遺伝的アルゴリズムによる最適化問題の数値解法

角　谷　　　敦

（受付　2000 年 5 月 8 日）

1. 序　　　論

自然科学や経済学において，数学的な考察を行うためにさまざまな現象を数理的なモデルで表現することがある。そのようなモデルの中には，最適化問題で記述されるものもある。

本論文では，次のような形で表現される最適化問題について考えていくことにする：

$$\max_{x \in X} f(x). \tag{1}$$

ただし，$f(x)$ は与えられた関数，X は R^L 上の有界閉領域とする ($L \geq 1$)。

本来，汎関数 $f(x)$ は具体的な現象から導かれたものを用いる方が望ましいが，本論文では，具体的な現象とは無関係な関数を用いることにする。

本研究の最終的な目標は，遺伝的アルゴリズムを偏微分方程式の数値解法や偏微分方程式に対する最適制御問題の数値シミュレーションに応用するところにある。しかし，遺伝的アルゴリズムはまだ新しい考え方なのでその性質があまり知られていない。そこで本論文では，(1)のような形の最適化問題の数値解法に遺伝的アルゴリズムを応用して，遺伝的アルゴリズムがどのような性質を持っているか調べることにする。

最適化問題の最適解を数値計算をする方法としては，2 分法，最急降下法，……などいろいろな方法がある。これらの方法では最適化問題の汎関数が凸関数であることを仮定している場合が少なくない。それは汎関数が凸関数の場合，凸関数の性質を利用することで比較的簡単に最適解が得られるためと思われる。しかし，実際に現象を記述してみると最適化問題の汎関数が凸関数でない場合も少なくない。このような場合には，汎関数のうち凸関数になっている部分に限定してこれらの方法を適用するなどの工夫をする必要があった。本論文では，そのような場合にも遺伝的アルゴリズムが特に工夫することなく機能するか調べてみることにする。

2. 遺伝的アルゴリズム

ここでは，遺伝的アルゴリズムについて説明することにする。

交叉，突然変異などのような生物の遺伝に関するメカニズムを利用したアルゴリズムを遺

伝的アルゴリズムという。

生物の場合いろいろな性質を持つ染色体がひとつの組になり，染色体の組み合わせがそれぞれの個体の個性を決定する。遺伝的アルゴリズムでは単純化して，ひとつの個体の個性をひとつの染色体で決定することにする。染色体は 2 進数（0 と 1 の列）で表される。

遺伝的アルゴリズムに関する基本的な用語について説明をしながら，遺伝的アルゴリズムの処理手順を説明をすることにする。

まず，初期の個体集団としていくつかの染色体を準備する。最初に準備する染色体はなるべくいろいろなものがある方が望ましいので，乱数を用いて発生させることにする。

次に，選択，交叉，突然変異という操作を順に行って，つぎの世代の染色体を発生させる。これらの操作について以下で説明することにする。

選択は，評価関数を使って初期の染色体の集団から次の世代へ残す染色体を選び出す作業を行う。最適化問題(1)の場合，評価関数を

$$\frac{1}{1+\exp(-f(x))}$$

とすればよい。染色体は N 個あり，1 から N までの番号が順にふられているとする。また，第 i 番目の染色体に対する評価関数の値が p_i であるとする。

まず，それぞれの染色体に対する評価関数の値の総和 P を求め，

$$P = \sum_{i=1}^{N} p_i \tag{2}$$

とする。次に，それぞれの染色体の選択確率 q_i $(i=1, 2, \cdots, \mathrm{N})$ を求める。選択確率は

$$q_i = \frac{p_i}{P} \tag{3}$$

と定義する（$q_0=0$ とする）。さらに，累積確率 r_i $(i=0, 1, 2, \cdots, \mathrm{N})$

$$r_i = \sum_{j=0}^{i} q_i \tag{4}$$

を求める。

これで準備作業が終了したので，次の世代に残す染色体を選んでいくことにする。乱数を発生させ，発生させた乱数 s が

$$r_i < s \leq r_{i+1} \tag{5}$$

を満たしているとき，$i+1$ 番目の染色体を選択する。この操作を N 回繰りかえし，N 個の染色体を選ぶ。このとき，同じ染色体を重複して選んでもかまわない。選ばれた順に染色体には 1 から N までの番号を付けておくことにする。

次に，この N 個の染色体に対して交叉を行う。交叉とは，選択で得られた染色体の中から

いくつかのペアを選び出し，新しい染色体を作り出す作業のことをいう。

あらかじめ交叉の発生する確率 p_c を決めておく。交叉の対象とする染色体を決めるために乱数を N 個発生させる（i 番目に発生させた乱数を s_i とする）。もし乱数 s_i が次の式を満たしているならば

$$s_i < p_c \tag{6}$$

i 番目の染色体を交叉の対象とする。交叉は 2 つの染色体に対して行うので，選ばれた染色体の数は偶数でないといけない。もし選ばれた染色体の数が奇数の場合，染色体をもう 1 つ追加して染色体の数が偶数になるようにする。

交叉する染色体の数が偶数になったら選ばれた順に 2 つずつのペアを作り,それぞれのペアごとに交叉を行う。交叉する位置は乱数を用いて決める。2 つの染色体が

$$\begin{aligned} &\text{A: 1100/10001} \\ &\text{B: 0100/11010} \end{aligned} \tag{7}$$

となっているとき（/ で交叉の位置を表す），次のような染色体を発生させることにする。

$$\begin{aligned} &\text{A': 1100/11010} \\ &\text{B': 0100/10001} \end{aligned} \tag{8}$$

つまり，交叉位置より右にある部分を入れ替えればよい。

このようにして発生させた染色体に，選択で選んだ染色体のうち交叉の対象にならなかったものをあわせて N 個の染色体の組ができる。

最後に，突然変異の処理をする。

突然変異が起こる確率 p_m を与えておく。染色体を表している 2 進数の 1 つ 1 つの数字をビットという。N 個の染色体のすべてのビットに対して乱数 s を 1 つずつ発生させ，この乱数が

$$s < p_m \tag{9}$$

を満たしているとき，対応する位置のビットを変化させることにする（もし 1 なら 0 に，0 なら 1 にする）。

このとき得られている N 個の染色体を次の世代の染色体とする。この染色体の組を新しい初期の個体集団として今と同様の手順で次の世代の染色体の組を得る。これを繰り返してあらかじめ指定した世代の染色体が得られるまで計算すればよい。

3. 遺伝的アルゴリズムの最適化問題への適用

最適化問題に遺伝的アルゴリズムを適用する際のポイントは何を染色体にするかである。最適化問題(1)では，変数 x を染色体として考えればよい。2 進数で表される染色体と変数 x を

以下のように対応させることにする。

まず，X が閉区間 $[a, b]$ である場合を考える。この場合，実数 $x \in [a, b]$ と M 桁の 2 進数を対応させる方法を考えればよい。M 桁の 2 進数は10進数で表すと 0 から 2^M-1 までの自然数となる。これを閉区間 $[a, b]$ と対応させる方法は何通りかあるが，本論文では

$$x = a + \frac{(b-a)z}{2^M - 1} \tag{10}$$

とする。ただし，M 桁の 2 進数が10進数で z と表せることにする。

次に，X が R^L の長方形領域，つまり

$$X = \{(x_1, x_2, \cdots, x_L) \mid a_i \le x_i \le b_i,\ i = 1, 2, \cdots, L\} \tag{11}$$

となっているとする。各 x_i $(i=1, 2, \cdots, \mathrm{L})$ を M_i 桁の 2 進数 d_i と

$$x_i = a_i + \frac{(b_i - a_i)z_i}{2^{M_i} - 1} \quad (i = 1, 2, \cdots, L) \tag{12}$$

と対応させることにする。ただし，2 進数 d_i が10進数で z_i と表せることにする。L 個の変数の組 $(x_1, x_2, \cdots, x_L)$ を d_1 から d_L までを書き並べた $\sum_{i=1}^{L} M_i$ 桁の 2 進数と対応させることにする。

ここで注意する必要があるのは，染色体を表す 2 進数列の桁数と数値計算によって求めた最適解の精度が無関係ではないということである。従って，汎関数 f の変数を何桁の 2 進数に変換するかは慎重に考える必要がある。

あとは，第 2 節で説明した手順どおりにアルゴリズムを構成すればよい。

4. 数値シミュレーション

まず，最適化問題(1)の汎関数が 1 変数関数の場合について考えることにする。

遺伝的アルゴリズムを用いる場合，以下に示すことをあらかじめ設定しておく必要がある。

- 染色体の総数 N
- 1 つの染色体に含まれるビットの総数 M
- 交叉の起こる確率 q_c
- 突然変異の起こる確率 q_m
- 区間 $[a, b]$
- 第何世代まで計算するか

区間 $[a, b]$ は汎関数を与えたときに決定すればよい。染色体の総数 N や 1 つの染色体に含まれるビットの総数 M の大小は最適解の精度，計算に必要な時間等に影響するので，うまく決める必要がある。交叉の起こる確率 q_c や突然変異の起こる確率 q_m は大きすぎたり，小さす

ぎたりするとアルゴリズムがうまく動かない可能性があるので注意する必要がある。

遺伝的アルゴリズムは乱数を用いているため，乱数の発生の仕方によって求まる結果が異なる可能性がある。そのため以下で行う数値実験では，同一の設定で100回計算して最適解とそれを与える x の値を考察していくことにする。

数値実験 1

ここでは，$M=N=40$ とし，最適解を求めるアルゴリズムに適した q_c と q_m の値を決定することを目的とする。汎関数は

$$f(x)=-x^2+0.5 \quad x\in[-10,10]$$

とする。数値計算で求まった最適解の最小値，最大値，平均値を表 1 に示してある。欄の上から順に最適解の最小値，最大値，平均値が書いてある。

表 1 を見てわかるように，交差が発生する確率，突然変異が発生する確率ともに値が大きすぎても小さすぎても計算が安定せずうまくいかないことがわかる。交差が発生する確率が

表 1

		q_m				
		0.01	0.05	0.1	0.2	0.3
q_c	0.01	0.481454 0.500000 0.496993	0.401604 0.500000 0.497809	0.455798 0.500000 0.496821	0.429822 0.500000 0.494020	0.153724 0.500000 0.477612
	0.1	0.423370 0.500000 0.494677	0.477036 0.500000 0.498411	0.432488 0.500000 0.497209	0.414876 0.500000 0.492623	0.108156 0.500000 0.473097
	0.2	0.400655 0.500000 0.495049	0.480552 0.500000 0.498051	0.462158 0.500000 0.497494	0.418811 0.500000 0.494285	0.224163 0.499987 0.473747
	0.25	0.442068 0.500000 0.495583	0.479901 0.500000 0.498531	0.380361 0.500000 0.496583	0.423285 0.500000 0.493866	0.123767 0.500000 0.466944
	0.3	0.385833 0.500000 0.494753	0.475056 0.500000 0.498110	0.419789 0.500000 0.497091	0.379477 0.500000 0.490368	0.192284 0.500000 0.474463
	0.35	0.471652 0.500000 0.497062	0.480085 0.500000 0.498763	0.451360 0.500000 0.496626	0.453296 0.500000 0.493469	0.131260 0.499998 0.473259
	0.4	0.462383 0.500000 0.496378	0.446118 0.500000 0.497830	0.441587 0.500000 0.496869	0.388207 0.499998 0.490878	0.294540 0.499999 0.478380

0.25ぐらい，突然変異が発生する確率が0.05ぐらいが最適解の精度がいいように思われる。

以下の計算では，$q_c = 0.25$，$q_m = 0.05$ として数値計算していくことにする。

数値実験 2

$M = 40$，$q_c = 0.25$，$q_m = 0.05$ とする。ここでは，染色体の総数 N が最適解の精度にどのように影響するか調べ，染色体の数はどのくらいが適当か考えることにする。汎関数は，前の数値実験と同じ

$$f(x) = -x^2 + 0.5 \quad x \in [-10, 10]$$

とする。数値計算で求まった最適解の最小値，最大値，平均値を表 2 に示してある。

表 2

N	最小値	最大値	平均値
5	−0.457399	0.499995	0.393436
10	0.377602	0.499998	0.469759
20	0.405746	0.500000	0.490747
40	0.485546	0.500000	0.498717
60	0.492117	0.500000	0.499324
80	0.492354	0.500000	0.499540
100	0.497029	0.500000	0.499769
200	0.498289	0.500000	0.499923
400	0.499892	0.500000	0.499989

表 2 に示された計算結果からもわかるように，染色体の数が多い方が最適解の精度がいいことがわかる。ただし，染色体の数が増えると必要なメモリーの量や計算にかかる時間が多くなるので，以下に行う数値実験では染色体の数を200で行うことにする。

数値実験 3

$N = 200$，$q_c = 0.25$，$q_m = 0.05$ とする。ここでは，染色体のビット数 M が最適解の精度にどのように影響するか調べ，染色体のビット数はどのくらいが適当か考えることにする。汎関数は，前の数値実験と同じ

$$f(x) = -x^2 + 0.5 \quad x \in [-10, 10]$$

とする。数値計算で求まった最適解の最小値，最大値，平均値を表 3 に示してある。

表 3 に示された計算結果からもわかるように，染色体のビット数が少ないと最適解の精度が悪くなることがある。染色体のビット数が100以上ならば，最適解の精度が安定していいと

表 3

N	最小値	最大値	平均値
5	0.395942	0.395942	0.395942
10	0.499140	0.499904	0.499897
20	0.499551	0.500000	0.499967
40	0.498289	0.500000	0.499917
60	0.499145	0.500000	0.499931
80	0.498247	0.500000	0.499839
100	0.498540	0.500000	0.499943
200	0.498782	0.500000	0.499943
400	0.497612	0.500000	0.499926

思われる。以下で行う数値実験では $M=200$ とする。

数値実験 4

$M=N=200$，$q_c=0.25$，$q_m=0.05$ とする。ここでは，区間 $[a, b]$ の大きさが最適解の精度にどのように影響するか調べることが目的である。汎関数は，

$$f(x)=-x^2+0.5 \quad x\in[a, b]$$

とする。このとき，最適解は $x=0$ のとき0.5となる。数値計算で求まった最適解の最小値，最大値，平均値を表 4 に示す。

区間 $[a, b]$ の幅が小さいほど最適解の精度がよくなることがわかる。a, b の絶対値が0.001より小さくなる場合，最適解の精度が変化していないように見えるが，a, b の絶対値が小さくなるほど最適解を与える x の値の精度がよくなっている。

表 4

$b\ (=-a)$	最小値	最大値	平均値
10	0.498782	0.500000	0.499939
5	0.498200	0.500000	0.499945
1	0.499274	0.500000	0.499958
0.5	0.499746	0.500000	0.499980
0.1	0.499987	0.500000	0.499999
0.05	0.499993	0.500000	0.500000
0.01	0.500000	0.500000	0.500000
0.005	0.500000	0.500000	0.500000
0.001	0.500000	0.500000	0.500000

最適解を精度よく求めるために次のような手順で計算することにする。

1. まず大きめに区間 $[a, b]$ の幅を設定して最適解の存在する位置をおよそ求める。
2. もし最適解が精度よく求まっていれば，終了する。そうでなければ，区間 $[a, b]$ の幅を小さく設定し直す。
3. 設定し直した $[a, b]$ に対して最適解を求め，2 へ。

このようにすることで，最適解の位置が全くわからない場合や最適解を与える点が 2 ヶ所以上ある場合でも最適解が精度よく求められると思われる。

表 5

世代数	最小値	最大値	平均値
10	0.499113	0.500000	0.499907
20	0.499203	0.500000	0.499962
40	0.499311	0.500000	0.499946
60	0.498853	0.500000	0.499933
80	0.499439	0.500000	0.499965
100	0.498782	0.500000	0.499946
200	0.499606	0.500000	0.499969
400	0.499607	0.500000	0.499978
600	0.498599	0.500000	0.499918
800	0.499139	0.500000	0.499925
1000	0.499112	0.500000	0.499952

数値実験 5

$M = N = 200$，$q_c = 0.25$，$q_m = 0.05$ とする。ここでは，何世代まで計算するかが最適解の精度にどのように影響するか調べることが目的である。汎関数は，前の数値実験と同じ

$$f(x) = -x^2 + 0.5 \quad x \in [a, b]$$

とする。数値計算で求まった最適解の最小値，最大値，平均値を表 5 に示してある。

計算する世代数が少ないと，精度がよくないことがわかる。以下の数値シミュレーションでは200世代まで計算することにする。

以下，さまざまな汎関数を設定し，遺伝的アルゴリズムを利用した最適値問題の数値解法がどのような場合に適用できるか調べてみることにする。特に断らなければ，$M = N = 200$，$q_c = 0.25$，$q_m = 0.05$とする。

数値実験 6

汎関数は

$$f(x)=3x-4 \quad x\in[-10,10]$$

とする。このとき，最大値は $x=10$ のとき 26 であることがわかる。

区間 $[a, b]=[-10, 10]$ として数値計算すると，求まった100個の最適解の最小値，最大値，平均値は

24.760220, 25.998092, 25.829099

となり，最適解を与える x は $9.5<x<10$ の範囲に存在していることがわかる。最適解の精度をあげるために，$[a, b]=[9.5, 10]$ として数値計算すると，最適解の最小値，最大値，平均値は

25.916744, 25.999809, 25.989131

となり，最適解を与える x は $9.95<x<10$ の範囲に存在している。さらに，最適解の精度をあげるために，区間 $[a, b]=[9.95, 10]$ として数値計算すると，求まった最適解の最小値，最大値，平均値は

25.992252, 25.999978, 25.998784

となり，最適解を与える x は $9.997<x<10$ の範囲に存在していることがわかる。そこで $[a, b]=[9.997, 10]$ として数値計算すると，求まった最適解の最小値，最大値，平均値は

25.999748, 25.999998, 25.999932

となり，最適解を与える x は $9.9999<x<10$ の範囲に存在している。そこで $[a, b]=[9.9999, 10]$ として数値計算すると，求まった最適解の最小値，最大値，平均値は

25.999989, 26.000000, 25.999997

となり，最適解は $x=10$ のとき 26 と推定できる。

数値実験 7

汎関数は

$$f(x)=-\frac{1}{4}x^4-\frac{7}{18}x^3+\frac{7}{36}x^2+\frac{4}{18}x+1 \quad x\in[-10,10]$$

とする。このとき，最大値は $x=-\dfrac{4}{3}$ のとき $\dfrac{287}{243}$ であることがわかる。

数値実験 6 のように，区間の幅をだんだん小さくして計算すると，表 6 のような結果が得られた。得られた結果から推定すると，最適解は $x=-\dfrac{4}{3}$ のとき 1.181070 と推定できる。$\dfrac{287}{243}=1.18069958\cdots$ より，汎関数が凸関数でない場合にもきちんと計算できていることがわかる。

表 6

[a, b]	最小値	最大値	平均値
[−10, 10]	1.178044	1.181070	1.180912
[−1.4, −1.2]	1.181035	1.181070	1.181068
[−1.34, −1,33]	1.181070	1.181070	1.181070
[−1.334, −1.333]	1.181070	1.181070	1.181070
[−1.33336, −1.3333]	1.181070	1.181070	1.181070

数値実験 8

汎関数は

$$f(x) = -\frac{1}{4}x^4 + x^3 \quad x \in [-10, 10]$$

とする。このとき，最大値は $x=3$ のとき $\dfrac{27}{4}$ である。

区間の幅をだんだん小さくして計算すると，表 7 のような結果が得られた。得られた結果から推定すると，最適解は $x=3$ のとき 6.75 と推定できる。

表 7

[a, b]	最小値	最大値	平均値
[−10, 10]	6.731311	6.750000	6.749142
[2.9, 3.05]	6.749972	6.750000	6.749997
[2.995, 3.002]	6.750000	6.750000	6.750000
[2.9998, 3.0001]	6.750000	6.750000	6.750000

数値実験 9

汎関数は

$$f(x) = \begin{cases} -x^2+1 & x \le -\sqrt{2} \\ -1 & -\sqrt{2} \le x \le 0 \\ -x^2+2\sqrt{2}x-1 & 0 < x \end{cases}$$

とする。このとき，最大値は $x=\sqrt{2}$ のとき 1 であることがわかる。

区間の幅をだんだん小さくして計算すると，表 8 のような結果が得られた。得られた結果から推定すると，最適解は $x=1.4142$ のとき 1 と推定できる。この汎関数のようにグラフの一部で傾きが 0 となっている部分があっても最適解が求められることがわかった。

表 8

$[a, b]$	最小値	最大値	平均値
[−10, 10]	0.999043	1.000000	0.999952
[1.38, 1.43]	0.999999	1.000000	1.000000
[1.413, 1.416]	1.000000	1.000000	1.000000
[1.4141, 1.4143]	1.000000	1.000000	1.000000
[1.41420, 1.41422]	1.000000	1.000000	1.000000

数値実験10

汎関数は

$$f(x)=\begin{cases} x+\dfrac{5}{2} & x<-\dfrac{1}{2} \\ -2x+1 & -\dfrac{1}{2}\le x<\dfrac{3}{2} \\ 3x-\dfrac{13}{2} & \dfrac{3}{2}\le x<\dfrac{5}{2} \\ -x+\dfrac{7}{2} & \dfrac{5}{2}\le x \end{cases}$$

とする。この汎関数はグラフが折れ線になっていて，最大値は $x=-\dfrac{1}{2}$ のとき 2 であることがわかる。

区間の幅をだんだん小さくして計算すると，表 9 のような結果が得られた。得られた計算結果から最適解は $x=-0.5$ のとき 2 であることが推定できる。この汎関数のように微分可能でない点に最適解が存在しても最適解が求められることがわかった。

表 9

$[a, b]$	最小値	最大値	平均値
[−10, 10]	1.941547	1.999946	1.990491
[−0.56, −0.47]	1.998249	1.999989	1.999598
[−0.502, −0.499]	1.999921	2.000000	1.999985
[−0.50007, −0.49998]	1.999998	2.000000	2.000000

数値実験11

汎関数は

$$f(x)=\begin{cases} x-2 & x<\dfrac{1}{3} \\ -2x+3 & \dfrac{1}{3}\le x \end{cases}$$

とする。この汎関数は $x=\frac{1}{3}$ で不連続になっている。最大値は $x=\frac{1}{3}$ のとき $\frac{7}{3}$ であることがわかる。

区間の幅をだんだん小さくして計算すると，表10のような結果が得られた。

表10

$[a, b]$	最小値	最大値	平均値
[−10, 10]	2.227306	2.333304	2.322370
[0.333, 0.387]	2.328895	2.333326	2.332398
[0.3333, 0.336]	2.333171	2.333333	2.333290
[0.33333, 0.3335]	2.333321	2.333333	2.333330
[0.333333, 0.33334]	2.333332	2.333333	2.333333

計算結果から，最適値は $x=\frac{1}{3}$ のとき $\frac{7}{3}$ であることが推定できる。従って，不連続点に最適解がある場合でも最適解を求めることができることがわかった。

以上の結果から，遺伝的アルゴリズムを利用すると従来の方法では最適解が容易に求められなかった場合でも簡単に最適解が求められることがわかった。また，最適解を与える位置が2ヶ所以上あってもすべて求められると思われる。

次に，汎関数が2変数関数の場合について考察することにする。簡単のため，最適化問題(1)で X が長方形領域 $[a, b]\times[c, d]$ であったとする。

遺伝的アルゴリズムを偏微分方程式の数値解法や最適制御問題の数値シミュレーションに応用することを考える場合，汎関数が多変数関数のときに利用できることを確認するのは重要なポイントとなる。

1変数関数のときと同様に，数値実験をして調べると $N=M=200$，$q_c=0.25$，$q_m=0.05$ とすればよいことがわかる。また，200世代まで計算することにする。

数値実験12

汎関数は

$$f(x, y)=\frac{1}{2}x+\frac{3}{2}y-5 \quad (x, y)\in[-10,10]\times[-10,10]$$

とする。このとき，最大値は $(x, y)=(10, 10)$ のとき 15 であることがわかる。

長方形領域をだんだん小さくして計算すると，表11のような結果が得られた。計算結果か

表11

$[a, b] \times [c, d]$	最小値	最大値	平均値
$[-10, 10] \times [-10, 10]$	12.749796	14.958406	14.230008
$[5.5, 10] \times [8.5, 10]$	14.475253	14.947668	14.792121
$[9.3, 10] \times [9.6, 10]$	14.892216	14.996959	14.957741
$[9.89, 10] \times [9.94, 10]$	14.983488	14.998828	14.993561
$[9.98, 10] \times [9.99, 10]$	14.996132	14.999924	14.998850
$[9.99, 10] \times [9.998, 10]$	14.999049	14.999922	14.999639
$[9.998, 10] \times [9.9995, 10]$	14.999795	14.999987	14.999920
$[9.9997, 10] \times [9.9998, 10]$	14.999950	14.999999	14.999981
$[9.9999, 10] \times [9.99998, 10]$	14.999989	15.000000	14.999996

ら最適値は $(x, y) = (10, 10)$ のとき 15 であることが推定できる。

数値実験13

汎関数は

$$f(x, y) = \frac{1}{4}x^2 - y^2 \quad (x, y) \in [-10, 10] \times [-10, 10]$$

とする。このとき，最大値は $(x, y) = (\pm 10, 0)$ のとき 25 であることがわかる。

$[a, b] \times [c, d] = [-10, 10] \times [-10, 10]$ として計算すると最適解を与える点が，$[9.5, 10] \times [-1.1, 1.5]$ 内と $[-10, -9.5] \times [-1.1, 1.5]$ 内に存在することがわかる。それぞれの領域に対して計算を行うと，表12，表13のような結果が得られた。計算結果から最適値は $(x, y) = (\pm 10, 0)$ のとき 25 であることが推定できる。つまり，2 変数の場合でも最適解を与える点が有限個の孤立点ならば，すべての点を求められる可能性があることになる。また，汎関数が鞍点を持っていても，問題なく最適解が求められるとも確認できた。

表12

$[a, b] \times [c, d]$	最小値	最大値	平均値
$[-10, 10] \times [-10, 10]$	22.181291	24.959092	24.131829
$[9.5, 10] \times [-1.1, 1.5]$	24.733068	24.994633	24.913264
$[9.94, 10] \times [-0.4, 0.5]$	24.969059	24.998970	24.989381
$[9.993, 10] \times [-0.14, 0.14]$	24.996387	24.999958	24.998824
$[9.999, 10] \times [-0.05, 0.05]$	24.999519	24.999987	24.999843
$[9.9999, 10] \times [-0.02, 0.02]$	24.999943	24.999999	24.999980
$[9.99998, 10] \times [-0.008, 0.006]$	24.999988	25.000000	24.999996

表13

$[a, b] \times [c, d]$	最小値	最大値	平均値
$[-10, 10] \times [-10, 10]$	22.181291	24.959092	24.131829
$[-10, -9.5] \times [-1.1, 1.5]$	24.708884	24.992196	24.888663
$[-10, -9.94] \times [-0.4, 0.5]$	24.957801	24.999161	24.987083
$[-10, -9.99] \times [-0.15, 0.14]$	24.994175	24.999882	24.998262
$[-10, -9.999] \times [-0.05, 0.07]$	24.999497	24.999982	24.999779
$[-10, -9.9999] \times [-0.017, 0.021]$	24.999930	24.999999	24.999979

表14

$[a, b] \times [c, d]$	最小値	最大値	平均値
$[-10, 10] \times [-10, 10]$	920.540973	1023.457503	995.071716
$[9.5, 10] \times [-1, 10]$	1016.018196	1024.465712	1021.531981
$[9.97, 10] \times [2.5, 7.5]$	1023.964184	1024.976430	1024.656383
$[9.996, 10] \times [4, 6]$	1024.861846	1024.996814	1024.949117
$[9.9995, 10] \times [4.5, 5.3]$	1024.976040	1024.999258	1024.993190
$[9.9999, 10] \times [4.9, 5.2]$	1024.994051	1024.999845	1024.998669
$[9.99998, 10] \times [4.95, 5.05]$	1024.999378	1024.999985	1024.999812
$[9.99999, 10] \times [4.98, 5.02]$	1024.999797	1024.999993	1024.999935
$[9.99999, 10] \times [4.985, 5.015]$	1024.999826	1024.999993	1024.999949
$[9.99999, 10] \times [4.99, 5.012]$	1024.999825	1024.999991	1024.999950
$[9.99999, 10] \times [4.990, 5.01]$	1024.999833	1024.999993	1024.999959

数値実験14

汎関数は

$$f(x, y) = x^3 + xy - y^2 \quad (x, y) \in [-10, 10] \times [-10, 10]$$

とする。このとき，最大値は $(x, y) = (10, 5)$ のとき 1025 であることがわかる。

長方形領域をだんだん小さくして計算すると，表14のような結果が得られた。得られた計算結果から，最適値は $(x, y) = (10, 5)$ のとき 1025 であることが推定できる。

数値実験15

汎関数は

$$f(x, y) = x^3 - 3xy + y^3 \quad (x, y) \in [-10, 10] \times [-10, 10]$$

とする。このとき，最大値は $x = y = 10$ のとき 1700 であることがわかる。

表15

$[a, b]\times[c, d]$	最小値	最大値	平均値
$[-10, 10]\times[-10, 10]$	1004.602560	1680.332814	1402.874951
$[7, 10]\times[-4, 10]$	1285.647433	1688.641955	1536.893807
$[9, 10]\times[8.5, 10]$	1623.763838	1697.945402	1669.926441
$[9.8, 10]\times[9.7, 10]$	1683.501575	1699.588811	1693.663695
$[9.95, 10]\times[9.95, 10]$	1696.854953	1699.905691	1698.738978
$[9.99, 10]\times[9.985, 10]$	1699.226084	1699.978535	1699.695954
$[9.997, 10]\times[9.997, 10]$	1699.811103	1699.995856	1699.924338
$[9.999, 10]\times[9.999, 10]$	1699.937032	1699.998619	1699.975249
$[9.9998, 10]\times[9.9997, 10]$	1699.984519	1699.999589	1699.993882
$[9.99995, 10]\times[9.99995, 10]$	1699.996852	1699.999906	1699.998810

長方形領域をだんだん小さくして計算すると，表15のような結果が得られた。計算結果から，最適値は $x=y=10$ のとき 1700 と推定できる。

5. 結　　　論

最適化問題の数値シミュレーションに遺伝的アルゴリズムが利用できるかどうかについて基本的な事項について考察を行った。汎関数が凸関数でない場合でも，数値シミュレーションによって最適解を求めることができることがわかった。しかも，最適解を与える点が2ヶ所以上あっても従来の方法より簡単にすべての点を求められると思われる。また，従来の方法では最適解が求めにくいような状況でも，特に工夫せずに最適解が求められることも確認できた。

従って遺伝的アルゴリズムは，汎関数が凸関数でない場合や連続性，微分可能性がない場合などには，特に有効な方法であると思われる。

汎関数が多変数の場合にも利用できることがわかったので，偏微分方程式の解の挙動の数値シミュレーションや最適制御問題の最適解の数値シミュレーションにおいても遺伝的アルゴリズムを利用できると思われる。ただし，汎関数の変数の数が多くなるため計算の精度を確保するのが難しくなるのではないかと考えられる。これらの点については引き続き研究することにする

終わりに

本論文は広島修道大学総合研究所調査研究費(1995年度－1997年度)の援助のもと，「貨幣的市場交換を前提とした均衡価格体系のサーチ問題の研究—遺伝的アルゴリズムによる最適化の検討を中心に—」というテーマで行われた高藪　学氏（東京学芸大学）との共同研究のうち，「遺伝的アルゴリズムに関する基礎研究」の部分に関する数学的な面からの考察についての報告である。

参考文献

パソコンで学ぶ遺伝的アルゴリズムの基礎と応用，石田良平・村瀬治比古・小山修平，森北出版，1997
遺伝的プログラミング，伊庭斉志，東京電機大学出版局，1996
C 言語による初めてのアルゴリズム，河西朝雄，技術評論社，1992
遺伝的アルゴリズム，北野宏明 編著，産業図書，1993
遺伝的アルゴリズム 2，北野宏明 編著，産業図書，1995
遺伝的アルゴリズム 3，北野宏明 編著，産業図書，1997