

鈍感な差分進化に関する研究

半田, 斎 / HANDA, Itsuki

(発行年 / Year)

2012-03-24

(学位授与年月日 / Date of Granted)

2012-03-24

(学位名 / Degree Name)

修士(工学)

(学位授与機関 / Degree Grantor)

法政大学 (Hosei University)

2011年度 修士論文

論文題名 鈍感な差分進化に関する研究

指導教授 斎藤 利通 教授

法政大学大学院工学研究科
電気工学専攻修士課程

学生証番号： 10R3133

ハンダ イツキ
氏名 半田 斎

あらまし

本修士論文では、進化的手法である差分進化 (Differential Evolution, DE) を改良し、考察する。

初めに、成長構造を持たせた差分進化について考察する。粒子が局所最適解に陥ったとき新しい粒子を追加することにより、粒子の位置情報を共有し陥った局所最適解以外の点が探索可能になり脱出が可能になる。ベンチマークによる数値実験を行い、典型的な探索例と探索成功率、探索に所要する計算回数と粒子数等を指標として、アルゴリズムの基本性能を考察する。

次に、複数最適化探索問題に対して適用した粒子を鈍感にした差分進化について考察する。このアルゴリズムは、global search と local search の二つのステージによって構成されている。global search では、それぞれ任意の解を持つ local subregions (LSR) を作成する。local search では、並列にそれぞれの LSR を操作し、すべての最適解を探索する。このアルゴリズムには、鈍感さをコントロールする二つの重要なパラメータが存在するもし、鈍感さが適切であれば、適切な LSR を作成でき、すべての最適解を探索することができる。典型的な探索例で数値実験を行い、このアルゴリズムの基本性能を考察する。

Abstract

In this thesis we consider two improve versions of the differential evolution. First, the differential evolution with growing structure is considered. If a particle exploring the optimum fall into a local optimal solution then a new particle is born and particles share the position imformation. And particles can help escape from the trap of the local optimal solution. Performing numerical experiments for basic benchmarks, the algorithm efficiency is evaluated in several key indicators such as typical search example, succes rate, the number of interation and number of particle.

Next, insensitive differential evolution for multi-solution problems is considered. The algorithm consists of global and local searches. In the global search, the algorithm tries to construct local sub-regions (LSRs) each of which includes either solution. In the local search, the algorithm operates on all the LSRs in parallel and tries to find all the solutions. The algorithm has a key parameter that controls the algorithm insensitivity. If the insensitivity is suitable, the algorithm can construct all the LSRs before trapping into either solution and can find all the solutions. Performing basic numerical experiments where parameters are adjusted by trial-and-errors, basic performance of the algorithm is investigated.

目次

第1章	まえがき	8
第2章	成長型差分進化	10
2.1	まえがき	10
2.2	成長型 DE アルゴリズム	11
2.3	数値実験	13
2.3.1	実験 1	13
2.3.2	実験 2	14
第3章	Insensitive Differential Evolution	35
3.1	まえがき	35
3.2	IDE アルゴリズム	36
3.2.1	global search	36
3.2.2	Local subregions	37
3.2.3	Local search	38
3.3	数値実験	39
3.3.1	実験 1	39
3.3.2	実験 2	40
第4章	様々な IDE による探索性能	42
4.1	まえがき	42
4.2	数値実験	43
第5章	むすび	85

目 次

2.1	粒子が局所解に陥る様子	16
2.2	粒子が局所解から脱出した様子	17
2.3	Sphere function の形状	18
2.4	非成長型 DE による Sphere function の探索過程	19
2.5	非成長型 DE による Sphere function の粒子の動き	20
2.6	Normalized Schwefel function の形状	21
2.7	非成長型 DE による Normalized Schwefel function の探索過程 ($N = 10$)	22
2.8	非成長型 DE による Normalized Schwefel function の粒子の動き ($N = 10$)	23
2.9	非成長型 DE による Normalized Schwefel function の探索過程 ($N = 20$)	24
2.10	非成長型 DE による Normalized Schwefel function の粒子の動き ($N =$ 20)	25
2.11	非成長型 DE による Normalized Schwefel function の探索過程 ($N = 30$)	26
2.12	非成長型 DE による Normalized Schwefel function の粒子の動き ($N = 30$)	27
2.13	成長型 DE による Normalized Schwefel function の探索過程, 粒子数 ($M =$ 1)	28
2.14	成長型 DE による Normalized Schwefel function の粒子の動き ($M = 1$)	29
2.15	成長型 DE による Normalized Schwefel function の探索過程, 粒子数 ($M =$ 2)	30
2.16	成長型 DE による Normalized Schwefel function の粒子の動き ($M = 2$)	31
2.17	成長型 DE による Normalized Schwefel function の探索過程, 粒子数 ($M =$ 3)	32

2.18	成長型 DE による Normalized Schwefel function の粒子の動き ($M = 3$)	33
4.1	LSR の作成図	47
4.2	Branin function の形状	48
4.3	Global search における粒子の動き ($\varepsilon=3.0, T_G=30$)	49
4.4	Local search における粒子の動き ($\varepsilon_2=0.01, T_L=70$)	50
4.5	Global search の探索過程	51
4.6	local search の探索過程	52
4.7	Global search における粒子の動き ($\varepsilon=0, T_G=30$)	53
4.8	Global search における粒子の動き ($\varepsilon=3.0, T_G=50$)	54
4.9	Himmelblau function の形状	55
4.10	Global search における粒子の動き ($\varepsilon=3.0, T_G=40$)	56
4.11	Local search における粒子の動き ($\varepsilon_2=0.01, T_L=60$)	57
4.12	Global search の探索過程	58
4.13	local search の探索過程	59
4.14	Global search における粒子の動き ($\varepsilon=0, T_G=40$)	60
4.15	Global search における粒子の動き ($\varepsilon=3.0, T_G=50$)	61
4.16	(a) ~ (d) の最適解の位置	62
4.17	(e) ~ (g) の最適解の位置	63
4.18	(a) 探索成功例 (DE/rand/1, $\varepsilon=0.3$)	64
4.19	(b) の探索成功例 (DE/rand/1, $\varepsilon=0.3$)	65
4.20	(b) の探索失敗例 (DE/rand/2, $\varepsilon=0.3$)	66
4.21	(c) の探索成功例 (DE/rand/2, $\varepsilon=3.0$)	67
4.22	(c) の探索失敗例 (DE/best/2, $\varepsilon=3.0$)	68
4.23	(d) の探索成功例 (DE/rand/2, $\varepsilon=3.0$)	69
4.24	(d) の探索失敗例 (DE/best/2, $\varepsilon=3.0$)	70
4.25	(e) の探索成功例 (DE/current-to-best, $\varepsilon=0.3$)	71

4.26 (e) の探索失敗例 (DE/best/1, $\varepsilon=0.3$)	72
4.27 (f) の探索成功例 (DE/rand/1, $\varepsilon=0.3$)	73
4.28 (f) の探索失敗例 (DE/best/2, $\varepsilon=0.3$)	74
4.29 (g) の探索成功例 (DE/rand/2, $\varepsilon=0.3$)	75
4.30 (g) の探索失敗例 (DE/best/2, $\varepsilon=0.3$)	76

表 目 次

2.1	f_2 に対する成長型 DE の探索結果	34
4.1	Branin function における Global search の成功率	77
4.2	Branin function における Local search の成功率 ($\varepsilon=3, T_G = 30$) $SR=$ 成 成功率	77
4.3	Himmelblau function における Global search の成功率	78
4.4	Himmelblau functionn における Local search の成功率 (%) ($\varepsilon=3, T_G =$ 40) $SR=$ 成功率	78
4.5	(c) における探索した最適解の個数の内訳 (%) ($\varepsilon=3, T_G = 30$)	79
4.6	(d) における探索した最適解の個数の内訳 (%) ($\varepsilon=3, T_G = 30$)	80
4.7	(e) における探索した最適解の個数の内訳 (%) ($\varepsilon=0.3, T_G = 30$)	81
4.8	(f) における探索した最適解の個数の内訳 (%) ($\varepsilon=0.3, T_G = 30$)	82
4.9	(g) における探索した最適解の個数の内訳 (%) ($\varepsilon=0.3, T_G = 30$)	83
4.10	様々な形式におけるベンチマークの探索成功率 (%)	84

第1章 まえがき

複雑な目的関数において局所解に陥らず大域的に最適解を探索でき、評価回数を減らすことに優れたアルゴリズムとして、集団的降下法に基づくアルゴリズムが注目されている。集団的降下法とは、複数の解の集合により集団を形成し、集団から得られる情報に基づき、集団の各要素に対して順次新しい解を生成し、新しい解がよければ古い解と置換する方法である [1]-[4]。

Differential evolution(DE)はこの集団的降下法を利用した最適化法である。1995年に K. Price と R. Storn により導入された。DE は確率的な直接探索法であり、解集団を用いた多点探索を行う。DE は非線形問題、微分不可能な問題、非凸問題、多峰性問題などの様々な最適化問題に適用されており、これらの問題に対して高速で頑健なアルゴリズムであることがしめされている [5]-[7]。

他にも同じ最適化法的一种である遺伝的アルゴリズム (Genetic Algorithm, GA)[8] や粒子群最適化 (Particle Swarm Optimization, PSO)[9] などがあるが、DE は関数最適化問題この二つにも劣らない性能を示している。DE を含めた最適化アルゴリズムの応用例として、AD 変換器 [10] やデジタルフィルタ [11], $\Sigma\Delta$ 変調器 [12] などがあり、設計の際のパラメータ最適化に役立てることができる。

DE にはいくつかの形式が提案されており、DE/best/1/bin や DE/rand/1/exp などがよく知られている。

これらは、DE/base/num/cross という記法で表現される。base は基本ベクトルとなる親の選択方法を指定する。例えば、best は集団の最良個体を選択し、rand は基本ベクトルための親を集団からランダムに選択する。num は基本ベクトルを変異させるための差分ベクトルの個数を選択する。cross は子を生成するために使用する交叉方法を

指定する。例えば、bin は一定の確率で遺伝子を交換する交叉 (binomial crossover) を用い、exp は指数関数的に減少する確率で遺伝子を交換する交叉 (exponential crossover) を用いる [13]。

DE では、探索空間中にランダムに初期個体を生成し、初期集団を構成する。各個体は決定ベクトルに対応し、 n 個の決定変数を遺伝子として持つ。各世代において、全ての個体を親して選択する。各親に対して、次のような処理が行なわれる。突然変異のために、選択された親を除く個体群から互いに異なる $1 + 2\text{num}$ 個の個体の個数を選択する。最初の個体が基本ベクトルとなり、残りの個体対が差分ベクトルとなる。差分ベクトル (scaling factor) が乗算され基本ベクトルに加える。その結果得られたベクトルと親が交叉し、交叉率 (crossover rate) により指定された確率で親の遺伝子ベクトルの要素で置換することにより、子のベクトルが生成される。最後に、生存者選択として、子が親より良ければ、親を子で置換する。

第 2 章では、差分進化アルゴリズム (Differential Evolution, DE) に成長構造をもたせた成長型 DE について考察する。DE には、粒子が高次元の多峰性関数の最適解を探索する時に、局所解に陥ってしまうことがある。成長型 DE は、粒子が局所解に陥ってしまった時に新しい粒子を追加することで、局所解からの脱出を促進する。成長構造は、Self-Organizing maps (SOM) を含むいくつかの学習アルゴリズムにおいて用いられている手法である [14] [15]。

典型的なベンチマークを用いて数値実験を行い、探索成功率、探索に所要する計算回数や粒子数等を指標として、アルゴリズムの性能を考察する。

第 3 章では、DE に粒子の動きを制限する鈍感パラメータを追加した、IDE (Insensitive Differential Evolution) を提案する。DE を含む最適化アルゴリズムには、複数最適解問題を扱う場合に、一点に収束してしまう欠点がある。IDE は、Global search と Local search の二つの探索によって構成されている。探索の時には、粒子の動きを鈍くし、位置情報の更新を遅らせることにより複数の最適解を探索する。様々な IDE の探索方法と複数の基本的なベンチマークにより、IDE の探索性能を考察する。

第2章 成長型差分進化

2.1 まえがき

DE の重要な特徴として、進化的戦略ではガウス突然変異のステップ幅を制御する必要があるが、DE はこのような制御が不要となる単純な数学的演算を用いている。一般に、ガウス突然変異における理想的なステップ幅は、遺伝子あるいは各次元毎に異なる。また進化の状態によっても異なるため、何らかの方法でステップ幅を適応的に調整する必要がある。これに対して、DE はガウス突然変異の代わりに、基本ベクトルと差分ベクトルとの重み付き和を突然変異として採用している。集団から選択された1個体が基本ベクトルとなり、集団からランダムに選択された個体対の差が差分ベクトルとなる。世代を経るに従い、解集団が探索空間中で収縮したり拡張したりする。そして、差分ベクトルが変化し、差分ベクトルとして与えられる各次元におけるステップ幅が自動的に調整される [16]。

本章では、DE に成長構造を持たせた成長型 DE を提案する。DE は、扱う関数が複雑になっていくと粒子が局所解に陥ってしまう可能性が高くなっていく。一度、局所解に陥るとそこから脱出することがとても難しくなる。図 2.1 は、粒子が局所解に陥ってしまった様子を示す。このアルゴリズムでは、粒子が局所解に陥ったときに新たな粒子を追加する。粒子群は、新たに追加された粒子と位置情報を共有することで、局所解からの脱出が可能になり、再び最適解を探索することができる。数値実験を通して、成長型 DE の探索性能を従来手法の DE と比較しながら考察していく。

2.2 成長型 DE アルゴリズム

本修士論文では基本的な最適化問題として、 m 次元の関数 $F(\vec{x})$ の最小値探索を行う ($\vec{x} \equiv (x_1, \dots, x_m) \in R^m$ 、 $F(\vec{x}) \in R$)。時間 t における粒子数は $N(t)$ で表す。 i ($i = 1, \dots, N(t)$) 番目の粒子の位置情報は $\vec{x}_i \equiv (x_{i1}, \dots, x_{im}) \in R^m$ 、カウンターは $c_i(t)$ で表される。成長型 DE では、位置情報の更新の有無について調べ、更新されない場合は、カウンター $c(t)$ を増やしていく。以下で関数 $F(\vec{x})$ の最小値を求める問題における成長型 DE の探索アルゴリズムを定義する。2.2 では、粒子が局所解から脱出した様子である。

形式は、DE/rand/1/bin とする。

Step 1 (初期化): $t = 0$ において、粒子数 $N(t)$ を設定。各粒子 $i = 1 \sim N(t)$ について、位置ベクトル $\vec{x}_i(t)$ を探索範囲 D_s 内にランダムに配置する。カウンター $c(t)$ を初期化する。

Step 2 (突然変異): 各個体 $\vec{x}_i(t)$ に対して、 $i = 1 \sim N(t)$ の中から 3 個体 $\vec{x}_{p_1}(t), \vec{x}_{p_2}(t), \vec{x}_{p_3}(t)$ をランダムに選択する。ここで、 $\vec{x}_i(t)$ と $\vec{x}_{p_1}(t), \vec{x}_{p_2}(t), \vec{x}_{p_3}(t)$ は互いに重複しないようにする。新しいベクトル $\vec{y}_i(t)$ を基本ベクトル $\vec{x}_{p_1}(t)$ および差分ベクトル $\vec{x}_{p_2}(t) - \vec{x}_{p_3}(t)$ から以下のように生成する。

$$\vec{y}_i(t) = \vec{x}_{p_1}(t) + B(\vec{x}_{p_2}(t) - \vec{x}_{p_3}(t)) \quad (2.1)$$

ここで、 B はスケーリングパラメータである。

Step 3 (交叉): $\vec{y}_i(t_1)$ を親 $\vec{x}_i(t_1)$ と交叉し、子ベクトル $\vec{x}_i^{new}(t_1)$ を生成する。交叉点 j を全ての次元 $[1, m]$ からランダムに選択する。子ベクトル $\vec{x}_i^{new}(t_1)$ の j 番目の要素を $\vec{y}_i(t_1)$ の j 番目の要素から継承する。それ以降の次元は、交叉率 CR の確率で、 $\vec{y}_i$ の要素から継承し、 $1 - CR$ の確率で親 $\vec{x}_i(t_1)$ から継承する。

Step 4 (生存者選択): 親 $\vec{x}_i(t)$ と子ベクトル $\vec{x}_i^{new}(t)$ の適応度を比較して、適応度の高いベクトルを次世代に残す。

$$\begin{aligned}
& \text{if } f(\vec{x}_i^{new}(t)) < f(\vec{x}_i(t)) \text{ then} \\
& \quad \vec{x}_i(t) = \vec{x}_i^{new}(t) \\
& \text{if } f(\vec{x}_i(t)) < f(\vec{x}_i^{new}(t)) \text{ then} \\
& \quad \vec{x}_i(t) = \vec{x}_i(t)
\end{aligned} \tag{2.2}$$

Step 5(カウンターの更新): 更新後の粒子全体の中で最良の値を $gbest(t)$ とする。

$$\begin{aligned}
& \text{if } gbest(t) = gbest(t-1) \text{ then} \\
& \quad c(t) = c(t) + 1 \\
& \text{if } gbest(t) < gbest(t-1) \text{ then} \\
& \quad c(t) = 0
\end{aligned} \tag{2.3}$$

Step 6 (粒子の追加): カウンター $c(t)$ とカウンターのしきい値 T_{int} を比較し、 $c(t)$ が T_{int} に達したらランダムに粒子を M 個追加する。また、その際にカウンターはリセットする。もし、新しい粒子を追加してから次の粒子の追加まで $gbest(t)$ に変化がないときは前に追加した粒子の削除を行う。

$$\begin{aligned}
& \text{if } c(t) = T_{int} \text{ then} \\
& \quad N = N + 1 \\
& \quad c(t) = 0
\end{aligned} \tag{2.4}$$

Step 7 (探索回数の更新): 探索回数 t を、指定した探索回数 t_{max} に至るまで更新する。

$$\begin{aligned}
& \text{If } t < t_{max} \text{ then } t = t + 1 \text{ Step2 } \curvearrowright \\
& \text{If } t = t_{max} \text{ then 探索終了}
\end{aligned} \tag{2.5}$$

成長型 DE により粒子が局所解から脱出した様子を図 2.2 に示す。 図 2.2(a) では、粒子が局所解に陥ってしまっている。しかし、図 2.2(b) で新しい粒子を新しく追加することによって図 2.2(c) で新しく追加した粒子に他の粒子が引き寄せられ、局所解から抜け出し新たに探索を続け始めている。図 2.2(d) では、局所解から脱出した粒子がさらなる探索により最適解を見つけ出すことに成功している。

2.3 数値実験

成長型 DE の性能を検証するために二つのベンチマークで実験を行う。また、非成長型 DE と比較ををするために非成長型 DE の典型的な探索結果も見てみる。今回の実験での形式は、DE/rand/1/bin を用いた。

初期パラメータは、 $B = 0.7, CR = 0.9$ とし、最大反復回数 T_{max} を 1000 回とした。 t は、探索回数を表し、 $N(0)$ は $t = 0$ のときの粒子の数、 $N(T_{last})$ は試行を終了したときの粒子の数を表す。

2.3.1 実験 1

まず初めに、非成長型 DE がどのような探索を行うかを見るために、単峰性関数である Sphere function(2.6) を用いて、実験を行った。

$$\begin{aligned} f_1(x_1, x_2) &= x_1^2 + x_2^2 \\ -50.0 &\leq x_1, x_2 \leq 50.0 \end{aligned} \tag{2.6}$$

この関数の形状を図 2.3 に示す。

この関数の最適解は $(x_1, x_2) = (0, 0)$ のとき $f_1(0, 0) = 0$ である。終了条件は (1) 評価値が 1.0×10^{-2} に達したとき、(2) 反復回数が $T_{max} = 3000$ に達したときのどちらかを満たした場合とし、 $T_{int} = 20$ とした。図 2.4 は、非成長型 DE の典型的な探索例、図 2.5 はこのときの粒子の動きである。

初めはランダムに散らばっていた粒子が探索回数を重ねるごとに最適解に収束していつていることがわかる。また、評価値も基準値である 1.0×10^{-2} に近づいていき、最終的には基準値を満たしている。関数 (2.6) のような単峰性関数なら非成長型 DE でも、最適解を探索することができる。

2.3.2 実験2

次に、多峰性関数である Normalized Schwefel function(2.7) を用いて実験を行った。

$$f_2(\vec{x}) = -418.9 + \frac{\sum_{i=1}^D -x_i \sin(\sqrt{|x_i|})}{D} \quad (2.7)$$
$$-512.0 \leq x_i \leq 512.0$$

ただし、 $i = 1, \dots, D$ で今回は $D = 2$ とする。

この関数の形状を図 2.6 に示す。

また、この関数の最適解は、 $(x_1, x_2) = (420.9, 420.9)$ のとき $f_2(420.9, 420.9) = 0$ である。終了条件は実験 1 と同様で、 $T_{int} = 20, N(0) = 10$ とした。図 2.6 を見ると関数 (3.7) には、複数の局所解が存在する。図 2.7 ~ 図 2.12 は、それぞれ非成長型 DE の $N = 10, N = 20, N = 30$ の典型的な探索例, 粒子の動きである。見てわかる通りどの探索例も反復回数が 10 回あたりで局所解に陥ってしまい、以後局所解から脱出することができず最適解を探索することができていないことがわかる。次に、成長型 DE の実験結果を見てみる。図 2.13 ~ 図 2.18 は、 $M = 1, M = 2, M = 3$ のときの典型的な探索例と t と粒子の増加の関係, 粒子の動きである。探索例を見てみると、一旦は局所解に陥ってしまっているが、新しい粒子を追加することにより局所解から脱出が可能になっている。

次に、表 2.1 に数値実験の結果を示す。この表は、試行回数 50 回における $M = 0 \sim 3$ のときの成功率、平均粒子数、平均探索回数である。各項目とも、初期値を変えて 50 回の試行を行い、 $M = 0$ では $N = 30, M = 1 \sim 3$ では $N(t) = 10$ とした。他のパラメータは上記で説明した数値で固定した。また、 $M = 0$ というのは、粒子を追加しないので非成長型 DE と同じである。各項目を詳しく説明すると、SR は成功率で、最大探索回数 $T_{max} = 3000$ 回以内に評価値が 1.0×10^{-2} に達した場合を成功とし、他は失敗とする。 $\sharp ITL$ は、平均反復回数で、探索を終了したときの反復回数の平均で、最大は、 $T_{max} = 500$ 回である。 $\sharp PCL$ は平均粒子数で、探索を終了したときの粒子数 $N(T_{last})$ の平均である。

表 2.1 を詳しく見てみると、 $M = 0$ では粒子が局所解に陥ってしまい成功率が成長型

DE に比べ低くなっている。成長型 DE は、局所解に陥っても脱出ができるので 100% の成功率になっている。しかし、同じ成長型 DE でも追加するときの粒子が探索回数によって大きく変わっていることがわかる。 $M = 1$ に比べ、 $M = 2, M = 3$ の方が探索回数がそれぞれ 3 倍, 10 倍近く少なくなっている。このことより、探索回数は追加する粒子に大きく依存しているがわかる。

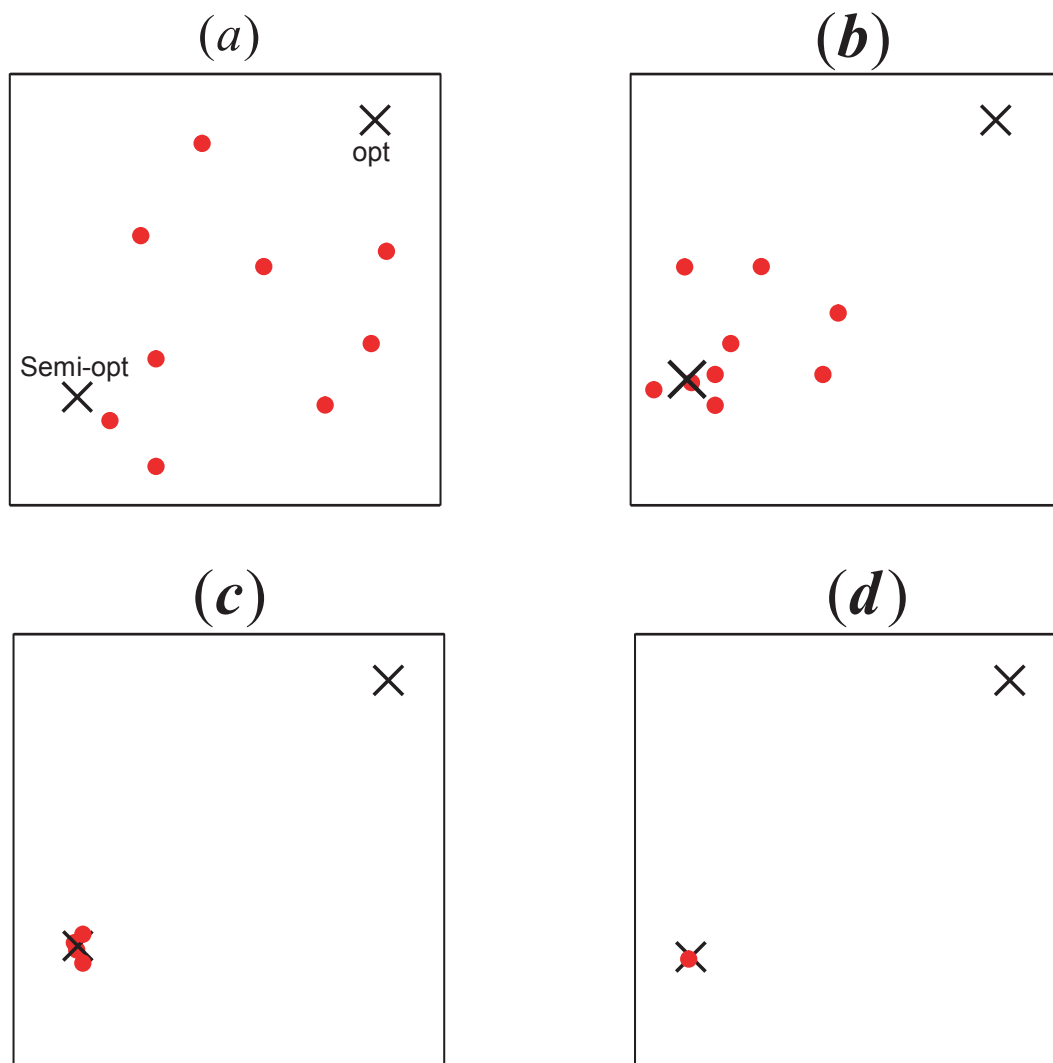


図 2.1: 粒子が局所解に陥る様子

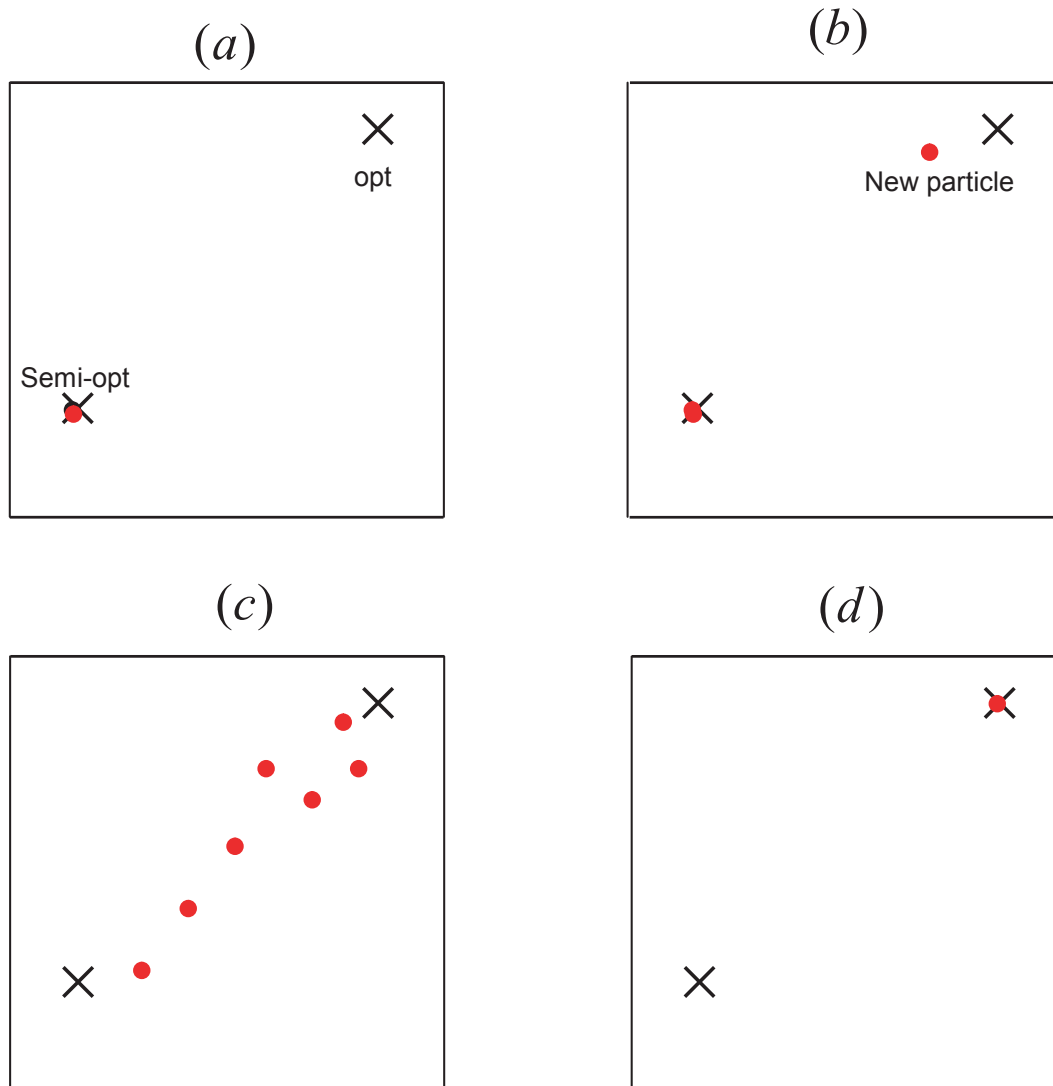


図 2.2: 粒子が局所解から脱出した様子

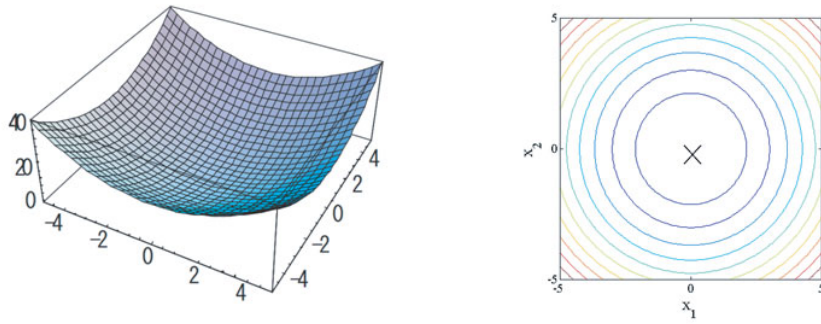


図 2.3: Sphere function の形状

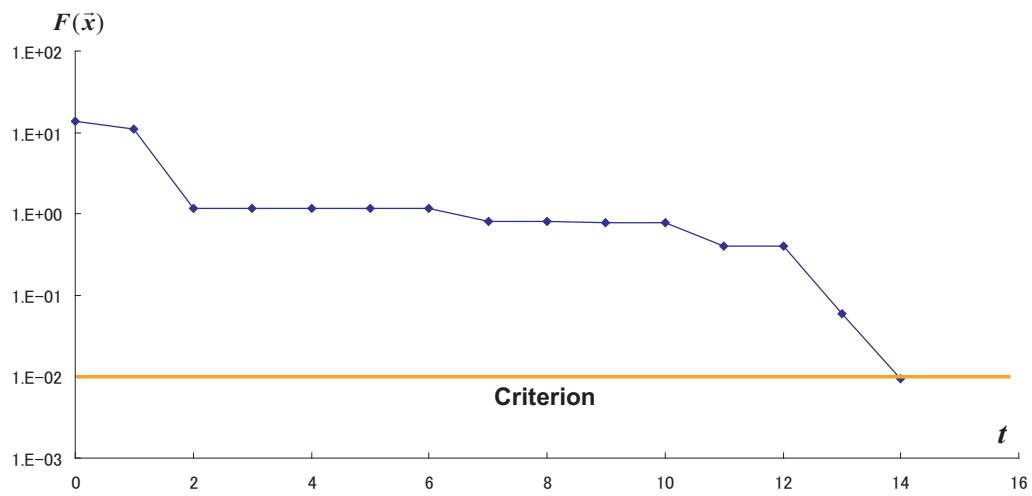


図 2.4: 非成長型 DE による Sphere function の探索過程

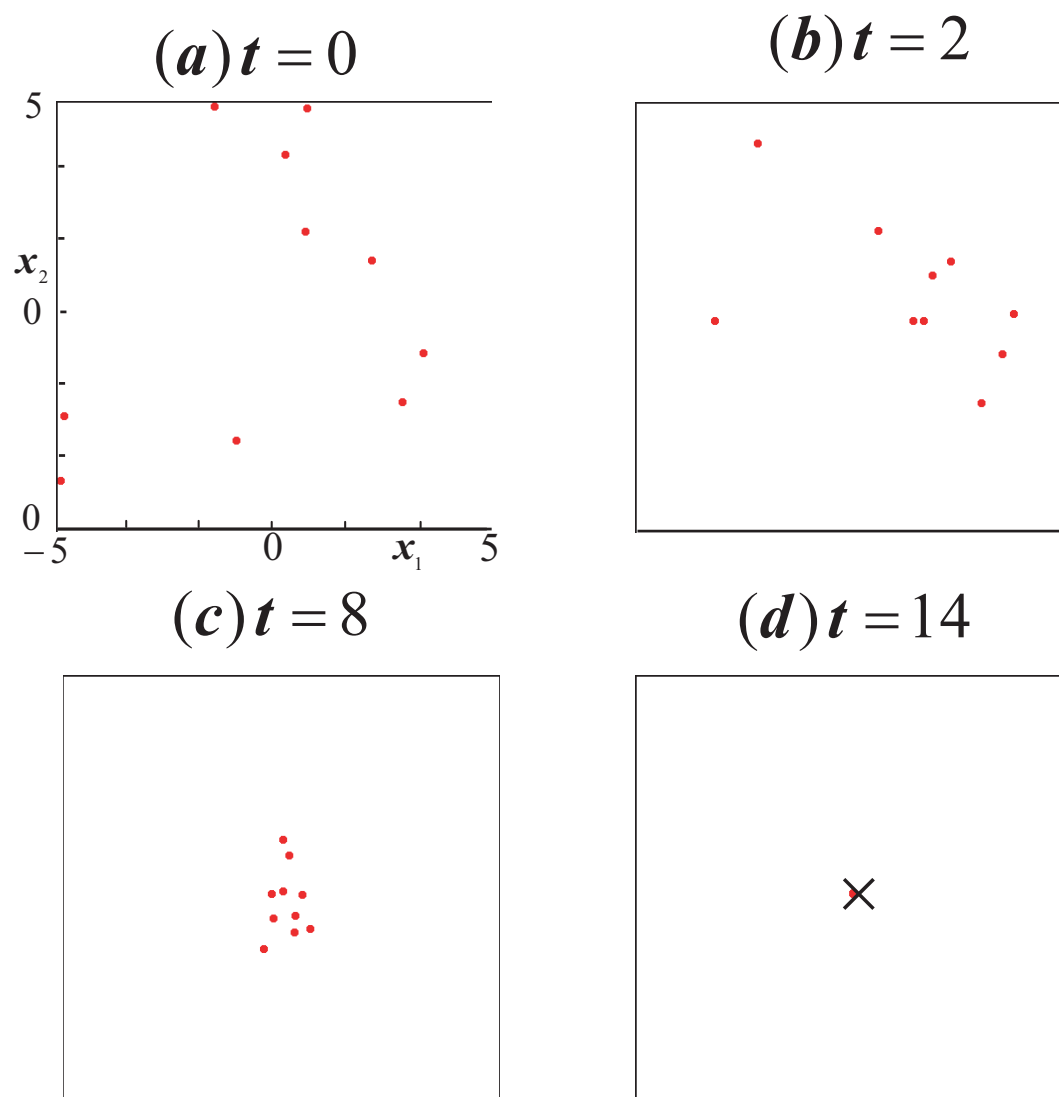


図 2.5: 非成長型 DE による Sphere function の粒子の動き

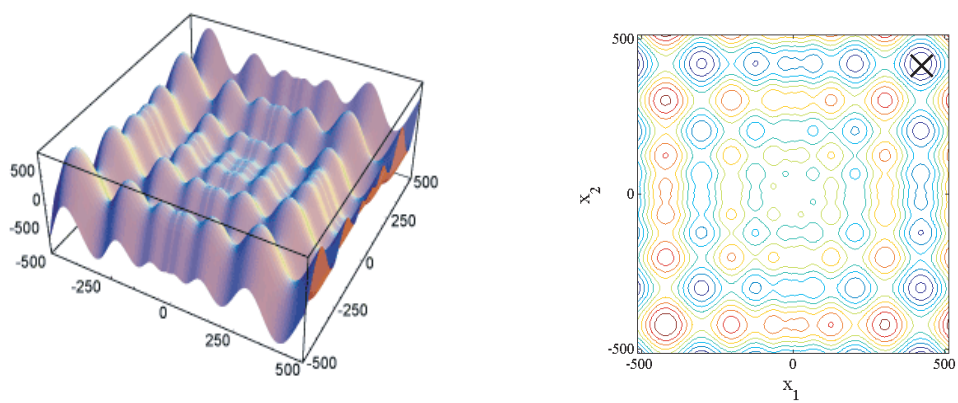


図 2.6: Normalized Schwefel function の形状

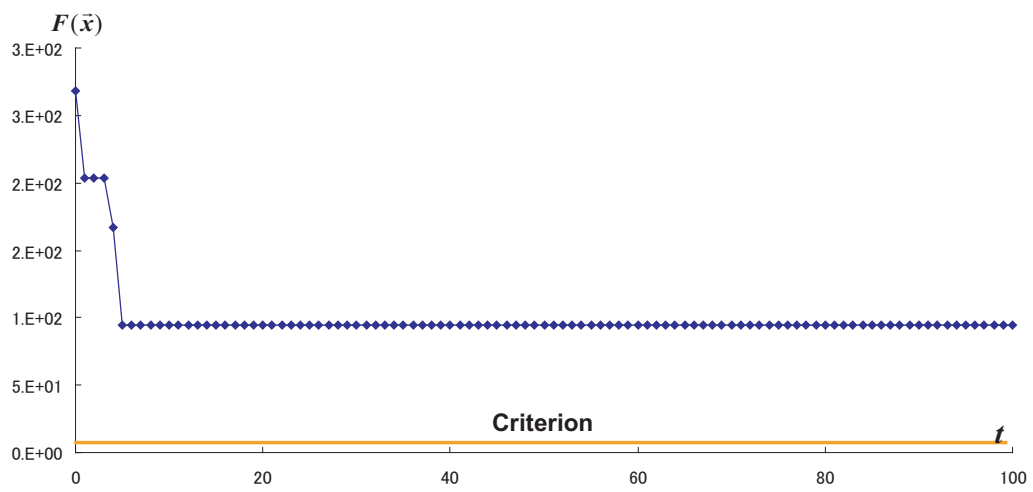


図 2.7: 非成長型 DE による Normalized Schwefel function の探索過程 ($N = 10$)

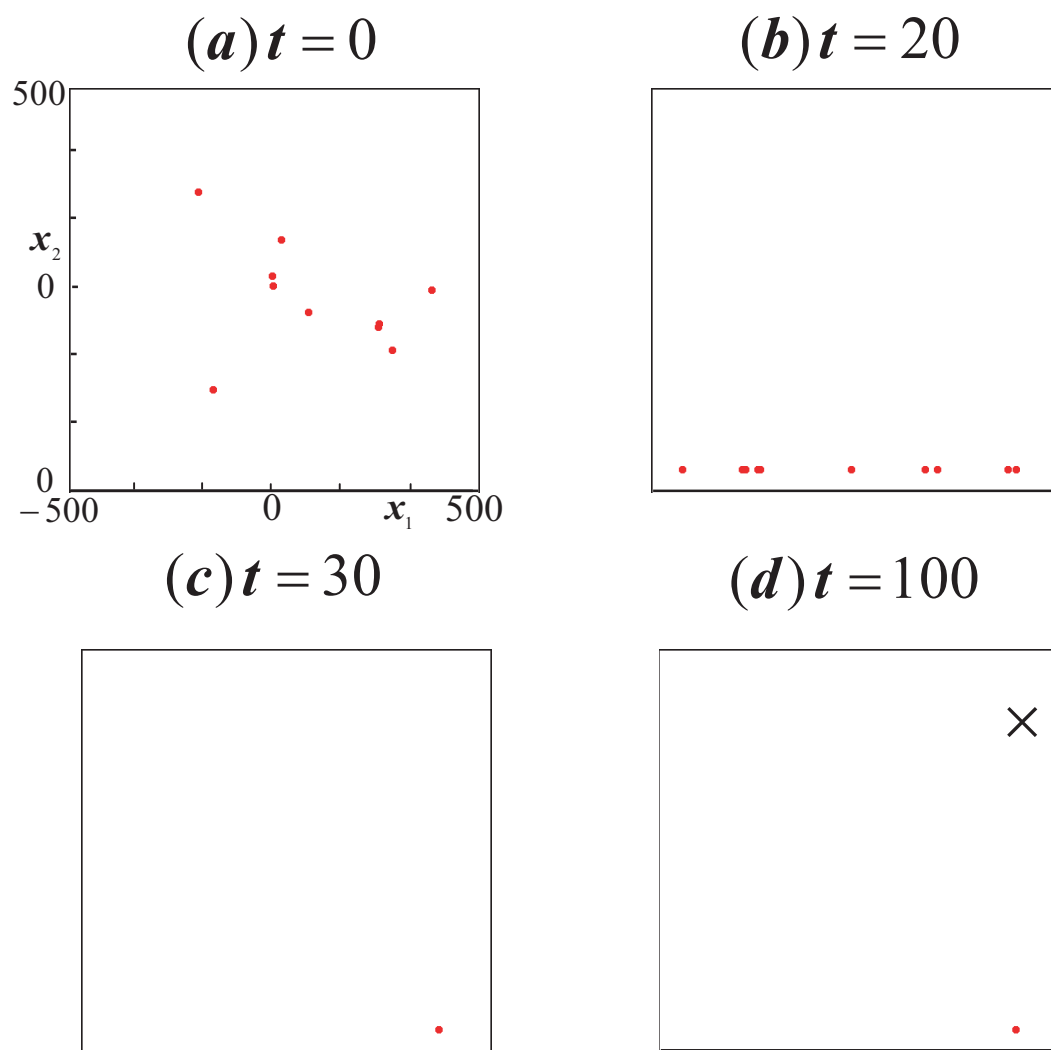


図 2.8: 非成長型 DE による Normalized Schwefel function の粒子の動き ($N = 10$)

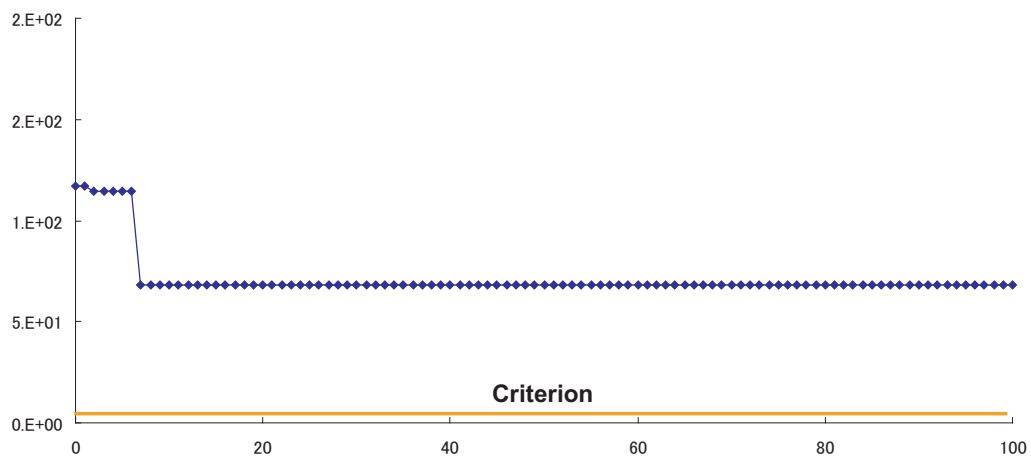
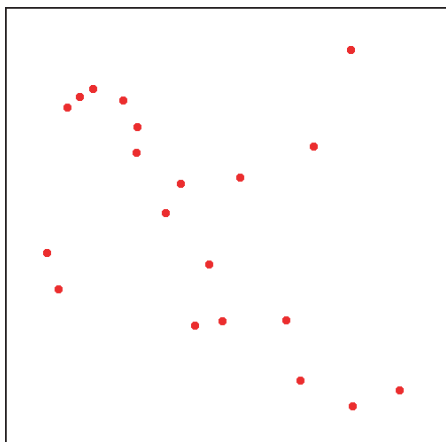
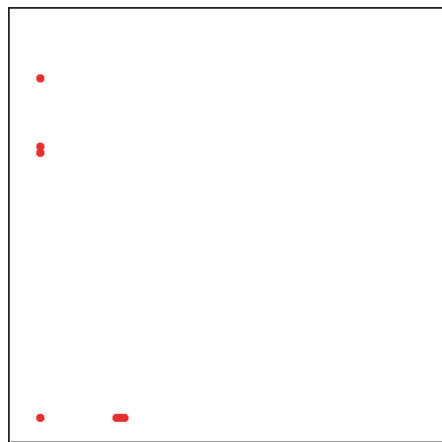


図 2.9: 非成長型 DE による Normalized Schwefel function の探索過程 ($N = 20$)

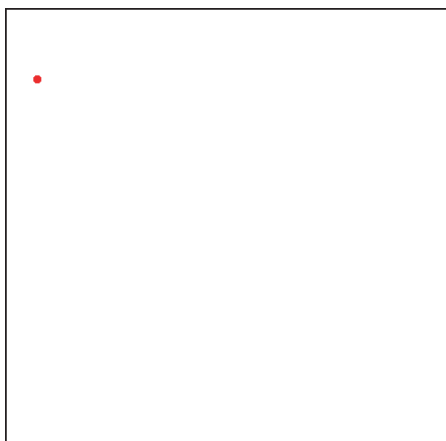
(a) $t = 0$



(b) $t = 10$



(c) $t = 50$



(d) $t = 100$

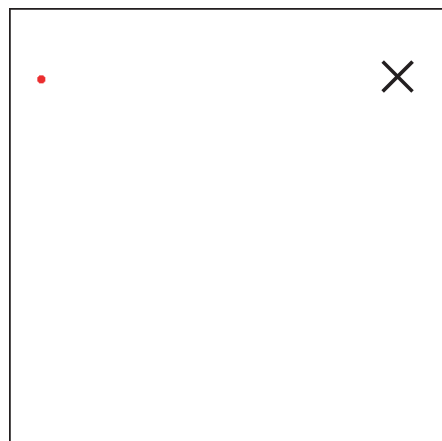


図 2.10: 非成長型 DE による Normalized Schwefel function の粒子の動き ($N = 20$)

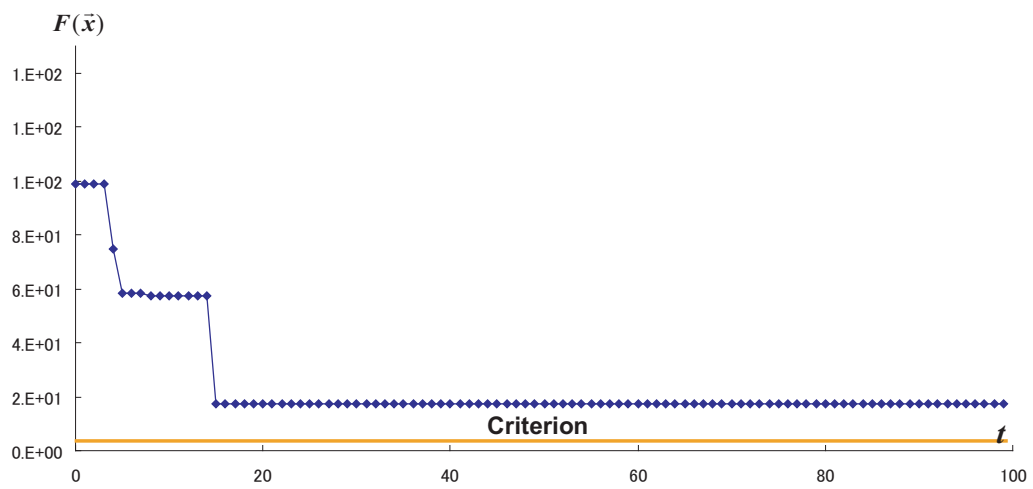
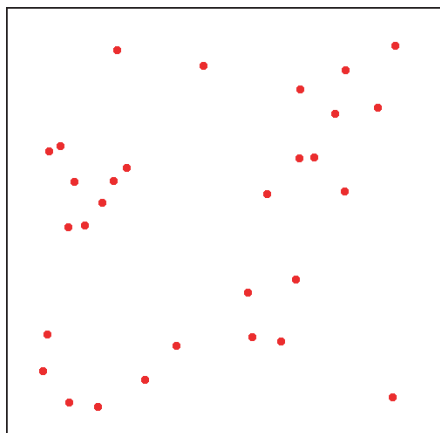
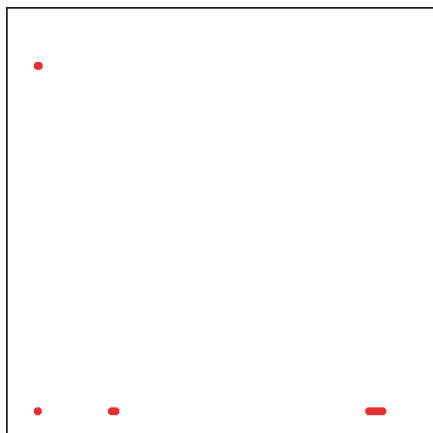


図 2.11: 非成長型 DE による Normalized Schwefel function の探索過程 ($N = 30$)

(a) $t = 0$



(b) $t = 10$



(c) $t = 50$



(d) $t = 100$

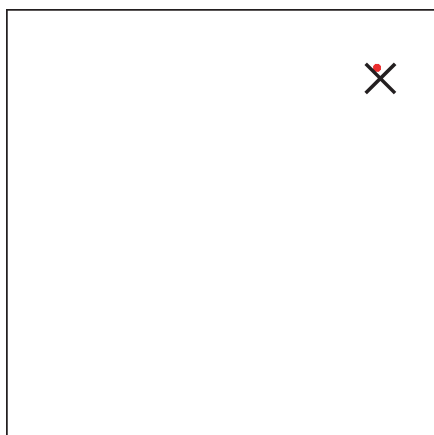


図 2.12: 非成長型 DE による Normalized Schwefel function の粒子の動き ($N = 30$)

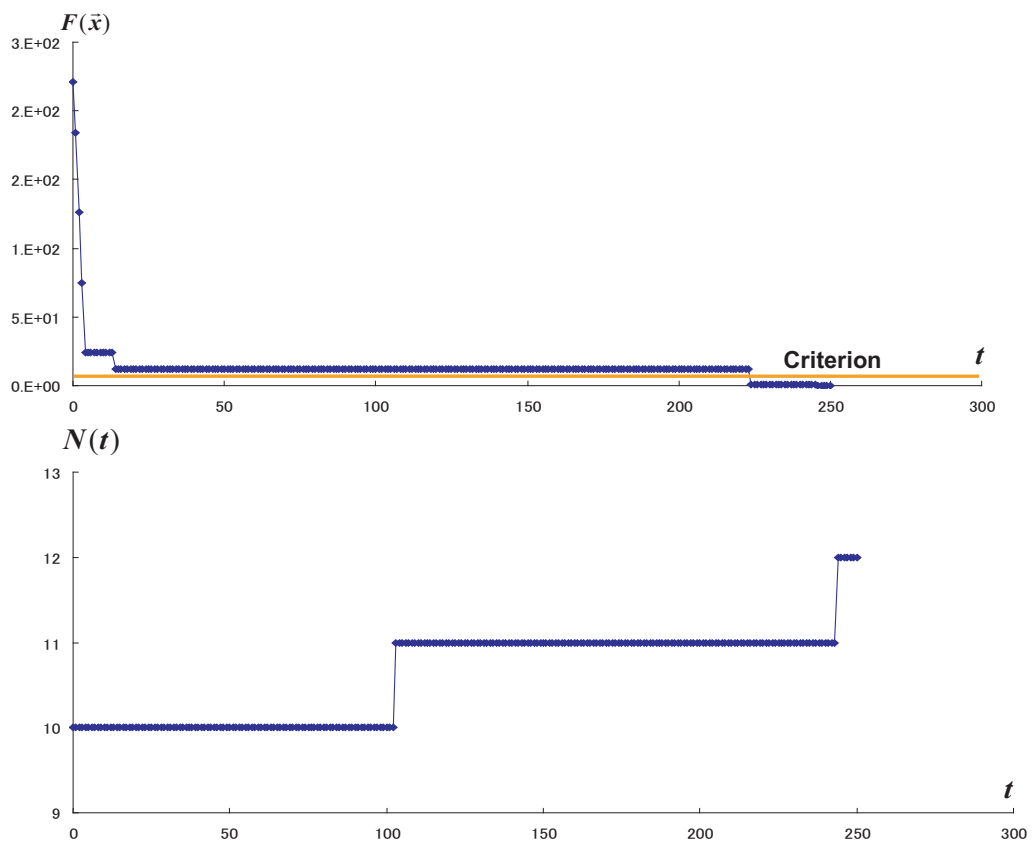
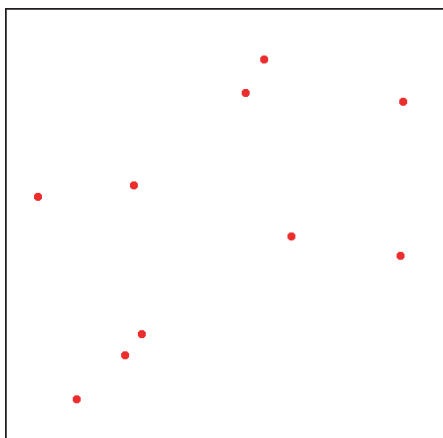
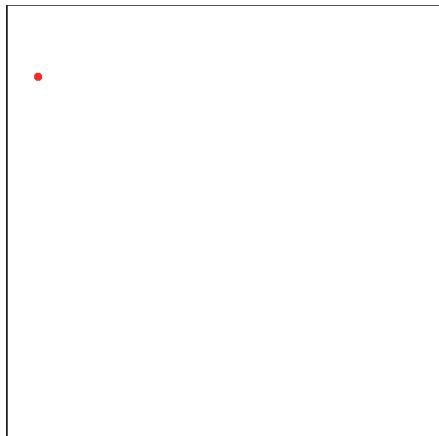


図 2.13: 成長型 DE による Normalized Schwefel function の探索過程, 粒子数 ($M = 1$)

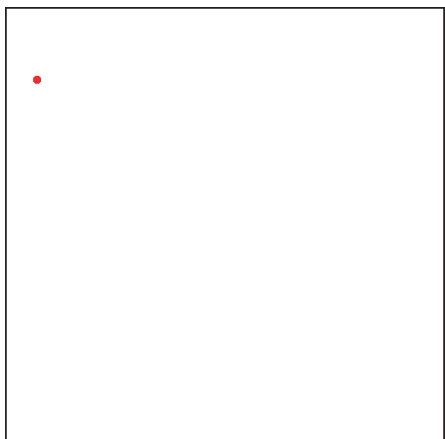
(a) $t = 0$



(b) $t = 50$



(c) $t = 150$



(d) $t = 258$

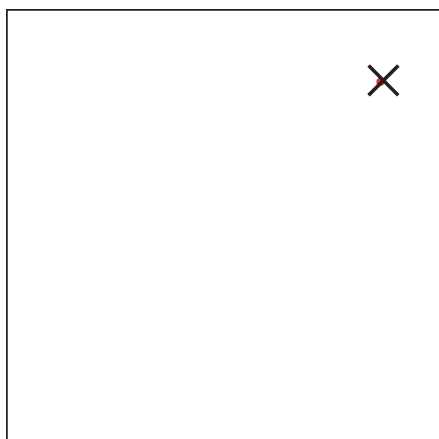


図 2.14: 成長型 DE による Normalized Schwefel function の粒子の動き ($M = 1$)

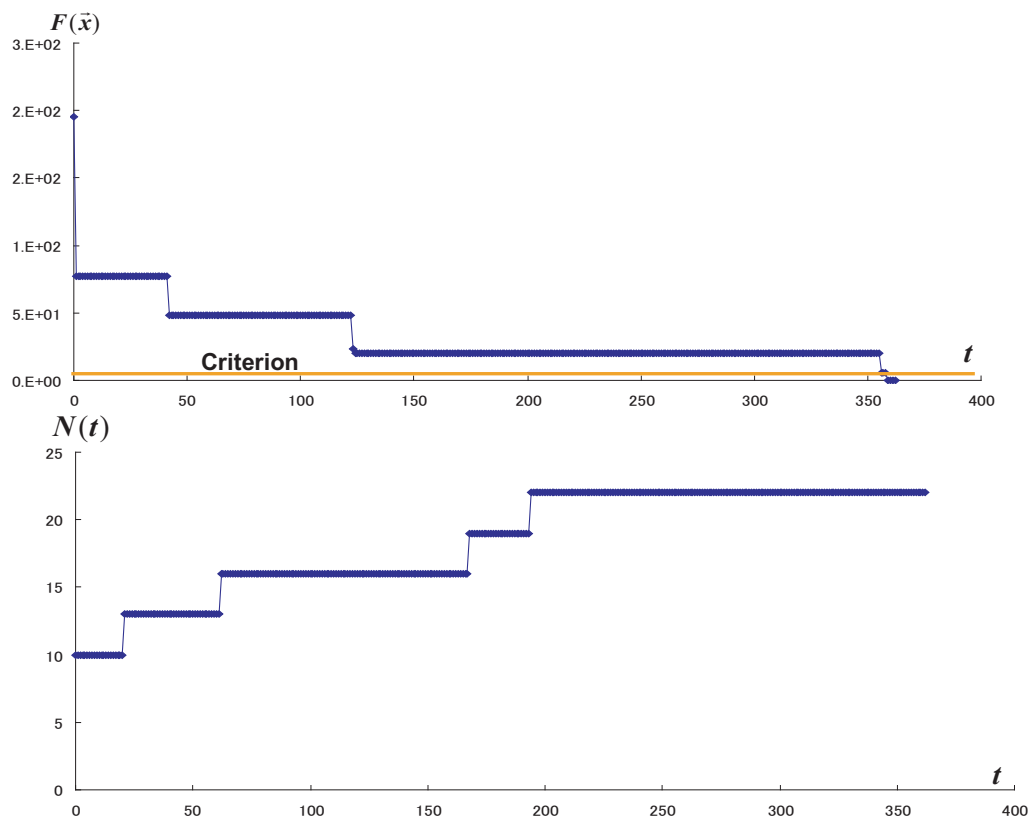


図 2.15: 成長型 DE による Normalized Schwefel function の探索過程, 粒子数 ($M = 2$)

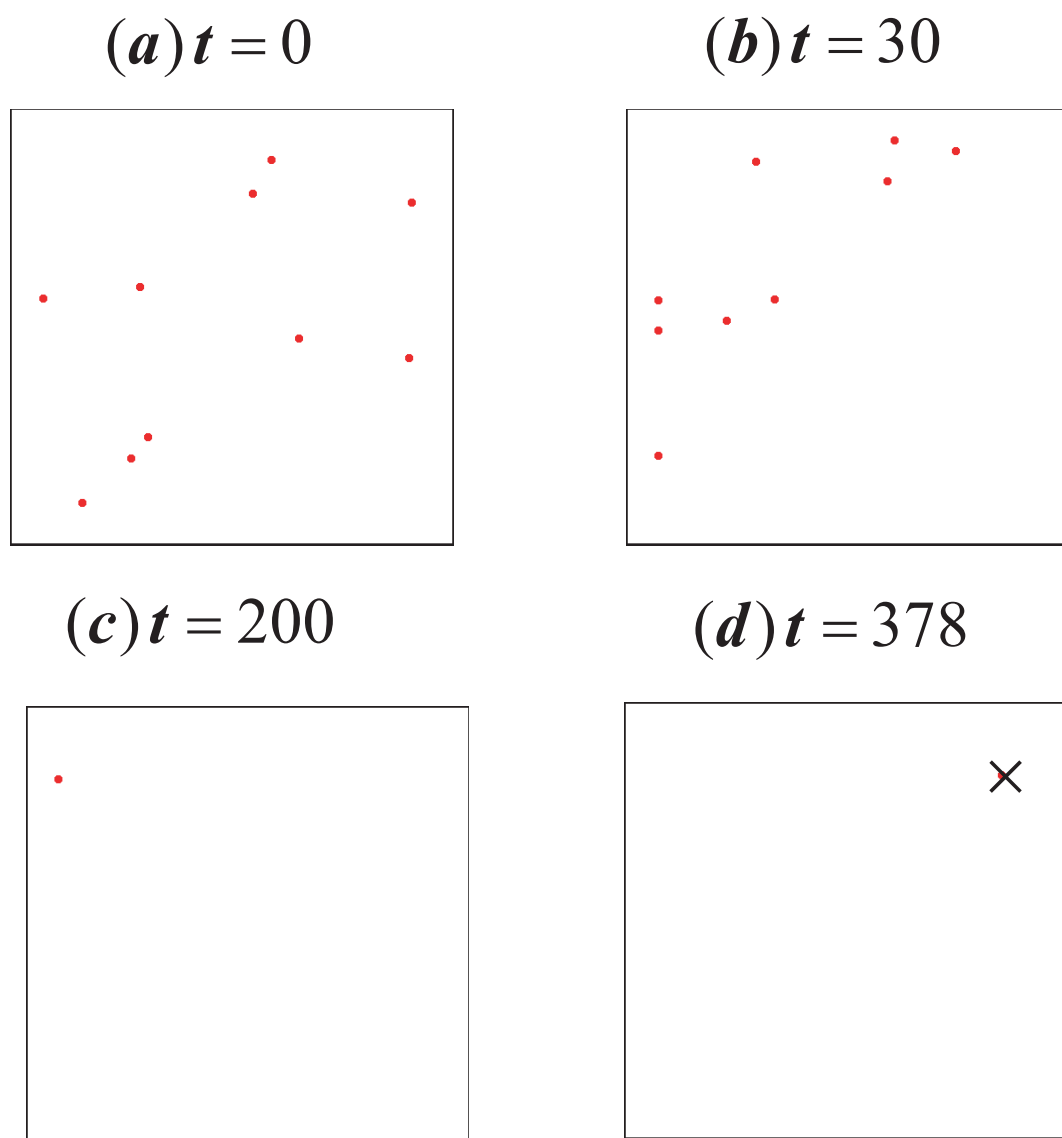


図 2.16: 成長型 DE による Normalized Schwefel function の粒子の動き ($M = 2$)

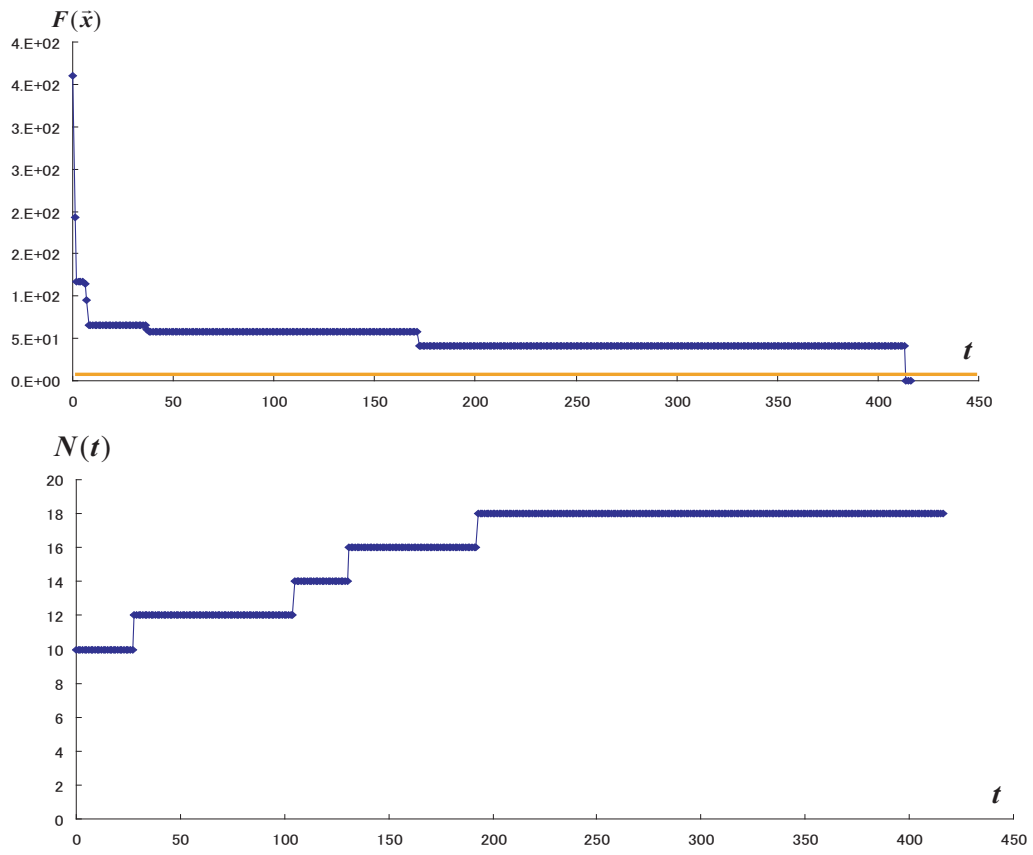


図 2.17: 成長型 DE による Normalized Schwefel function の探索過程, 粒子数 ($M = 3$)

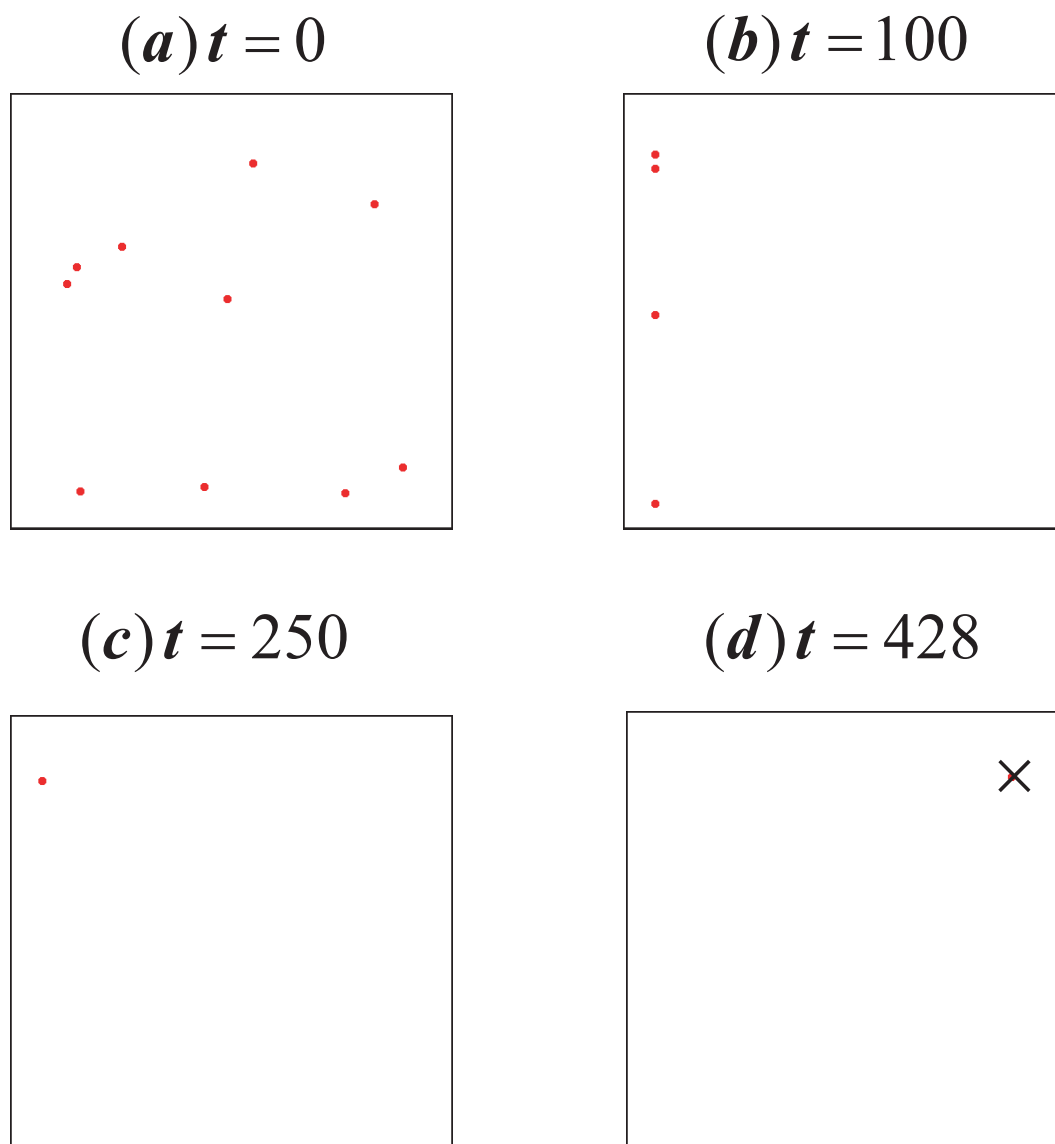


図 2.18: 成長型 DE による Normalized Schwefel function の粒子の動き ($M = 3$)

表 2.1: f_2 に対する成長型 DE の探索結果

M	成功率	粒子数	探索回数
0	92	30.0	83
1	100	22.4	1945
2	100	23.5	603
3	100	24.0	206

第3章 Insensitive Differential Evolution

3.1 まえがき

DE は各粒子の粒子と位置情報を共有し、自身と他の粒子の情報を差分方程式によって最適解に向かって更新していく。この集団での探索を行うことで1点探索と比較して幅広い探索を行なえ、最良の情報を粒子間で共有する事で群全体が最良な方へ移動する事が出来る。しかし最適解を複数持つ問題に対しては、従来手法のDEでの探索では全ての最適解を求められないケースがある。例えばある粒子が複数ある最適解の中の1つを見つけ出すと、その粒子と情報を共有した他の粒子も発見された最適解へと収束してしまい、他の最適解の発見が困難になってしまう。この問題点の解決のために、タブー探索[17]などのアルゴリズムが提案されている。

本章では、複数最適化探索問題の適用に粒子を鈍感にしたDE(Insensitive Differential Evolution, IDE)を提案する。IDEは、global search と local search の二つによって構成されている。global search では、それぞれ任意の解を持つ local subregions(LSR)を作成する。IDEには、粒子の位置情報の更新を鈍感にするパラメータ ε とlocal searchに切り替えるタイミングをコントロールする T_G という二つの重要なパラメータが存在する。 ε の値を小さくしすぎてしまうと、粒子は従来のDEと似た動きをしてしまい、すべての粒子が一箇所の最適解に収束してしまう可能性が高くなる。このような場合、最適解を含むすべてのLSRを作成するのは難しくなる。 ε の値を増加させていくと、粒子の鈍感さが大きくなっていくので、一箇所の最適解に収束する前にLSRを作成できる。 T_G は、global search を止め、local search に切り替えるためのパラメータである。local search では、並列にそれぞれのLSRを操作し、すべての最適解を探索する。も

し、global search が成功し、すべての LSR を作成することができれば、local search によってすべての最適解を探索していく。探索を成功させるには、 ε と T_G の二つのパラメータの決定が重要になってくる。

3.2 IDE アルゴリズム

IDE は、global search と local search の二つのステージで構成されている。global search では、それぞれ任意の解を持つ local subregions(LSR) を作成する。local search では、並列にそれぞれの LSR を操作し、すべての最適解を探索する。

本論文では、 m 次元関数 $F : \vec{R}^m \rightarrow \vec{R}$ 、 $\vec{x} = (x_1, x_2, \dots, x_m)$ の最適化を対象とする。粒子の位置を $\{\vec{x}_i, i = 1, 2, \dots, N\}$ 、粒子数を N とする。以下で IDE のアルゴリズムについて説明する。また、アルゴリズムの説明には、DE/rand/1/bin を用いた。

3.2.1 global search

global search では、それぞれ任意の解を持つ local subregions(LSR) を作成する。探索回数 t_1 における粒子を以下のように定義する。

$$X \equiv \{\vec{x}_1(t_1), \dots, \vec{x}_N(t_1)\}$$

N は、粒子数である。以下で global search のアルゴリズムを 5 つのステップで定義する。

Step 1 (初期化): $t_1 = 0$ のとき、各粒子の位置 $\vec{x}$ を初期化する。各粒子の位置 $\vec{x}_i(0)$ をランダムに決定する。

Step 2 (突然変異): 各個体 $\vec{x}_i(t)$ に対して、 $i = 1 \sim N$ の中から $x_{p_1}(t)$ 、 $x_{p_2}(t)$ 、 $x_{p_3}(t)$ をランダムに選択する。ここで、 $\vec{x}_i(t)$ と $x_{p_1}(t)$ 、 $x_{p_2}(t)$ 、 $x_{p_3}(t)$ は互いに重複しないようにする。新しいベクトル $\vec{y}_i(t)$ を基本ベクトル $x_{p_1}(t)$ および差分ベクトル $x_{p_2}(t) - x_{p_3}(t)$

から以下のように生成する。

$$\vec{y}_i(t_1) = \vec{x}_{p1}(t_1) + B(\vec{x}_{p2}(t_1) - \vec{x}_{p3}(t_1)) \quad (3.1)$$

ここで、 B はスケーリングパラメータである。

Step 3 (交叉): $\vec{y}_i(t_1)$ を親 $\vec{x}_i(t_1)$ と交叉し、子ベクトル $\vec{x}_i^{new}(t_1)$ を生成する。交叉点 j を全ての次元 $[1, m]$ からランダムに選択する。子ベクトル $\vec{x}_i^{new}(t_1)$ の j 番目の要素を $\vec{y}_i(t_1)$ の j 番目の要素から継承する。それ以降の次元は、交叉率 CR の確率で、 $\vec{y}_i$ の要素から継承し、 $1 - CR$ の確率で親 $\vec{x}_i(t_1)$ から継承する。

Step 4 (生存者選択): 親 $\vec{x}_i(t_1)$ と子ベクトル $\vec{x}_i^{new}(t_1)$ の適応度を比較して、適応度の高いベクトルを次世代に残す。この時に、パラメータ ε_1 を用いて、粒子の動きを鈍感にする。

$$\begin{aligned} &\text{if } (f(\vec{x}_i(t_1)) - f(\vec{x}_i^{new}(t_1))) > \varepsilon_1 \text{ then} \\ &\quad \vec{x}_i(t_1) = \vec{x}_i^{new}(t_1) \\ &\text{if } (f(\vec{x}_i(t_1)) - f(\vec{x}_i^{new}(t_1))) < \varepsilon_1 \text{ then} \\ &\quad \vec{x}_i(t) = \vec{x}_i(t_1) \end{aligned} \quad (3.2)$$

Step 5 (探索回数の更新): 探索回数 t を、指定した探索回数 T_G に至るまで更新する。

$$\begin{aligned} &\text{If } t_1 < T_G \text{ then } t_1 = t_1 + 1 \text{ Step2} \curvearrowright \\ &\text{If } t_1 = T_G \text{ then 探索終了} \end{aligned} \quad (3.3)$$

3.2.2 Local subregions

Step 1: i 番目の LSR を LSR_i とし、 $i = 1$ とする。

Step 2: 粒子の中で最良の粒子 $\vec{x}_g$ を選ぶ。以下の式のように、 $\vec{x}_g$ を中心に半径 r の領域を作り、 LSR_i を作成する。

$$LSR_i = \{ \vec{x} \mid \| \vec{x} - \vec{x}_g \| < r \}$$

LSR_i の粒子を取り除く。図 4.1 は LSR の作成図である。

Step 3: すべての粒子がどこかの LSR に属するまで更新する。

3.2.3 Local search

local search では、並列にそれぞれの LSR を操作し、すべての最適解を探索する。 t_2 は、探索回数である。 LSR_i の粒子を以下のように定義する。

$$X_i \equiv \{\vec{x}_1(t_2), \dots, \vec{x}_{N_i}(t_2)\}$$

LSR_i における、local search のアルゴリズムを以下の 5 つのステップで定義する。

Step 1 (初期化): $t_2 = 0$ とする。

Step 2 (突然変異): 2.1 の Step2 の t_1 、 X 、 N を t_2 、 X_i 、 N_i に置き換え同様に突然変異を行う。

Step 3 (交叉): 2.1 の Step3 の t_1 、 X 、 N を t_2 、 X_i 、 N_i に置き換え同様に交叉を行う。

Step 4 (生存者選択): 親 $\vec{x}_i(t_2)$ と子ベクトル $\vec{x}_i^{new}(t_2)$ の適応度を比較して、適応度の高いベクトルを次世代に残す。この時に、パラメータ ε_2 を用いて、粒子の動きを鈍感にする。

$$\begin{aligned} &\text{if } (f(\vec{x}_i(t_2)) - f(\vec{x}_i^{new}(t_2)) > \varepsilon_2) \text{ then} \\ &\quad \vec{x}_i(t_2) = \vec{x}_i^{new}(t_2) \\ &\text{if } (f(\vec{x}_i(t_2)) - f(\vec{x}_i^{new}(t_2)) < \varepsilon_2) \text{ then} \\ &\quad \vec{x}_i(t_2) = \vec{x}_i(t_2) \end{aligned} \tag{3.4}$$

LSR_i の最良の値を $f(\vec{x}_{Li})$ とし、 $f(\vec{x}_{Li}) < C_1$ を満たしたら探索を終了する。それ以外の場合は、Step5 に進む。 C_1 は、近似基準解である。

Step 5 (探索回数の更新): 探索回数 t_2 を、指定した探索回数 T_L に至るまで更新する。

$$\begin{aligned}
&\text{If } t_2 < T_L \text{ then } t_2 = t_2 + 1 \text{ Step2} \rightarrow \\
&\text{If } t_2 = T_L \text{ then 探索終了}
\end{aligned} \tag{3.5}$$

3.3 数値実験

IDE の性能を検証するために二つのベンチマークで実験を行う。今回の実験での形式は、DE/rand/1/bin を用いた。初期パラメータは、交叉率 $CR = 0.9$ 、スケーリングパラメータ $B = 0.7$ 、近似基準解 $C_1 = 0.01$ 、LSR の半径 $r = 1$ 、local search の最大探索回数 $T_L = 70$ と固定して用いた。

3.3.1 実験 1

まず、Branin function(3.6) に IDE の適用を行った:

$$\begin{aligned}
f_1(\vec{x}) &= -0.397 + (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 \\
&+ (1 - \frac{1}{8\pi})\cos(x_1) + 10 \\
&-5.0 \leq x_1 \leq 10.0 \\
&0.0 \leq x_2 \leq 15.0 \\
\min(f_1(\vec{x})) &= 0 \\
x_{opt1}^{\rightarrow} &= (-\pi, 12.275) \\
x_{opt2}^{\rightarrow} &= (\pi, 2.275) \\
x_{opt3}^{\rightarrow} &= (9.42478, 2.475)
\end{aligned} \tag{3.6}$$

この関数は 3 つの最適解を持つ。この関数の形状を図 4.2 に示す。

図 4.3 は、IDE の Global search における粒子の動き ($\varepsilon=3.0$ 、 $T_G=30$) である。図 4.4 は、IDE の Local search における粒子の動き ($\varepsilon_2=0.01$ 、 $T_L=70$) である。4.3(b)-(d) では、global search を行っていて、粒子はそれぞれ 3 つの群ごとに分かれてきていることが分かる。図 4.4(a) では、それぞれの群ごとに LSR を作り、local search に切り替えている。図 4.4(b) では、粒子が LSR ごとにそれぞれの最適解を探索している。Fig.4.4(a')-(b') は、図 4.4(a)-(b) の拡大図である。図 4.5 は、Global search の探索過程である。図 4.6 は、local search の探索過程である。global search では 3 つの LSR が作成できたら成功

とし、local search では、3 つすべての LSR の粒子が最適解を探索できたら成功とする。表 4.1 は、試行回数 50 回における global search の成功率 (SR) である。表を見てみると、 $(\varepsilon, T_G)=(3, 30)$ の周りで成功率が高くなっていることがわかる。もし、 ε を小さく設定しすぎてしまうと、図 4.7 のように一点に収束になってしまう確率が高くなってしまふ。同様に、 T_G を大きくしすぎてしまっても、図 4.8 のように粒子が他の群に引きつけられてしまう確率が高くなってしまふ。表 4.2 は試行回数 50 回における Local search の成功率 ($\varepsilon=3$ and $T_G = 30$) である。 ε_2 を小さくすると、成功率が高くなっていく傾向がある。

3.3.2 実験 2

次に、Himmelblau function(3.7) に IDE の適用を行った:

$$\begin{aligned}
 f_2(\vec{x}) &= (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \\
 -6.0 &\leq x_1, x_2 \leq 6.0 \\
 \min(f_2(\vec{x})) &= 0 \\
 \vec{x}_{opt1} &= (-2.80, 3.13) \\
 \vec{x}_{opt2} &= (3, 2) \\
 \vec{x}_{opt3} &= (-3.77, -3.28) \\
 \vec{x}_{opt4} &= (3.58, -1.85)
 \end{aligned} \tag{3.7}$$

この関数は 4 つの最適解を持つ。この関数の形状を図 4.9 に示す。

図 4.10 は、IDE の Global search における粒子の動き ($\varepsilon=3.0$ 、 $T_G=40$) である。図 4.11 は、IDE の Local search における粒子の動き ($\varepsilon_2=0.01$ 、 $T_L=60$) である。実験 1 と同様に、4.10(b)-(d) では、global search を行っていて、粒子はそれぞれ 4 つの群ごとに分かれてきていることが分かる。図 4.11(a')-(b') は、図 4.11(a)-(b) の拡大図である。図 4.12 は、Global search の探索過程である。図 4.13 は、local search の探索過程である。global search では 4 つの LSR が作成できたら成功とする。

表 4.3 は、試行回数 50 回における Global search の成功率 (SR) である。表を見てみると、全体的に成功率が下がってしまっていることがわかる。これは、最適解の個数が増えると探索成功率が下がってしまうからだと考えられる。探索 $(\varepsilon, T_G)=(3, 40)$ の周

りで成功率が高くなっていることがわかるが、これは、実験 1 とは違うパラメータである。このことより、LSR を構成するのに最適なパラメータはベンチマークによって異なることがわかる。また、実験 1 と同様に ε を小さく設定しすぎてしまうと、図 4.14 のように探索に失敗してしまう確率が高くなってしまう。同様に、 T_G を大きくしすぎてしまっても、図 4.15 のように粒子が他の群に引きつけられてしまう確率が高くなってしまふ。表 4.10 は local search の成功率 ($\varepsilon=3.0$ 、 $T_G=40$) である。

第4章 様々なIDEによる探索性能

4.1 まえがき

第1章でも述べたように、DEには様々な形式が存在する。本章では、親の選択方法である `base` と差分ベクトルの個数である `num` を変え、様々な差分方程式で情報を更新していき探索性能を比較していく。今回用いる差分方程式を (4.1) ~ (4.5) に示す。 $\vec{x}_{best}$ は、 $\vec{x}_i$ の中で最良の粒子である。

DE/best/1

$$\vec{y}_i(t) = \vec{x}_{best}(t) + B(\vec{x}_{p_1}(t) - \vec{x}_{p_2}(t)) \quad (4.1)$$

DE/best/2

$$\vec{y}_i(t) = \vec{x}_{best}(t) + B(\vec{x}_{p_1}(t) - \vec{x}_{p_2}(t) + \vec{x}_{p_3}(t) - \vec{x}_{p_4}(t)) \quad (4.2)$$

DE/rand/1

$$\vec{y}_i(t) = \vec{x}_{p_1}(t) + B(\vec{x}_{p_2}(t) - \vec{x}_{p_3}(t)) \quad (4.3)$$

DE/rand/2

$$\vec{y}_i(t) = \vec{x}_{p_1}(t) + B(\vec{x}_{p_2}(t) - \vec{x}_{p_3}(t) + \vec{x}_{p_4}(t) - \vec{x}_{p_5}(t)) \quad (4.4)$$

$$\vec{y}_i(t) = \vec{x}_i(t) + B(\vec{x}_{best}(t) - \vec{x}_{p_i}(t)) + \vec{x}_{p_1}(t) - \vec{x}_{p_2}(t) \quad (4.5)$$

4.2 数値実験

様々な IDE の探索性能を検証するために、7 つのベンチマークを用いる。用いるベンチマークは、(a) Sphere function (b) Normalized Schwefel function (c) Six-hump Camel Back function (d) Branin function (e) 分岐図 1 (f) 分岐図 2 (g) 分岐図 3 である。(a),(b) は最適解が 1 つであるが (b) には局所解が存在する。(c) は 2 つ、(d),(e),(g) は 3 つ、(f) は 4 つの最適解を持っている。図 4.16, 図 4.17 は、(a) ~ (g) の最適解の位置を示したものである。また、比較のために従来手法の DE で、(4.1) と (4.3) を使い、IDE は (4.1) ~ (4.5) 使い実験する。実験では、global search の鈍感パラメータ ε 、初期粒子 N を (a), (b) で $N = 10$, (c) で $N = 20$, (d), (e), (g) で $N = 30$, (f) で $N = 40$ を用いている。他のパラメータは、local search の鈍感パラメータ $\varepsilon_2 = 0.01$ 、交叉率 $CR = 0.9$ 、スケールパラメータ $B = 0.7$ 、近似基準解 $C_1 = 0.01$ 、LSR の半径 $r = 1$ 、global search の最大探索回数 $T_G = 30$ 、local search の最大探索回数 $T_L = 70$ を固定して用いた。

(a) Sphere function

$$\begin{aligned} f_1(x_1, x_2) &= x_1^2 + x_2^2 \\ -50.0 &\leq x_1, x_2 \leq 50.0 \\ \min(f_1(\vec{x})) &= 0 \\ \vec{x}_{opt1} &= (0, 0) \end{aligned} \quad (4.6)$$

(b) Normalized Schwefel function

$$\begin{aligned}
f_2(\vec{x}) &= -418.9 + \frac{\sum_{i=1}^D -x_i \sin(\sqrt{|x_i|})}{D} \\
-512.0 &\leq x_i \leq 512.0 \\
\min(f_1(\vec{x})) &= 0 \\
\vec{x}_{opt1} &= (420.9, 420.9)
\end{aligned} \tag{4.7}$$

(c) Six-hump Camel Back function

$$\begin{aligned}
f_3(\vec{x}) &= -1.03 + 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \\
-5.0 &\leq x_i \leq 5.0 \\
\min(f_3(\vec{x})) &= 0 \\
\vec{x}_{opt1} &= (-0.089842, 0.712656) \\
\vec{x}_{opt2} &= (0.089842, -0.712656)
\end{aligned} \tag{4.8}$$

(d) Branin function

$$\begin{aligned}
f_4(\vec{x}) &= -0.397 + (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 \\
&+ (1 - \frac{1}{8\pi})\cos(x_1) + 10 \\
-5.0 &\leq x_1 \leq 10.0 \\
0.0 &\leq x_2 \leq 15.0 \\
\min(f_4(\vec{x})) &= 0 \\
\vec{x}_{opt1} &= (-\pi, 12.275) \\
\vec{x}_{opt2} &= (\pi, 2.275) \\
\vec{x}_{opt3} &= (9.42478, 2.475)
\end{aligned} \tag{4.9}$$

(e) 分岐図 1

$$\begin{aligned}
x &= y + 0.55x + 4\tan^{-1}(x) \\
y &= -0.9x \\
f_5(x, y) &= |y + 0.55x + 4\tan^{-1}(x) - x| + |-0.9x - y| \\
-5.0 &\leq x_i \leq 5.0 \\
\min(f_5(\vec{x})) &= 0 \\
\vec{x}_{opt1} &= (-3.9128, 3.5215) \\
\vec{x}_{opt2} &= (0, 0) \\
\vec{x}_{opt3} &= (3.9128, -3.5215)
\end{aligned} \tag{4.10}$$

(f) 分岐図 2

$$\begin{aligned}
x_{t+1} &= 1 - ax_t^2 + y_t \quad a = 1.4 \\
y_{t+1} &= bx_t \quad b = 0.3 \\
f_6(\vec{x}, y) &= |x_{t+2} - x_t| + |y_{t+2} - y_t| \\
&= |1 - a(1 - ax_t^2 + y_t)^2 + bx_t - x_t| + |b(1 - ax_t^2 + y_t) - y_t| \\
-1.5 &\leq x_i \leq 1.5 \\
\min(f_6(\vec{x})) &= 0 \\
\vec{x}_{opt1} &= (-1.13135, -0.33940) \\
\vec{x}_{opt2} &= (-0.47580, 0.29274) \\
\vec{x}_{opt3} &= (0.63135, 0.18940) \\
\vec{x}_{opt4} &= (0.97580, -0.14274)
\end{aligned} \tag{4.11}$$

(g) 分岐図 3

$$\begin{aligned}
x_{t+1} &= u(x_t \cos \tau - y_t \sin \tau) + 1 \quad \tau_t = 0.4 - \frac{6}{1+x_t^2+y_t^2} \\
y_{t+1} &= u(x_t \sin \tau + y_t \cos \tau) \quad u = 0.9 \\
f_7(\vec{x}, y) &= |x_{t+2} - x_t| + |y_{t+2} - y_t| \\
&= |1 - a(1 - ax_t^2 + y_t)^2 + bx_t - x_t| + |b(1 - ax_t^2 + y_t) - y_t| \\
-2.5 &\leq x_i \leq 4.5 \\
\min(f_7(\vec{x})) &= 0 \\
\vec{x}_{opt1} &= (2.9721, 4.1459) \\
\vec{x}_{opt2} &= (0.5327, 0.2468) \\
\vec{x}_{opt3} &= (1.1142, -2.2856)
\end{aligned} \tag{4.12}$$

図 4.18 ~ 図 4.30 は、それぞれ (a) ~ (g) の成功時の典型的な粒子の動き、(b) ~ (g) の失敗時の典型的な粒子の動きである。表 4.5 ~ 表 4.9 は、試行回数 50 回での (c) ~ (g) の探索できた最適解の個数の内訳である。表 4.10 は、様々な形式におけるベンチマークの探索成功率である。(a), (b) のように最適解が 1 つしかないベンチマークは、IDE の効果が薄い、(c) ~ (g) のような複数最適化問題は従来手法の DE に比べ、IDE の方が全体的に高い成功率である。これは、粒子を鈍感にすることにより更新を遅らせ粒子が広い範囲を探索できるからだと考えられる。また、rand/2 の成功率が高く、best/1 の成功率が低い。これは、粒子が rand/2 は、広い範囲をまんべんなく探索し、best/1 は 1 点に収束をする動きをしているからだと考えられる。best/1 と best/2、rand/1 と rand/2 を比較してみると差分の数が多いほうが高い探索性能を誇っている。ベンチマークは、最適解の数が多いほど探索が難しくなっているが、最適解が 3 つしかない分岐

図3が最適解が4つである分岐図2より成功率が低い。これは、最適解同士の距離が近いことや、ベンチマークの高さが全体的にバラバラだからだと考えられる。

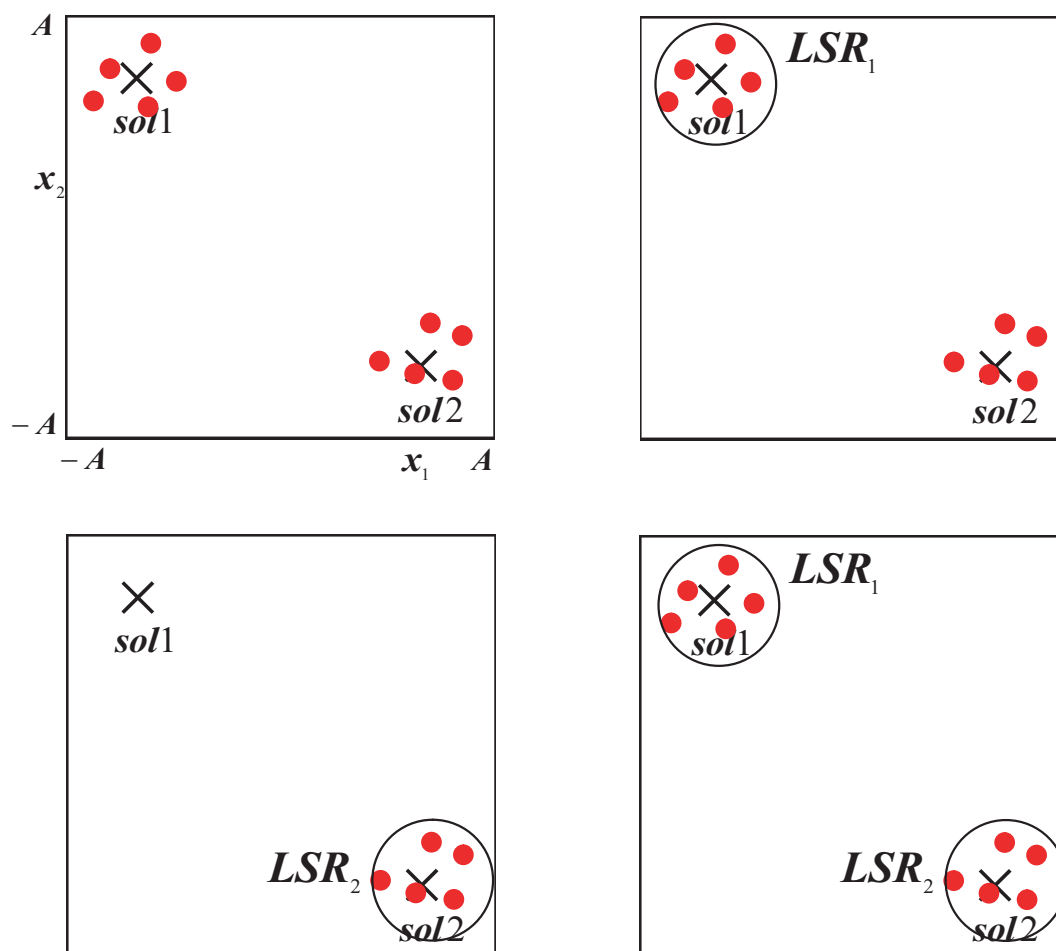


図 4.1: LSR の作成図

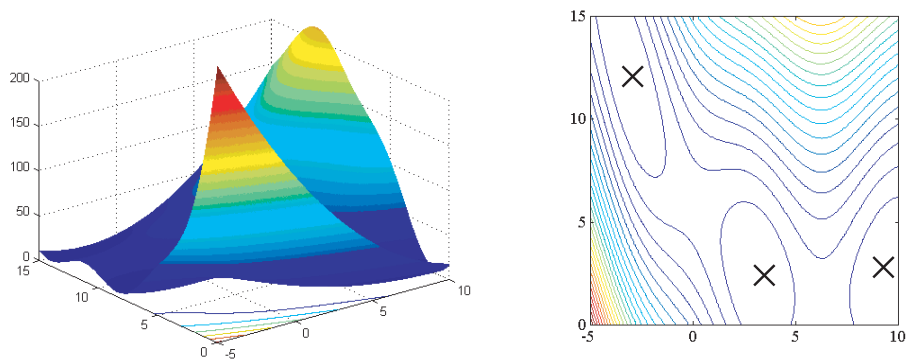


図 4.2: Branin function の形状

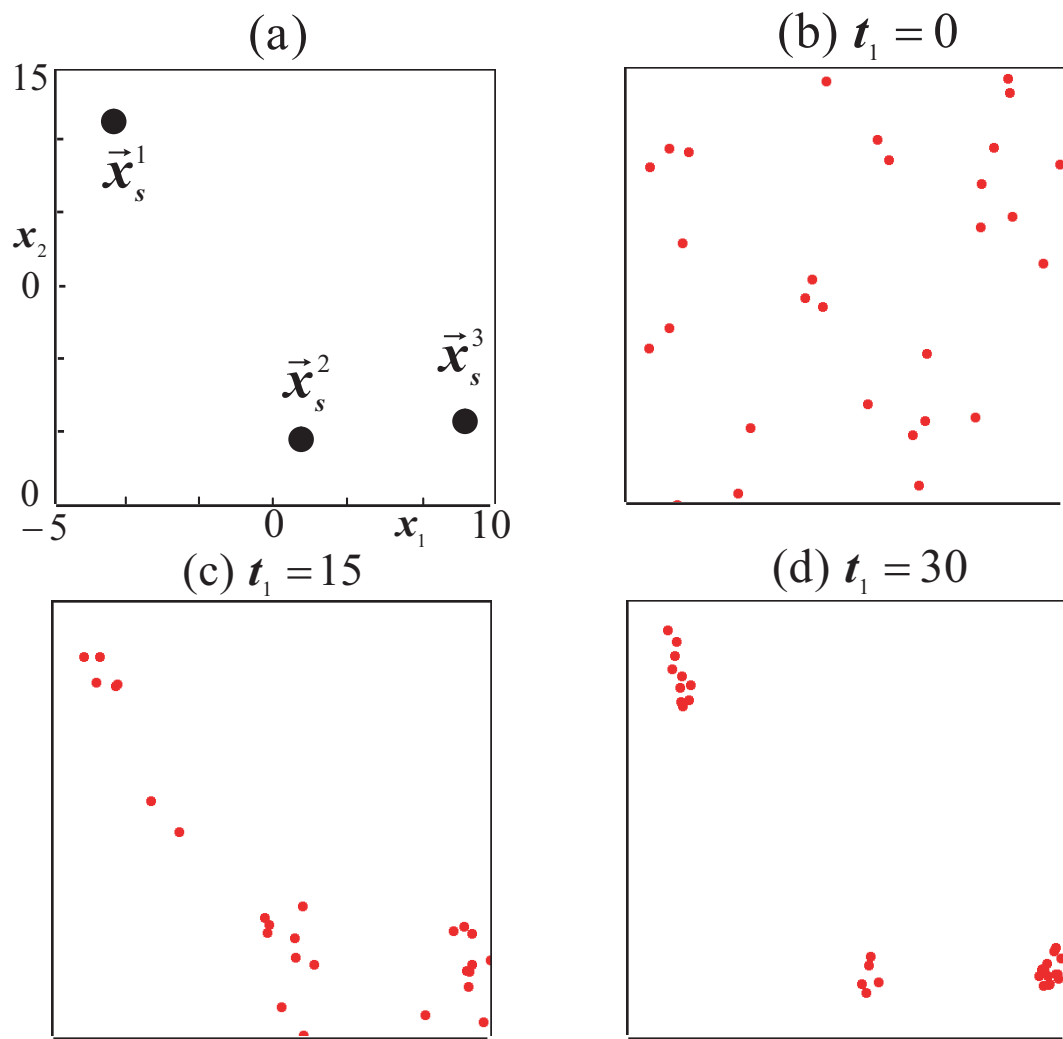


図 4.3: Global search における粒子の動き ($\varepsilon=3.0$, $T_G=30$)

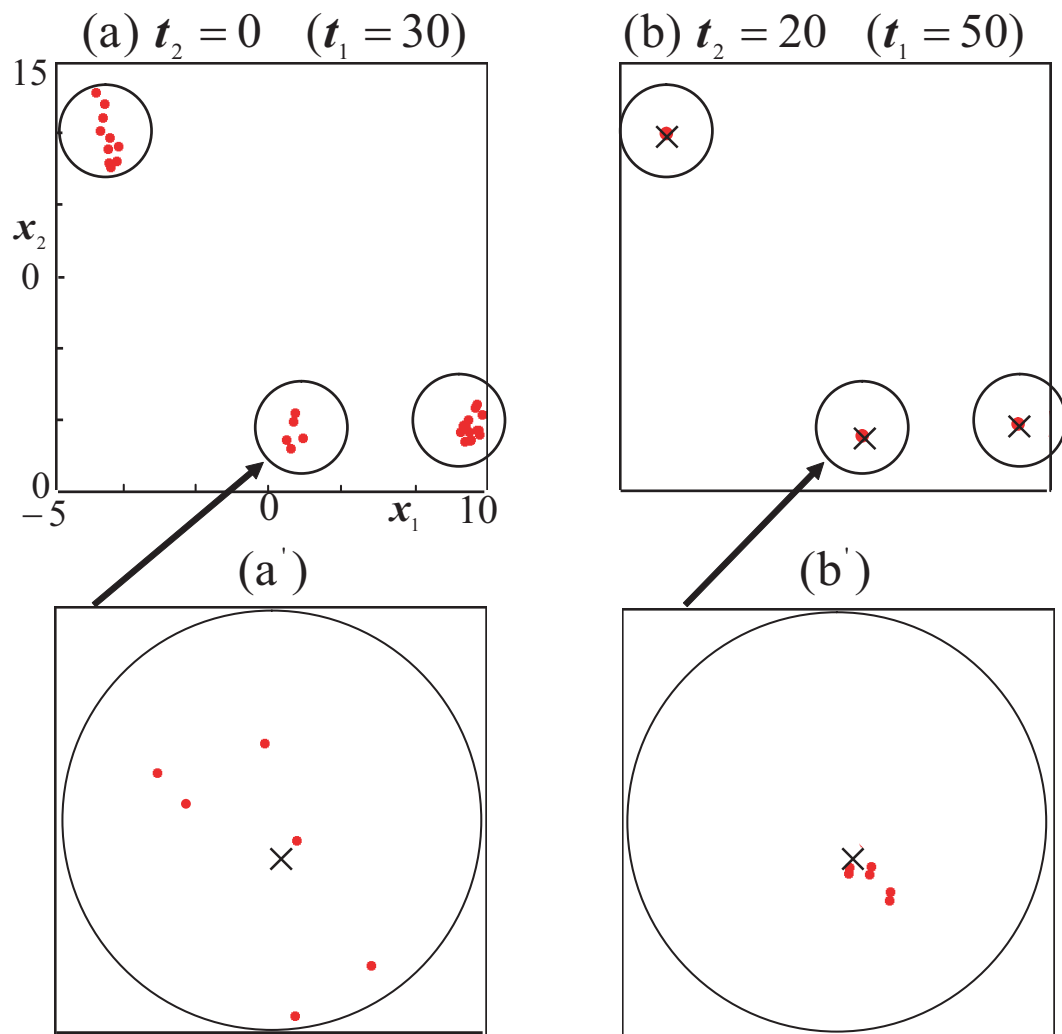


図 4.4: Local search における粒子の動き ($\varepsilon_2=0.01$, $T_L=70$)

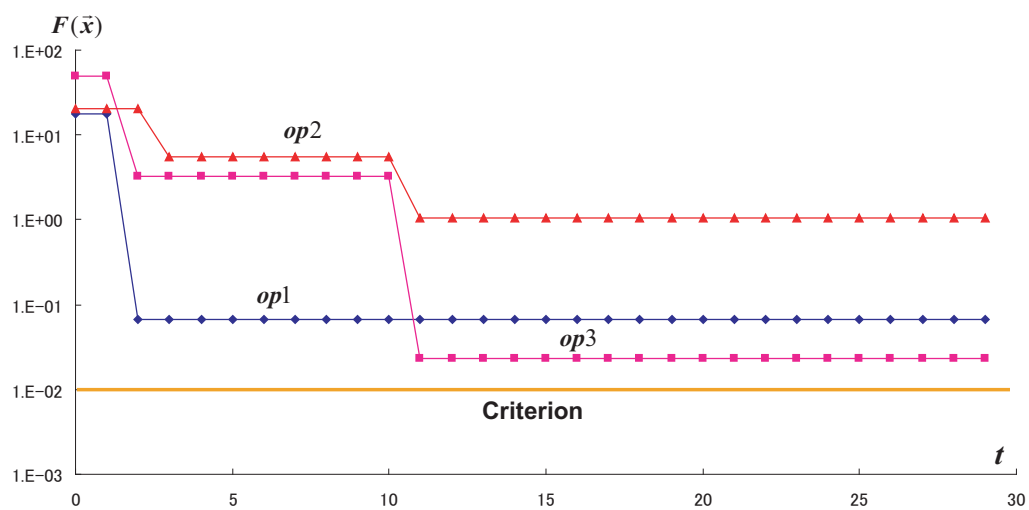


図 4.5: Global search の探索過程

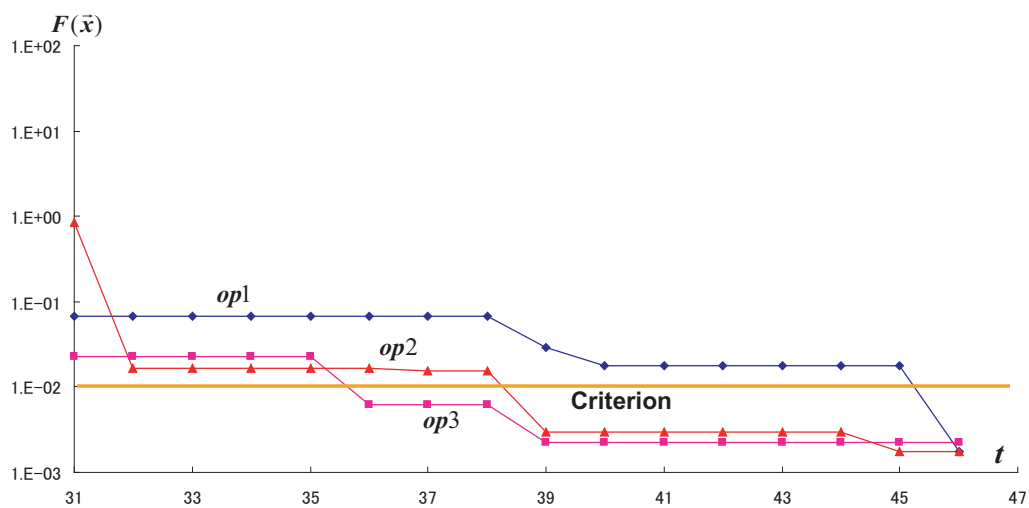


図 4.6: local search の探索過程

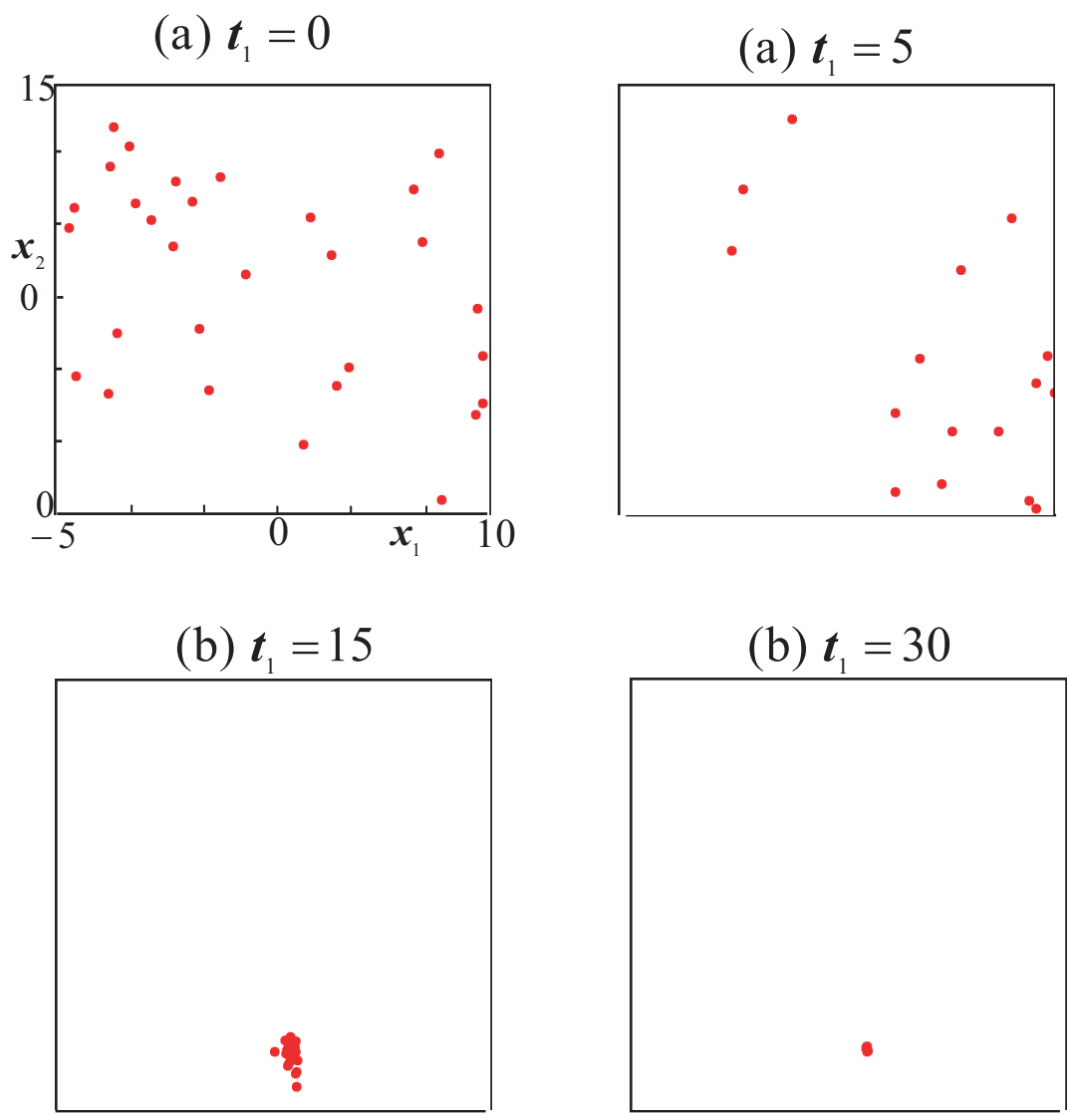


図 4.7: Global search における粒子の動き ($\varepsilon=0, T_G=30$)

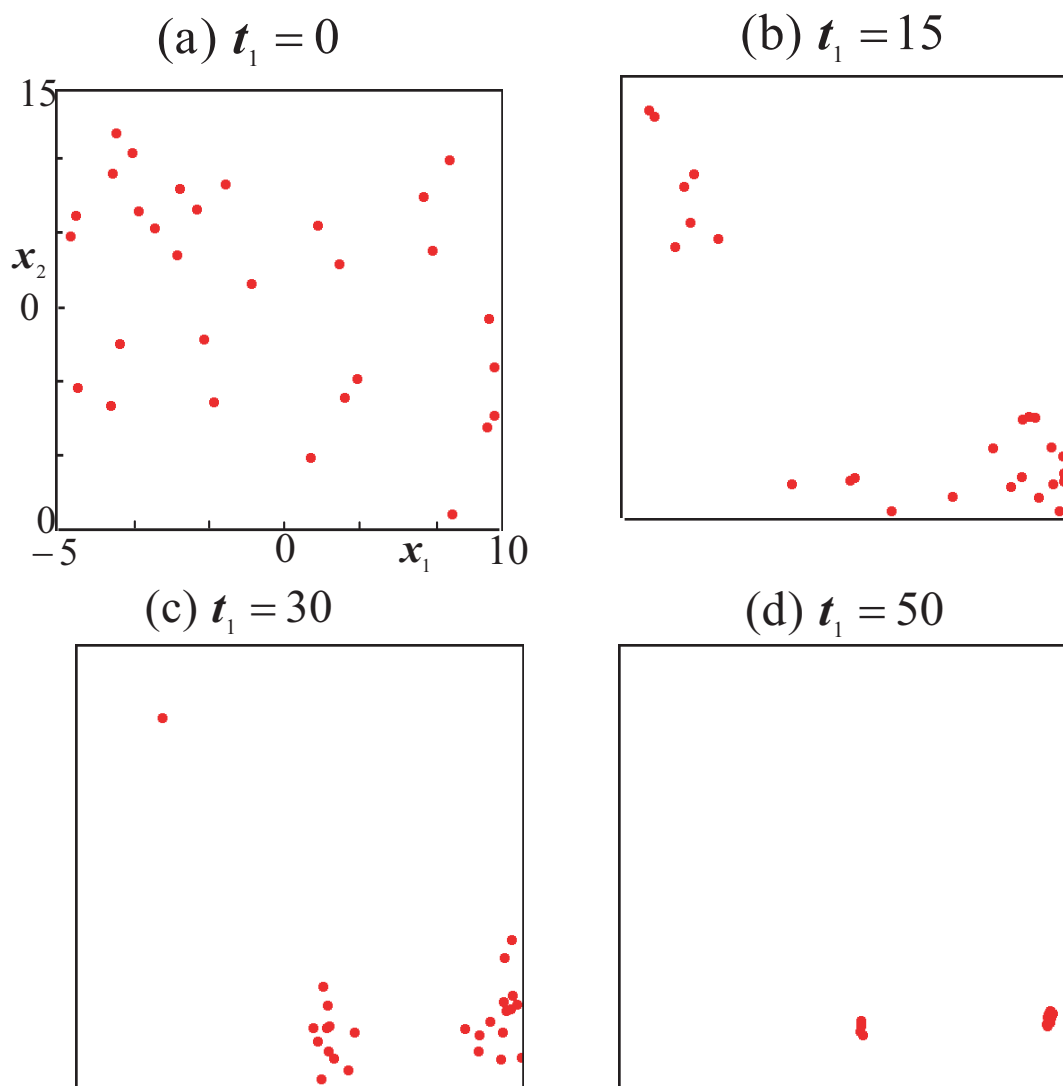


図 4.8: Global search における粒子の動き ($\varepsilon=3.0$, $T_G=50$)

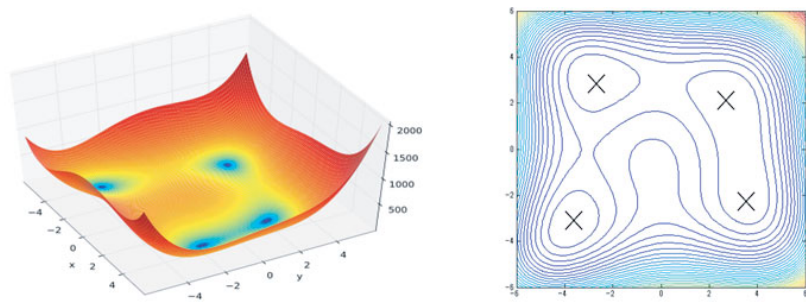


図 4.9: Himmelblau function の形状

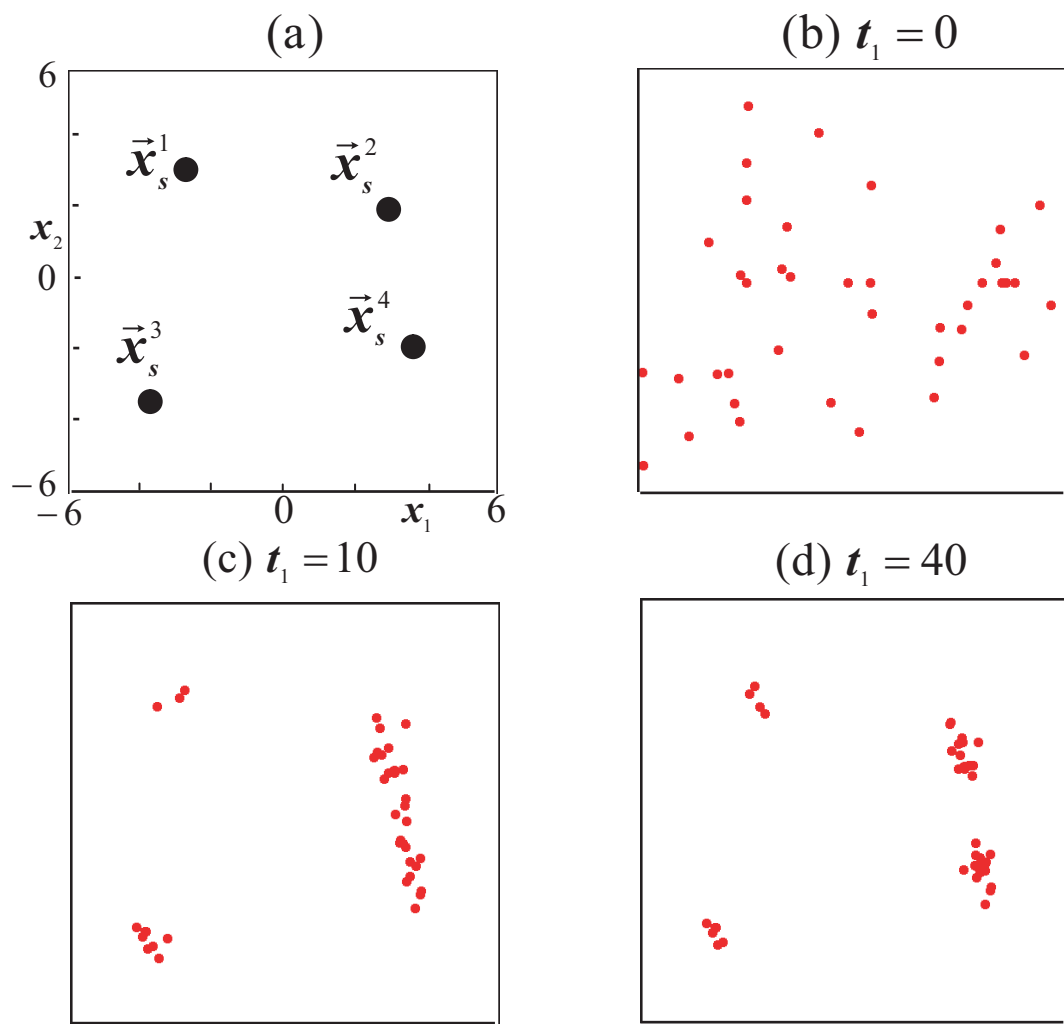


図 4.10: Global search における粒子の動き ($\varepsilon=3.0$, $T_G=40$)

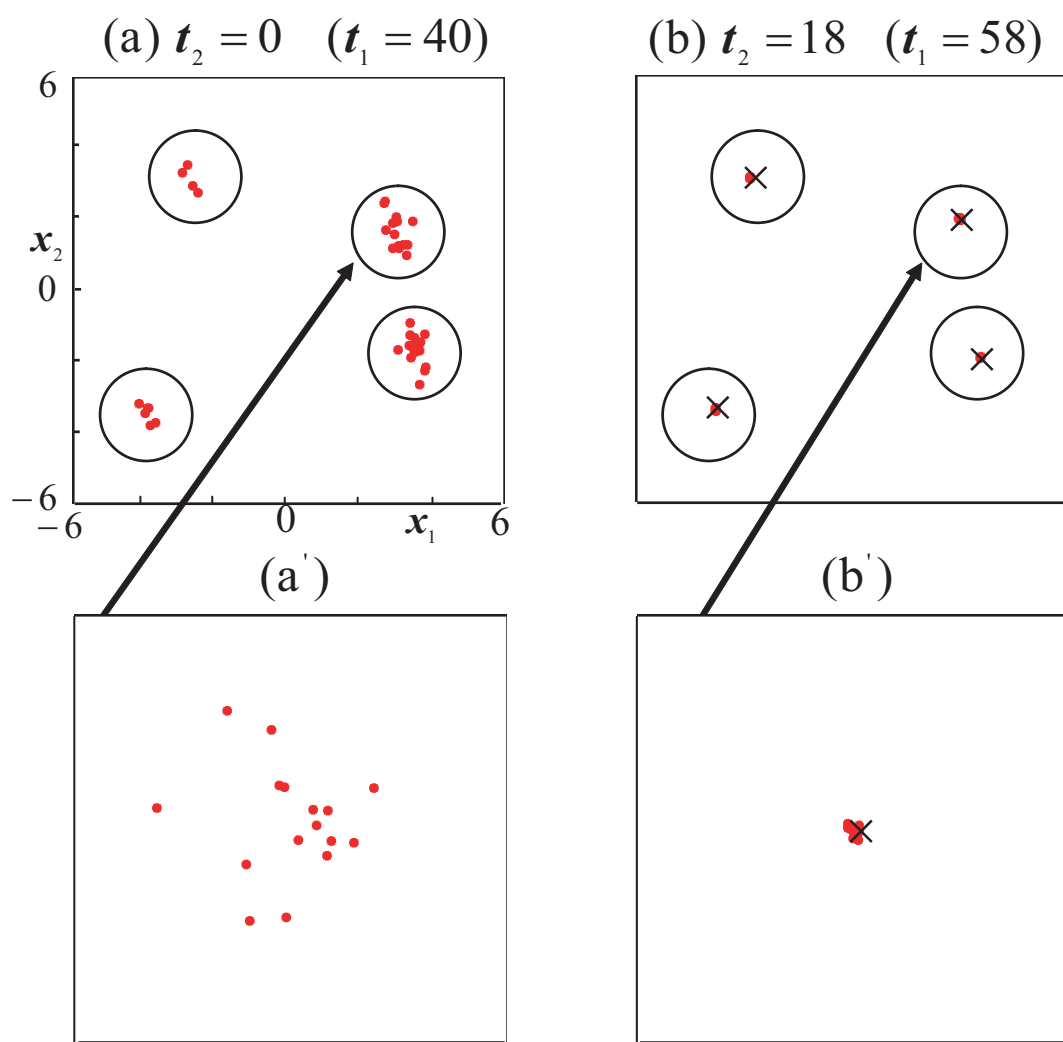


図 4.11: Local search における粒子の動き ($\varepsilon_2=0.01$, $T_L=60$)

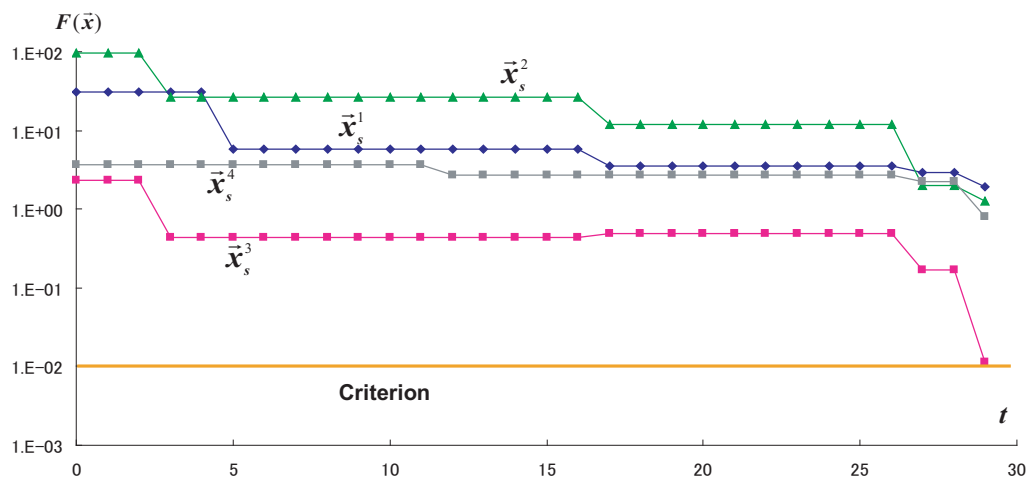


図 4.12: Global search の探索過程

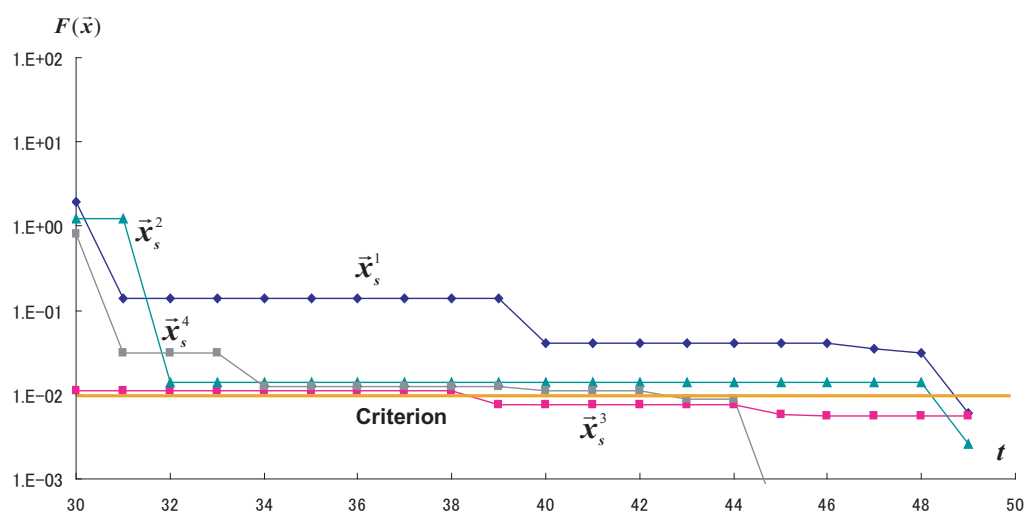


図 4.13: local search の探索過程

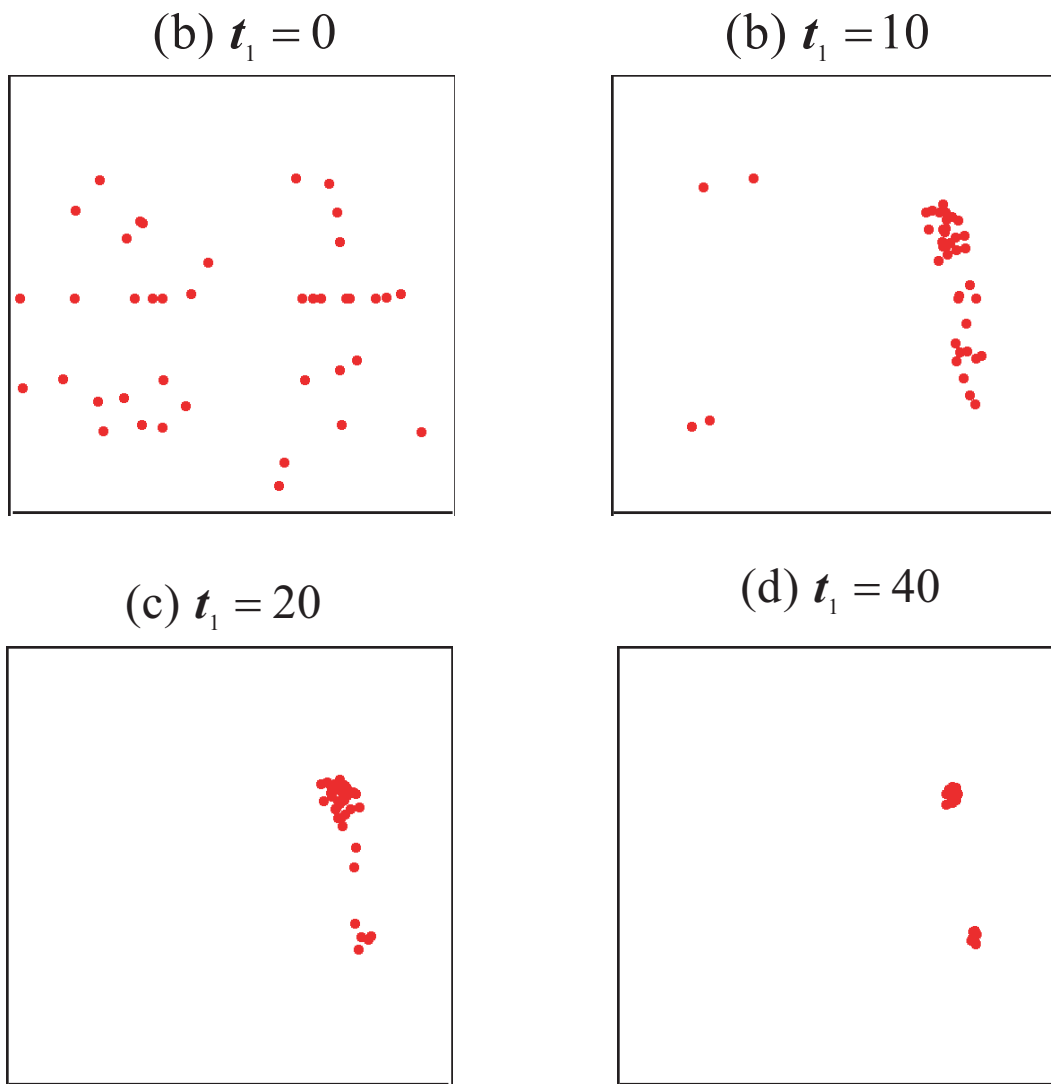
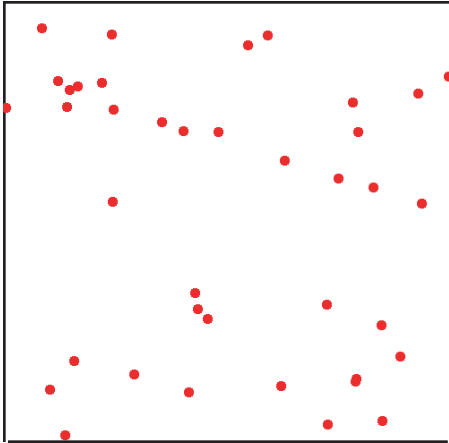
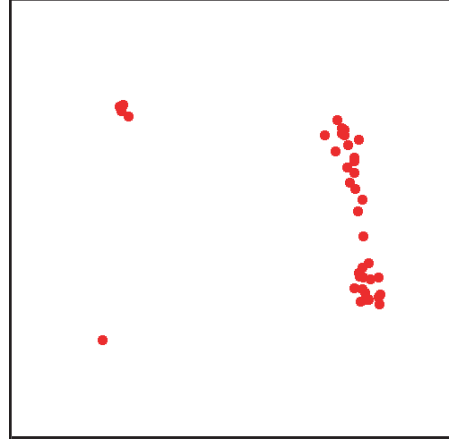


図 4.14: Global search における粒子の動き ($\varepsilon=0$, $T_G=40$)

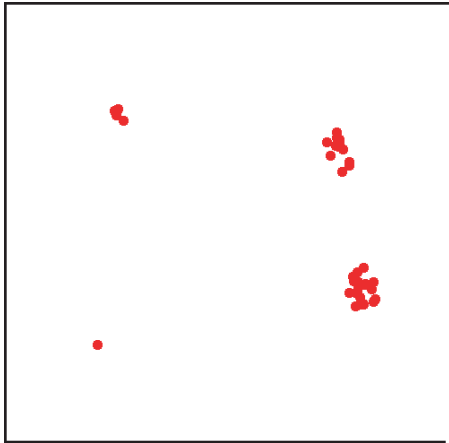
(b) $t_1 = 0$



(b) $t_1 = 20$



(c) $t_1 = 40$



(d) $t_1 = 50$

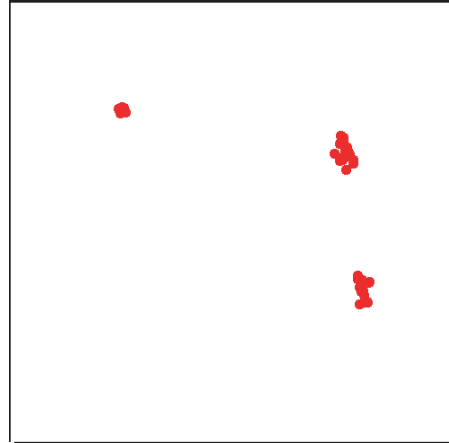
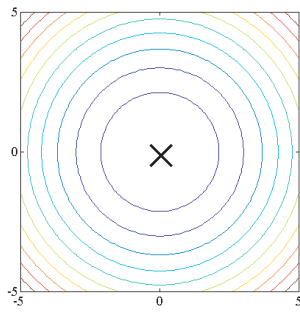
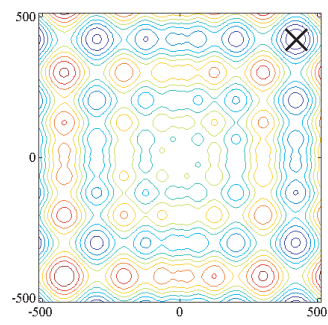


図 4.15: Global search における粒子の動き ($\varepsilon=3.0$, $T_G=50$)

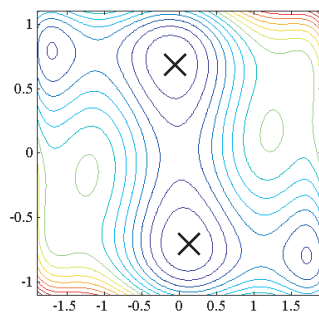
(a) Sphere function



(b) Normalized Schwefel function



(c) Six-hump Camel Back function



(d) Branin function

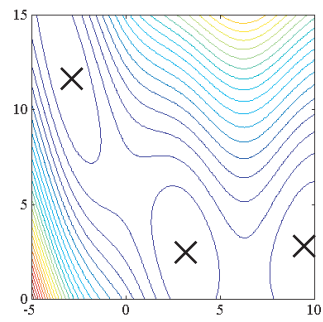
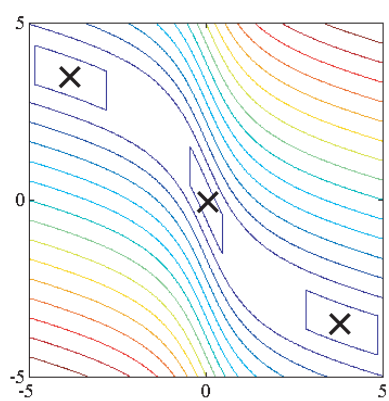
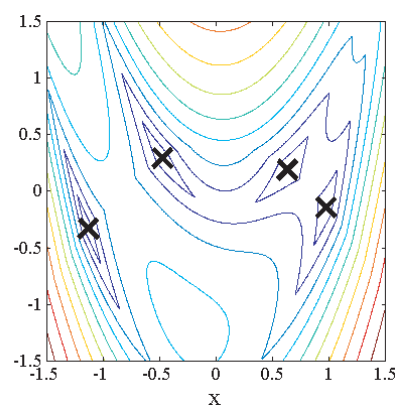


図 4.16: (a) ~ (d) の最適解の位置

(e) 分岐図1



(f) 分岐図2



(g) 分岐図3

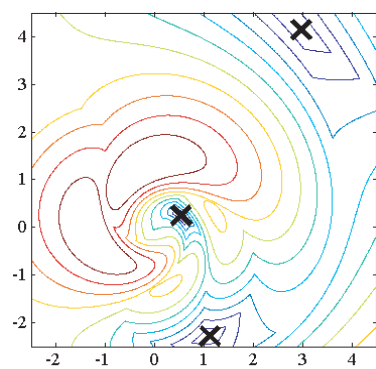


図 4.17: (e) ~ (g) の最適解の位置

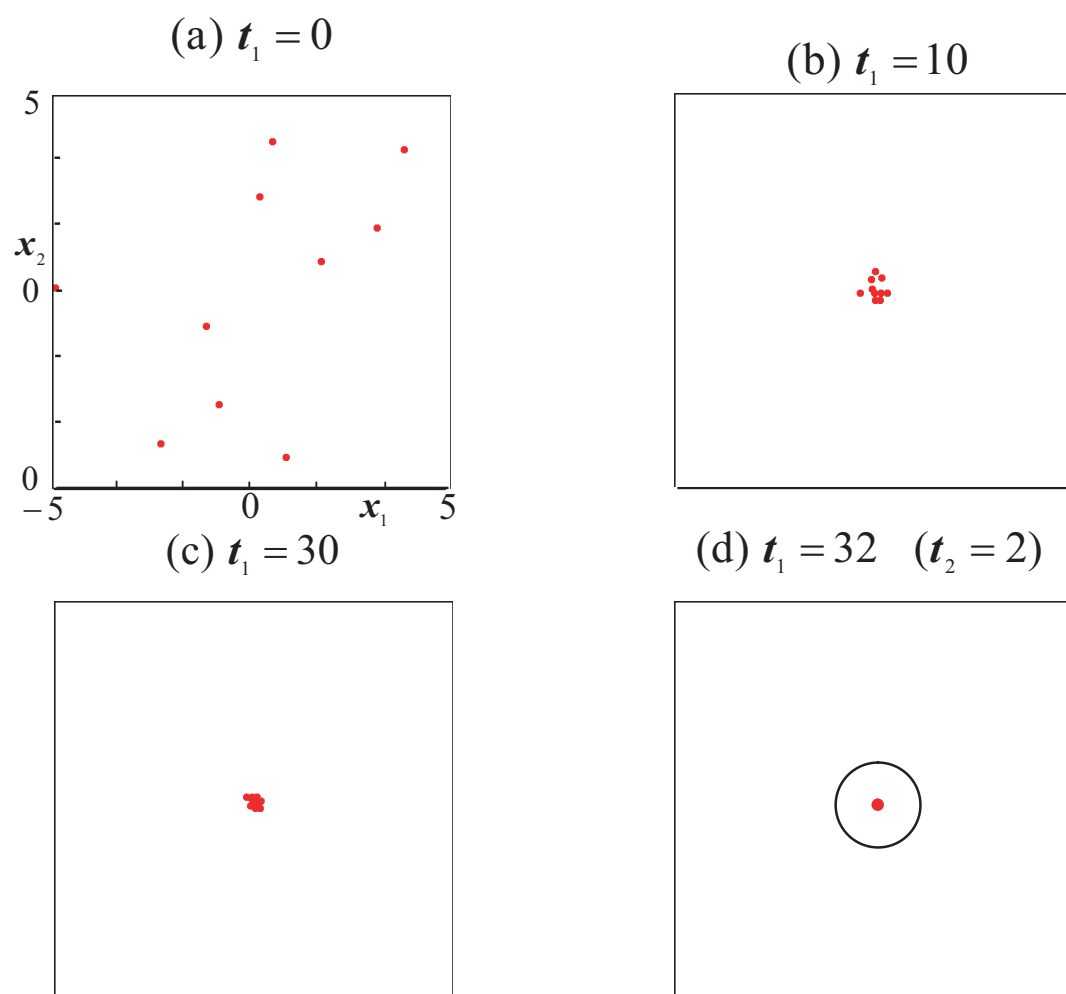


图 4.18: (a) 探索成功例 (DE/rand/1, $\varepsilon=0.3$)

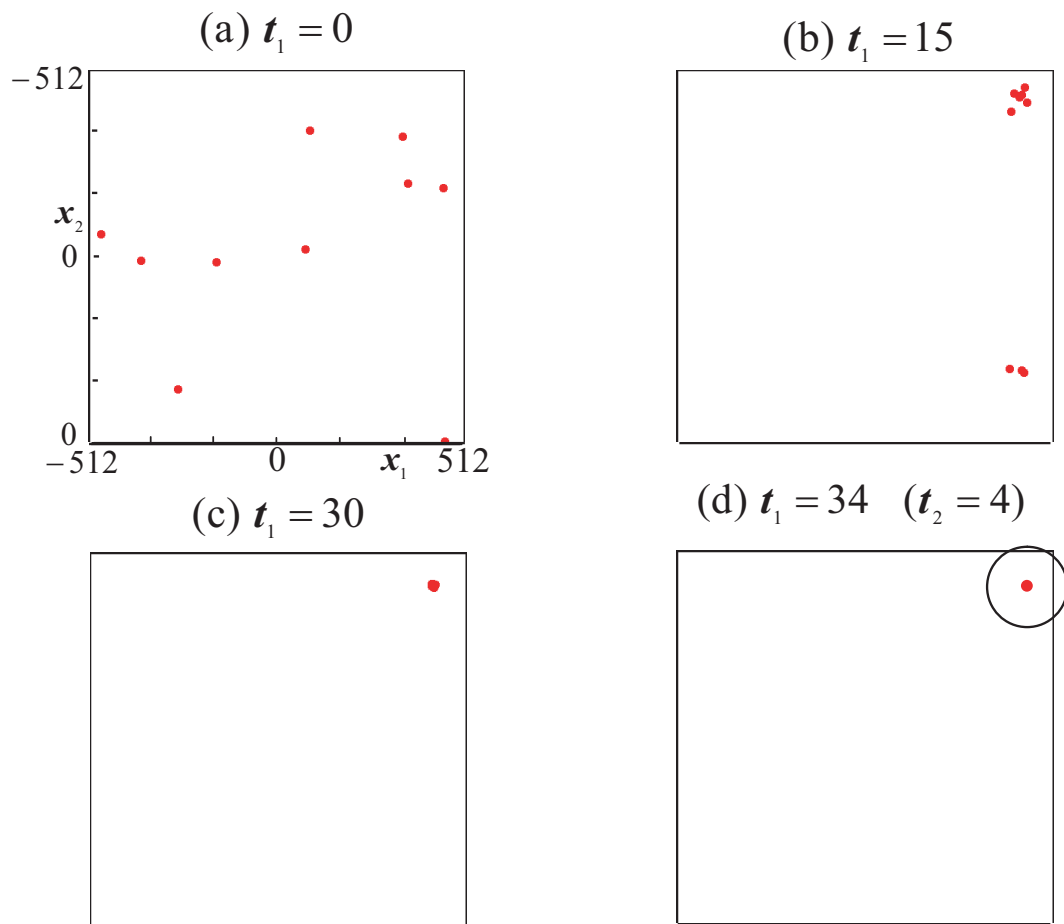


図 4.19: (b) の探索成功例 (DE/rand/1, $\varepsilon=0.3$)

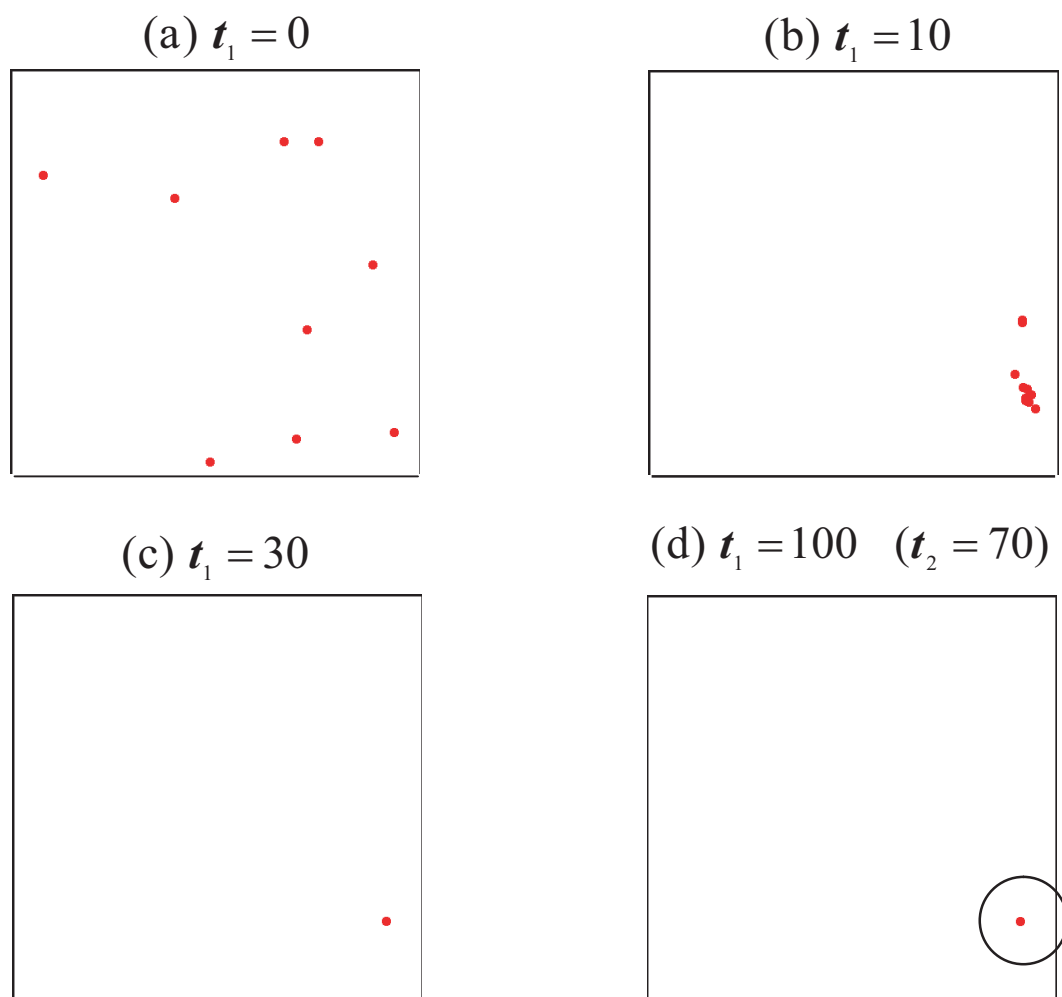


図 4.20: (b) の探索失敗例 (DE/rand/2, $\varepsilon=0.3$)

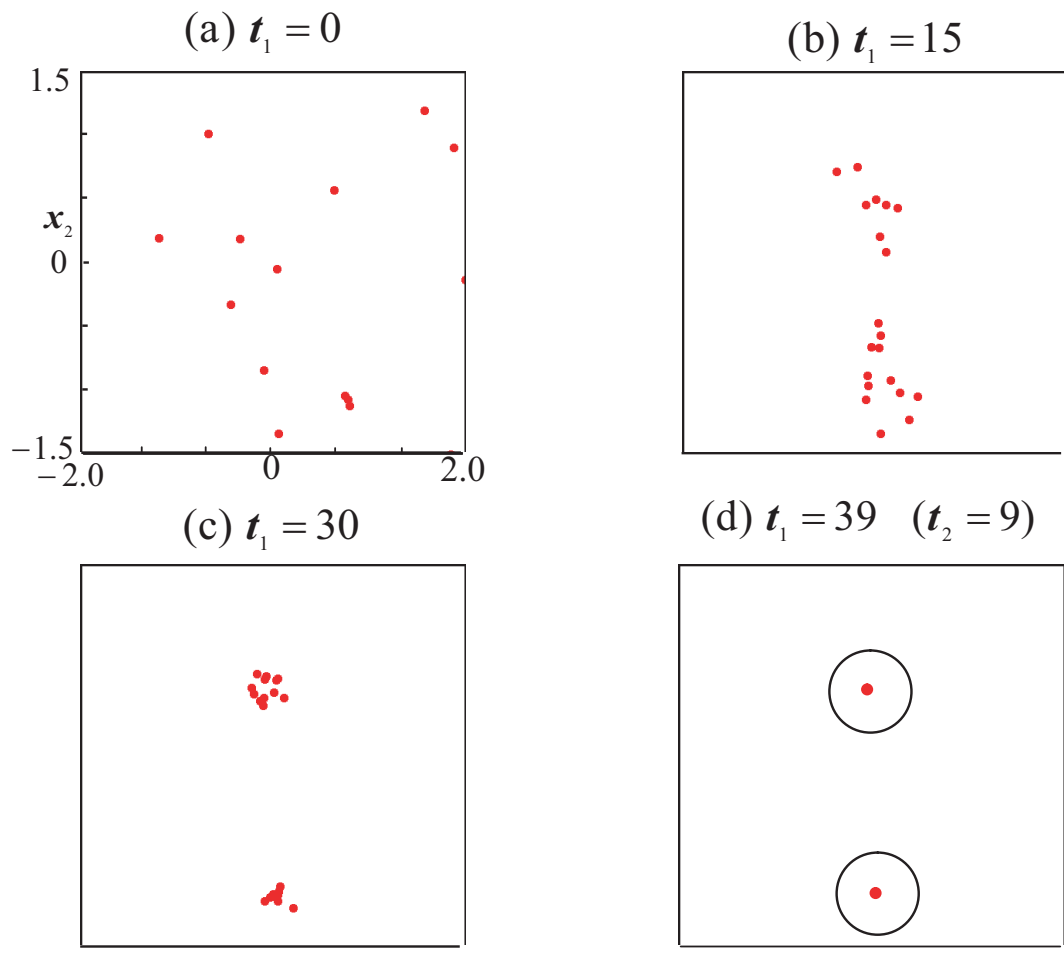


図 4.21: (c の) 探索成功例 (DE/rand/2, $\varepsilon=3.0$)

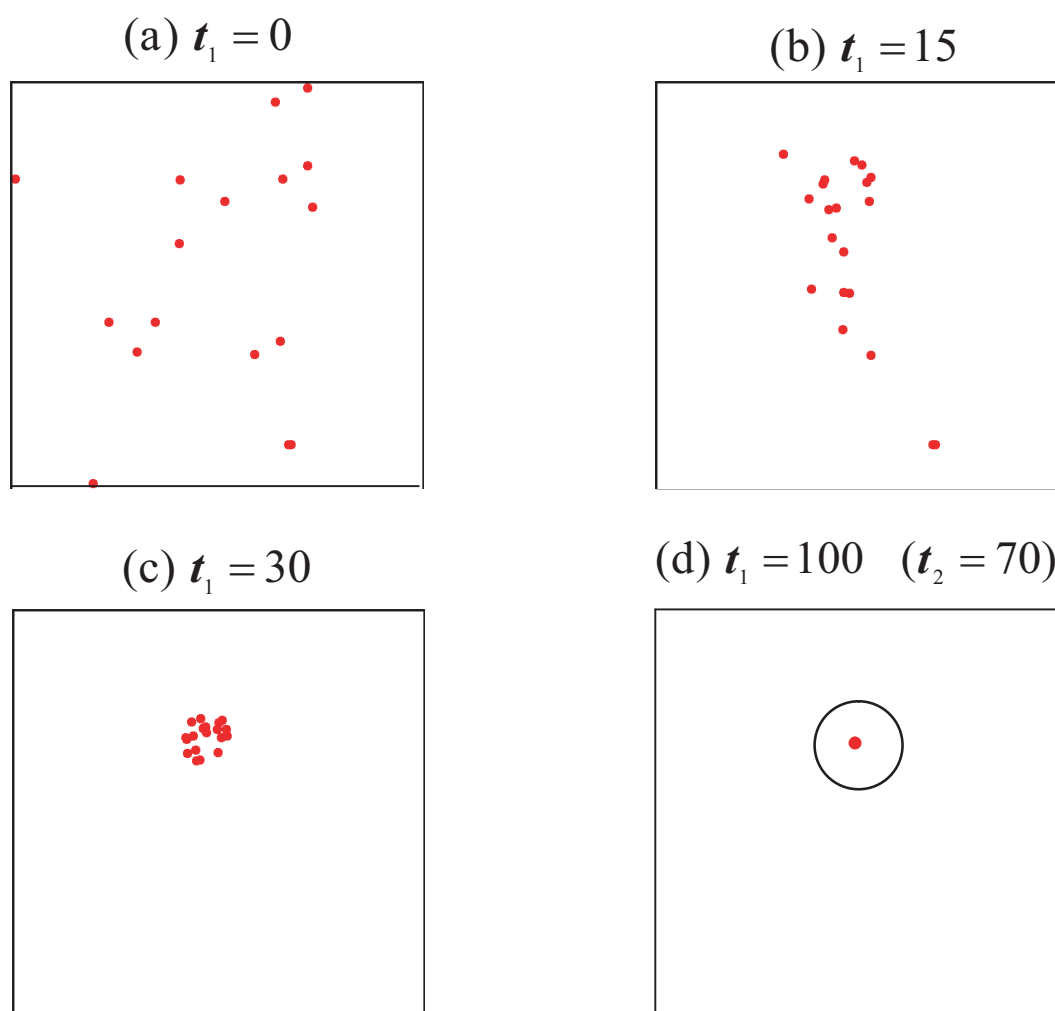


図 4.22: (c) の探索失敗例 ($DE/best/2$, $\varepsilon=3.0$)

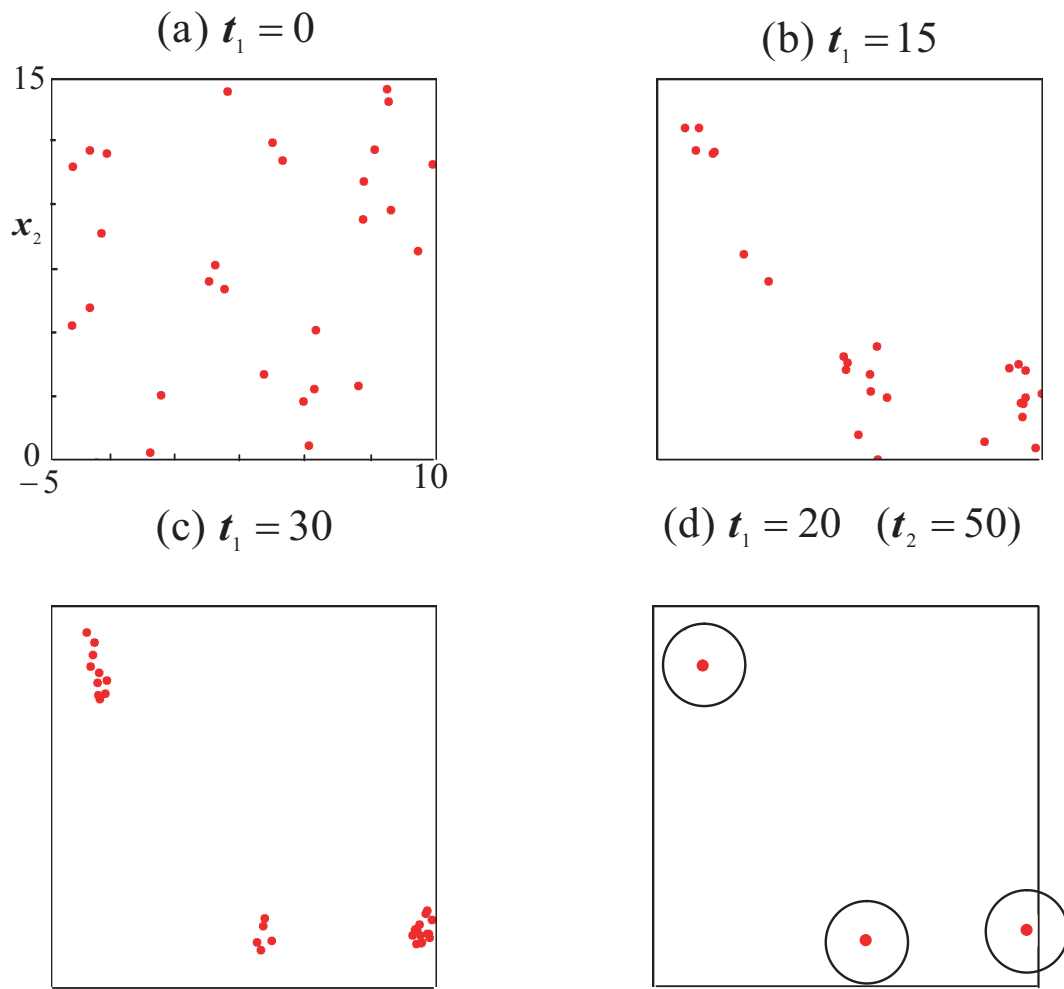


図 4.23: (d) の探索成功例 (DE/rand/2, $\varepsilon=3.0$)

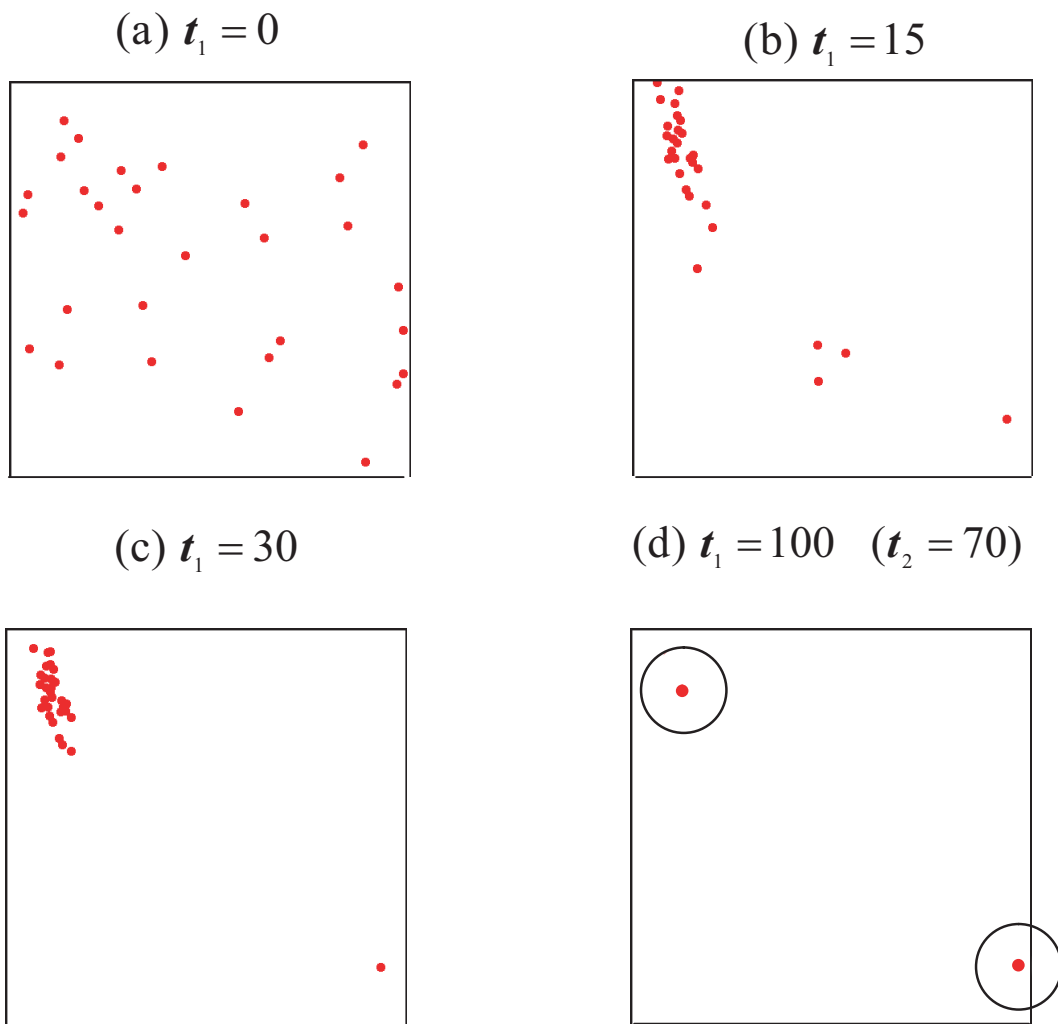


図 4.24: (d) の探索失敗例 (DE/best/2, $\varepsilon=3.0$)

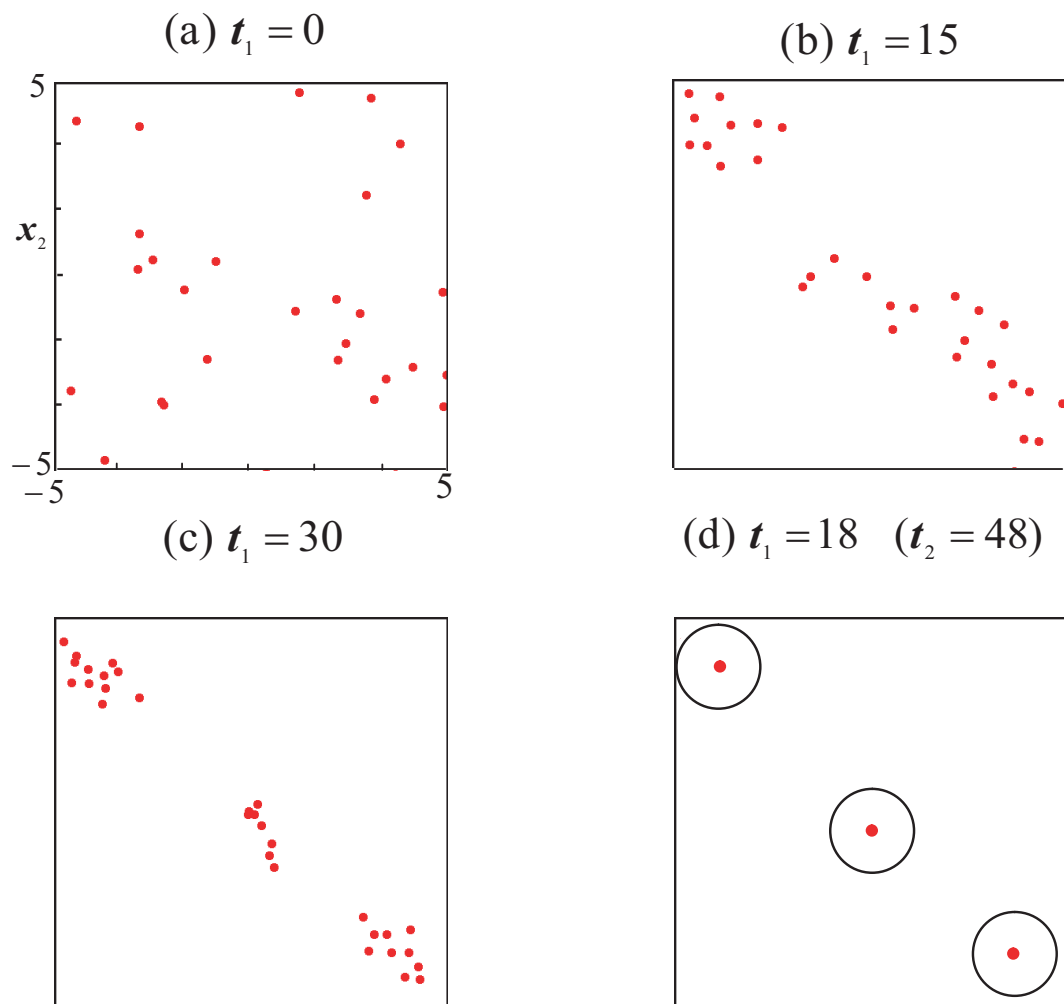


図 4.25: (e) の探索成功例 (DE/current-to-best, $\varepsilon=0.3$)

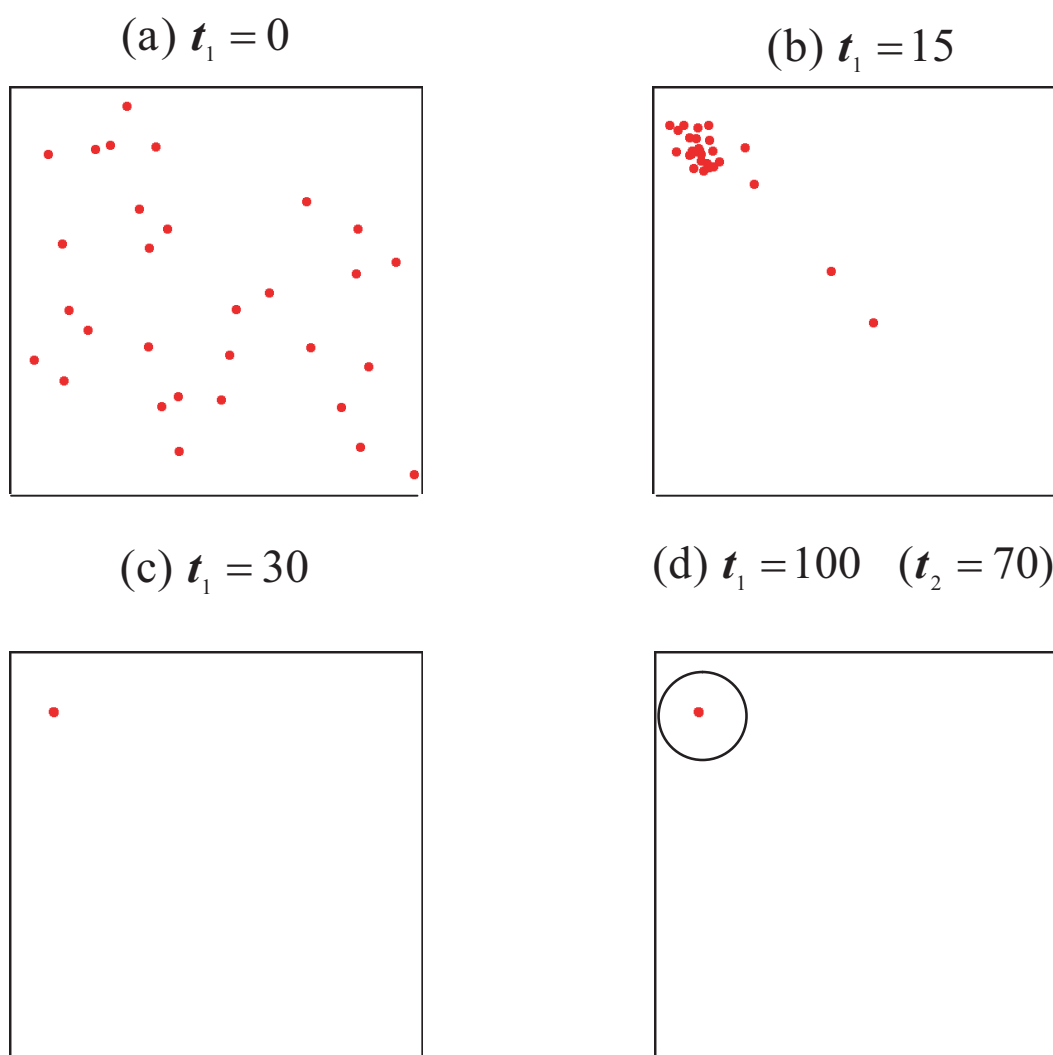


図 4.26: (e) の探索失敗例 (DE/best/1, $\varepsilon=0.3$)

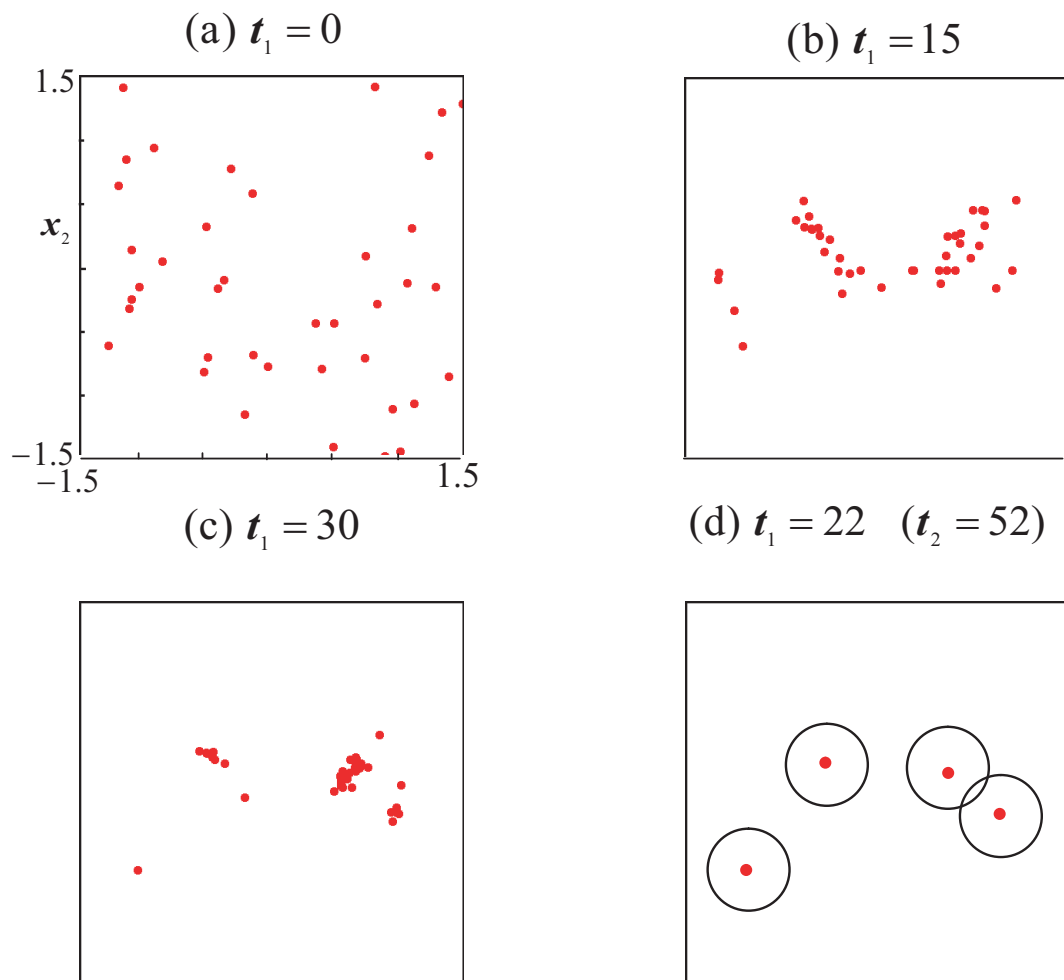


図 4.27: (f) の探索成功例 (DE/rand/1, $\varepsilon=0.3$)

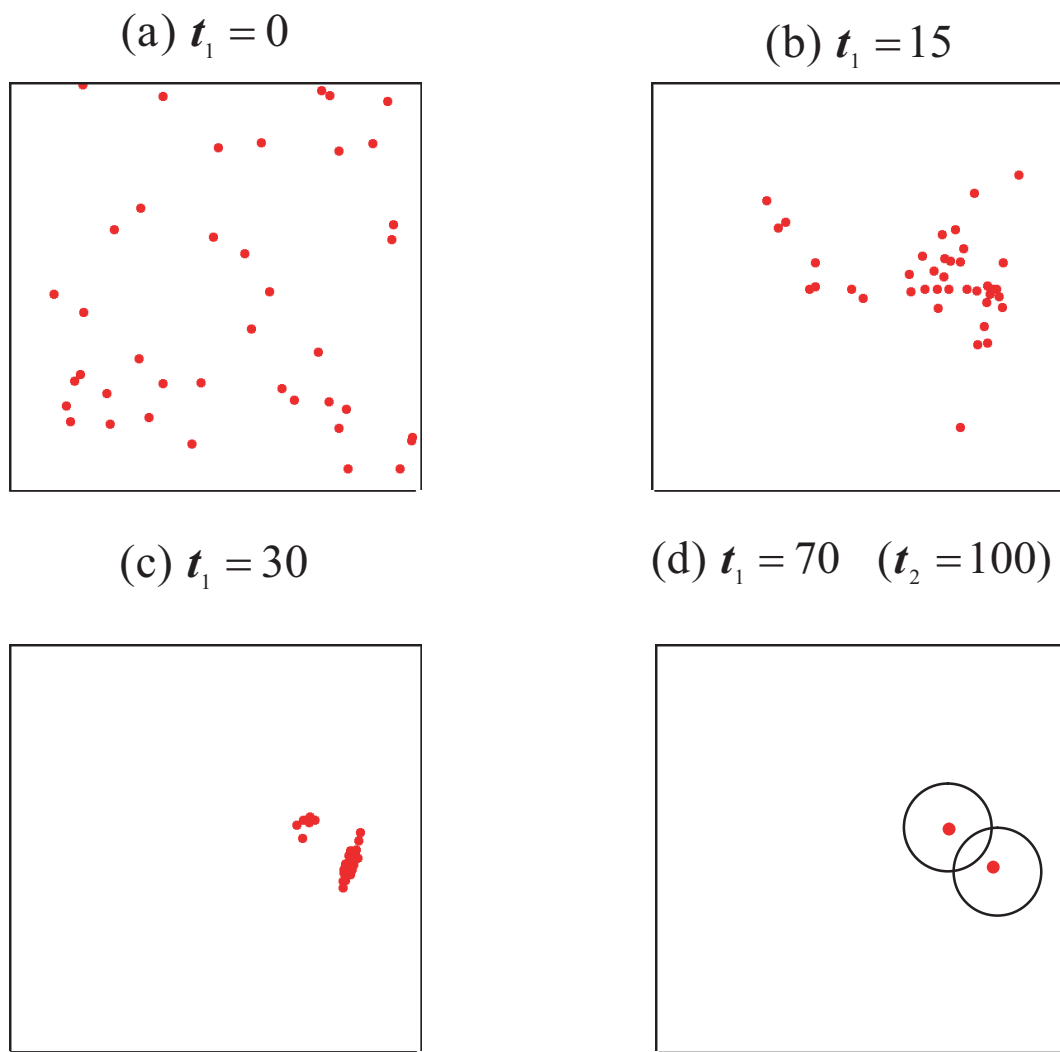


図 4.28: (f) の探索失敗例 ($DE/best/2$, $\varepsilon=0.3$)

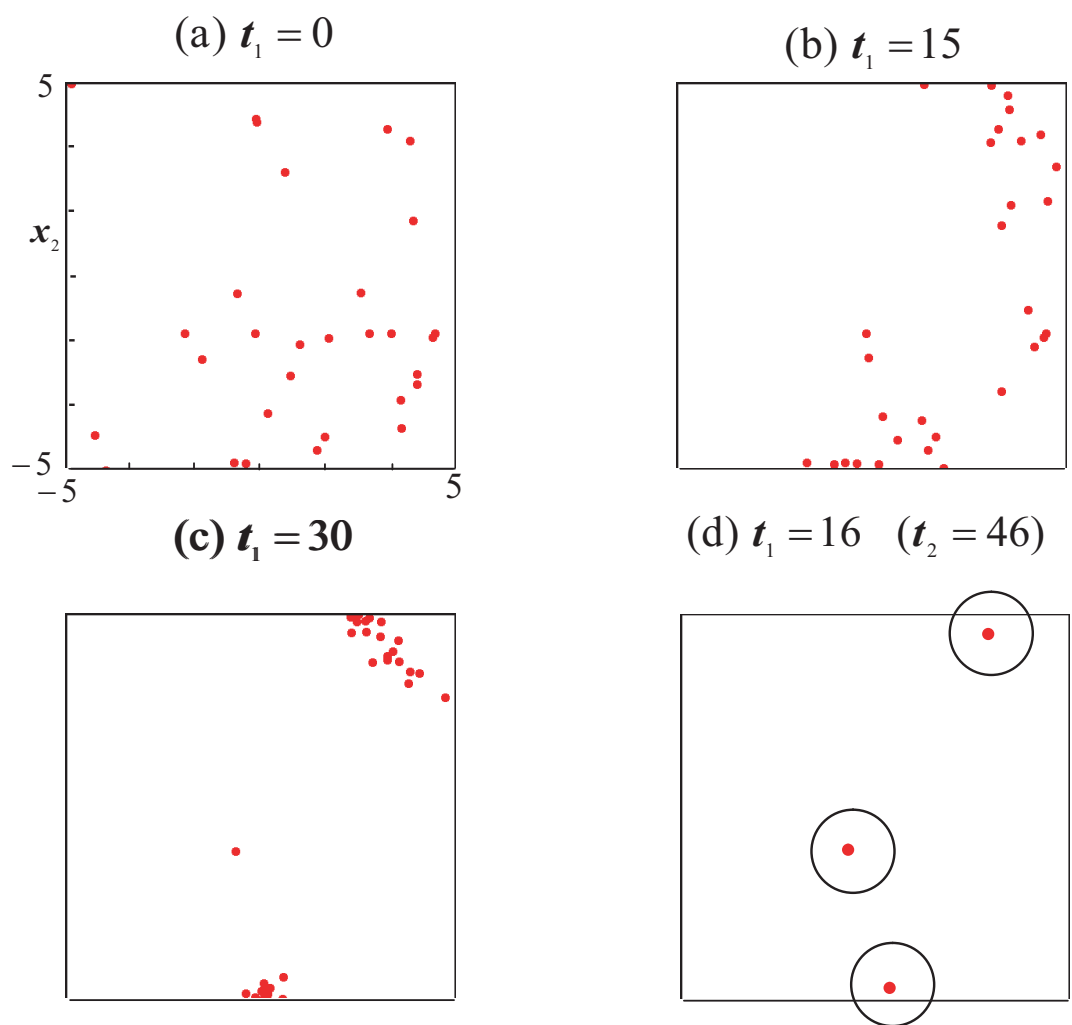


図 4.29: (g) の探索成功例 (DE/rand/2, $\varepsilon=0.3$)

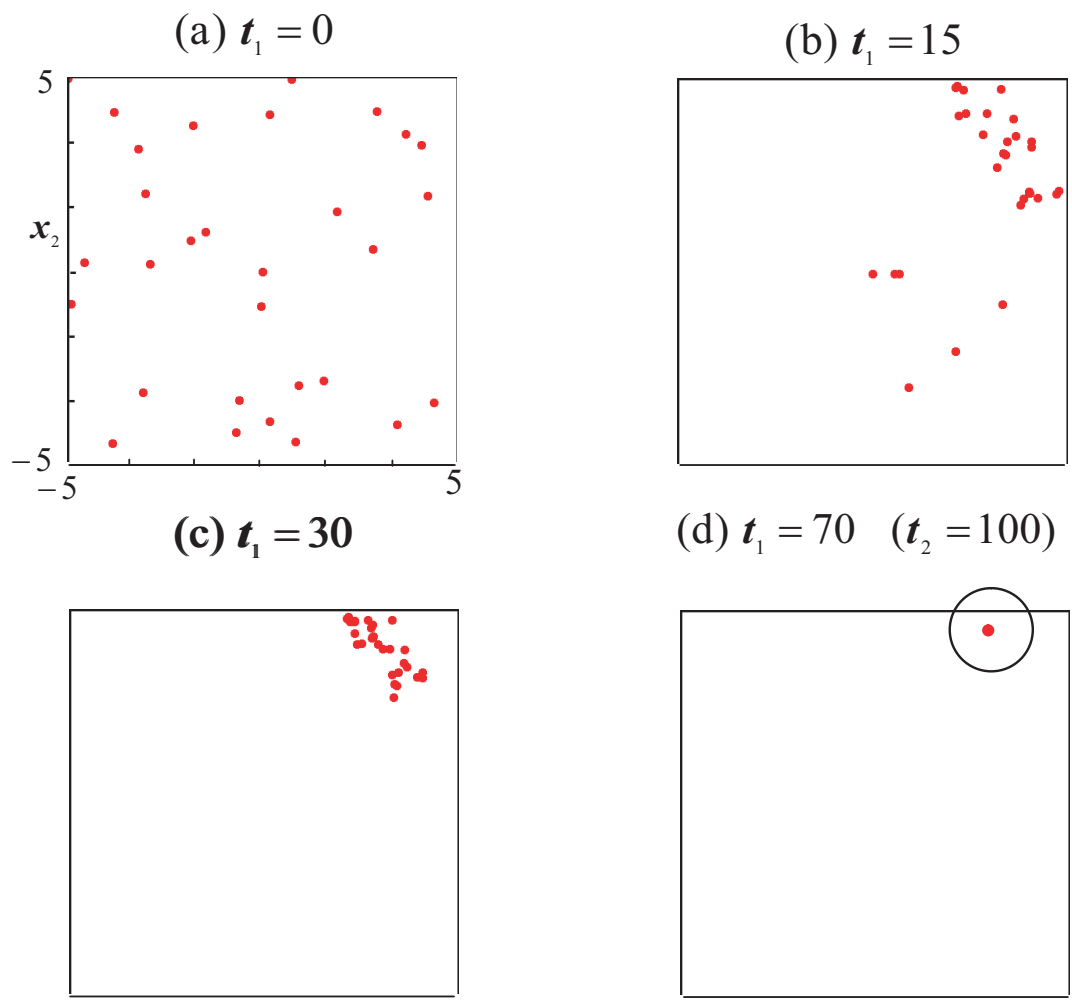


図 4.30: (g) の探索失敗例 ($DE/best/2$, $\varepsilon=0.3$)

表 4.1: Branin function における Global search の成功率

$T_G \backslash \varepsilon$	0	1	2	3	4	5
10	36	72	80	80	72	68
20	36	68	68	84	84	72
30	44	74	84	88	80	76
40	26	64	70	80	88	82
50	4	68	72	78	76	80

表 4.2: Branin function における Local search の成功率 ($\varepsilon=3, T_G=30$) SR =成功率

ε_2	0.01	0.02	0.1	0.5	1.0	2.0	3.0
SR	100	100	60	10	1	0	0

表 4.3: Himmelblau function における Global search の成功率

$T_G \backslash \varepsilon$	0	1	2	3	4	5
10	0	10	16	10	4	2
20	0	22	52	56	40	36
30	8	28	44	60	64	58
40	12	24	24	40	42	46
50	4	24	20	38	48	50

表 4.4: Himmelblau function における Local search の成功率 (%) ($\varepsilon=3, T_G = 40$) SR =成功率

ε_2	0.01	0.02	0.1	0.5	1.0	2.0	3.0
SR	100	100	56	0	0	0	0

表 4.5: (c) における探索した最適解の個数の内訳 (%) ($\varepsilon=3$, $T_G = 30$)

	最適解 0 個	最適解 1 個	最適解 2 個
best/1(DE)	0	100	0
rand/1(DE)	0	36	64
best/1(IDE)	0	74	26
best/2(IDE)	0	40	60
rand/1(IDE)	0	4	96
rand/2(IDE)	0	0	100
current-to-best(IDE)	0	28	72

表 4.6: (d) における探索した最適解の個数の内訳 (%) ($\varepsilon=3, T_G = 30$)

	最適解 0 個	最適解 1 個	最適解 2 個	最適解 3 個
best/1(DE)	0	100	0	0
rand/1(DE)	0	12	44	0
best/1(IDE)	0	30	30	40
best/2(IDE)	0	4	48	48
rand/1(IDE)	0	0	12	88
rand/2(IDE)	0	0	0	100
current-to-best(IDE)	0	0	44	56

表 4.7: (e) における探索した最適解の個数の内訳 (%) ($\varepsilon=0.3$, $T_G = 30$)

	最適解 0 個	最適解 1 個	最適解 2 個	最適解 3 個
best/1(DE)	0	100	0	0
rand/1(DE)	0	20	20	60
best/1(IDE)	0	16	48	36
best/2(IDE)	0	4	38	58
rand/1(IDE)	0	0	10	90
rand/2(IDE)	0	0	0	100
current-to-best(IDE)	0	0	20	80

表 4.8: (f) における探索した最適解の個数の内訳 (%) ($\varepsilon=0.3$, $T_G = 30$)

	最適解 0 個	最適解 1 個	最適解 2 個	最適解 3 個	最適解 4 個
best/1(DE)	0	82	28	0	0
rand/1(DE)	0	0	20	64	16
best/1(IDE)	0	4	20	56	20
best/2(IDE)	0	0	0	58	42
rand/1(IDE)	0	0	6	38	56
rand/2(IDE)	0	0	0	24	76
current-to-best(IDE)	0	0	8	64	28

表 4.9: (g) における探索した最適解の個数の内訳 (%) ($\varepsilon=0.3$, $T_G = 30$)

	最適解 0 個	最適解 1 個	最適解 2 個	最適解 3 個
best/1(DE)	0	100	0	0
rand/1(DE)	0	28	64	8
best/1(IDE)	0	88	12	0
best/2(IDE)	0	24	60	16
rand/1(IDE)	0	12	36	52
rand/2(IDE)	0	0	36	64
current-to-best(IDE)	0	8	68	24

表 4.10: 様々な形式におけるベンチマークの探索成功率 (%)

	DE		IDE				
	best/1	rand/1	best/1	best/2	rand/1	rand/2	current-to-best
(a)	100	100	100	100	100	100	100
(b)	48	84	52	64	82	100	28
(c)	0	64	26	60	80	100	72
(d)	0	44	40	48	88	100	56
(e)	0	60	32	58	76	100	80
(f)	0	16	20	42	76	76	28
(g)	0	8	0	78	16	64	24

第5章 むすび

本修士論文では差分進化に成長構造を持たせたアルゴリズムと粒子を鈍感にした差分進化のアルゴリズムについて考察した。成長構造を持たせた差分進化では、局所解に陥ったときに、新しい粒子を追加することにより脱出が可能であることがわかった。また、粒子の追加する数によって探索性能がどのように変わるかを考察した。

今後の課題として、以下に示すような課題が挙げられる。:

- 1) 粒子の増やし方の工夫、
- 2) 異なる形式での実験、
- 3) 実用的な問題への適用、
- 4) より複雑なベンチマーク問題での実験。

粒子を鈍感にした差分進化では、Global search と Local search の二つの探索により複数最適解探索問題に適用が可能であった。異なるパラメータで実験を繰り返し、ベンチマークごとの最適なパラメータを考察した。また、様々な形式により実験を行い、複数最適解探索により効果的な形式を考察した。今後の課題として、以下に示すような課題が挙げられる、

- 1) 問題ごとによる、最適な ε と T_G の設定、
- 2) 実用的な問題への適用。
- 3) より複雑なベンチマーク問題での実験。
- 4) アルゴリズムの更なる改良。

参考文献

- [1] R. Storn and K. Price: "Minimizing the real functions of the ICEC'96 contest by differential evolution", Proc. of the International Conference on Evolutionary Computation, pp. 842-844(1996)
- [2] R.Storn and K.Price: "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization , 11, pp. 341-359(1997)
- [3] T.Takahama, S.Sakai and N.Iwane: "Solving nonlinear constrained optimization problems by the ε constrained differential evolution", Proc. of the 2006 IEEE Conference on System, Man, and Cybernetics, pp2322-2327(2006)
- [4] T.Takahama and S.Sakai: "Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites", Proc. of the 2006 World Congress on Computational Intelligence, pp308-315(2006)
- [5] 伊藤 稔, 田中雅博 "関数値最適化のための Particle Swarm Optimization, Differential Evolution, 実数値遺伝的アルゴリズムの探索性能に関する検討 "(2005)
- [6] R. Storn and K. Price: " Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces , Technical report , International Computer Science Institute , 11 , Berkley , (1995)
- [7] T. Takahama and S. Sakai, " Constrained optimization by combining the α constrained method with particle swarm optimization " , Proc. of Joint 2nd Interna-

tional Conference on Soft Computer and Intelligent Systems and 5th International Symposium on Advanced Intelligent Systems(2004)

- [8] John Henry Holland, "The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance", in: Francisco J. Varela, Paul Bourguine, editors. Toward a Practice of Autonomous Systems: proceedings of the first European conference on Artificial Life. MIT Press. 1992
- [9] J. Kennedy and R. Eberhart, "Particle Swarm Optimization" , Proc of IEEE/ICNN, pp.1942-1948, 1995
- [10] H. Lampinen and O. Vainio: " An optimization approach to designing OTAs for low-voltage sigma-delta modulators ", Proc. of WCCI, pp. 1665-1671 (2001)
- [11] Luitel, B. and Venayagamoorthy, G.K.: " Differential evolution particle swarm optimization for digital filter design ", Proc. of the 2008 IEEE, pp. 3954-3961 (2008)
- [12] Lampinen, H. and Vainio, O.: " An optimization approach to designing OTAs for low-voltage sigma-delta modulators ", Proc. of the 2001 WCCI, pp. 1665-1671 (2001)
- [13] 高濱徹行 "集団的降下法による最適化 -差分進化と粒子群最適化-" (2007)
- [14] B. Fritzke, "Growing cell structures - a self-organizing network for unsupervised and supervised learning", Neural Networks, 7, pp. 1441-1460, 1994.
- [15] M. Takanashi, H. Torikai and T. Saito, "An approach to fusion of growing self-organising maps and adaptive resonance theory maps", IEICE Trans. Fundamentals, E90-A, 9, pp. 2047-2050, 2007.

- [16] T. Takahama and S. Sakai: " Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites ", Proc. of the 2006 WCCI, pp. 308-315 (2006)
- [17] Chunjie Li and Huiru Zhao: " The Hybrid Differential Evolution Algorithm for Optimal Power Flow Based on Simulated Annealing and Tabu Search ", Proc. of the 2010 IEEE, pp. 1-7 (2010)

研究業績

(国際会議)

I. Handa, T. Saito, " Insensitive Differential Evolution and Multi-Solution Problems, "Proc. International Conference on Evolutionary Computation Theory and Applications(ECTA 2011),pp. 292-295.

(国内発表)

半田 斎, 斎藤利通, 黒須智博, "成長型差分進化アルゴリズムの基本性能について," 電子情報通信学会 総合大会, ISS-P-104, pp. 4, 東北大学 川内キャンパス, 宮城, 2010 年 3 月.

半田 斎, 斎藤利通, 黒須智博, "成長型差分進化の基本特性について," 電子情報通信学会 技術研究報告 (NLP 研究会), vol. 110, no. 82, NLP2010-21, pp. pp.161-164, 琉球大学 50 周年記念館, 沖縄, 2010 年 6 月.

半田 斎, 斎藤利通, "成長型差分進化の探索能力," 電子情報通信学会 ソサイエティ大会, A-1-28, 大阪府立大学 中百舌鳥キャンパス, 大阪, 2010 年 9 月.

半田 斎, 斎藤利通, "複数解探索のための鈍感な差分進化について," 電子情報通信学会 技術研究報告 (NLP 研究会), vol. 110, no. 465, NLP2010-167, pp. pp.23-27, 東京理科大学森戸記念館, 東京, 2011 年 3 月.

半田 斎, 斎藤利通, "鈍感な差分進化による複数最適解探索へのアプローチ," 電子情報通信学会 第 24 回 回路とシステムワークショップ (KWS), A1-4-2, pp. 46-51, 淡路夢舞台国際会議場, 兵庫, 2011 年 8 月.

丸山一紀, 半田 斎, 斎藤利通, "鈍感な粒子群最適化について," 電子情報通信学会 ソサイエティ大会, A-2-15, 北海道大学 札幌キャンパス, 北海道, 2011 年 9 月.

半田斎, 斎藤利通, ”様々な鈍感差分進化の探索能力について,” 電子情報通信学会
ニューロコンピューティング研究会 (NC), 玉川大学, 東京, 2012 年 3 月.

謝辞

本研究は著者が法政大学大学院工学研究科電気工学専攻に在学中に行ったものである。この研究は同大学理工学部電気電子工学科 斎藤利通教授の指導下で行ったもので、全ての研究活動を遂行するにあたり同教授から大変御参考になる御指導・御鞭撻を沢山賜りました。ここに心から深謝いたします。

また、研究活動中に貴重な御助言・御討論を賜りました日本工業大学 神野健哉教授、本田技研工業株式会社 黒須智博氏には感謝の意を表明いたします。最後に法政大学理工学部電気電子工学科 斎藤利通研究室の皆様にはいろいろな有益な御討論・ご助言を戴きました。ここに感謝の意を表します。