数理解析研究所講究録
第359巻 1979年 40-58

最適化手法の比較

原研 東海研　　鈴木忠和

最適化問題は数学的には数理計画法の問題として扱われ，一般に以下のように定式化される。

$$\left.\begin{array}{l} \underset{x \in E^n}{\text{Minimize}}\ F(x) \\ \\ \text{subject to} \\ \quad x \in R \end{array}\right\} \qquad (1)$$

（1）においてFは最適性の基準で，目的関数と呼ばれる。またRは制約領域であり，一般的には制約条件式と呼ばれる関数の集まりである。最適化問題はこのFやRの形によって

（I）線形の問題

（II）非線形で制約条件を持たない問題

（III）非線形で制約条件を持つ問題

の三通りに分類することができる。（I）は線形計画法の問題

であり，数理計画法の問題の中でも最も基礎的な問題である。この場合目的関数や制約条件式が独立変数 $x=(x_1,\cdots,x_n)^t$ の一次式で表わされ，

$$\left.\begin{aligned} &\underset{x\in E^n}{\text{Minimize}}\quad F(x)=c^t\cdot x \\ &\text{subject to} \\ &\qquad\qquad Ax=b \\ &\qquad\qquad x\geqq 0 \end{aligned}\right\}\quad(2)$$

という形になる。但し A は $m\times n$ 係数行列で，コスト係数 c，右辺ベクトル b はそれぞれ n, m 次元列ベクトルで，肩の t は転置を意味している。

問題(2)に対する解法としては Dantzig によって開発された単体法という確固としたものがあり，(2)が有界な実行可能領域を持つなら有限回の探索で最適解が得られるという保証が与えられている。したがって非線形問題と比べると解法上の問題点はなく，ただ大規模な問題や整数計画問題に対して単体法を効率的に適用するための問題操作が提案されている。

(ii)は変数 x の制約がなく，最適性の基準だけが与えられている問題で，非線形計画法の特殊な場合と考えられている。解法は目的関数の形に応じて広く研究されており，それら

は大別して直接探索法（direct search method）と傾斜法あるいは降下法（descent search method）に分類される。

直接探索法は一般に過去の反復で得られた情報を使わないことが多く，各反復では目的関数の値だけが計算される。したがって記憶容量が少なくて済むし，アルゴリズムも簡単で使い易いという利点を持っている。原研ではALPS，ROTAX，ALSIM，SPASE，CORASEの5つのプログラムが開発・整備されている。

ALPSはHookeとJeevesにより紹介されたパターン探索法に基づくプログラムで，これは与えられた基点の回りでの目的関数の局所的な動きを探索するパターン決定と，その局所的な探索により到達した点から大域的な動きを探索するために次の基点へと移動するパターン移動という二つの操作から成っている。

ROTAXはRosenbrockによって紹介された座標回転法を用いたプログラムで，変数空間での直交座標系の1つの座標軸が目的関数の最急降下方向に向くように座標系を回転させながら探索を行なうものである。

ALSIMはNelderとMeadによって開発されたシンプレックス法を採用したプログラムで，変数空間に張られる正則単体の各端点上での目的関数値を比較しながら，この単体

を求める最適点へと動かしていく方法で，この移動は折り返し(reflection)，縮少(contraction)，拡大(expansion)という三つの基本操作により行なわれる。

SPASE は筆者により提案された球面探索法(spherical search method)によるプログラムで，変数空間に生成された超球面上に一様に分布する探索点を発生させ，最小値を与える点と球の中心を結ぶ方向に球を移動させながら最適点を探索していこうというものである。

CORASE は Price によって紹介された制御ランダム法(controlled random search procedure)を用いたプログラムで，変数空間内に十分多くの点をランダムに生成し，それらの点の中から任意の点を選んで単体を生成し，その極点を重心に関して折り返しながら最適点の探索を行なうもので，従来のランダム探索法と，前述のシンプレックス法を組み合わせた手法と考えられる。

一方降下法は直接探索法と比べ，各反復の過程で用いる情報量が多く，したがって1回の反復で目的関数をより良く改善できるという利点を持っている。この場合の反復計算は三つの部分から構成されている。即ち，

1. 降下方向の計算。
2. 降下方向に沿った降下ステップ刻の幅の計算。

3. 求められた降下方向と刻み幅から次の点を計算する。

このような三つの部分から構成される降下法は方向の定め方，刻み幅の定め方によって種々の方法が提案されている。

計量行列を単位行列とし，方向として目的関数が局所的に最も減少する方向 $-\nabla F(x^k)$ と定めるのが最急降下法（steepest descent method）である。Shahらはこれをさらに加速した傾斜PARTAN法を提案した。計量行列としてヘッセ行列を用い，降下方向として

$$s^k = -H(x^k)^{-1}\cdot\nabla F(x^k)$$

を用いるのがNewton法である。但し上式において $H(x^k)$ は点 x^k におけるヘッセ行列，$\nabla F(x^k)$ は F の傾斜ベクトルである。また各反復毎に計量行列を前の計量行列と，

$$\Delta x^k = x^k - x^{k-1}$$

$$y^k = \nabla F(x^k) - \nabla F(x^{k-1})$$

の関数として更新していくのが可変計量法である。この他共役方向法，共役傾斜法などが良く知られている。なお原研においては共役方向法によるALCODR，共役傾斜法によるVA08A，CGD，PARTAN法によるALPART，修正Newton法によるMINIM，可変計量法によるVA01A，VA06A，VA09A，VA10A，FPD，ALVAMなどが整備されている。

以上の制約条件を持たない問題に対する最適化プログラム

の収束効率，安定性を調べるために用いたテスト関数は，

1. Rosenbrock の関数

$$F(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

2. Beale の関数

$$F(x) = \sum_{i=1}^{3} \{c_i - x_1(1 - x_2^i)\}^2, \quad c_i : \text{given}$$

3. Box の関数

$$F(x) = \sum_{i=1}^{10} \{(e^{-x_1 y_i} - e^{-x_2 y_i}) - x_3(e^{-y_i} - e^{-10 y_i})\}^2,$$

y_i : given

4. Enzyme の関数

$$F(x) = \sum_{i=1}^{11} \{V_i - x_1(y_i^2 + x_2 y_i)/(y_i^2 + x_3 y_i + x_4)\}^2$$

V_i, y_i : given

5. Watson の関数

$$F(x) = \sum_{i=1}^{30} \{\sum_{j=1}^{6} (j-1) x_j y_i^{j-2} - (\sum_{j=1}^{m} x_j y_i^{j-1})^2\}^2 + x_1^2$$

y_i : given

6. Powell の関数

$$F(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

7. Wood の関数

$$F(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2) + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$$

8. Gauss分布関数

$$F(x) = \sum_{i=1}^{15} \{ x_1 \cdot e^{-(z_i - x_3)^2 \cdot x_2/2} - y_i \}^2 .$$

$z_i,\ y_i$: given

9. 拡張されたRosenbrockの関数

$$F(x) = \sum_{i=1}^{n/2} \{ 100 (x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2 \}$$

の9つで，いずれも最小値とその点が既知な関数である。テストは各プログラムの出発点への依存性（安定性）を調べるため，出発点は問題1〜8に対してランダムに10点ずつ生成した。また収束効率を調べるために

$$C.E. = \ln \frac{|FO - FM|}{|FF - FM|} / T \qquad (3)$$

によって収束効率を定義した。但しFOは出発点における目的関数の値，FMは既知な最小値，FFは収束した点における関数値でTは収束に要した計算時間（秒）である。

Table 1からTable 5には前記の問題に対する計算結果が示されているが，結果の詳細については省き，結論を述べると降下法によるプログラムでは

1. 全般的に収束効率の点で優れ，とくにMINIMが良い。
2. 一般に可変計量法のプログラムの安定性，収束効率が良い。
3. 直線探索のルーチンが必要で，収束効率や解の精度もと

の影響を受ける

4. ユーザーズ・ルーチンとして目的関数の1階あるいは高階の導関数の計算ルーチンを用意する必要がある。

5. 収束点が出発点の位置に左右されやすい。

が言える。また直接探索法では，

1. 目的関数に対する微分可能性の要求はなく，アルゴリズムが簡単で，使い易さの点で優れている。

2. 安定性の点で優れており，大域的な探索に適している。

3. 反面収束効率の点で劣る。

4. 多変数問題に対する効率が悪い。

5. 各手法に固有の係数，例えばシンプレックス法の折り返し，縮少，拡大係数などの定め方が難かしい。

が言える。

(iii)は一般的な非線形計画法の問題であり，解法は大別して発見的方法によるもの，変換法によるもの，線形近似によるものに分類される。この問題に対して原研において開発・整備されたプログラムは，発見的方法によるものではSIMPLEX, COMPLX, CORASEである。

SIMPLEXは前述のシンプレックスを flexible tolerance 法に組み入れて作られたプログラムである。

COMPLXはBOXにより紹介されたコンプレックス法を

採用したプログラムで，シンプレックス法では幾何形として単体を考えるのに対しこれは実行可能領域の中に複体を生成して探索を行なうものである。

線形近似によるものではMAPとCUTPLNが開発されている。MAPはGriffithとStewartにより提案された近似計画法によるプログラムで，目的関数と制約条件式を各反復の過程で得られる点の回りでTaylor展開し，2次以上の項を無視することにより線形化し，LPを用いて解くものである。またCUTPLNは制約領域が凸領域になる凸計画法の問題や，整数計画法の問題を扱うための切除平面法を用いたプログラムである。

変換法は，制約条件式を目的関数の中に組み込んだ変換関数を考え，それを新しい目的関数として制約条件無しの問題に還元するもので，広範囲に研究されている制約条件無しの問題に対する最適化手法を利用できるという利点がある。変換関数の定め方により種々の方法があるが，我々が開発・整備したのはAllranとJohnsenによるAJM，PowellによるPWMである。

これらのプログラムの中で我々はSIMPLEX，COMPLX，CORASE，MAP，AJMのベンチマークテストを行なった。テスト問題は以下の6題である。

1. (Box の問題)

$$\text{Min} \quad b_0 + a_{01}x_1 + \left(\sum_{j=2}^{5} a_{0j}x_j\right)x_1$$

subject to

$$0 \leq a_{i1}x_1 + \left(\sum_{j=2}^{5} a_{ij}x_j\right)x_i \leq b_i \quad , i = 1, 2, 3$$

b_i, a_{ij} : given

2. (Wilde)

$$\text{Min} \quad -e^{(x_1-1)^2 + (x_2-2)^2}$$

subject to

$$x_1 - x_2^2 \geq 0 \quad ,$$

$$-e^{-x_1} + x_2 \geq 0 \quad ,$$

$$-2(x_1-1)^2 + x_2 \geq 0 \quad , \quad \text{and } x_1, x_2, x_3 \geq 0$$

3. (Wood)

$$\text{Min} \quad 100(x_2 - x_1^2)^2 + (1-x_1)^2 + 90(x_4 - x_3^2)^2$$
$$+ (1-x_3)^2 + 10.1[(x_2-1)^2 + (x_4-1)^2]$$
$$+ 19.8(x_2-1)(x_4-1)$$

subject to

$$-10 \leq x_i \leq 10 \quad , i = 1, \cdots, 4$$

4.

$$\text{Min} \quad \sum_{j=1}^{5} e_j x_j + \sum_i \sum_j c_{ij} x_i x_j + \sum_j d_j x_j^3$$

subject to

$$\sum_j a_{ij} x_j - b_i \geq 0 \quad , i = 1, \cdots, 5$$

and $x_j \geq 0$

5. Min $\sum_{i=1}^{10} \{ [\ln(x_i - 2)]^2 + [\ln(10 - x_i)]^2 \} - (\prod_{i}^{10} x_i)^{0.2}$

subject to

$2.001 < x_i < 9.999 \quad , \quad i = 1 \cdots 10$

6. Min $1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$

subject to

$x_1^2 + x_2^2 + x_3^2 - 25 = 0$

$8x_1 + 14x_2 + 7x_3 - 56 = 0$

$x_1, x_2, x_3 \geq 0$

Table 6にはこれらの関数に対する各プログラムの計算結果が示されている。表の中でNFEは要した目的関数計算ルーチンの呼び出し回数，F-finalは収束した点における目的関数値，X-erは収束点と最適点の相対誤差の最大値である。この結果から得られる結論として，

(1) 全体としてMAPの効率が良い。ユーザーズ・ルーチンとしてLPルーチンと目的関数と制約条件式の計算ルーチンが必要。

(2) 発見的方法によるものの中ではCOMPLXの効率が良い。しかしこれは等号制約を扱えないということと複体の折り返し係数の定め方が困難である。

(3) SIMPLEX は安定した手法で，等号制約を扱えるが，
やはり手法に固有の係数の定め方が困難。

(4) 使い易さの点ではCORASEが良い。

(5) AJMは制約条件無しの他の任意の手法を用いることができるが，収束効率，安定性もその影響を受ける。

が言える。

Table 1. Computational Result for Rosenbrock and Beale

ROUTINE	Rosenbrock		Beale	
	Stability	C.E.	Stability	C.E.
VA01A	10	0.1	7	0.1
VA06A	10	0.2	7	0.1
VA08A	10	0.4	5	0.2
VA09A	10	0.1	1	0.5
VA10A	10	0.1	3	0.1
MINIM	10	0.2	6	1.0
CGD	10	0.6	3	0.4
FPD	10	1.0	5	0.5
ALSIM *	10	0.2	7	0.2
ALPS *	10	0.1	6	0.7
ALAG	10	0.1	7	0.1
ALPART	9	0.02	6	0.04
ALCODR	10	0.1	1	0.2
ALVAM	10	0.05	7	0.05
ROTAX *	10	0.5	10	0.3
SPASE *	10	0.2	10	0.1

Table 2. Computational Result for Box and Enzyme

ROUTINE	Box		Enzyme	
	Stability	C.E.	Stability	C.E.
VA01A	10	0.4	7	0.4
VA06A	9	0.5	7	0.3
VA08A	1	0.2	6	0.2
VA09A	5	1.0	6	0.3
VA10A	8	0.3	4	0.4
MINIM	8	0.9	3	1.0
ÇGD	10	0.9	0	——
FPD	9	0.7	3	0.4
ALSIM *	10	0.1	7	0.2
ALPS *	10	0.05	5	0.1
ALAG	10	0.03	3	0.1
ALPART	10	0.03	7	0.03
ALÇODR	5	0.1	4	0.1
ALVAM	9	0.1	6	0.01
ROTAX *	10	0.6	10	0.4
SPASE *	10	0.01	10	0.04

Table 3. Computational Result for Watson and Powell

ROUTINE	Watson		Powell	
	Stability	C.E.	Stability	C.E.
VA01A	10	0.3	10	0.3
VA06A	10	0.2	10	0.3
VA08A	6	0.05	10	0.4
VA09A	10	0.5	10	0.5
VA10A	10	0.3	10	0.7
MINIM	10	1.0	10	1.0
CGD	0	—	10	0.4
FPD	10	0.3	10	0.9
ALSIM *	5	0.1	10	0.4
ALPS *	5	0.2	10	0.3
ALAG	0	—	10	0.1
ALPART	0	—	10	0.1
ALCODR	0	—	10	0.3
ALVAM	0	—	10	0.04
ROTAX *	10	0.4	10	0.8
SPASE *	—	—	10	0.1

Table 4. Computational Result for Wood and Gauss

ROUTINE	Wood		Gauss	
	Stability	C.E.	Stability	C.E.
VA01A	10	0.2	10	0.7
VA06A	10	0.3	10	0.5
VA08A	10	0.3	7	0.5
VA09A	10	0.5	10	0.7
VA10A	10	0.4	10	0.3
MINIM	10	1.0	10	1.0
CGD	10	0.6	10	0.5
FPD	9	0.9	10	0.6
ALSIM *	10	0.2	10	0.04
ALPS *	10	0.4	10	0.1
ALAG	9	0.1	10	0.1
ALPART	10	0.1	10	0.1
ALCODR	10	0.2	10	0.2
ALVAM	10	0.05	10	0.2
ROTAX *	10	0.7	10	0.4
SPASE *	10	0.1	10	0.1

Table 5. Computational Result for Extended Rosenbrock

Routine \ m	4	8	10	20
VA01A	2.15	0.80	0.60	0.17
VA06A	0.54	0.17	0.08	0.02
VA08A	1.99	0.80	0.68	0.81
VA09A	1.23	0.51	0.34	0.09
VA10A	1.27	0.45	0.29	0.05
MINIM	3.70	1.78	1.24	0.33
CGD	1.93	1.18	1.07	0.62
FPD	2.44	1.25	0.62	0.22
ALSIM *	0.36	0.05	0.004	0.001
ALPS *	2.79	1.22	0.20	0.07
ALAG	0.25	0.20	0.19	0.13
ALPART	0.29	0.26	0.25	0.19
ALCODR	0.42	0.06	0.03	0.01
ALVAM	0.12	0.06	0.05	0.01
ROTAX *	2.53	0.73	0.11	0.004
SPASE *	0.48	——	——	——

Table 6.

Resuts of constrained problems

No.	Program	NFE	F-final	X-er.*	T (sec)
1	SIMPLEX	2442	-5.280+6	1.43-3	96.451
	COMPLX	4026	〃	1.83-3	7.403
	CORASE	6001	〃	3.33-3	24.951
	MAP	4	〃	0.0	0.600
	AJM	4017	〃	1.67-3	11.341
2	SIMPLEX	126	-23.722	3.89-4	0.789
	COMPLX	37	〃	3.89-4	0.187
	CORASE	3004	-23.705	1.47-4	0.801
	MAP	8	-23.722	3.89-4	0.134
	AJM	312	-23.107	2.72-2	0.270
3	SIMPLEX	665	0.321-15	0.	2.507
	COMPLX	754	0.730-9	0.	2.536
	CORASE	4001	0.374-8	4.0-3	3.188
	MAP	32	0.609-11	0.	1.273
	AJM	1812	0.449-9	0.	2.175
4	SIMPLEX	1174	-32.449	4.67-4	27.363
	COMPLX	1739	-32.332	3.57-2	11.223
	CORASE	6001	-32.337	2.67-2	18.147
	MAP	64	-32.349	4.66-4	8.473
	AJM	3803	-32.344	1.20-3	13.335
5	SIMPLEX	1154	-45.778	8.55-5	13.843
	COMPLX	1927	〃	7.48-5	20.886
	CORASE	8001	-42.329	5.24-2	42.331
	MAP	61	-45.778	7.48-5	6.128
	AJM	3411	〃	8.55-5	14.854

* X-er. $= \| x^{*} - x^{\dagger} \|_{\infty} / | x^{*} |$

Table 6. (continued)

No.	Program	NFE	F-final	X-final	$h_i(x)$	T(sec)
6	SIMPLEX	222	961.717	3.547 0.2143 3.517	h_1=3.57-3 h_2=4.8-3	1.169
	MAP	5	961.632	3.444 0.2176 3.628	h_1=7.1-2 h_2=7.6-3	0.537