^a がマックスミニ配分となることに反する。また $x_j^b > u_j (j \in I)$ なら (20) より $f_j(x_j^b) > v_b$ となり、 x^b がミニマックス配分であることに反する。また細かい証明は省略するが、

$$(23) \quad x_j^a = l_j, \quad x_j^b = u_j \quad j \in I$$

となるマックスミニ解、ミニマックス解 x^a, x^b は必ず存在し、それはたとえば 3 節の逐次増分法で得られることが (18) の仮定を用いると容易に示される。(18) の仮定を除くとこれは一般にはいえない。以下 (23) が成り立っているものとする。

$\{f_j(u_j) \mid j \in I\}$ の値を小さい順に

$$(24) \quad d_1 < d_2 < \dots < d_k \quad (v_a)$$

とする。ただし $k = |I|$ 。アルゴリズムは以下のようになる。

Algorithm FAIR

Step 1 : MAXIMIN, MINIMAX を解き、 v_a, v_b を求める。 l_j, u_j を (15), (16) によって求める。

補題の条件が満たされれば、マックスミニ解またはミニマックス解を最適解として出力し、終了。そうでなければ Step 2 へゆく。

Step 2 : (19)式で定まる集合 I を求め、(24)で定まる $d_1, \dots, d_k$ を求める。 $x = (x_1, \dots, x_n)$ をミニマックス解とし、 x を記憶する。 $i=1, \dots, k$ に対して Step 3 を実行する。

Step 3 : $f_j(x_j) = d_i$ となる $j \in I$ に対して、 $x_j := x_j + 1$ とする。 $x_j' > l_j'$ なる $j' \notin I$ を 1 つ選び、 $x_{j'} = x_{j'} - 1$ とし、新しく得られた解 x を記憶し、Step 3 にもどる。

Step 4 : 記憶された $k+1$ 個の解のなかで目標関数 g を最小にする解を最適解として出力する。終了。□

詳しい説明は省くが、Step 1 は MINIMAX, MAXIMIN を解くのと同じオーダの計算手間を要し、Step 2 はソートが必要なので $O(n \log n)$ を要する。Step 3 を 1 回実行するには n 次元ベクトルを記憶する必要があるので $O(n)$ を要し、Step 3 は高々 n 回くりかえされるので全体で $O(n^2)$ を要する。Step 4 は $O(n)$ を要する。少し工夫すれば Step 3 ではベクトル全体を記憶する必要はなくその目標関数値だけ記憶すれば十分であることがわかるので、全体の計算量はもう少し減らすことができる。結論としては、ミニマックス、マックスミニ問題と同じ計算手間で最大格差最小問題を解くことができる。

(18)の仮定をゆるめても、上のアルゴリズムをほとんど修正することなしに最適解を計算できる。ただ注意すべき点は、Step 1 で求めるミニマックス解 x が、 $x_j = l_j (j \in I)$ を満たすとは限らないということである。これはそれほどやっかいな問題とはならず、 $x_j = l_j (j \in I)$ を満たすミニマックス解は常に存在し、しかも容易に求めることができる。

最大格差最小問題は最初 Zeitlin[14] によって $g(u, v) = u - v$ の特別な形に対して考察され、彼は有限時間アルゴリズムを与えた。その後、加藤、

茨木、三根[10]が(24)式の一般形を考察し、効率の良いアルゴリズムを考えた。その後、最近になって藤重、加藤、一森[4]によってより広い制約条件のもとで、[10]よりも見通しのよいアルゴリズムを与えた。本稿の説明はそのアルゴリズムにしたがっている。アルゴリズムの正当性の証明は[4]、[11]を参照してください。

5. おわりに

本報告では、離散資源の公平な配分方法に関するいくつかの評価規準を議員定数の公平な配分を例にとりて導入し、離散変数をもつ最適化問題として定式化した。そしてそれらを解く効率的解法を紹介した。

このような問題は昔から多くの研究者によって研究されてきているが([8]を参照)、最近の話題としては制約条件をサブモジュラー制約（またはポリマトロイド制約）まで拡張しても同様の効率的解法が構成されるという話がある([4, 6]を参照)が、専門的すぎるのでここではとりあげなかった。最後にこの方面の研究、応用がさらに発展することを期待する。

謝辞 日頃研究上大変お世話になっている関西大学教授（京都大学名誉教授）三根久先生、京都大学教授茨木俊秀先生に感謝いたします。また4章で紹介した研究結果は筑波大学藤重悟先生、大阪工業大学一森哲男先生との共同成果（論文[4]）によるものです。ここで改めてお2人に感謝いたします。

参考文献

- [1] A.V. Aho, J.E. Hopcroft and J.D. Ullman
The Design and Analysis of Computer Algorithms, Reading Mass: Addison-Wesley Co., 1974.
(邦訳「アルゴリズムの設計と解析 I・II」サイエンス社, 1977年).
- [2] B. L. Fox, Discrete optimization via marginal analysis, *Management Science*, Vol. 13

(1966), 210-216.

[3] G.N. Frederickson and D.B. Johnson, The complexity of selection and ranking in $X+Y$ and matrices with sorted columns, *Journal of Computer and System Sciences*, Vol. 24 (1982), 197-208.

[4] S.Fujishige, N. Katoh and T. Ichimori, The fair resource allocation problem with submodular constraints and its solution algorithm, Discussion Paper No. 291, Institute of Socio-Economic Planning, University of Tsukuba, 1986.

[5] Z.Galil and N. Megiddo, A fast selection algorithm and the problem of distribution of effort, *Journal of ACM*, Vol. 26 (1979), 58-64.

[6] H.Groenevelt, Two algorithms for maximizing a separable concave function over a polymatroid feasible region, Working Paper, The Graduate School of Management, The University of Rochester, 1985.

[7] O.Gross, A class of discrete type minimization problems, RM-1644, RAND Corp., 1956.

[8] 茨木俊秀, 資源配分問題, オペレーションズ・リサーチ, Vol. 25, No.12 (1980) 756-764.

[9] N.Katoh, T.Ibaraki and H. Mine, A polynomial time algorithm for the resource allocation problem with a convex objective function, *Journal of Operational Research Society*, Vol.30 (1979), 449-455.

[10] —, An algorithm for the equipollent resource allocation problem, *Mathematics of Operations Research*, Vol. 10 (1985), 44-53.

[11] 加藤直樹, 離散型配分とその応用, 日本OR学会第15回シンポジウム“離散システムとその応用”論文集, (1986), 21-27.

[12] 越山康, 一森哲男, 衆議院議員の定数配分の是正に関する緊急措置, 1986年度OR学会春季研究発表会アブストラクト集.

[13] W.Shih, A new application of incremental analysis in resource allocation, *Operational Research Quarterly*, Vol. 25 (1974), 587-597.

[14] Z.Zeitlin, Minimization of maximum absolute deviation in integers, *Discrete Applied Mathematics*, Vol. 3 (1981), 203-220.

[付録]

問題SR((6), (7), (8)式)が逐次増分法で正しく解けることは次のようにして証明される. はじめに,

$$(25) \quad M = \begin{pmatrix} d_1(1) & d_2(1) & \cdots & d_n(1) \\ d_1(2) & d_2(2) & \cdots & d_n(2) \\ \vdots & \vdots & \ddots & \vdots \\ d_1(N) & d_2(N) & \cdots & d_n(N) \end{pmatrix}$$

なる行列 M を定義する.

定理1 M における最小の N 個の要素の集合 D を作ると(ただし, $d_j(y-1)=d_j(y)$ のとき, $d_j(y-1)$ を優先する), 次式で定義される x^* はSRの最適解となる.

$$(26) \quad x_j^* = \begin{cases} 0, & d_j(1) \notin D \text{ の場合} \\ N, & d_j(N) \in D \text{ の場合} \\ y, & d_j(y) \in D \text{ かつ } d_j(y+1) \notin D \text{ の場合} \end{cases}$$

(証明) (26)式で定義される x^* が最適解でないと仮定する. $\bar{x}$ をSRの最適解で

$$(27) \quad \delta = \sum_{j=1}^n |\bar{x}_j - x_j^*|$$

が最小の $\bar{x}$ を考えよう. $\bar{x} \neq x^*$ であるので, $\sum_{j=1}^n \bar{x}_j = \sum_{j=1}^n x_j^* = N$ より, $x_i^* > \bar{x}_i$, $x_k^* < \bar{x}_k$ なる i, k ($1 \leq i, k \leq n$)が存在する. (10)より,

$$d_i(\bar{x}_i) \leq d_i(\bar{x}_i+1) \leq d_i(x_i^*)$$

$$d_k(x_k^*) \leq d_k(x_k^*+1) \leq d_k(\bar{x}_k)$$

が成り立つ. また $d_k(x_k^*+1) \notin D$, $d_i(x_i^*) \in D$ より,

$$d_i(x_i^*) \leq d_k(x_k^*+1)$$

であるから,

$$(28) \quad d_i(\bar{x}_i+1) \leq d_k(\bar{x}_k)$$

を得る. ここで実行可能解 $x'=(\bar{x}_1, \dots, \bar{x}_{i-1}, \bar{x}_i+1, \bar{x}_{i+1}, \dots, \bar{x}_{k-1}, \bar{x}_k-1, \bar{x}_{k+1}, \dots, \bar{x}_n)$ を考えると,

$$\sum_{j=1}^n f_j(x_j') - \sum_{j=1}^n f_j(\bar{x}_j) = d_i(\bar{x}_i+1) - d_k(\bar{x}_k) \leq 0$$

となる. したがって x' もまた最適解であるが, x' に対する δ ((27)式) は $\bar{x}$ のそれより小さい. これは $\bar{x}$ に対する δ の最小性に反する. $\square$

定理2 Algorithm INCREMENT はSRの最適解を $O(N \log n + n)$ の計算時間で正しく求める.

(証明) Step 2 を k 回実行した後に得られた解を x^k と書くことにする. 集合

$$D^k = \{d_j(y) \mid y=1, 2, \dots, x_j^k, j=1, 2, \dots, n\}$$

が行列 M の k 個の最小の要素からなることを k に関する帰納法によって示す. $k=1$ の時, Step 2 の手続きと, (10) より明らかに D^1 は M の最小の要素からなる. D^k が行列 M の k 個の最小の要素からなることを仮定して, D^{k+1} が M の $k+1$ 個の最小の要素からなることを示す. $k+1$ 回目の Step 2 の実行によって $x_{j^*}^{k+1} := x_{j^*}^k + 1$ とされたとすると $D^{k+1} = D^k \cup \{d_{j^*}(x_{j^*}^k + 1)\}$ である. 帰納法の仮定から,

$$d_{j^*}(x_{j^*}^k + 1) \geq \max \{d_j(x_j^k) \mid x_j^k \geq 1, 1 \leq j \leq n\}$$

であり, Step 2 の手続きより

$$d_{j^*}(x_{j^*}^k + 1) = \min \{d_j(x_j^k + 1) \mid x_j^k < N, 1 \leq j \leq n\}$$

である. したがって (10) の d_j の単調性より $d_{j^*}(x_{j^*}^k + 1)$ は M における $k+1$ 番目に小さい要素である. よって帰納法による証明が完成した. よって定理 1 より, Algorithm INCREMENT が出力する解 x は SR の最適解である.

次に計算手間であるが, $\{d_j(x_j + 1) \mid j=1, \dots, n\}$ をヒープ (heap [1]) などのデータ構造で保持しておけば Step 2 の j^* の計算やそれにとまらぬヒープの修正も $O(\log n)$ 時間で行なわれる. またヒープの初期設定は $O(n)$ 時間でできる. Step 2, 3 のくりかえしは $N-1$ 回であるので全体で $O(N \log n + n)$ 時間を要する. $\square$

● ミニミニOR ●

オレンジ・カードの経済計算

国鉄からオレンジ・カードというものが発売されている. 自動券売機用の磁気カードで, 小銭がなくても切符が買えるから, 少なくとも, そのための券売機が数多く置かれている大都会では至極便利だ.

筆者も金 5,000 円也をだして 1 枚購入した. 300 円分のオマケがついている. 使ってみると, 残額がみるみる減ってゆく様子がはっきりとわかり, 交通費, なかんずく国鉄サンのそれがかなり大きな累積額になっていることがわかった.

こうなると人間ケチになる. 300 円のオマケは乗客にとって得か損か? セコイ話とはいえ, 経済計算をしてみるのも一興であろう.

簡単に言ってしまうえば, 料金を先にとられてしまっても, その金利分がオマケでカバーされれば得になるし, そうでなければ損になる. 使う上での便利さなど他の要因もあろうが, ここではとりあえず経済的要因だけをとりあげることにする.

しかし, これはその乗客が常日頃どのような資金運用をしているのかによる. いいかえれば, その人にとっての金利はいくらかということである. しかし, これは人によってマチマチであるから, 年 1 分から 1 割までの値を想定して, その各々について計算してみることにしよう.

個人差の大きいもう 1 つの要因は使用の頻度である. 買うとすぐに使ってしまう人もあろうし, 何年にもわたって少しずつ使ってゆく人もあろう. その使い方だて時間的にみて均等というわけではなからう.

しかし, 大雑把な見積もりをするために, 毎日一定額分乗車することにして, ちょうど x 日で全額使いきるものとしよう. つまり, 当初獲得した交通費はオマケを含めて 5,300 円分になるから, カードを使わなければ, 毎日 5,300/ x 円を x 日間国鉄サンに支払うことになる.

このようなキャッシュ・フローの現在価値を求めるには, 経済性工学でよく知られている年金現価係数をこの値に乗ずればよい. すなわち,

$$\frac{5,300}{x} \cdot \frac{\{1 - (1+i)^{-x}\}}{i}$$

となる. ここに, i は日歩である.

さて, この現在価値が 5,000 円を上回れば, 乗客にとっては得な買い物, 下回れば損な買い物となるわけだからその損得の別れ目となる日数を計算しておけば, 各乗客に判断の材料が提供できる. つまり

$$5000 = \frac{5,300}{x} \cdot \frac{\{1 - (1+i)^{-x}\}}{i}$$

という方程式を満たす x を求めればよい. 均等に使ってこの期間以内で使いきれば得ということになる.

この方程式を解くには数値的方法を用いなければならぬ. いろいろな方法が考えられるが, ここではごく手軽に, 手近の関数電卓を使って 2 分法を試みた. 結果は表のとおりである. (計算は日歩で行なったが, 表ではわかりやすいように, 年利率に換算しておいた.) ただ, 精度はあまり高くなく, 信頼できるのはせいぜい 1~2 桁であるが, もともと大雑把な見積もりであるから問題にはならないだろう.

オレンジ・カードを買うか買わないか? あとの判断は読者各位におまかせしよう. (からくり堂主人)

年利率	使用期間
1 分	4200 日
2 分	2200 日
3 分	1500 日
4 分	1100 日
5 分	900 日
6 分	700 日
7 分	600 日
8 分	550 日
9 分	500 日
1 割	450 日