Non-Linear Programming の計算法について

穂　鷹　良　介

1. はしがき

L.P. (線型計画法) が出現して以来，simplex method を始めとして，種々の実用的な計算法が案出されている。しかしながら，非線型計画法（以下 N.L.P. と略記する) については，現在，その一般的且つ効率的な解法はないと言って良い。むしろ N.L.P. においては，一般的な解法を求めるより，個々の問題に即して，別々の計算法を組み立てることが，結局，実用的解法として価値のあるもののように思われる。この論文では，始めに N.L.P. の問題の特殊性を比較的考慮に入れない一般的な解法を二種類扱い，次に，若干問題の特殊性を考慮した解法を提案する。前者は，問題の特殊性を考慮しないがために，高速度計算機をもってしても，尨大な計算量，従って，多大な時間が，必要とされることが，実例をもって示される。後者はこの前者の欠点を救うべくして，必然的に出現した計算法である。なお，この論文を通じて計算法は凡て，digital computer にすぐ応用出来ることを考えたために，始めから，変数のとりうる値は discrete なものだけに考察を限っている。また筆者は，実際にこれらの algorithm を高速度 digital computer で解く機会に恵まれた。使用した計算機は HIPAC 103 で coding は筆者によって，HISIP 103 B を利用してなされた。[1] これらの計算結果は，夫々の節の終りにのせられている。

1) 実際の coding に際して，日立製作所の森岡紀夫氏に親切な指導を受けた。ここで感謝の意を表したい。

この論文の作成に当っては，小樽商科大学　古瀬大六教授，北海道大学　山元周行助教授に有益なる助言を頂いた。

2. Systematic and random neighbourhood search method

ここで目的としている問題は，最大値問題ではなくて，極大値問題である。しかし，時間の制約を考慮に入れなければ，この計算法は容易に全域的な最大値問題に作り変えることが出来る。考える問題は次の通りである。

条件　　$x \geqq 0$　　$(x \in R^n)$

　　$g(x) \geqq 0$　　$(g(x) \in R^m)$

の下に　　$f(x)$

を極大にせよ。

本来の目的は $f(x)$ の最大化なのであるが，これは，$f(x)$ に適当な条件（例えば，strict concavity）を与えるとき，上のようにして解かれた問題の解は最大値問題の解になる。

以下に述べる，二つの方法の計算 process は，まず最初に $x^0 \in \{x \in R_+^n | g(x) \geqq 0\} \equiv X$ が与えられたとして，x^0 の近傍の点を適当にさがしてもし $x \in X$ で且つ $f(x) > f(x^0)$ なる点が見つかったならば，新らたに x を出発点として，その近傍の点をさがして，更に $f(x)$ の値を改良する点をさがすというやり方で，従って，次々に得られる改良点 x^ν をとすると，$f(x^\nu)$ は単調増大数列をなすから，$f(x)$ がもし変域 X 内で上に有界ならば，必ず有限 step で計算は改良点のない状態に達する。[2] もし，いくらさがしても，改良点が近傍内になければ，計算は一つの極大点をさがしえたものとして終了する。

ここで提案する systematic neighbourhood（以下 nbd. と略す）search と random nbd. search とは，このある与えられた feasible な点即ち $x \in X$ に対する近傍のさがし方に相違があるだけで，他は同じである。

2) 理論上，もし，変域を連続体等にしておくと，$f(x^\nu) < f(x^{\nu+1}) \leqq M$（$M$ は定数）なる可能性があるが，$(\nu = 1, 2, \cdots\cdots)$ ここでは $f(x)$ のとりうる値も discrete なものと考えているために，上のような心配はない。

2.1 systematic nbd. search

x^0 の近傍は，$\{x\in R|\,\|x-x^0\|\leq\varepsilon\}$ なる集合を包含するから，いつでも上の x^0 の ε- 近傍で，x^0 の近傍の代表と考えておいて差支えない。ここで $\|x\|$ は R^n のノルムを表わしているが，ここではいわゆる maximum norm を採用しておく。これは主に計算上の便宜さから来た理由であって，他にどんな norm を考えても，理論の本質には変りがない。何故なら，有限次元線型位相空間に導入されうる位相は，一種類に限ることが証明出来るからである。[3)]

$V(x^0,\varepsilon)\equiv\{x\in R|\,\|x-x^0\|\leq\varepsilon\}$ の点凡てをさがしつくすことは不可能である。従って，ここでは，$V(x_0,\varepsilon)$ の表面つまり，$\{x\in R|\,\|x-x^0\|=\varepsilon\}$ の点を組織的に調べつくすのである。この表面だけの点をも，凡て調べつくすということは到底出来ることでない。そこで第一近似として次のように考える。説明の便宜上，$x^0=0$ (原点)，$\varepsilon=1$ とする。各次元の $[-1,\ +1]$ を等分割して，得られた小分割の幅を h とする。分割点には，i 番目の座標のときには，$\frac{2}{M}=h$ としたとき

$$-1=a_0^i<-1+h=a_1^i<\cdots\cdots<1-h=a_{M-1}^i<a_M^i=1$$

のように，a_j^i を定める。

$$\mathfrak{A}=(a_{j_1}^1,\ a_{j_2}^2,\cdots\cdots,\ a_{j_n}^n)$$

は，$V(0,\ 1)$ の点であるが，$a_{j_i}^i$ のどれかの絶対値を 1 としておけば，この点 $\mathfrak{A}$ は $V(0,1)$ の表面にのる。ここでとる方法は $\mathfrak{A}$ のどれか一つの component の絶対値を 1 にし，他の element は $(-1,1)$ の中から $a_{j_i}^i$ を選ぶようにして出来る勝手な組合せを順序良く作る方法である。このように $V(0,1)$ の表面を探索するとき，その任意の点は，距離 (maximum norm で定義されたもの) が h 以内の点で近似することが出来る。正確には

定理　すべての $V(0,\ 1)$ の点 x に対して，適当な $\mathfrak{A}=(a_{j_1}^1,\cdots\cdots,\ a_{j_n}^n)$ が

3) [1] p. 53, [2] p. 76 参照.

あって，$\|x-\mathfrak{A}\| \leqq h$，が成立する。

10次元の場合についてこの方法による，近傍の search を $h=0.2$ にとり，$x^0=0$，$\varepsilon=1$，として，HIPAC 103 にやらせて見たところ，近傍の表面を一わたりめぐって歩くのに，約7年間かかることが分った。この方法をもう少し改良した方法で（本質的には同じ）2二次元の問題を解いた結果が次の二例である。

なお x の initial condition は二例とも (0, 0) である。

[問題 1]

maximize $\quad 4x_1+3x_2$

subject to $\quad 2x_1+3x_2 \leqq 6$

$\qquad\qquad 2x_1+x_2 \leqq 4$

$\qquad\qquad x_1 \geqq 0,\ \ x_2 \geqq 0$

結果は第1表に示す通りで，h の値を小にするほど，解が精密になっていく。

第 1 表

	Iterationの 回数	x_1	x_2	$f(x)$
$h=1$	1	1.0	0.1	4.3
	2	1.25	1.1	8.3
$h=\frac{1}{100}$	1	1.26	1.09	8.31
	2	1.27	1.08	8.32
	3	1.28	1.07	8.33
	⋮	⋮	⋮	⋮
	10	1.35	1.0	8.3999999999
	11	1.36	0.99	8.41
	⋮	⋮	⋮	⋮
	70	1.4976	1.0	8.9904
$h=\frac{1}{10000}$	1	1.4977	0.9999	8.9905
	2	1.4978	0.9998	8.9906
	⋮	⋮	⋮	⋮
	4400	1.499998	1.00	8.999992

正解は　$x_1=1.5,\ x_2=1,\ f(x)=9$

であるが，$h=\frac{1}{10000}$でかなりの精度がある。

[問題 2]

$$\max \quad f(x)=-x_1^2+2x_1-x_2^2+2x_2$$

$$\text{subject to} \quad \begin{cases} 2x_1+3x_2\leq 6 \\ 2x_1+x_2\leq 4 \\ x_1,\ x_2\geq 0 \end{cases}$$

この計算結果は第2表に示す通りで，この方法が non-linear な目的函数の際にもうまく動いたことが分る。正解は $x_1=x_2=1,\ f(x)=2$ である。

第 2 表

	Iteration の回数	x_1	x_2	$f(x)$
$h=1$	1	1.0	0.1	1.19
	2	1.25	1.1	1.9275
$h=\frac{1}{100}$	1	1.2415	1.11	1.92957775
	2	1.233	1.12	1.931311
	⋮	⋮	⋮	⋮
	157	0.996499999	0.997499999	1.9999815
$h=\frac{1}{10000}$	1	0.9965999	0.997399	1.99998168
	2	0.9966999	0.997299	1.99998182
	⋮	⋮	⋮	⋮
	208	0.9999899942	0.99995499	1.99999999979

2.2 random nbd. search

§2.1 で扱った近傍の search の仕方を乱数を作ることに変更したのがこの方法である。

乱数の作り方は von Neumann の創始したという二乗採中法を応用したもので，あるでたらめな数(2進法で 48 桁)を二乗して，その真中の 48 桁をとることによって [−1, 1] 区間に一様に分布する乱数を得た。その一様性の

テストとして，このようにして作られた乱数10000個を $[-1, +1]$ を100等分してヒストグラムを作った。第3表で，左上から順に，このようにして作られた乱数が $\left[-1, -1+\frac{1}{100}\right)$, $\left[-1+\frac{1}{100}, -1+\frac{2}{100}\right)$, …… $\left[1-\frac{1}{100}, +1\right]$ なる区間に入った回数を示している。なお，+. 970+02 は 0.97×10^2，+. 106+03 は 0.106×10^3 の表現方法である。これによって，χ^2-test による test をすると自由度は $100-1=99$ で，このとき χ^2- 分布はほ

第 3 表 乱 数 の 一 様 性 テ ス ト

+.106+03	+.880+02	+.109+03	+.990+02	+.960+02
+.104+03	+.113+03	+.820+02	+.940+02	+.990+02
+.910+02	+.930+02	+.100+03	+.100+03	+.104+03
+.980+02	+.910+02	+.850+02	+.113+03	+.900+02
+.950+02	+.790+02	+.102+03	+.115+03	+.106+03
+.125+03	+.106+03	+.116+03	+.102+03	+.950+02
+.109+03	+.100+03	+.120+03	+.112+03	+.910+02
+.950+02	+.860+02	+.103+03	+.103+03	+.104+03
+.930+02	+.117+03	+.110+03	+.108+03	+.930+02
+.121+03	+.104+03	+.101+03	+.111+03	+.990+02
+.109+03	+.122+03	+.930+02	+.104+03	+.870+02
+.990+02	+.108+03	+.890+02	+.930+02	+.102+03
+.990+02	+.980+02	+.930+02	+.940+02	+.102+03
+.950+02	+.920+02	+.890+02	+.102+03	+.910+02
+.106+03	+.950+02	+.102+03	+.103+03	+.920+02
+.960+02	+.940+02	+.950+02	+.101+03	+.900+02
+.970+02	+.870+02	+.970+02	+.950+02	+.950+02
+.990+02	+.950+02	+.118+03	+.980+02	+.970+02
+.110+03	+.970+02	+.112+03	+.100+03	+.860+02
+.100+03	+.113+03	+.105+03	+.100+03	+.880+02

CHI SQARE=+.84480000000+02

ぼ正規分布と見做して，片側テストを行なうと，$\chi^2=\sum_{i=1}^{100}\frac{(f_i-100)^2}{100}=84.48$ となり，有意水準5%でも，この乱数が一様分布からの sample であるという仮説は棄却出来ない。以上の乱数は +314159265358979 という数を2進法に直して二乗採中法によったものであるが，他にでたらめな数として，+21414213560787 を最初の数としてやって見た所 χ^2 は 97.02 を得た。この場合も仮説は棄却出来ない。このようにして二乗採中法による乱数は乱数としての資格を持つことが分ったので，これを用いて計算を次のように進める。簡単のため例を $x^0=0$ の場合とする。原点の近傍は $V(0,\varepsilon)=\{x|\|x\|\leq\varepsilon\}$ であるが，この近傍の探索として上のようにして作られる乱数列を $\theta_\nu(\nu=1,2,\cdots\cdots)$ とするとき，$x^1=(\theta_1,\ \theta_2\ \cdots\cdots,\ \theta_n)$ $x^2=(\theta_{n+1},\ \theta_{n+2}\cdots\cdots,\ \theta_{2n})\cdots\cdots x_\nu=(\theta_{(\nu-1)n},\ \theta_{(\nu-1)n+1},\ \cdots\cdots,\ \theta_{\nu_n})\cdots\cdots$ のような n 次元 vector $\{x^\nu\}$ に ε の scalar multiplication を施したものを採用するのである。勿論 $\varepsilon x_\nu\in V(0,\ \varepsilon)$ で，上に述べたように $\{\theta_\nu\}$ が一様な乱数のときには，$V(0,\ \varepsilon)$ 内を一様に探索することが期待される。この方法を用いて計算した例を次に示す。

[問題 3]

$$\begin{aligned}
&\max && -x_1^2-2x_2^2+2x_1+4x_2\\
&\text{subject to} && 6x_1+3x_2\leqq 18\\
&&& 4x_1+5x_2\leqq 20\\
&&& 7x_1+2x_2\leqq 14\\
&&& x_1,\ x_2\geqq 0
\end{aligned}$$

この正解は $x_1=1$，$x_2=1$，$f(x)=3$ であるが途中の approximation の状態は第4表にある通りである。これを2次元にグラフを書いて見ると，systematic search と異って，乱数独特のジグザグ運動をしながら最適解に近ずく模様が観察出来る。なお initial condition は (0, 0) である。

なお，詳述は避けるが，L.P. 問題に対してもこの方法は N.L.P. と同様に解くことが出来た。いずれの場合にも，函数形等の知識を使っていないので，一般的ではあるが，あまり効率的な解法ではなく，十分な精度に迄収束

第 4 表

	Iteration の 数	x_1	x_2	$f(x)$
$h=1$	1	0.0011978	0.65733	1.767548
	2	0.9250086	0.509436	2.5130709
	3	0.7668044	0.977303	2.9445896
	4	0.9191936	0.889562	2.9690776
	⋮	⋮	⋮	⋮
	12	1.0137148	0.9952382	2.999766
$h=\frac{1}{10}$	1	1.0041928	0.9937598	2.999904
	2	0.9954970	1.000465	2.9999792
	⋮	⋮	⋮	⋮
	5	1.0014668	0.999575	2.9999974

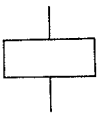
させるためには二次元で数十分を要した。

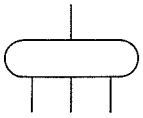
3. 資源配分問題型の N.L.P. 解法

§2 で論じたように systematic nbd. search も random nbd. search も共に一般的であるが，函数形の知識を利用していないので時間がかかるという欠点があった。この欠点をいくらかでも埋めるために考えられたのが次に述べる計算法である

この計算法の数学的表現はおそらく定差方程式で与えることが可能であろうと思われるが，ここでは，これも良く計算機の programming もしくは coding に採用されている flow-chart もしくは block-diagram の形で述べる。この方がより直感的に理解しやすい表現方法であり，また実際の programming もしくは coding と直結していると思われるからである。以下はこの block-diagram の図式の説明である。

1. flow-chart の流れの向きは原則として上から下へ縦軸方向に向う。
2. $A=B$ は A が B によって代入されることを示す。

3. 
操作 box；この過程で演算，代入等の操作がさされることを示す。

4. 
条件判定 box；一つの入口から入った情報に対していくつかの条件に従って流れが分かれることを示す。

さて我々が考えている N.L.P. のここでの形は，第2節のものと同じで再説すれば次の如くである。

[問題 A]

$$\begin{aligned} &\text{maximize} && f(x) \\ &\text{subject to} && g(x)\geqq 0 \\ & && x\geqq 0 \\ & && x\in R^n \quad g(x)\in R^m \end{aligned}$$

ここで R^n は n 次元ユークリッド空間である。又，$x\geqq x'$ は各座標 x_j，x'_j について同様の不等式が成立することを表わし，e_j は単位ベクトル $(0,0,\cdots,\overset{j}{1},\cdots 0)$ を表わす。

問題 A は次のような経済 model と解釈出来る。(そのためには色々と函数形についての制限，条件が課せられるべきであるが，その吟味はここでは省く) x はある経済主体の activity を表現し，それは更に n 個の数，x_1, x_2, ……x_n で完全に記述しつくされるものとする。この x_1, ……，x_n を一まとめにしたもの $x=(x_1, x_2, \cdots\cdots x_n)$ に対して，この経済主体の目的とする成果が，これも又，実数で測定出来るとし，$f(x)$ で与えられるものとする。当然経済主体は $f(x)$ を最大ならしめるように行動する訳であるが，それは無制限にではなく，経済的に意味のある変数という意味で，$x\geqq 0$ の条件が，更に，資源，能力等の制限として $g(x)\geqq 0$ が課せられている。$g(x)=(g_1(x), \cdots\cdots, g_m(x))$ はその意味を考えるならば，経済主体 e が x なる activity を行なう時の，資源ないしは能力等の余剰を示す。$g_1(x)$, ……，$g_m(x)$ を一まとめにして，$g(x)=(g_1(x), \cdots\cdots, g_m(x))$ とベクトル表示したものである。

我々の考えた計算法は，まず feasible な activity $x^0(x^0\geqq 0,\ g(x^0)\geqq 0$ な

るとき x^0 を feasible と呼ぶ) が与えられたときに，これを出発点として，変域内での $f(x)$ の値を大きくする点を次々に見つけて行って，行く所迄行きついた時に，終るのである。 従って，$f(x)$, もしくは $g(x)$ の定義如何によっては，最大値を必ず見つけうる計算法ではなく，$f(x)$, $g(x)$ に何等かの制限を加えなければいけない。しかしながら，ここでは，その証明は省略し，計算法とその経済的意味とを同時に説明し，更に，実例について述べることにする。計算手続の block-diagram は第1図に示す通りである。

ある feasible point x が与えられたとき，もし更に feasible でしかも $f(x')>f(x)$ となる x' が何等かの方法で見出されたとき，この計算法ではあらたな出発点として，x' を考えることにしていることは第2節と同様である。以下第1図の各段階を順を追って説明する。

第1段階　Gradient Motion

与えられた feasible な activity x に対して，その第 j 成分を h だけ増加させたときの収穫の増分は $\Delta f_j \equiv f(x_1, \cdots\cdots x_j+h, \cdots\cdots, x_n)-f(x_1, \cdots\cdots x_n) = f(x+e_jh)-f(x)$ と考えられる。 $\Delta f_j>0$ の場合には，当然経済主体 e は activity の第 j component x_j をふやそうと考えるであろう。各成分について同様の考察を行なえば

$$x'_j = x_j+\theta[f(x+e_jh)-f(x)]$$

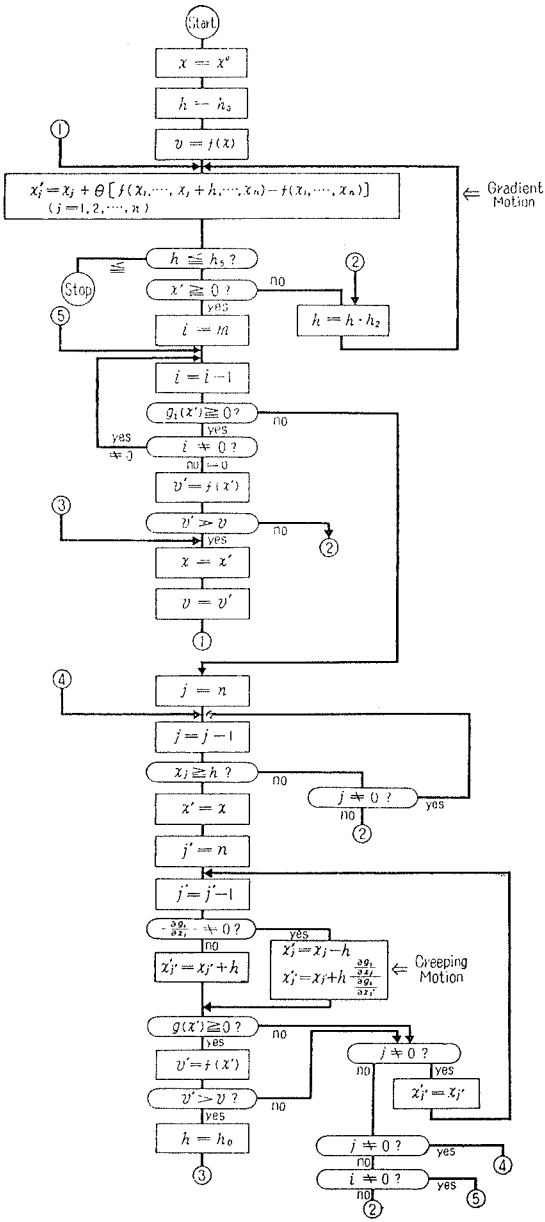
$$(j=1, 2, \cdots\cdots, n)$$

なる成分をもつ x' を，経済主体 e は取ることが考えられる。ここでは θ は適当な正の反応係数である。 そして，この新らたな x' が feasible で且つ，$f(x')>f(x)$ ならば，経済行動の次の立脚点として x の代りに x' を代入して，最初から始める。

以上の process において，Δf_i は近似的には $\dfrac{\partial f}{\partial x_i}\cdot h$ に等しいから，f の勾配を考えることになり，Gradient Motion と呼んで良かろう。

第2段階　Creeping Motion

第1段階による改良解の模索過程が，達成されない場合が三つある。第



第 1 図　Block Diagram of the program

1は以上のようにして作った x' が $x' \geqq 0$ を充たさぬ場合，第2は $g(x') \geqq 0$ が成立しない場合，即ち，ある i に対して $g_i(x') < 0$ となる場合である。以上の二つは，trial に作った x' の feasibility が充たされぬ場合である。最後の第三の場合は x' が，改良解としての条件 $f(x') > f(x)$ を充たさぬ場合である。

x がすでに最適解である場合もしくは，最適解に非常に近い場合を除いて，第1段階をパス出来ない場合，以上三つの場合が一般には起りうる。第1の場合は，$f(x)$ が x の各成分に対して単調増加であるような通常の場合には起り得ない。($f(x)$ は収穫量を示し，activity の各成分が増加すれば増加するものとここでは仮定する。) 第2の場合，即ち，$g(x') \geqq 0$ が成立しないときには，ある資源もしくは能力 i に対して $g_i(x') < 0$ なる時である。つまり，新らたな activity x' に従うと第 i 番目の資源に不足を来すことを意味する。このときに考えられる改良解を求める方法としては，$x_j \geqq h$ を充たす x_j のどれでも勝手な j について $x'_j = x_j - h$，また，j と異なる勝手な j' に対して $x'_{j'} = x_{j'} + h \cdot \left(\frac{\partial g_i}{\partial x_j} \Big/ \frac{\partial g_i}{\partial x_{j'}} \right)$ として見ることである。(但し $\frac{\partial g_i}{\partial x_{j'}} = 0$ の場合には別の吟味を要するので後述) この経済的意味は次の通りである。x_j を h だけ減少させることによって，現在稀少となっていた資源 i を

$$g_i(x - e_j h) - g_i(x)$$

だけ使わないで済むことになる。この余った資源をもし activity の第 j' 成分の増加によって使い果すとするといくら j' 成分を増すことが出来るかを考えて見る。各 i に対して $g_i(x)$ の偏導函数は凡て連続と仮定すると，第 j' 成分を h' だけ増加させることによって必要な資源の量は

$$\begin{aligned} & g_i(x - e_j h) - g_i(x - e_j h + e_{j'} h') \\ & = -(g_i(x_1, \cdots\cdots x_j - h, \cdots\cdots, x_{j'} + h', \cdots\cdots, x_n) - g_i(x_1, \cdots\cdots, x_j - h, \\ & \cdots\cdots, x_n)) \\ & \fallingdotseq -\frac{\partial g_i(x)}{\partial x_{j'}}\Big|_{x = x - e_j h} \cdot h' \fallingdotseq -\frac{\partial g_i(x)}{\partial x_{j'}} \cdot h' \end{aligned}$$

他方　$g_i(x-e_jh)-g_i(x)\fallingdotseq -\dfrac{\partial g_i}{\partial x_j}\cdot h$

この両者を等しいものと置くと，

$$\frac{\partial g_i}{\partial x_{j'}}\neq 0 \text{ のときに } h'=h\left(\frac{\partial g_i}{\partial x_j}\Big/\frac{\partial g_i}{\partial x_{j'}}\right)$$

を得る。よって，$x_{j'}=x_j+h'$ とすれば良い。$x'=(x_1, \cdots, x_j-h, \cdots, x_{j'}+h', \cdots, x_n)$ に対して第一段階の後に行なったと同様の feasibility の吟味を行なう。ここで述べた motion は制限となっている不等式の壁に沿って動く motion となるため，creeping motion と呼んで良かろう。

上で $\dfrac{\partial g_i}{\partial x_{j'}}=0$ となる場合は，activity の第 j' 成分が若干ふえても第 i 資源を使わない場合と解釈出来る。従って，$x'_{j'}=x_{j'}+h$ という動きをさせることにする。この操作が微妙に作用する例を以下の例問題5で見る。計算の難かしい所は，初期条件が変域の特殊な境界に位置する場合等であった。この計算法では，上に述べた creeping motion がこの困難さを克服した。第二節で述べた方法に頼るとき，もし初期条件が，変域の隅の方にあると，次の改良解を見出すのに狭い範囲の中からさがすので非常に時間がかかったのである。

[問題 4]

max　　$4x_1+x_2$

subject to　　$\begin{pmatrix}6 & 3\\ 4 & 5\\ 7 & 2\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}\leqq\begin{pmatrix}18\\ 20\\ 14\end{pmatrix}$

$x_1,\ x_2\geqq 0$

初期条件　　$x^0=(0,\ 4)$

x^0 は変域の一方の隅にあって，改良解は常に制限不等式の壁に沿って見出されている。計算過程は第5表に示す通りである。

x_1 軸を横軸，x_2 軸を縦軸にとって，これらの点を plot して見れば，点は左上の変域から壁を伝って，右下の変域の角に迄下りてきていることが分

第 5 表

Iteration 回数	x_1	x_2	$f(x)$
0	0	4	4
1	0.285714	3.0	4.14286
2	0.685714	3.1	5.84286
3	0.714286	3.0	5.85714
4	0.742857	2.9	5.87143
5	1.14286	3.0	7.57143
⋮	⋮	⋮	⋮
10	1.28571	2.5	7.64286
11	1.31429	2.4	7.65714
12	1.34286	2.3	7.67143
13	1.37143	2.2	7.68571
⋮	⋮	⋮	⋮
最 終 解	2.0	0.00000999	8.00000

る。勿論，正解は $x_1=2$, $x_2=0$, $f(x)=8$ である。x_2 は計算結果では 10^{-5} の order に迄小さくなっているから，0 と考えて良い。勿論この order は flow-chart 上での h_5 に依存する訳で，この計算例では $h_5=10^{-6}$ としたためにこの答を得たのである。

[問題 5]

maximize $\quad 4x_1+2x_2$

subject to $\quad \begin{pmatrix}0 & 1\\ 4 & 5\\ 7 & 3\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}\leqq\begin{pmatrix}3\\ 20\\ 21\end{pmatrix}$

$$x_1,\ x_2\geqq 0$$

初期条件 $\quad x^0=(0,\ 0)$

この問題は第二段階の creeping motion で問題となった $\frac{\partial g_i}{\partial x_{j'}}=0$ の成立する場合である。$i=1$, $j'=1$ の場合 $g_1=3-x_2$ であるから $\frac{\partial g_1}{\partial x_1}=0$ である。つまり，activity 1 は，資源 1 を全然必要としないのである。この際 activity 2 を h だけ節約するしないに拘わらず activity 1 は他の制約にぶつか

らぬ範囲内で増加させうる訳である。計算過程及び $h_5=10^{-6}$ での最終結果は第6表に示す通りである。

第 6 表

Iteration 回数	x_1	x_2	$f(x)$
0	0	0	0
1	0.3	0.5	3.4
2	0.6	1.0	6.8
3	0.9	1.5	10.2
4	1.2	2.0	13.6
5	1.5	2.5	17.0
6	1.4	2.73333	17.8667
7	1.3	2.81333	17.9667
8	1.2	2.89333	18.0667
9	1.1	2.97333	18.1667
10	1.2	2.97333	18.4667
⋮	⋮	⋮	⋮
不　明	1.248	2.99893	18.7387
不　明	1.247	2.99973	18.7397
⋮	⋮	⋮	⋮
最終解	1.25	3.0	18.75

正解は　$x_1=1.25$,　$x_2=3$,　$f(x)=18.75$ である。

[問題 6]

$$\max \quad 4x_1+5x_2+3x_3+2x_4+10x_5$$

$$\text{subject to} \quad \begin{pmatrix}3&0&2&0&6\\1&1&0&4&4\\2&2&5&1&0\end{pmatrix}\begin{pmatrix}x_1\\\vdots\\x_5\end{pmatrix}\leqq\begin{pmatrix}24\\8\\45\end{pmatrix}$$

$$x_1,\ \cdots\cdots,\ x_5\geqq 0$$

これは単純な L.P. 問題であるが，20分間では $h_5=10^{-6}$ のとき最終解にまで到達出来ず，途中で打切ったのであるが，その時の答として，$x=(2.96\times10^{-10},\ 7.9967,\ 5.801,\ 5.99\times10^{-4},\ 3.58\times10^{-10})$ $f(x)=57.3877$ を得た，正解は $x=(0, 8, 5.8,\ 0, 0)$ $f(x)=57.4$ である。

以上で，具体的な計算結果の詳述は終るが，非線型目的函数として，$x'Ax+B'x$ のようなものを $b-Cx \geqq 0\, x \geqq 0$ で最大化する類の二，三の例題についても，この計算法は，予想された精度で正解への近似解を与えた。

4. あ と が き

第3節で述べた解法は，筆者が第2節で試みた方法に計算時間の点であき足らなく思い，考案した方法であったが，確かに，§3でとり扱った種類の例題では計算時間を1/3～1/4に縮少し得たと思う。しかしながら，後者の方法ですら，5次元で，制約不等式を3個にした時，満足な精度のものを得るのに30分位はかかる。§2で述べた方法は，これに比し，更に莫大な時間を必要とするが，random search の方法を使えば，問題の特殊性をあまり考慮しないでも容易に適用出来るから，将来において，更に高速，大記憶容量の計算機が出現すれば，実用的計算法となる可能性がある。

1962. Dec. 10.

参 照 文 献

[1] K. J. Arrow and others: *Studies in Linear and Non-linear Programming*, Stanford Univ. Press. 1958.

[2] 二階堂副包「現代経済学の数学的方法」岩波書店 昭和34年