

# 低精度の近似モデルを用いた比較推定法による Differential Evolution における関数評価回数の削減

高濱 徹行<sup>†a)</sup>      阪井 節子<sup>††</sup>      原 章<sup>†</sup>

Reducing the Number of Function Evaluations in Differential Evolution by Estimated Comparison Method Using an Approximation Model with Low Accuracy

Tetsuyuki TAKAHAMA<sup>†a)</sup>, Setsuko SAKAI<sup>††</sup>, and Akira HARA<sup>†</sup>

あらまし Differential Evolution (DE) は探索効率や頑健性の高さから多くの分野で利用されているが、関数評価の計算コストが高い問題などに対応するために、更なる関数評価回数の削減が望まれている。関数評価回数を削減する方法として、関数の近似モデルを構成し、近似値(推定値)を利用して探索を行う研究が活発に行われている。しかし、近似モデルの汎化性能を十分に高めるように学習するのは容易ではなく、更に近似モデルの学習に時間が掛かるという問題がある。本研究では、推定値を用いた比較を導入し、近似誤差に配慮した余裕を与えることにより、低精度の近似モデルを用いても効率的に評価回数が削減できる比較推定法を提案する。低精度の近似モデルとしては、ポテンシャルを用いた学習不要なモデルを用いる。本方法を DE, DE に比較推定法を適用した方法、及び世代交代モデル MGG において関数評価回数を削減した Saving MGG と比較することにより、比較推定法の有効性を示す。

キーワード ポテンシャルモデル, differential evolution, 関数近似, 進化的アルゴリズム

## 1. ま え が き

近年、最適化問題が大規模化し、目的関数の評価コストが増大してきているため、目的関数の評価回数を抑えるために、効率的な最適化アルゴリズムの開発が期待されている。関数評価回数を削減するための代表的な方法として、関数の近似モデルを構成し、近似値を利用して探索を行う研究が進化的アルゴリズム [1], [2] を中心に活発に行われている [3] ~ [11]。確かに、精度の高い近似モデルを構築できれば、関数評価回数を大きく削減できる。しかしながら、汎化性能の高い近似モデルを学習することは非常に困難であり、訓練データに対して近似精度の高い近似モデルの汎化

性能が高いとは限らないこともよく知られている。また、近似する目的関数によって適切な学習方法が異なるため、試行錯誤的に学習パラメータを調整しなければならないという問題もある。更に、学習のためにかかりの時間量及び記憶量を要するという問題があるため、比較的成本の低い関数の最適化に利用すると学習コストが評価回数の削減効果を超えてしまうこともある。

本研究では、高精度の近似モデル (approximation model with high accuracy), すなわち近似誤差が小さく汎化能力の高いモデルを得るために学習を必要とする近似モデルではなく、低精度の近似モデル (approximation model with low accuracy) を用いて関数評価回数を削減することを提案する。低精度の近似モデルは、真の関数値に対する近似値(推定値)の誤差は大きいですが、近傍の点の大小関係のある程度表現していると期待できるので、誤差を考慮して近傍の点の比較に利用することは可能であると考えられる。学習が不要な低精度の近似モデルを有効に利用できれば、汎化能力の高い学習のために必要な様々な学習パラメー

<sup>†</sup> 広島市立大学大学院情報科学研究科知能工学専攻, 広島市 Graduate School of Information Sciences, Hiroshima City University, 3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima-shi, 731-3194 Japan

<sup>††</sup> 広島修道大学商学部経営学科, 広島市 Faculty of Commercial Sciences, Hiroshima Shudo University, 1-1-1 Ozukahigashi, Asaminami-ku, Hiroshima-shi, 731-3195 Japan

a) E-mail: takahama@its.hiroshima-cu.ac.jp

タの調整が不要となり、広範囲の問題に簡単に利用できる。更に、学習のための時間が不要となり、評価コストがそれほど高くない目的関数の最適化問題にも適用可能となる。非常に高コストの問題については高精度の近似モデルを使用する価値があるが、比較的低コストの問題については、低精度の近似モデルの使用が適していると考えられる。しかしながら、このような低精度近似モデルを利用した関数評価回数の削減に関する検討はほとんど行われていないのが現状である。

次に、低精度の近似モデルを利用する最適化アルゴリズムについて考える。一般に最適化アルゴリズムの良さを決定することは困難であるが、網羅性と効率性が求められる性質であると考えられる。網羅性とは、できる限り広範囲に探索を行うことにより、局所解に陥らず大域最適解を探すことである。特に多峰性の問題において局所解を避けるには網羅性が要求される。また、目的関数の評価回数を必要最小限に抑え、効率的に探索を行うことは非常に重要である。一般に網羅性と効率性は相反する場合が多いが、網羅性と効率性のバランスに優れたアルゴリズムとして、複数の解により集団を形成し、集団から得られる情報に基づき、集団中の各解に対して新たに解を生成し、この解が良ければもとの解と置換していくアルゴリズムが提案されている。本論文では、このようなアルゴリズムを集団的降下法と呼ぶことにする。代表的な集団的降下法として、Differential Evolution (DE) や Particle Swarm Optimization (PSO) などがある。

DE [12] ~ [15] では、個体により集団を形成し、各個体から交叉と突然変異により新しい個体を生成し、新しい個体が良ければもとの個体と置換する。PSO [16] ~ [19] では、解に対応する位置情報をもつエージェントにより集団を形成し、各エージェントの現在位置とエージェントの最良位置と集団の最良位置により新しい位置を生成し、その位置がエージェントの最良位置より良ければ古い最良位置と置換する。これらの方法は、集団の各要素が降下法により個別に最適化されるため、集団が急速に特定の解に集中することが少なく、網羅性の高い探索が行われる。また、各要素が新しい解を生成する際に集団の情報を利用するため、一点による降下法と比較すると局所解に陥りにくい効率の良い探索が行われる。

本研究では、集団的降下法における古い解と新しい解の比較操作に注目し、比較のみに近似モデルによる推定値を用いることにより関数評価回数を削減する比

較推定法を提案する。比較推定法では、新しい解が良いと推定された場合のみ、その解の真の関数値を求め、新しい解の真値が古い解の真値より良ければ古い解を置き換える。したがって、比較推定が正しければ、降下方向以外の解の評価を削減できることになる。集団的降下法においては、一般に降下方向の解が見つかる頻度は低いため、十分な削除効果が期待できる。

比較推定法の重要な特長として、低精度の近似モデルを使用できることが挙げられる。比較推定法では推定値同士を比較するため、推定値により大小関係が正しく判断できれば、真の関数値との誤差が大きくても、最適解を見つけることができる。更に、比較推定法では、近似誤差に配慮し、良くなりそうな解を排除しないための余裕パラメータを指定する。このため、低精度の近似モデルを用いても有効に関数評価回数の削減が可能となる。したがって、従来のように汎化性能の高い高精度の近似モデルを学習する必要がなくなり、推定値を効率良く求められるため、比較的低コストの低い関数の最適化にも利用可能となる。

本論文では、集団的降下法としてDEを、低精度の近似モデルとしてはポテンシャルモデルを用いる。ポテンシャルモデルは、各解に対してポテンシャルエネルギーを仮定し、解のポテンシャルにより解の良さを推定するモデルであり、パラメータのない学習不要なモデルである。DEに比較推定法を適用した方法を、DE、世代交代モデル MGG において関数評価回数を削減した Saving MGG と比較することにより、比較推定法の有効性を示す。

本論文は次のように構成されている。2. で近似モデルを利用した最適化について説明する。3. で最適化問題を定義し、集団的降下法について説明する。4. で比較推定法とポテンシャルモデルを提案する。5. で比較推定法をDEに組み込んだ potential DE 法を提案する。6. で実験結果を示す。7. はまとめである。

## 2. 近似モデルを利用した最適化

関数を近似するために、様々な近似モデルが使用されている。例えば、多項式モデル [20] では二次モデルがよく採用されている。Kriging モデル [20], [21] では、大域モデルと局所的偏差により目的関数を近似する。ニューラルネットワーク [22] では、階層型ネットワーク、Radial Basis Function (RBF) ネットワークなどが利用されている [3], [5] ~ [8], [11]。多くのモデルでは、最小二乗法、こう配法、最尤法などによりパ

ラメータを学習するが、次元が大きくなるに従い、更に、近似モデルの精度を高くするに従い、学習に時間が掛かるという課題がある。

近似モデルを利用して進化的アルゴリズムにより最適化を行う場合、近似モデルによる推定値のみを用いる方法と、真の関数値と推定値の両方を用いる方法がある。最も評価回数を削減する方法は、精度の高い近似モデルを構築し、近似モデルによる推定値のみを用いる方法である。しかし、この方法は目的関数の性質が詳細に分かる場合に有効であり、多様な問題を解くには適していない。通常は、真の値と近似モデルに基づく推定値の両方を利用する。この場合、進化制御 (evolution control) アプローチと代理 (surrogate) アプローチに大別される。進化制御アプローチは、一部の個体を評価し残りの個体は推定値を用いて進化する方法であり、世代ベース (generation-based) の方法と個体ベース (individual-based) の方法に分けられる。世代ベースの方法は、適当な世代ごとにすべての個体の真の評価値を求めることにより、真値の世代と推定値の世代を分ける方法である [3], [6]。しかし、訓練データに対して精度の高い近似モデルが偽の最適解を作り出したり、最適解を隠したりすることが報告されている。したがって、何世代かにわたり推定値のみで最適化を行う際に、間違っただけに収束してしまう危険性が高くなり、近似モデルの汎化性能に大きな影響を受けるという課題がある。個体ベースの方法では、各世代ごとに、推定値の良好な (あるいはランダム選択された) 個体の真の評価値を求め、他は推定値を使用する [3] ~ [5], [7]。通常、複製選択や生存者選択において、真値をもつ個体と推定値をもつ個体を比較する必要があるため、推定値と真値の誤差をかなり小さくする必要がある。代理アプローチは、近似モデル上で推定最適解を求め、推定最適解を評価し、その値で近似モデルを更新する方法である [9] ~ [11]。推定値のみで最適化を行う過程を有するため、世代ベースの方法と同様の課題がある。

比較推定法は、すべての世代において真値をもつ個体のみが存在する。したがって、進化制御アプローチではあるが、世代ベースでも個体ベースでもない方法である。また、比較推定法では、新しい解を評価するかどうかを決定する際に推定値同士の比較を行い、新しい解が良いと推定されれば解を評価し、真値同士で比較を行う。したがって、推定値と真値を比較する必要がなくなるため、推定値同士の大小関係が保持され

る程度に近似すれば十分となり、推定値と真値の誤差を最小化する必要もなくなる。これにより、低精度の近似モデルでも十分な最適化が可能であると考えられる。

### 3. 最適化問題と集団的降下法

#### 3.1 最適化問題

一般的な最適化問題 ( $P$ ) は、以下のように定義できる。

$$(P) \text{ minimize } f(x) \quad (1)$$

$$\text{subject to } x \in \mathcal{F}$$

ここで、 $x = (x_1, \dots, x_n)$  は  $n$  次元決定変数ベクトル、 $f(x)$  は目的関数、 $\mathcal{F}$  は実行可能領域である。実行可能領域は不等式制約、等式制約、及び上下限制約で与えられる。

目的関数及び制約条件がともに線形の場合が線形計画問題、その他の場合が非線形計画問題である。本研究では、不等式制約及び等式制約のない非線形計画問題を対象とする。

#### 3.2 集団的降下法

DE や PSO などのように解集団による最適化の際に降下法を利用した最適化法である集団的降下法について説明する。集団的降下法のアルゴリズムは一般に以下のように記述できる。

- (1) 初期化：集団に属する解をランダムに発生する
- (2) 評価：すべての解を評価する
- (3) 終了判定：終了条件を満足すれば終了する
- (4) 各解に対して、
  - (a) 生成：各解と集団の情報に基づき新しい解を生成する
  - (b) 評価：新しい解を評価する
  - (c) 更新：新しい解が古い解より良ければ、古い解を新しい解で置換する
- (5) (3) へ戻る

### 4. 比較推定法

#### 4.1 集団的降下法と比較推定法

本研究では、関数評価回数を削減する手法を提案するが、できる限り汎用的な手法とするために、導入が容易であり、それぞれのアルゴリズムの特徴を生かすことに配慮した。このため、集団的降下法における (4b) 及び (4c) の部分のみを変更し、以下のよう

に、新しい解を評価する前に、古い解より良いと推定された解だけを実評価するように変更する。

```

if 新しい解が古い解より良いと推定された then
  (b) 評価: 新しい解を実評価する
  (c) 更新: 新しい解が古い解より良ければ、古い
        解を新しい解で置換する
endif

```

すなわち、比較推定法では、(1) 新しい解と古い解の良さを推定し、(2) 新しい解が良いと推定されれば評価・更新を行い、そうでなければ、新しい解を実評価することなく拒否するということになる。解の良さを推定は目的関数値を求めずに決める必要があるため、本研究ではポテンシャルモデルを用いて推定する。

#### 4.2 ポテンシャルモデル

ポテンシャルエネルギーとは、物体がある位置に存在することで物体に蓄えられる位置エネルギーである。例えば、次のようなものがある。質量  $m$  の物体が存在すれば、その周りには重力ポテンシャル  $U_g$  が発生し、その物体から距離  $r$  離れた質量  $m'$  の物体には引力  $F_g$  が働く。

$$U_g = -G \frac{m}{r}, F_g = G \frac{mm'}{r^2} \quad (2)$$

ここで、 $G$  は万有引力定数である。

本研究では、ある解  $x$  が存在することにより、その解から  $r$  離れた位置に、目的ポテンシャル  $U_o$  (potential for objective) と混雑ポテンシャル  $U_c$  (potential for congestion) が発生すると仮定する。

$$U_o = \frac{f(x)}{r^p} \quad (3)$$

$$U_c = \frac{1}{r^p} \quad (4)$$

ここでは、簡単のため、比例係数を 1 とした。また、距離のべき乗数  $p$  は正数であり通常 1 あるいは 2 とする。

解集合  $X = \{x_1, x_2, \dots, x_N\}$  が存在し、関数値  $f(x_i)$ ,  $i = 1, 2, \dots, N$  が既知であるとき、ある点  $y$  におけるポテンシャルを、以下のように定義する。

$$U_o(y) = \sum_i \frac{f(x_i)}{d(x_i, y)^p} \quad (5)$$

$$U_c(y) = \sum_i \frac{1}{d(x_i, y)^p} \quad (6)$$

ただし、 $d(x, y)$  は点  $x$  と点  $y$  間の距離である。

このとき、 $U_o$  はある点  $y$  と各点  $x_i$  との距離の逆数で重み付けされた各点における関数値の和を示しており、 $U_c$  はこれら重みの和を示している。したがって、点  $y$  における関数の推定値  $\hat{f}(y)$  は、これらの商で与えられる。

$$\hat{f}(y) = U_o(y)/U_c(y) \quad (7)$$

なお、混雑ポテンシャルは近傍にどの程度他の点が存在し混雑しているかを示しており、探索の網羅性向上に利用できると考えられるが、本論文では推定値を求めるために使用している。

集団的降下法では、関数値が既知の解集合  $X$  中の古い解  $x_i$  から新しい解  $x'_i$  を生成する。このとき、 $x_i$  と  $x'_i$  におけるそれぞれの推定値は、以下のように古い解を除いた解集合によるポテンシャルで求める。

$$U_o(x_i^{(l)}) = \sum_{j \neq i} \frac{f(x_j)}{d(x_j, x_i^{(l)})} \quad (8)$$

$$U_c(x_i^{(l)}) = \sum_{j \neq i} \frac{1}{d(x_j, x_i^{(l)})} \quad (9)$$

$$\hat{f}(x_i^{(l)}) = U_o(x_i^{(l)})/U_c(x_i^{(l)}) \quad (10)$$

$x_i$  における関数値の推定に  $x_i$  自体を含めると、 $x_i$  における推定値は真値に一致する。このとき推定値は  $x_i$  の近傍で急激に真値に近づくため、推定値の変化の滑らかさを壊し、関数全体の概形の近似が困難になることがある。また、 $x_i$  における推定値の精度は高くなるが  $x'_i$  における推定値の精度は低いため、比較の際の精度が却って下がるという問題もある。このため、本研究では  $x_i$  を除いた推定値を用いた。

## 5. potential DE

Differential Evolution にポテンシャルモデルに基づく比較推定法を適用したアルゴリズムである potential DE を提案する。

### 5.1 Differential Evolution

Differential Evolution (DE) は進化的戦略 (evolution strategy) の一つであり、Storn & Price [12], [13] によって提案された。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きており、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示されてきている。

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、rand は基本ベクトルのための親を集団からランダムに選択し、best は集団の最良個体を選択する。“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指定する。“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。

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もつ。各世代において、すべての個体を親として選択する。各親に対して、次のような処理が行われる。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。差分ベクトルは  $F$  (scaling factor) が乗算され基本ベクトルに加えられる。その結果得られたベクトルと親が交叉し、 $CR$  (crossover factor) により指定された確率で親の遺伝子をベクトルの要素で置換することにより、子のベクトル (trial vector) が生成される。最後に、生存者選択として、子が親よりも良ければ、親を子で置換する。

本研究では、差分ベクトル数を 1 ( $\text{num} = 1$ ) とした DE/rand/1/exp を用いる。

## 5.2 potential DE のアルゴリズム

potential DE/rand/1/exp のアルゴリズムは以下のように記述できる [14], [15]。

Step0 初期化。  $N$  個の初期個体  $x_i$  を初期探索点として生成し、初期集団  $\{x_i, i = 1, 2, \dots, N\}$  を構成する。すべての個体を評価する。

Step1 終了判定。終了条件を満足すれば、アルゴリズムは終了する。終了条件としては、最大の繰返し回数や関数評価回数を用いることが多い。

Step2 突然変異。各個体  $x_i$  に対して、3 個体  $x_{p1}$ ,  $x_{p2}$ ,  $x_{p3}$  を  $x_i$  及び互いに重複しないようにランダムに選択する。新しいベクトル  $x'$  を基本ベクトル  $x_{p1}$  及び差分ベクトル  $x_{p2} - x_{p3}$  から以下のように生成する。

$$x' = x_{p1} + F(x_{p2} - x_{p3}) \quad (11)$$

ここで、 $F$  はスケーリングパラメータである。

Step3 交叉。ベクトル  $x'$  を親  $x_i$  と交叉し、子ベクトル  $x_i^{\text{new}}$  を生成する。交叉点  $j$  をすべての次元  $[1, n]$  からランダムに選択する。子ベクトル  $x_i^{\text{new}}$  の  $j$  番目の要素を  $x'$  の  $j$  番目の要素から継承する。それ以降の次元は、交叉パラメータ  $CR$  によって指数関数的に減少する確率で、 $x'$  の要素から継承する。残りの部分は、親  $x_i$  から継承する。実際の処理では、Step2 と Step3 は一まとまりの処理で実現される。

Step4 比較推定法。もし、子ベクトルが親ベクトルよりも良いと推定されれば、Step5 に進む。そうでなければ、Step6 に進む。

Step5 生存者選択。子ベクトルを評価する。子ベクトル  $x_i^{\text{new}}$  が親ベクトルよりも良ければ子ベクトルが生存者となり、親を子ベクトルで置換する。

Step6 Step1 に戻る。

以下に擬似コードを示す。

```
potential DE/rand/1/exp()
{
    P=Generate N individuals {x_i} randomly;
    Evaluate x_i, i = 1, 2, ..., N;
    for(t=1; 終了条件が満たされない; t++) {
        for(i=1; i ≤ N; i++) {
            (p1, p2, p3)=select randomly in [1, N] \ {i}
                s. t. p_j ≠ p_k (j, k = 1, 2, 3, j ≠ k);
            x_i^new = x_i ∈ P;
            j=select randomly from [1, n];
            k=1;
            do {
                x_ij^new = x_p1,j + F(x_p2,j - x_p3,j);
                j=(j + 1)%n;
                k++;
            } while(k ≤ n && u(0, 1) < CR);
            if(Better(x_i^new, x_i)) {
                Evaluate x_i^new;
                if(f(x_i^new) < f(x_i)) x_i = x_i^new;
            }
        }
    }
}
```

ここで、 $u(0, 1)$  は区間  $[0, 1]$  の一様乱数、 $\text{Better}(\cdot, \cdot)$

は、比較推定法に基づき、解の良さを比較する関数である。この関数値が常に真ならば、すなわち、 $\text{Better}(\cdot, \cdot) = 1$  ならば、常に関数値の評価が行われることになるため、通常の  $\text{DE}/\text{rand}/1/\text{exp}$  と一致する。

## 6. 実験

### 6.1 テスト問題

本実験では、変数間依存性、悪スケール性、及び大谷構造を有する問題を用いる [23]。変数間依存性の強い問題として、稜構造を有する問題を用いる。悪スケール性のある問題として、稜構造でかつ変数によりスケールが異なる問題を用いる。大谷構造とは、微視的に見れば局所解となる多数の小さな谷が存在するが、巨視的に見れば大きな谷が一つだけ存在し、その谷が最適解となっている構造であり、Rastrigin 関数とその典型例である。

以下に、関数とその初期化領域を示す。なお、 $n$  は次元数を表している。

- $f_1$ : Sphere 関数

$$f(x) = \sum_{i=1}^n x_i^2, \quad -5.12 \leq x_i \leq 5.12 \quad (12)$$

単峰性の関数で、点  $(0, 0, \dots, 0)$  で最小値 0 をとる。

- $f_2$ : Rosenbrock 関数

$$f(x) = \sum_{i=2}^n \{100(x_1 - x_i^2)^2 + (x_i - 1)^2\}, \quad (13)$$

$$-2.048 \leq x_i \leq 2.048$$

単峰性の稜構造を有する関数で、点  $(1, 1, \dots, 1)$  で最小値 0 をとる。

- $f_3$ : ill-scaled Rosenbrock 関数

$$f(x) = \sum_{i=2}^n \{100(x_1 - (ix_i)^2)^2 + (ix_i - 1)^2\}, \quad (14)$$

$$-2.048/i \leq x_i \leq 2.048/i$$

単峰性の稜構造を有する関数で、点  $(1, \frac{1}{2}, \dots, \frac{1}{n})$  で最小値 0 をとる。

- $f_4$ : Rastrigin 関数

$$f(x) = 10n + \sum_{i=1}^n \{x_i^2 - 10 \cos(2\pi x_i)\}, \quad (15)$$

$$-5.12 \leq x_i \leq 5.12$$

多峰性の大谷構造を有する関数で、点  $(0, 0, \dots, 0)$  で最小値 0 をとる。

図 1 に  $n = 2$  のときの関数  $f_1, f_2, f_4$  のグラフを示す。また、表 1 に各関数の特徴を示した [23]。

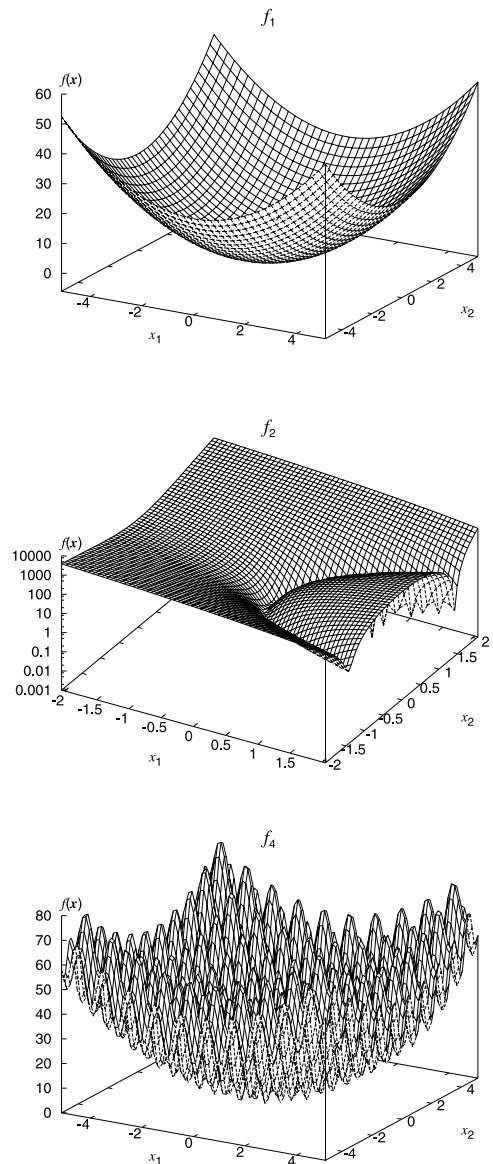


図 1 関数  $f_1, f_2, f_4$  のグラフ  
Fig.1 Graphs of  $f_1, f_2$  and  $f_4$ .

表 1 テスト問題の特徴  
Table 1 Features of test problems.

関数	変数間依存性	悪スケール性	大谷構造
$f_1$	なし	なし	なし
$f_2$	強い	なし	なし
$f_3$	強い	あり	なし
$f_4$	なし	なし	あり

表 2 DE で良い解を見つける頻度  
Table 2 Frequency of finding better solution in DE.

Func	eval	succ	fail	rate (%)
$f_1$	76,887.4	10,739.1	66,098.2	13.98
$f_2$	408,749.4	17,505.7	391,193.8	4.24
$f_3$	400,122.5	17,292.8	382,779.8	4.32
$f_4$	275,101.8	14,177.8	260,874.1	5.15

### 6.2 予備実験

DE/rand/1/exp を用い、次元数  $n = 30$  に設定し、 $f_1$  から  $f_4$  の関数を最適化する。DE の設定は、個体数  $N = 50$ ,  $F = 0.7$ ,  $CR = 0.95$  とし、各関数について 20 回の試行を行い、結果を考察する。個体数は  $2n$  から  $10n$  程度が良いとされているが、効率性を重視し少し小さい値とした。なお、試行打ち切り条件は、文献 [23] と同様に、(1) 評価値が  $1.0 \times 10^{-7}$  以下に達した、(2) 評価回数が  $f_1$  及び  $f_2$  は  $2n \times 10^5$ ,  $f_3$  は  $5n \times 10^5$ ,  $f_4$  は  $3n \times 10^5$  に達した、のいずれかの条件を満足する場合とした。

このとき、関数の評価回数を eval に、子ベクトルが親ベクトルより良い解となる成功した関数評価の回数を succ に、失敗した回数を fail に、成功の確率を rate として表 2 に示した。簡単な単峰性の関数でさえ、探索によってより良い解が見つかる確率は 13.98% であり、可能性はあまり高くない。また、より困難な問題では、良い解が見つかる確率は 5% 程度に過ぎない。このように関数評価の多くの部分が悪い解に対するものとなっている。したがって、もし悪い解であることを的確に判断できれば、関数評価回数を大きく削減することが可能となる。しかし、一般にこの判断は困難であり、精度の低い判断を行うと、少数の成功した解に対する評価を省略することになり、最適解を見つけれなくなる危険性が高くなる。推定には必ず誤差が存在するため、誤差を考慮して判断を行う必要がある。

### 6.3 potential DE の設定

potential DE の実験条件は、予備実験のときと同様とした。関数 Better は推定誤差を考慮して、以下のようにある程度の余裕を与える。

$$\text{Better}(x'_i, x_i) \Leftrightarrow \frac{\hat{f}(x'_i) - \hat{f}(x_i)}{|\hat{f}(x_i)|} \leq \delta \quad (16)$$

ただし、 $\delta \geq 0$  は誤差の余裕を決めるパラメータであり、推定値を決めるための解集合  $X$  は、各世代における集団  $P$  とした。 $\delta$  が 0 の場合は、推定値に基づく比較となり、多数のベクトルを拒否するため、良いベクトルも拒否してしまう可能性が高くなる。 $\delta$  を大き

くすると、推定値が少し悪いベクトルも受け入れるため、良いベクトルを拒否する可能性は低くなるが、逆に、拒否する数が減少する。したがって、 $\delta$  は適当な大きさにとる必要がある。

更に、各世代において集団に存在する解の次元ごとの幅、すなわち最大値と最小値の差により正規化した距離を導入する。これは、探索の進行により解の存在する探索領域が扁平的になることがあり、このような場合にも探索領域における関数形状をよりの確に表現するには、各次元をその大きさに合わせてスケールングする必要があるためである。

$$d(x, y) = \sqrt{\sum_j \frac{(x_j - y_j)^2}{(\max_i x_{ij} - \min_i x_{ij})^2}} \quad (17)$$

なお、距離のべき乗数は、最も計算量が少なくなる  $p = 2$  に設定した。

### 6.4 実験結果

本実験では、 $\delta = 0.001, 0.005, 0.01$  の 3 通りについて調べた。実験結果を表 3 に示す。実際に関数を評価した回数を eval、そのうち子ベクトルが良かった回数を succ、悪かった回数を fail、成功率を rate に示した。ecomp-succ は、拒否されたベクトルが親ベクトルよりも悪かった回数、ecomp-fail は、拒否されたベクトルが良かった回数、ecomp-rate は比較推定法の成功率、すなわち、拒否されたベクトルが実際に悪かった確率を示す。なお、比較を容易にするために、評価回数及び成功率が最良の結果を太字で示し、DE と表記した行に DE の結果である表 2 の内容を再掲した。

DE と比較して成功率が倍程度あるいは倍以上に向上し、関数評価回数もかなり減少しており、 $f_1, f_4$  では半分以下に減少している。比較推定法の成功率は 90% 以上であり、余裕 ( $\delta$ ) を大きくするにつれて増加するが、関数の成功率は逆に減少する傾向がある。実験では、 $f_1, f_3, f_4$  については余裕  $\delta = 0.001$  の場合、 $f_2$  については余裕  $\delta = 0.005$  の場合が最も効率が高い結果となった。

potential DE を評価するために、文献 [23] で提案された UNDX- $m$ /Saving MGG の結果と比較する。この手法は、多様性維持に優れた世代交代モデルである MGG (Minimal Generation Gap) [24] に対して、重み付きサンプリングに基づく分布推定により子個体の有望度を算出し、有望な子個体のみを評価する機能を付加したものである。これにより、高い網羅性を維持しながら、評価回数を削減することにより効率性を高

表 3 実験結果  
Table 3 Experimental results.

Func.	params	eval	succ	fail	rate (%)	ecom-p-succ	ecom-p-fail	ecom-p-rate (%)
$f_1$	0.001	<b>33,537.45</b>	9,852.5	23,634.95	<b>29.38</b>	55,491.85	4,061.5	93.18
	0.005	50,058.7	10,600.2	39,408.5	21.18	30,754.25	974	96.93
	0.01	61,135.75	10,642.35	50,443.4	17.41	16,364.4	222.25	<b>98.66</b>
	DE	76,887.4	10,739.1	66,098.2	13.98			
$f_2$	0.001	352,745	40,388.9	312,306.2	<b>11.45</b>	298,408.2	8,029.1	97.38
	0.005	<b>338,964.4</b>	27,303.35	311,611	8.05	172,743.9	2,040.3	98.83
	0.01	362,287.8	23,994.9	338,242.9	6.62	125,034.1	990.35	<b>99.21</b>
	DE	408,749.4	17,505.7	391,193.8	4.24			
$f_3$	0.001	<b>339,911.7</b>	38,727.6	301,134	<b>11.39</b>	295,146.6	7,885.05	97.40
	0.005	344,141.3	27,711	316,380.3	8.05	170,904.4	1,979.35	98.86
	0.01	363,576.5	24,233.75	339,292.7	6.67	123,862.9	1,009.65	<b>99.19</b>
	DE	400,122.5	17,292.8	382,779.8	4.32			
$f_4$	0.001	<b>139,045.2</b>	13,270	125,725.2	<b>9.54</b>	168,580.9	5,006.85	97.12
	0.005	228,197.7	14,236.05	213,911.6	6.24	82,484.15	1,028.45	98.77
	0.01	234,902.3	14,103.3	220,749	7.38	40,538.3	278.05	<b>99.32</b>
	DE	275,101.8	14,177.8	260,874.1	5.15			

表 4 Saving MGG, DE との評価回数の比較  
Table 4 Comparison of the number of function evaluations in DE and Saving MGG.

Algorithm	Parameters	$f_1$	$f_2$	$f_3$	$f_4$
DE	potential 0.001	<b><math>3.35 \times 10^4</math></b> (0.43, 0.07)	<b><math>3.53 \times 10^5</math></b> (0.86, 0.38)	<b><math>3.40 \times 10^5</math></b> (0.85, 0.56)	<b><math>1.39 \times 10^5</math></b> (0.51, 0.07)
DE	original	$7.69 \times 10^4$ (1, 0.17)	$4.09 \times 10^5$ (1, 0.45)	$4.00 \times 10^5$ (1, 0.66)	$2.75 \times 10^5$ (1, 0.14)
Saving MGG	UNDX- $m$	$4.62 \times 10^5$ (6.00, 1)	$9.18 \times 10^5$ (2.24, 1)	$6.03 \times 10^5$ (1.51, 1)	$2.00 \times 10^6$ (7.27, 1)
	SPX	$5.60 \times 10^5$ (7.28, 1.21)	$9.37 \times 10^5$ (2.29, 1.02)	$1.00 \times 10^6$ (2.50, 1.66)	$2.46 \times 10^6$ (8.95, 1.23)

めている。

表 4 に potential DE と DE 及び Saving MGG との評価回数の比較を示す。括弧内は DE を 1 としたとき、Saving MGG/UNDX- $m$  を 1 としたときの比率である。

potential DE を DE と比較すると、関数評価回数が  $f_2, f_3$  では約 15%削減され、 $f_1, f_4$  では約 50%削減されている。次に、Saving MGG と比較すると、関数評価回数が  $f_2, f_3$  では約 40~60%削減され、 $f_1, f_4$  では約 90%削減されている。したがって、potential DE は網羅性を損なうことなく、探索効率を大幅に向上させることに成功した、非常に効率性の高いアルゴリズムであるといえる。

DE 及び potential DE ( $\delta = 0.001$ ) を比較するために、二つの方法について各問題における関数値の変化及び成功率の変化を図 2~図 5 の上側、下側にそれぞれ示した。横軸は関数評価回数、縦軸は関数値及び成功率である。potential DE では、比較推定法の成功率を estimated comparison success rate に示した。なお、グラフは 20 回の実験の平均値を示しているた

め、グラフの終末付近では探索を打ち切った試行が多くなり、データ数が減少し平均値に乱れが出ている。

すべての問題において potential DE が DE よりも常に高速に関数値を減少させており、成功率についても、potential DE の方が常に高い値を示している。また、比較推定法の成功率についてもほとんどの場合に 90%を超えており、かなり精度が高いことが分かる。

## 6.5 議論

比較推定法が有効な最適化問題について検討する。比較推定法は、適用対象であるベースとなる最適化法の特徴を生かすように設計されているため、比較推定法が有効な問題はベースとなる最適化法に依存し、その最適化法が有効な問題とおおよそ一致する。

しかし、2. で述べたように一般に近似モデルを利用する場合には、近似モデルが偽の最適解を作り出したり、最適解を隠したりすることにより、本来の最適解を探索しにくい場合がある。偽の最適解を作り出した場合は、その解の付近に生成された新しい解は現在の解よりも良いと推定される。しかし、比較推定法では、真の関数値が評価されることになり、新しい解が良く



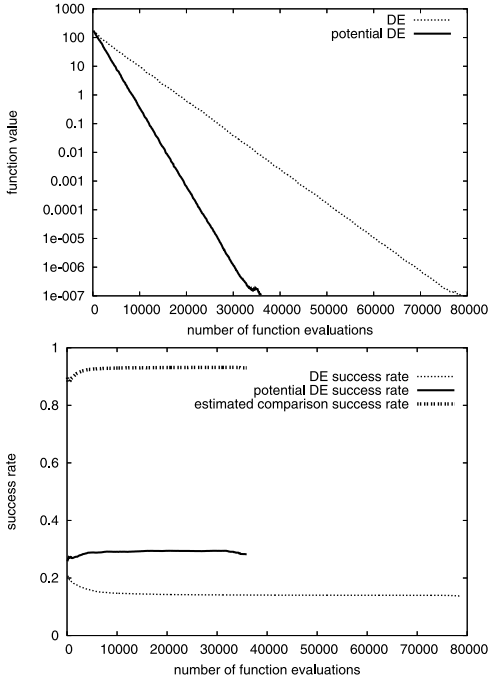


図 2  $f_1$  における関数値及び成功率の変化  
Fig. 2 Graphs of function value and success rate over the number of function evaluations in  $f_1$ .

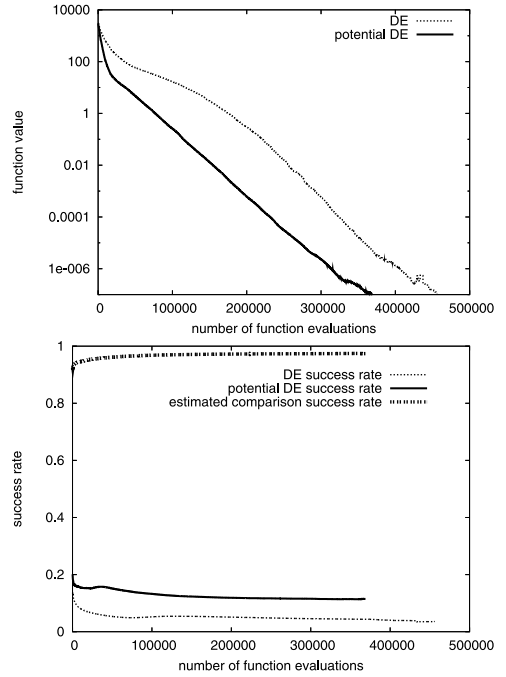


図 4  $f_3$  における関数値及び成功率の変化  
Fig. 4 Graphs of function value and success rate over the number of function evaluations in  $f_3$ .

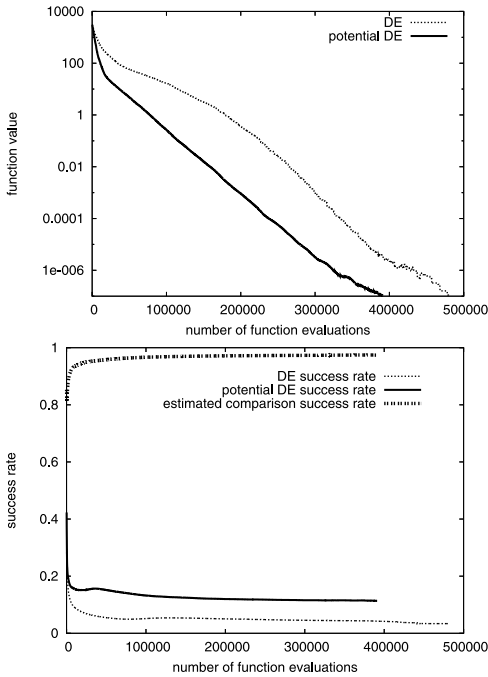


図 3  $f_2$  における関数値及び成功率の変化  
Fig. 3 Graphs of function value and success rate over the number of function evaluations in  $f_2$ .

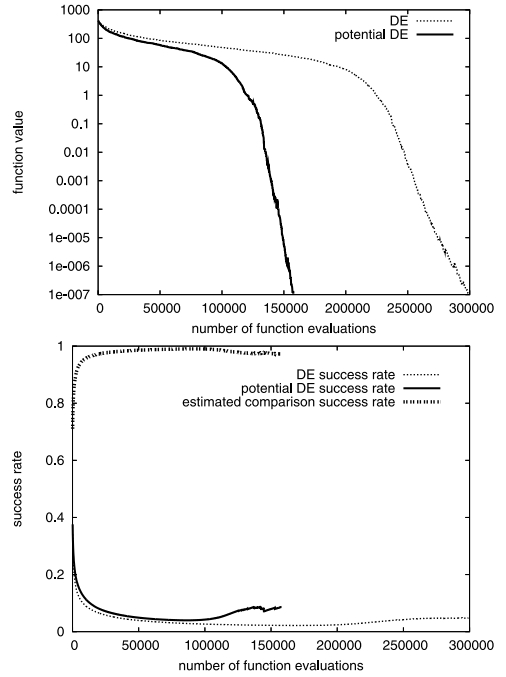


図 5  $f_4$  における関数値及び成功率の変化  
Fig. 5 Graphs of function value and success rate over the number of function evaluations in  $f_4$ .

ないことが判明し、偽の最適解の方に探索を行うことはないため、この影響はほとんど受けないと考えられる。これに対して、最適解が隠された場合には、最適解の付近に生成された新しい解は現在の解よりも良くないと推定される。したがって、比較推定法でも最適解の方への探索が行われない可能性がある。しかし、この場合でも比較推定法では誤差のための余裕を導入しているため、隠される度合が余裕の範囲内ならば、真の関数値が評価され、新しい解が良いと判定されるため、最適解への探索が可能となる。このように、比較推定法は他の近似モデルを用いる方法よりも偽の最適解や隠された最適解の問題への影響が少ない方法であると考えられる。

逆に、一般に近似モデルを用いることにより、真の関数値のみを用いるよりも最適解が探索しやすくなる場合のあることが指摘されている [10]。例えば、極端に凹凸の激しい曲面を滑らかに近似することにより隠された最適解をとらえやすくなるような場合である。比較推定法でも滑らかな近似モデルを採用することにより、このような効果を得られると考えられる。

最後に、ポテンシャルモデルによる近似に関する問題について検討する。比較推定法では、近似モデルの精度が高ければ、余裕を小さくし、関数評価回数を大きく削減することができる。しかし、ポテンシャルモデルは低精度の近似モデルであるため、上記のように、余裕を十分にとらないと最適解を発見できない場合がある。特に、最適解の近傍で急激な変化をもつ問題では、滑らかな近似を行うポテンシャル法では近似誤差が大きくなり、最適解を発見しにくくなる場合がある。また、探索の初期段階では少数の探索点で広い領域を近似するため、近似精度が低下しやすい。特に、局所解が広域に分布するような問題では、初期段階で最適解の領域を見逃す可能性があると考えられる。このような場合に対処するには余裕を適切に設定・制御することが有効であると考えられるため、今後更なる検討が必要である。

## 7. む す び

本研究では、集団的降下法において、関数評価回数を削減し効率性を向上する方法として、比較推定法を提案した。比較推定法は、近似モデルによる推定値に基づき解を比較し、不必要な解の評価をできる限り行わないようにすることで、関数評価回数を削減する方法であり、低精度の近似モデルを利用可能にする方

法である。更に、低精度の近似モデルとして、学習の不要なポテンシャルモデルを提案した。比較推定法を DE に適用し、DE 及び Saving MG と比較することにより、potential DE が他の方法と比較して効率性の高い方法であることを示した。これにより、近似精度が高くないポテンシャルモデルを利用しても十分に比較推定が可能であることを示し、低精度の近似モデルを利用して関数評価回数の削減が行えることを示した。

今後は、比較推定法を PSO などの他の集団的降下法へ適用すること、混雑ポテンシャルなどの混雑度を示す指標を利用することにより網羅性についても考慮した比較を導入すること、推定値に基づく比較に Simulated Annealing のような確率的な比較を取り入れることを予定している。

謝辞 この研究の一部は、日本学術振興会科学研究費補助金基盤研究 (c) (No.16500083, 17510139) 及び広島市立大学特定研究費 (一般研究) の援助のもとで行われた。

## 文 献

- [1] A.E. Eiben and J.E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003.
- [2] 高濱徹行, 阪井節子, “ $\epsilon$  制約遺伝的アルゴリズムによる制約付き最適化”, *情処学論*, vol.47, no.6, pp.1861–1871, 2006.
- [3] Y. Jin, M. Olhofer, and B. Sendhoff, “On evolutionary optimization with approximate fitness functions,” *Proc. Genetic and Evolutionary Computation Conference*, pp.786–792, Morgan Kaufmann, 2000.
- [4] 花木 康, 橋山智訓, 大熊 繁, “適応度の推論による進化的アルゴリズムの計算時間の短縮”, *電学論 (C)*, vol.120, no.1, pp.123–129, 2000.
- [5] 中山弘隆, 荒木雅生, 佐々木理恵, “RBF ネットワークと遺伝的アルゴリズムによる未知目的の最適化”, *システム制御情報学会論文誌*, vol.13, no.3, pp.152–154, 2000.
- [6] Y. Jin, M. Olhofer, and B. Sendhoff, “A framework for evolutionary optimization with approximate fitness functions,” *IEEE Trans. Evol. Comput.*, vol.6, no.5, pp.481–494, 2002.
- [7] Y. Jin and B. Sendhoff, “Reducing fitness evaluations using clustering techniques and neural networks ensembles,” *Genetic and Evolutionary Computation Conference*, vol.3102 of LNCS, pp.688–699, Springer, 2004.
- [8] 田中雅晴, 溝口正信, 高見 勲, “適応度予測型遺伝的アルゴリズムにおける探索性能の向上”, *電学論 (C)*, vol.124, no.9, pp.1853–1860, 2004.
- [9] D. Büche, N.N. Schraudolph, and P. Koumoutsakos, “Accelerating evolutionary algorithms with gaussian process fitness function models,” *IEEE Trans. Syst. Man Cybern., C, Appl. Rev.*, vol.35, no.2, pp.183–

- 194, 2005.
- [10] Y.S. Ong, Z. Zhou, and D. Lim, "Curse and blessing of uncertainty in evolutionary algorithm using approximation," 2006 IEEE Congress on Evolutionary Computation, pp.9833-9840, Vancouver, BC, Canada, 2006.
- [11] F.G. Guimarães, E.F. Wanner, F. Campelo, R.H. Takahashi, H. Igarashi, D.A. Lowther, and J.A. Ramírez, "Local learning and search in memetic algorithms," 2006 IEEE Congress on Evolutionary Computation, pp.9841-9848, Vancouver, BC, Canada, 2006.
- [12] R. Storn and K. Price, "Minimizing the real functions of the ICEC'96 contest by differential evolution," Proc. International Conference on Evolutionary Computation, pp.842-844, 1996.
- [13] R. Storn and K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," J. Global Optimization, vol.11, pp.341-359, 1997.
- [14] T. Takahama and S. Sakai, "Constrained optimization by the  $\epsilon$  constrained differential evolution with gradient-based mutation and feasible elites," Proc. 2006 IEEE Congress on Evolutionary Computation, pp.308-315, 2006.
- [15] T. Takahama, S. Sakai, and N. Iwane, "Solving nonlinear constrained optimization problems by the  $\epsilon$  constrained differential evolution," Proc. 2006 IEEE Conference on Systems, Man, and Cybernetics, pp.2322-2327, 2006.
- [16] J. Kennedy and R.C. Eberhart, "Particle swarm optimization," Proc. IEEE International Conference on Neural Networks, vol.IV, pp.1942-1948, Perth, Australia, 1995.
- [17] J. Kennedy and R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann, San Francisco, 2001.
- [18] T. Takahama and S. Sakai, "Constrained optimization by the  $\alpha$  constrained particle swarm optimizer," J. Advanced Computational Intelligence and Intelligent Informatics, vol.9, no.3, pp.282-289, 2005.
- [19] T. Takahama and S. Sakai, "Constrained optimization by  $\epsilon$  constrained particle swarm optimizer with  $\epsilon$ -level control," Proc. 4th IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST'05), pp.1019-1029, 2005.
- [20] A. Giunta and L. Watson, "A comparison of approximation modeling techniques: Polynomial versus interpolating models," AIAA Technical Report, 98-4755, 1998.
- [21] T.W. Simpson, T.M. Mauery, J.J. Korte, and F. Mistree, "Comparison of response surface and kriging models in the multidisciplinary design of an aerospike nozzle," AIAA Technical Report, 98-4758, 1998.
- [22] W. Shyy, P.K. Tucker, and R. Vaidyanathan, "Response surface and neural network techniques for

rocket engine injector optimization," AIAA Technical Report, 99-2455, 1999.

- [23] 田中雅晴, 土谷千加夫, 佐久間淳, 小野 功, 小林重信, "Saving MGG:実数値 GA/MGG における適応度評価回数の削減," 人工知能誌, vol.21, no.6, pp.547-555, 2006.
- [24] 佐藤 浩, 小野 功, 小林重信, "遺伝的アルゴリズムにおける世代交代モデルの提案と評価," 人工知能誌, vol.12, no.7, pp.734-744, 1997.

(平成 19 年 4 月 13 日受付, 9 月 13 日再受付)



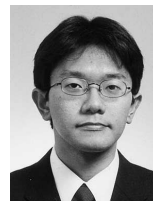
高濱 徹行 (正員)

1982 京大・工・電気第二工学卒。1987 同大大学院博士課程研究認定退学。同年福井大学工学部助手。1994 同大講師。1998 広島市立大学情報科学部知能情報システム工学科助教授。2005 より同大教授。進化的計算及び群知能を含む Natural Computing, 非線形最適化, 機械学習などに関する研究に従事。情報処理学会, 人工知能学会, 教育システム情報学会, 言語処理学会, IEEE 各会員。工博。



阪井 節子

1979 福井大・教育卒。1984 阪大大学院基礎工学研究科数理系後期課程単位取得退学。1986 甲子園大学経営情報学部講師。1990 福井大教育学部助教授。1998 より広島修道大学商学部経営学科助教授。ゲーム理論, 意思決定, ファジィ数理計画, GA によるファジィ制御, CAI などに関する研究に従事。日本 OR 学会, 日本知能情報ファジィ学会, 日本生産管理学会, IEEE 各会員。工博。



原 章 (正員)

1997 東工大・工・電気・電子卒, 2002 同大大学院総合理工学研究科博士課程了。2002 広島市立大学情報科学部知能情報システム工学科助手。2007 より同大講師。生物の適応・学習のメカニズムに興味をもち, 進化計算論やマルチエージェントシステム, 人工社会に関する研究に従事。情報処理学会, 日本知能情報ファジィ学会, IEEE 各会員。博士(工学)。