

パラメータの相関を考慮した 適応型差分進化アルゴリズム JADE の改良

広島修道大学商学部
Faculty of Commercial Sciences, Hiroshima Shudo University
広島市立大学大学院 情報科学研究科 高濱 徹行 (Tetsuyuki Takahama)
Graduate School of Information Sciences, Hiroshima City University

1 はじめに

差分進化 (Differential Evolution, DE) は 1995 年に Storn と Price [1, 2] によって提案された実数空間における最適化アルゴリズムであり, 進化的アルゴリズム (Evolutionary Algorithm, EA) の一つである. 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 [3]. また, DE は進化的計算に関する国際会議のコンペティションにおいて優秀な成績を収めている [4, 5, 6].

DE の利用が増加してきている主な理由としては, 単純な算術演算に基づいているため高速に動作すること, 制御パラメータがスケールリングファクター F , 交叉率 CR , 集団サイズ NP の 3 つと単純であることが挙げられる. しかし, 制御パラメータについては, 問題によって適切なパラメータ設定が異なり, パラメータ設定によって DE の性能に大きな差が出るため, 非常に重要な検討課題となっている. 制御パラメータを調整する主な方法は, 下記のように大別できる.

- (1) 試行錯誤的調整: 推奨されたパラメータから始めてパラメータを試行錯誤的に調整する.
- (2) 観測による調整 (observation-based tuning): 探索状況を観測し, 観測量に応じて適切なパラメータ値を推論し, パラメータを動的に調整する. ファジィ推論を用いる FADE (Fuzzy Adaptive DE) [7] や ファジィクラスタリングを用いる DESFC (DE with Speciation and Fuzzy Clustering) [8], 単峰性・多峰性を検出する LMDE (DE with detecting Landscape Modality) [9, 10], ランク情報に基づきパラメータ値を選択する RDE (Rank-based DE) [11, 12] が提案されている. FADE では世代間における探索点の移動量と関数値の変化量を, DESFC では探索点の分割エントロピー (partition entropy) を, LMDE では直線上に生成したサンプリング点における関数値の変化を, RDE では探索点のランク情報を観測量として用いている.
- (3) 成功による調整 (success-based tuning): 良い探索点を生成した場合を成功と捉え, 成功したときのパラメータ値が使用されやすいようにパラメータを動的に調整する. なお, 個体の遺伝情報に制御パラメータを含む自己適応 (self-adaptation) も成功による調整の一種であると考えられる. 自己適応により F, CR, NP を調整する DESAP (Differential Evolution with Self-Adapting Populations) [13], 自己適応により F, CR を調整し成功率により変異戦略の選択確率を調整する SaDE (Self-adaptive DE) [14], 成功に応じて F, CR の平均値を調整する JADE (adaptive DE with optional external archive) [15] や MDE_pBX (modified DE with p -best crossover) [16] などが提案されている.

しかし, (1) はユーザの負荷が大きいという課題, (2) は問題や問題のスケールに依存しない観測量を設定するのが困難であるという課題がある. (3) では, 探索点の近傍で良い探索点を発見した場合, 集団が収束する方向にパラメータが調整される. このため, 良い探索点が存在する範囲の狭い稜構造問題や多峰性

の問題において、小さな成功 (small success) の方向にパラメータが調整され、大きな成功 (big success) を見逃し、局所解に収束してしまうことがある。

本研究では、局所解への収束を避けるための工夫がなされている (3) に属する JADE を対象とし、JADE を改良する方法を提案する。成功に基づく調整を行う手法では、JADE を含めてパラメータを独立的に調整することが多く、パラメータ間の依存関係を考慮して調整を行う手法はほとんど提案されていない。本研究では、2つのアルゴリズムパラメータ F と CR の依存関係に着目する。最初に生成されたパラメータのパラメータ値と2つのパラメータ間の相関係数を用いて、後に生成するパラメータの確率分布を修正して2つ目のパラメータの値を決定するという方法を提案する。これにより、パラメータ間の依存関係が実現できると考えられる。幾つかのベンチマーク問題を最適化し、性能を比較することにより、本手法の有効性を示す。

以下、2. で DE を簡潔に紹介する。3. で、JADE について説明する。4. で、本手法のアルゴリズムを説明する。5. で他の方法と比較した性能を示す。6. はまとめである。

2 差分進化 (Differential Evolution)

2.1 概要

Differential evolution (DE) は Storn and Price[1, 2] によって提案された進化的アルゴリズムである。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 の重要な特徴として、遺伝的アルゴリズムや進化的戦略ではガウス突然変異などのステップ幅を制御する必要があるが、DE で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ことが挙げられる。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り、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、DE はガウス突然変異の代わりに、基本ベクトル (base vector) と差分ベクトル (difference vectors) との重み付き和を突然変異として採用している。基本的には、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ことにより、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のである。

DE には幾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、DE/rand/num/cross 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、DE/best/num/cross 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、DE/base/num/bin は一定の確率で遺伝子を交換する交叉 (binomial crossover) を用い、DE/base/num/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。

2.2 アルゴリズム

DE/rand/1/bin のアルゴリズムは以下のように記述できる [17, 5].

Step0 初期化. N 個の初期個体 \mathbf{x}_i を探索空間 S 内に生成し, 初期集団 $P = \{\mathbf{x}_i, i = 1, 2, \dots, N\}$ 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, 最大の繰り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 (target vector) \mathbf{x}_i に対して, 3 個体 $\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, \mathbf{x}_{r_3}$ を \mathbf{x}_i および互いに重複しないようにランダムに選択する. 基本ベクトル \mathbf{x}_{r_1} および差分ベクトル $\mathbf{x}_{r_2} - \mathbf{x}_{r_3}$ から変異ベクトル (mutant vector) \mathbf{m}_i を以下のように生成する.

$$\mathbf{m}_i = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (1)$$

ここで, F はスケールリングパラメータである.

Step3 交叉. 変異ベクトル \mathbf{m}_i と親 \mathbf{x}_i を交叉し, 子ベクトル (trial vector) $\mathbf{x}_i^{\text{child}}$ を生成する. 交差点 j を全ての次元 $[1, n]$ からランダムに選択する. 子ベクトル $\mathbf{x}_i^{\text{child}}$ の j 番目の要素を \mathbf{m}_i の j 番目の要素から継承する. それ以外の次元は, 交叉パラメータ CR の確率で, \mathbf{m}_i の要素から継承する. 残りの部分は, 親 \mathbf{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 生存者選択. 子ベクトルを評価する. 子ベクトル $\mathbf{x}_i^{\text{child}}$ が親ベクトルよりも良ければ子ベクトルが生存者となる. そうでなければ, 親ベクトルが生存者となる. すべての個体が選択されたならば, Step5 に進む. そうでなければ, Step2 に戻る.

Step5 世代交代: 生存者によって集団 P を置換する. Step1 に戻る.

以下に DE/rand/1/bin の擬似コードを示す.

```

DE/rand/1/bin()
{
  P=Generate N individuals {x_i} randomly;
  Evaluate x_i, i = 1, 2, ..., N;
  for(t=1; !終了条件; t++) {
    for(i=1; i ≤ N; i++) {
      (p1, p2, p3)=select randomly from [1, N] s.t. p1 ≠ p2 ≠ p3 ≠ i;
      j=select randomly from [1, n];
      for(k=1; k ≤ n; k++) {
        if(k == j || u(0,1) < CR)
          x_{ik}^{child} = x_{r1,k} + F(x_{r2,k} - x_{r3,k});
        else
          x_{ik}^{child} = x_{ik};
      }
      Evaluate x_i^{child};
      if(f(x_i^{child}) ≤ f(x_i)) z_i = x_i^{child};
    }
    P={z_i};
  }
}

```

ここで, $u(0, 1)$ は区間 $[0, 1]$ の一様乱数である.

2.3 戦略とパラメータ

一般に, 大域探索性 (多様性) を重視すれば局所解に陥りにくくなるが, 探索速度は低下し, 局所探索性 (収束性) を重視すれば, 探索速度は向上するが, 局所解に陥りやすくなる. DE では, 基本ベクトルの選択

によって、大域探索性と局所探索性のどちらを重視するかが決まる。以下が代表的な戦略である。

- rand/1 (強い大域探索)

$$\mathbf{m}_i = \mathbf{x}_{r1} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3})$$
- best/1 (強い局所探索)

$$\mathbf{m}_i = \mathbf{x}^{\text{best}} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3})$$
- best/2 (強い局所探索)

$$\mathbf{m}_i = \mathbf{x}^{\text{best}} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3}) + F(\mathbf{x}_{r4} - \mathbf{x}_{r5})$$
- current-to(target-to)/1 (弱い大域探索)

$$\mathbf{m}_i = \mathbf{x}_i + F(\mathbf{x}_{r2} - \mathbf{x}_{r3})$$
- current-to-best(target-to-best)/1 (弱い局所探索)

$$\mathbf{m}_i = \mathbf{x}_i + F'(\mathbf{x}^{\text{best}} - \mathbf{x}_i) + F(\mathbf{x}_{r2} - \mathbf{x}_{r3})$$
- rand-to-best (大域・局所探索)

$$\mathbf{m}_i = \mathbf{x}_{r1} + F'(\mathbf{x}^{\text{best}} - \mathbf{x}_{r1}) + F(\mathbf{x}_{r2} - \mathbf{x}_{r3})$$

また、スケーリングファクター F を大きくすれば大域探索性が強まり、小さくすれば局所探索性が強まる。同様に、集団サイズ N を大きくすれば多様性が強まり、小さくすれば収束性が強まるため、 F と集団サイズはトレードオフの関係にある。すなわち、小さな集団サイズでは F を大きくし、大きな集団サイズでは F を小さくすれば、同等の結果を得ることができる。交叉パラメータ CR は、同時に変化させる変数の数を制御するため、変数分離型の問題では小さく、変数間依存性の強い問題では大きくする必要がある。

JADE では、全ての個体に対して固定されたパラメータ値を用いるのではなく、各個体毎に異なるパラメータ値を生成し、子が生存者となる成功した場合のパラメータ値から優れたパラメータ値を学習することにより、適応的な調整を実現している。

3 JADE

JADE では、スケーリングファクターの平均値 μ_F と交叉率の平均値 μ_{CR} によって良好なパラメータ値の確率分布を表現し、成功した時のパラメータ値を用いてそれぞれの平均値を学習する。初期値は、 $\mu_F = \mu_{CR} = 0.5$ である。各個体 \mathbf{x}_i のために、異なるスケーリングファクター F_i と交叉率 CR_i が次式に従って独立に生成される。

$$F_i \sim C(\mu_F, \sigma_F) \quad (2)$$

$$CR_i \sim N(\mu_{CR}, \sigma_{CR}^2) \quad (3)$$

ここで、 $C(\mu_F, \sigma_F)$ は位置パラメータ μ_F 、尺度パラメータ $\sigma_F = 0.1$ の Cauchy 分布に基づく確率分布である。 $N(\mu_{CR}, \sigma_{CR}^2)$ は平均 μ_{CR} 、標準偏差 $\sigma_{CR} = 0.1$ の正規分布に基づく確率分布である。 CR_i は区間 $[0, 1]$ となるように切り捨てられる。 F_i は負の値の場合は再生成され、それ以外の場合は 1 以下となるように切り捨てられる。位置パラメータ μ_F と平均 μ_{CR} は、安定した学習を実現するために、指数移動平均を用い

て更新される.

$$\mu_F = (1-c)\mu_F + cS_{F2}/S_F \quad (4)$$

$$\mu_{CR} = (1-c)\mu_{CR} + cS_{CR}/S_N \quad (5)$$

ここで, S_N は親より良い子が生成された成功した回数, S_F, S_{F2}, S_{CR} はそれぞれ親よりよい子が生成された成功時の F_i, F_i^2, CR_i の和である. すなわち, μ_{CR} は成功時の単純な算術平均により更新される. これに対して, μ_F は多様性を保持するために, 大きな値を重視した重み付き平均によって更新される. 定数 c は値を更新する際に使用される区間 $(0,1]$ の重みであり, 推奨値は 0.1 である.

JADE では “current-to-pbest” と呼ばれる突然変異戦略が提案され, 親と上位個体の中間点が基本ベクトルとなる. 変異ベクトルは次式で生成される.

$$m_i = x_i + F_i(x^{pbest} - x_i) + F_i(x_{r2} - x_{r3}) \quad (6)$$

ここで, x^{pbest} は上位 100p% 個体からランダムに選択された個体である.

JADE ではアーカイブを使用する方法も提案されているが, 本研究ではアーカイブを使用しないため, 説明は省略する.

4 パラメータの相関を考慮した適応型 DE

本研究では, 適応型 DE である JADE を参考に, パラメータ間の相関を考慮した適応型差分進化 CADE (Correlation based Adaptive DE) を提案する.

4.1 条件付き正規分布

一般に, k 次元確率変数ベクトル $\mathbf{u} = (U_1, U_2, \dots, U_k)$ の正規分布は次のように記述される.

$$\mathbf{u} \sim N(\boldsymbol{\mu}, \Sigma) \quad (7)$$

$$f(\mathbf{u}) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{u} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{u} - \boldsymbol{\mu})\right) \quad (8)$$

ここで, $\boldsymbol{\mu}$ は平均値ベクトル, Σ は分散共分散行列, f は確率密度関数, $|\Sigma|$ は Σ の行列式, Σ^{-1} は Σ の逆行列である. $k=2$ の 2 変数の場合, 分散共分散行列は次のようになる.

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \quad (9)$$

ここで, ρ は U_1, U_2 の相関係数である. U_1 が与えられたとき, U_2 の条件付き分布は次のようになる.

$$U_2 | U_1 = u_1 \sim N(\mu_2 + \rho \frac{\sigma_2}{\sigma_1} (u_1 - \mu_1), (1 - \rho^2)\sigma_2^2) \quad (10)$$

4.2 アルゴリズムパラメータ間の条件付き分布

本研究では, アルゴリズムパラメータ F, CR の 2 変数条件付き分布を考える. 成功時のパラメータ値が $S = \{(F_i, CR_i)\}$ で与えられると仮定する. このとき, F と CR 間の相関係数 ρ は次のようになる.

$$\rho = \frac{\sum_{(F_i, CR_i) \in S} (F_i - \mu_{SF})(CR_i - \mu_{SCR})}{\sigma_{SF}\sigma_{SCR}} \quad (11)$$

ここで、 μ_{SF} と μ_{SCR} は S における F と CR の平均、 σ_{SF} と σ_{SCR} は S における F と CR の標準偏差である。 F と CR が正規分布にしたがっているならば、式 (10) を F と CR で書き直すと、次のようになる。

$$CR | F = F_i \sim N(\mu_{SCR} + \rho \frac{\sigma_{SCR}}{\sigma_{SF}} (F_i - \mu_{SF}), (1 - \rho^2) \sigma_{SCR}^2) \quad (12)$$

$$F | CR = CR_i \sim N(\mu_{SF} + \rho \frac{\sigma_{SF}}{\sigma_{SCR}} (CR_i - \mu_{SCR}), (1 - \rho^2) \sigma_{SF}^2) \quad (13)$$

論文 [18] では式 (12) を使用する方法について検討したが³、本論文では式 (13) を使用する方法について検討する。条件付き確率を用いる際に、成功回数が少ないと標準偏差が大きく変化し、安定性が失われることがある。また、 F は正規分布ではなく、Cauchy 分布で生成されている。これに対応するために、本論文では以下の対策を採用した。

- μ_{SCR} と σ_{SCR} の推定値として、 CR_i 生成時の平均 μ_{CR} と標準偏差 $\sigma_{CR} = 0.1$ を用いることにより、安定性を高める。
- F_i は Cauchy 分布で生成されているため、 μ_{SF} を μ_F で推定することは困難である。ただし、安定性を高めるために、 μ_{SF} の重み付き指数移動平均である μ_F を用いる。
- パラメータ値の多様性を保持するために、標準偏差 $\sqrt{(1 - \rho^2) \sigma_{SF}^2}$ の代わりに $\sigma_F = 0.1$ を用いる。
- ρ の代わりにその指数移動平均 $\bar{\rho}$ を用いて安定性を高める。
- 成功回数が非常に小さい場合、相関係数 ρ が -1 あるいは 1 に近い値となり、パラメータ値の多様性が失われる。本論文では、成功回数が 5 以上の時のみ ρ と $\bar{\rho}$ を更新する。

すなわち、以下の式を採用することになる。

$$F | CR = CR_i \sim C(\mu_F + \bar{\rho} \frac{\bar{\sigma}_{SF}}{\sigma_{CR}} (CR_i - \mu_{CR}), \sigma_F) \quad (14)$$

$$\bar{\sigma}_{SF} = (1 - c) \bar{\sigma}_{SF} + c \sigma_{SF} \quad (15)$$

$$\bar{\rho} = (1 - c) \bar{\rho} + c \rho \quad (16)$$

4.3 アルゴリズム

提案手法 CADE のアルゴリズムは以下の通りである [18]。

Step0 パラメータの初期化

スケールリングファクターの平均値 $\mu_F = 0.5$ 、交叉率の平均値 $\mu_{CR} = 0.5$ とする。パラメータ生成時の標準偏差を $\sigma_F = 0.1$ 、 $\sigma_{CR} = 0.1$ とする。成功時のパラメータ値を保持するリスト S を空にする。指数移動平均の初期値 $\bar{\sigma}_{SF} = \sigma_F$ とする。

Step1 個体の初期化

初期集団 $P = \{x_i | i = 1, 2, \dots, N\}$ を探索空間 S 中でランダムに生成する。 N は集団サイズである。

Step2 終了条件

関数評価回数が最大評価回数 FE_{\max} を超えれば、アルゴリズムは終了する。

Step3 DE 操作

交叉率 CR_i を正規分布 $N(\mu_{CR}, \sigma_{CR}^2)$ で生成する。スケーリングファクター F_i を式 (14) に基づき Cauchy 分布で生成する。パラメータを F_i, CR_i として DE/current-to-pbest/1/bin 操作を実行し、子 $\mathbf{x}^{\text{child}}$ を生成する。子が親より良ければ、成功と判断し、子を生存者として選択し、成功時のパラメータ値 (F_i, CR_i) をリスト S に追加する。成功でなければ、親 \mathbf{x}_i を生存者とする。

Step4 パラメータの更新

JADE と同様に μ_F と μ_{CR} を S に基づいて更新する。もし、成功回数が十分ならば、 $\bar{\rho}$ と $\bar{\sigma}_{SF}$ を S に基づいて更新する。

Step5 Step2 へ戻る。

CADE の擬似コードを図 1 に示す。‘+’で始まる行は、JADE に対する変更点を示している。

5 実験**5.1 テスト問題**

本実験では、単峰性関数である Sphere, Schwefel 2.22, Schwefel 1.2, 多峰性関数である Rastrigin, Ackley, Griewank を用いる [19]。表 1 に、関数定義とその初期化領域を示す。なお、 D は次元数を表している。

表 1: Test functions of dimension D . These are sphere, Schwefel 2.22, Schwefel 1.2, Rastrigin, Ackley, and Griewank, respectively[19]

Test functions	Bound constraints
$f_1(\mathbf{x}) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$
$f_2(\mathbf{x}) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]^D$
$f_3(\mathbf{x}) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^D$
$f_4(\mathbf{x}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$
$f_5(\mathbf{x}) = -20 \exp\left(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^D$
$f_6(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^D$

図 2 に $n = 2$ のときの関数 $f_1 \sim f_6$ のグラフを示す。

```

CADE/current-to-pbest/1/bin()
{
   $\mu_F = \mu_{CR} = 0.5$ ;  $\sigma_F = \sigma_{CR} = 0.1$ ;  $S = \phi$ ;
+  $\bar{\rho} = 0$ ;  $\bar{\sigma}_{SF} = \sigma_F$ ;
// Initialize a population
   $P = NP$  individuals generated randomly in  $S$ ;
   $FE = FE + N$ ;
  for( $t=1$ ;  $FE < FE_{max}$ ;  $t++$ ) {
    for( $i=1$ ;  $i \leq N$ ;  $i++$ ) {
       $CR_i = \mu_{CR} + N(0, \sigma_{CR})$ ;
      if( $CR_i < 0$ )  $CR_i = 0$ ;
      else if( $CR_i > 1$ )  $CR_i = 1$ ;
      do {
+        $F_i = \mu_F + \bar{\rho} \frac{\bar{\sigma}_{SF}}{\sigma_{CR}} (CR_i - \mu_{CR}) + C(0, \sigma_F)$ ;
      } while( $F_i \leq 0$ );
      if( $F_i > 1$ )  $F_i = 1$ ;
       $x^{pbest}$  = Randomly selected from top 100% in  $P$ ;
       $x_{r1}$  = Randomly selected from  $P(r1 \notin \{i\})$ ;
       $x_{r2}$  = Randomly selected from  $P(r2 \notin \{i, r1\})$ ;
       $m_i = x_i + F_i(x^{pbest} - x_i) + F_i(x_{r1} - x_{r2})$ ;
       $x^{child}$  = generated from  $x_i$  and  $m_i$  by binomial crossover as a trial vector;
       $FE = FE + 1$ ;
// Survivor selection
      if( $f(x^{child}) < f(z)$ ) {
         $z_i = x^{child}$ ;
         $S = S \cup \{(F_i, CR_i)\}$ ; // a success case is added to  $S$ 
      }
      else  $z_i = x_i$ ;
    }
     $P = \{z_i\}$ ;
    if( $|S| > 0$ ) {
       $\mu_F = (1 - c)\mu_F + c \sum_{F_i \in S} F_i^2 / \sum_{F_i \in S} F_i$ ;
       $\mu_{CR} = (1 - c)\mu_{CR} + c \sum_{CR_i \in S} CR_i / |S|$ ;
+     if( $|S| \geq 5$ ) {
+        $\sigma_{SCR}$  = standard deviation of  $CR$  in  $S$ ;
+        $\sigma_{SF}$  = standard deviation of  $F$  in  $S$ ;
+        $\bar{\sigma}_{SF} = (1 - c)\bar{\sigma}_{SF} + c\sigma_{SF}$ ;
+        $\rho$  = correlation between  $F$  and  $CR$  in  $S$  using  $\sigma_{SCR}$  and  $\sigma_{SF}$ ;
+        $\bar{\rho} = (1 - c)\bar{\rho} + c\rho$ ;
+     }
  }
}

```

図 1: The pseudo-code of CADE

5.2 実験条件

次元数 $D = 30$ に設定し, f_1 から f_6 の関数を最適化する. CADE のアルゴリズムパラメータは JADE と同じものを採用した. すなわち, 個体数 $N = 100$, 平均値の初期値 $\mu_F = \mu_{CR} = 0.5$, パラメータ生成時の固定された標準偏差 $\sigma_F = \sigma_{CR} = 0.1$ とする. CADE のアルゴリズムパラメータとして, 相関係数の初期値 $\bar{\rho} = 0$ とした. 各関数について 50 回の試行を行い, 主として JADE と CADE の実験結果について考察する.

5.3 実験結果

CADE, JADE, jDE, SaDE, DE/rand/1/bin, PSO の実験結果を表 2 に示す. CADE 以外の結果は論文 [15] の結果を用いた. 上段の数値は平均値, 下段の数値は標準偏差である. また, 全てのアルゴリズム

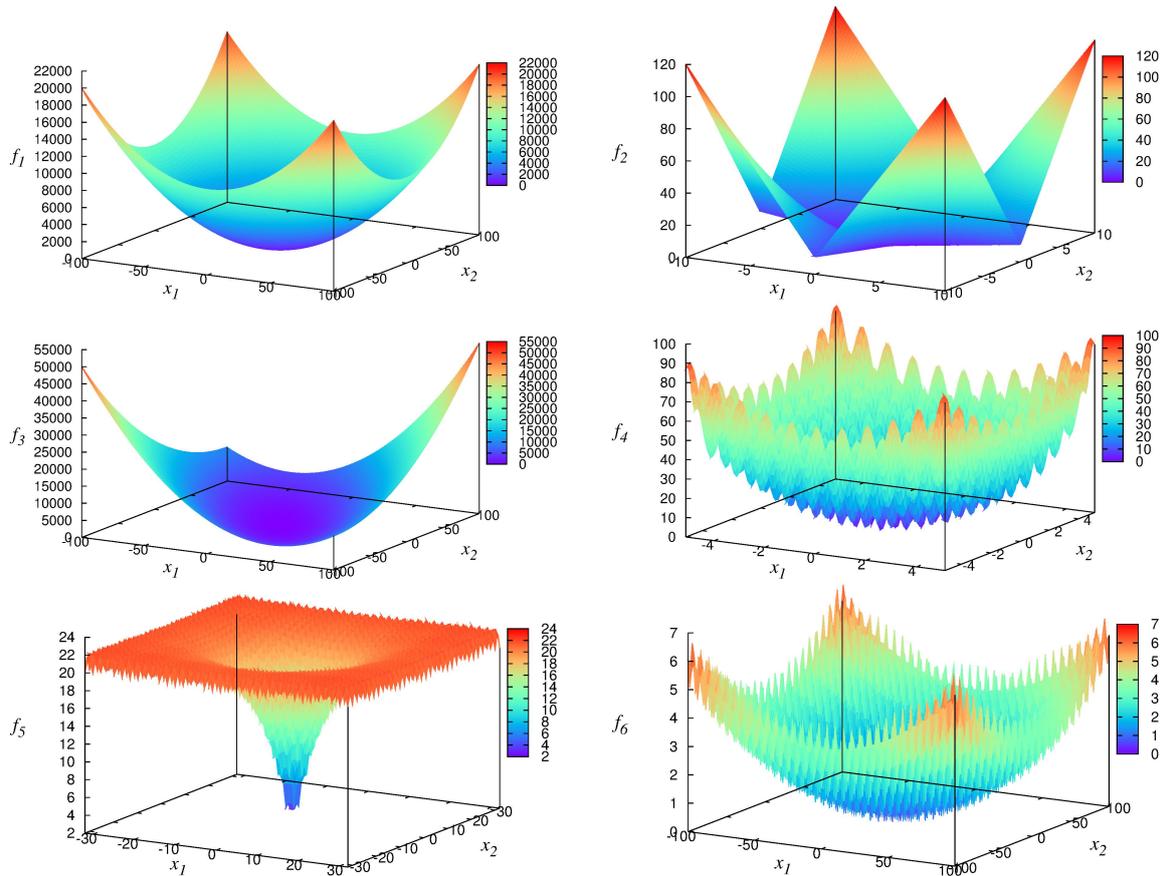


図 2: 関数 $f_1 \sim f_6$ のグラフ

中で最良の結果を太字で示した。

CADE は、6 問題中で f_3 を除く f_1, f_2, f_4, f_5, f_6 の 5 関数で最良の結果を示した。 f_3 については、アーカイブを用いた JADE が最良の結果を示した。また、全ての問題において CADE はアーカイブを用いない JADE より優れた結果を示した。したがって、条件付き分布によるパラメータの適応的制御は、独立的なパラメータ制御よりも優れており、条件付き分布の有効性が示された。

6 おわりに

本研究では、2つのパラメータ F と CR を独立に生成するのではなく、2つのパラメータの相関を考慮して生成する方法を提案した。本論文では、最初に CR を正規分布により生成し、次にその値と相関係数に基づいて F を生成するという方法について検討した。提案手法は、6 問題全てにおいて、本手法の基本モデルである JADE より優れた性能を示した。

F は Cauchy 分布に基づいているが、本研究では 2 変数正規分布による条件付き確率のアイデアに基づいて F_i を生成しており、理論的な不整合が存在する。今後は、理論的な不整合を解消する方法について検討する予定である。

表 2: 実験結果

	FE_{max}	CADE	JADE w/o archive	JADE w archive	jDE	SaDE	DE/rand/1/bin	PSO
f_1	150,000	2.59e-71	1.8e-60	1.3e-54	2.5e-28	4.5e-20	9.8e-14	9.6e-42
		(8.22e-71)	(8.4e-60)	(9.2e-54)	(3.5e-28)	(6.9e-20)	(8.4e-14)	(2.7e-41)
f_2	200,000	6.58e-50	1.8e-25	3.9e-22	1.5e-23	1.9e-14	1.6e-09	9.3e-21
		(2.72e-49)	(8.8e-25)	(2.7e-21)	(1.0e-23)	(1.05e-14)	(1.1e-09)	1(6.3e-20)
f_3	500,000	6.38e-62	5.7e-61	6.0e-87	5.2e-14	9.0e-37	6.6e-11	2.5e-19
		(2.27e-61)	(2.7e-60)	(1.9e-86)	(1.1e-13)	(5.43e-36)	(8.8e-11)	(3.9e-19)
f_4	100,000	2.44e-05	1.0e-04	1.4e-04	1.5e-04	1.2e-03	1.8e+02	5.2e+01
		(2.43e-05)	(6.0e-05)	(6.5e-05)	(2.0e-04)	(6.5e-04)	(1.3e+01)	(1.6e+01)
f_4	500,000	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	7.1e+01	5.2e+01
		(0.0e+00)	(0.0e+00)	(0.0e+00)	(0.0e+00)	(0.0e+00)	(2.1e+01)	(1.6e+01)
f_5	50,000	9.96e-11	8.2e-10	3.0e-09	3.5e-04	2.7e-03	1.1e-01	4.6e-01
		(1.17e-10)	(6.9e-10)	(2.2e-09)	(1.0e-04)	(5.1e-04)	(3.9e-02)	(6.6e-01)
f_5	200,000	4.30e-15	4.4e-15	4.4e-15	4.7e-15	4.3e-14	9.7e-11	4.6e-01
		(1.77e-15)	(0.0e+00)	(0.0e+00)	(9.6e-16)	(2.6e-14)	(5.0e-11)	(6.6e-01)
f_6	50,000	2.09e-10	9.9e-08	2.0e-04	1.9e-05	7.8e-04	2.0e-01	1.3e-02
		(1.46e-09)	(6.0e-07)	(1.4e-03)	(5.8e-05)	(1.2e-03)	(1.1e-01)	(1.7e-02)
f_6	300,000	0.0e+00	0.0e+00	2.0e-04	0.0e+00	0.0e+00	0.0e+00	1.1e-02
		(0.0e+00)	(0.0e+00)	(1.4e-03)	(0.0e+00)	(0.0e+00)	(0.0e+00)	(1.6e-02)

謝辞 この研究の一部は、本研究は JSPS 科研費 24500177, 26350443 および広島修道大学 2014 年度調査研究費の援助を受けた。

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] U. K. Chakraborty Ed.: "Advances in Differential Evolution", Springer (2008).
- [4] S. Das and P. Suganthan: "Differential evolution: A survey of the state-of-the-art", IEEE Transactions on Evolutionary Computation, 15, 1, pp. 4–31 (2011).
- [5] T. Takahama and S. Sakai: "Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites", Proc. of the 2006 IEEE Congress on Evolutionary Computation, pp. 308–315 (2006).
- [6] T. Takahama and S. Sakai: "Constrained optimization by the ϵ constrained differential evolution with an archive and gradient-based mutation", Proc. of the 2010 IEEE Congress on Evolutionary Computation, pp. 1680–1688 (2010).
- [7] J. Liu and J. Lampinen: "A fuzzy adaptive differential evolution algorithm", Soft Comput., 9, 6, pp. 448–462 (2005).
- [8] T. Takahama and S. Sakai: "Fuzzy c-means clustering and partition entropy for species-best strategy and search mode selection in nonlinear optimization by differential evolution", Proc. of the 2011 IEEE International Conference on Fuzzy Systems, pp. 290–297 (2011).
- [9] T. Takahama and S. Sakai: "Differential evolution with dynamic strategy and parameter selection by detecting landscape modality", Proc. of the 2012 IEEE Congress on Evolutionary Computation, pp. 2114–2121 (2012).

- [10] T. Takahama and S. Sakai: "Large scale optimization by differential evolution with landscape modality detection and a diversity archive", Proc. of the 2012 IEEE Congress on Evolutionary Computation, pp. 2842–2849 (2012).
- [11] 高濱, 阪井, 原: "RDE:探索点のランク情報を利用した効率的な differential evolution の提案", 電子情報通信学会論文誌 D, **95**, 5, pp. 1196–1205 (2012).
- [12] T. Takahama and S. Sakai: "Efficient constrained optimization by the ϵ constrained rank-based differential evolution", Proc. of the 2012 IEEE Congress on Evolutionary Computation, pp. 62–69 (2012).
- [13] J. Teo: "Exploring dynamic self-adaptive populations in differential evolution", Soft Comput., **10**, 8, pp. 673–686 (2006).
- [14] A. K. Qin and P. N. Suganthan: "Self-adaptive differential evolution algorithm for numerical optimization", Proc. of the 2005 IEEE Congress on Evolutionary Computation, pp. 1785–1791 (2005).
- [15] J. Zhang and A. C. Sanderson: "JADE: Adaptive differential evolution with optional external archive", IEEE Transactions on Evolutionary Computation, **13**, 5, pp. 945–958 (2009).
- [16] S. M. Islam, S. Das, S. Ghosh, S. Roy and P. N. Suganthan: "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, **42**, 2, pp. 482–500 (2012).
- [17] 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 Systems, Man, and Cybernetics, pp. 2322–2327 (2006).
- [18] T. Takahama and S. Sakai: "An adaptive differential evolution considering correlation of two algorithm parameters", Proc. of the Joint 7th International Conference on Soft Computing and Intelligent Systems and 15th International Symposium on Advanced Intelligent Systems (SCIS&ISIS2014) (2014). to appear.
- [19] X. Yao, Y. Liu, K.-H. Liang and G. Lin: "Fast evolutionary algorithms", Advances in Evolutionary Computing: Theory and Applications (Eds. by A. Ghosh and S. Tsutsui), Springer-Verlag New York, Inc., New York, NY, USA, pp. 45–94 (2003).