

適応型差分進化 JADE における個体順位に基づく グループ別パラメータ制御

Parameter Control According to Groups Defined by Individual Ranking for Adaptive Differential Evolution JADE

広島修道大学商学部

阪井 節子 (Setsuko Sakai)

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 [4-6]。

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 N 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 では探索点の関数値に対する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, N 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], JADE の改良である MDE- p BX (modified DE with p -best crossover) [16], SHADE (Success-History based Adaptive DE) [17], CADE (Correlation-based Adaptive DE) [18] 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。JADE では、成功時のパラメータ値を用いて、パラメータを生成するための確率分布における平均値を学習する。このとき、一種類のパラメータに対して一つの確率分布を用いている。しかし、探索点の性質によって異なるパラメータ分布を学習することにより探索性能を向上できる可能性がある。本研究では、この可能性を検証するために、関数値のランクに基づき探索点をグループに分割し、グループ毎にパラメータ分布を学習する方法を提案する。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 について、3. で JADE について簡潔に説明する。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)

DE のアイデアとアルゴリズムについて説明する。

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り、解集団を用いた多点探索を行う確率的な直接探索法である。

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) を使い、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 [5, 19]。

Step0 初期化。 N 個の初期個体 x_i を探索空間 S 内に生成し、初期集団 $P = \{x_i \mid i = 1, 2, \dots, N\}$ を構成する。

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を用いることが多い。

Step2 突然変異。各個体 (target vector) x_i に対して、3 個体 $x_{r_1}, x_{r_2}, x_{r_3}$ を 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 x_{r_1} および差分ベクトル $x_{r_2} - x_{r_3}$ から変異ベクトル (mutant vector) m_i を以下のように生成する。

$$\mathbf{m}_i = \mathbf{x}_{r1} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (1)$$

ここで、 F はスケーリングファクターである。

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 \mathbf{x}_i 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。

Step5 Step1 に戻る。

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、親より良い子が生成された場合を成功とし、成功した時のパラメータ値を用いて平均値を学習する。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。

$$F_i \sim C(\mu_F, \sigma_F), \quad CR_i \sim N(\mu_{CR}, \sigma_{CR}^2) \quad (2)$$

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。

$$\mu_F = (1 - c)\mu_F + cS_{F^2}/S_F, \quad \mu_{CR} = (1 - c)\mu_{CR} + cS_{CR}/S_N \quad (3)$$

ここで、 S_N は親より良い子が生成された成功回数、 S_F 、 S_{F^2} 、 S_{CR} はそれぞれ親より良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。

$$\mathbf{m}_i = \mathbf{x}_i + F_i(\mathbf{x}^{\text{pbest}} - \mathbf{x}_i) + F_i(\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (4)$$

ここで、 $\mathbf{x}^{\text{pbest}}$ は上位 $100p\%$ 個体からランダムに選択された個体であり、 p の推奨値は 0.05 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 提案手法

4.1 ランクによるグループ化

本研究では、関数値のランクによって個体をグループ化する。このため、各個体 $\mathbf{x}_i \in \{\mathbf{x}_i \mid i = 1, 2, \dots, N\}$ に対して関数値の良い順にランク r_i , ($r_i = 1, 2, \dots, N$) を付与する。ここで、最良個体のランクは1である。グループ数を K とすると、個体 \mathbf{x}_i のグループ番号 $group(\mathbf{x}_i)$ は、以下のようになる。

$$group(\mathbf{x}_i) = \left\lceil \frac{r_i}{N} K \right\rceil \quad (5)$$

本研究では、 $K=2$ の場合について検証する..

JADE のパラメータ制御をグループ別に行うために、各グループ $k = 1, \dots, K$ に対して以下の式を用いる。

$$F_i \sim C(\mu_F^k, \sigma_F), \quad \mu_F^k = (1-c)\mu_F^k + cS_{F^2}^k/S_F^k, \quad (6)$$

$$CR_i \sim N(\mu_{CR}^k, \sigma_{CR}^2), \quad \mu_{CR}^k = (1-c)\mu_{CR}^k + cS_{CR}^k/S_N^k \quad (7)$$

ここで、 μ_F^k はグループ k に対する F の Cauchy 分布の位置パラメータ、 μ_{CR}^k はグループ k に対する CR の正規分布の平均である。 S_N^k はグループ k において親より良い子が生成された成功回数、 $S_F^k, S_{F^2}^k, S_{CR}^k$ はそれぞれグループ k において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。JADE と同様に、 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。

4.2 最悪個体に対するパラメータ制御

JADE の収束性を高めるために、最悪個体 \mathbf{x}_{worst} のパラメータ制御を以下のように修正する。

$$F_{worst} \sim u(0.9, 1.1), \quad CR_{worst} \sim u(\mu_{CR}^K, 1) \quad (8)$$

ただし、 $u(l, h)$ は区間 $[l, h]$ の一様分布である。 F を1程度にし、 CR を平均値より大きくすることにより、 $pbest$ 個体を中心にした大きな範囲で新しい個体を生成することができ、大域的な探索と最悪個体の高速な収束の実現が期待できる。

4.3 アルゴリズム

提案手法のアルゴリズムは以下の通りである。

Step0 パラメータの初期化

スケールングファクターの平均値 $\mu_F^k = 0.5$ 、交叉率の平均値 $\mu_{CR}^k = 0.5$ 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。ただし、 k はグループ番号である。

Step1 個体の初期化

初期集団 $P = \{\mathbf{x}_i \mid i = 1, 2, \dots, N\}$ を探索空間 \mathcal{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。
成功時のパラメータ値を保持するリスト S^k を空にする。

<pre> modified JADE/current-to-pbest/1/bin() { + $\mu_F^k = \mu_{CR}^k = 0.5$ ($1 \leq k \leq K$); + $\sigma_F = \sigma_{CR} = 0.1$; // Initialize a population + $P = N$ individuals generated randomly in S; + $FE = FE + N$; for($t=1$; $FE < FE_{max}$; $t++$) { + $S^k = \phi$ ($1 \leq k \leq K$); + sort P and obtain rank values r_i; for($i=1$; $i \leq N$; $i++$) { + $k = group(x_i)$; + if($r_i = N$) { // worst + $F_i = u(0.9, 1.1)$; + $CR_i = u(\mu_{CR}^k, 1)$; + } + else { + $CR_i = \mu_{CR}^k + N(0, \sigma_{CR}^2)$; + if($CR_i < 0$) $CR_i = 0$; + else if($CR_i > 1$) $CR_i = 1$; + do { + $F_i = \mu_F^k + C(0, \sigma_F)$; + } while($F_i \leq 0$); + if($F_i > 1$) $F_i = 1$; + } } </pre>	<pre> 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($f(x^{child}) < f(x_i)$) { + $z_i = x^{child}$; + $S^k = S^k \cup \{(F_i, CR_i)\}$; + // a success case is added to S^k + } else $z_i = x_i$; } $P = \{z_i\}$; + for($k=1$; $k \leq K$; $k++$) { + if($S^k > 0$) { + $\mu_F^k = (1 - c)\mu_F^k + c \sum_{F_i \in S^k} F_i^2 / \sum_{F_i \in S^k} F_i$; + $\mu_{CR}^k = (1 - c)\mu_{CR}^k + c \sum_{CR_i \in S^k} CR_i / S^k$; + } + } } </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

図 1: The pseudo-code of proposed method

Step3 グループ化

個体集団 $\{x_i\}$ を関数値についてソートし、ランク r_i を求め、グループ番号 $group(x_i)$ を決定する。

Step4 DE 操作

各個体 x_i について、交叉率 CR_i を正規分布 $N(\mu_{CR}^k, \sigma_{CR}^2)$ で生成する。スケールリングファクター F_i を $C(\mu_F^k, \sigma_F)$ に基づき Cauchy 分布で生成する。ここで、 k は個体 x_i のグループ番号 ($k = group(x_i)$) である。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 x^{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^k に追加する。成功でなければ、親 x_i を生存者とする。

Step5 パラメータの更新

全てのグループにおいて、 μ_F^k と μ_{CR}^k を S^k に基づいて更新する。

Step6 Step2 へ戻る。

提案手法の擬似コードを図 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 [20]。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。

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

表 1: D 次元テスト関数 (Sphere, Schwefel 2.22, Schwefel 1.2, Rastrigin, Ackley, and Griewank [20])

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

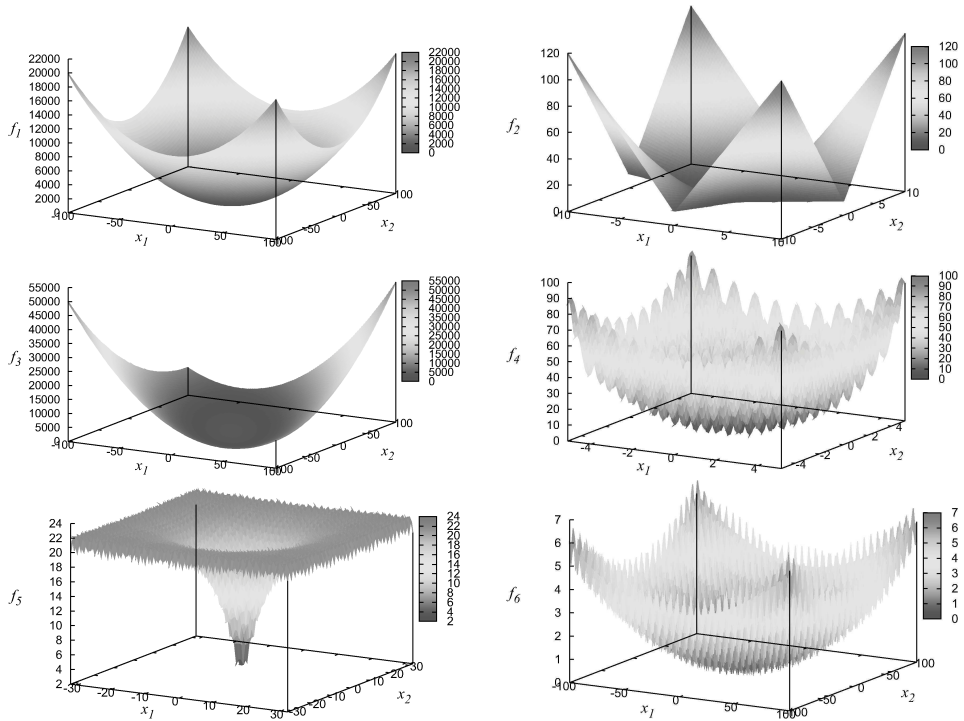
図 2: 関数 $f_1 \sim f_6$ のグラフ

表 2: 実験結果

	FE_{max}	JADE	Proposed (group)	Proposed (worst)	Proposed (group and worst)
f_1	150,000	9.379e-59 ± 6.53e-58	4.318e-66 ± 1.29e-65	6.928e-65 ± 2.87e-64	4.910e-68 ± 1.70e-67
		—	++	++	++
f_2	200,000	4.194e-31 ± 2.37e-30	5.103e-32 ± 2.73e-31	1.777e-41 ± 1.10e-40	3.549e-37 ± 1.31e-36
		—	=	++	++
f_3	500,000	8.171e-62 ± 3.01e-61	1.765e-59 ± 1.23e-58	1.088e-62 ± 3.61e-62	1.113e-62 ± 5.61e-62
		—	=	+	++
f_4	100,000	1.009e-04 ± 3.91e-05	5.643e-05 ± 2.76e-05	1.109e-04 ± 8.43e-05	5.680e-05 ± 2.79e-05
		—	++	=	++
	500,000	0.000e+00 ± 0.00e+00	0.000e+00 ± 0.00e+00	0.000e+00 ± 0.00e+00	0.000e+00 ± 0.00e+00
		—	=	=	=
f_5	50,000	9.199e-10 ± 6.43e-10	4.217e-10 ± 3.02e-10	1.133e-09 ± 8.53e-10	4.003e-10 ± 3.26e-10
		—	++	=	++
	200,000	2.665e-15 ± 0.00e+00	2.665e-15 ± 0.00e+00	2.665e-15 ± 0.00e+00	2.665e-15 ± 0.00e+00
		—	=	=	=
f_6	50,000	1.154e-08 ± 6.91e-08	1.973e-04 ± 1.38e-03	4.001e-15 ± 1.87e-14	2.457e-12 ± 1.24e-11
		—	-	+	+
	300,000	0.000e+00 ± 0.00e+00	1.973e-04 ± 1.38e-03	0.000e+00 ± 0.00e+00	0.000e+00 ± 0.00e+00
		—	=	=	=
+		—	3	4	6
=		—	2	2	0
-		—	1	0	0

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. JADE と共通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 $\mu_F^k = \mu_{CR}^k = 0.5$ ($k = 1, \dots, K$), パラメータ生成時の固定された尺度 $\sigma_F = 0.1$, 標準偏差 $\sigma_{CR} = 0.1$ とする. なお, グループ数 $K=2$ である. 各関数について 50 回の試行を行い, JADE の結果と比較を行う.

5.3 実験結果

JADE と提案手法の比較結果を表 2 に示す. 提案手法については, グループ化のみを用いた場合 (group), 最悪個体の制御のみを用いた場合 (worst), 両者を用いた場合 (group and worst) を示した. 各関数毎に最大評価回数 FE_{max} および各試行における最良値の平均値と標準偏差を示した. さらに, Wilcoxon signed rank test を行い, JADE に対して有意に優れていた場合に +, 有意に劣っていた場合に -, 有意差がない場合に = を付与した. なお, 有意水準 5% の場合は +, -, 有意水準 1% の場合は ++, -- で表現している. 全てのアルゴリズム中で最良の結果を太字で示した.

提案手法 (group and worst) は, 全ての関数において JADE と比較して優位に優れている. グループ化のみの場合は, f_1, f_4, f_5 について優れているが, f_6 は劣っている. 最悪個体のみの制御の場合は, f_1, f_2, f_3, f_6 について優れており, 劣った関数はない. したがって, グループ化と最悪個体の制御を共に導入することにより, 安定したアルゴリズムを実現できたと考えられる.

各問題における最良個体の関数値の平均値および μ_F^k と μ_{CR}^k の値の変化を図 3, 図 4 および図 5 に示す。横軸は関数評価回数である。縦軸については, 上段の図では関数値, 中段の図では μ_F^k の値, 下段の図では μ_{CR}^k の値である。

単峰性の関数 $f_1 \sim f_3$ では, JADE の μ_F の値と比較して, 優良なグループの μ_F^1 の値は小さく, 優良でないグループの μ_F^2 の値が大きくなる傾向がある。また, μ_{CR}^1 の値は小さく, μ_{CR}^2 の値は大きくなる傾向がある。多峰性の関数 $f_4 \sim f_6$ でも同様の傾向にある。したがって, グループ化によって JADE と異なるパラメータの制御が実現され, 探索効率の向上に繋がったと考えられる。

6 おわりに

本研究では, 個体集団を関数のランクに基づいてグループ化し, グループ毎に JADE のパラメータを制御する方法を提案した。さらに, 最悪個体に対する特別な制御方法を提案した。提案手法であるグループ別の制御と最悪個体の制御により, 6 つのベンチマーク関数全てにおいて, 本手法の基本モデルである JADE より優れた性能が実現できることを示した。

本論文では, グループ数が 2 の場合を検討したが, 適切なグループ数についてさらに検討する必要がある。現在, μ_F^k や μ_{CR}^k の初期値を 0.5 としているが, より適切な初期値の設定方法について検討する必要がある。また, 最良個体に対する特別な制御についても検討したいと考えている。

謝辞 この研究の一部は, 本研究は JSPS 科研費 26350443 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 Computing, 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).

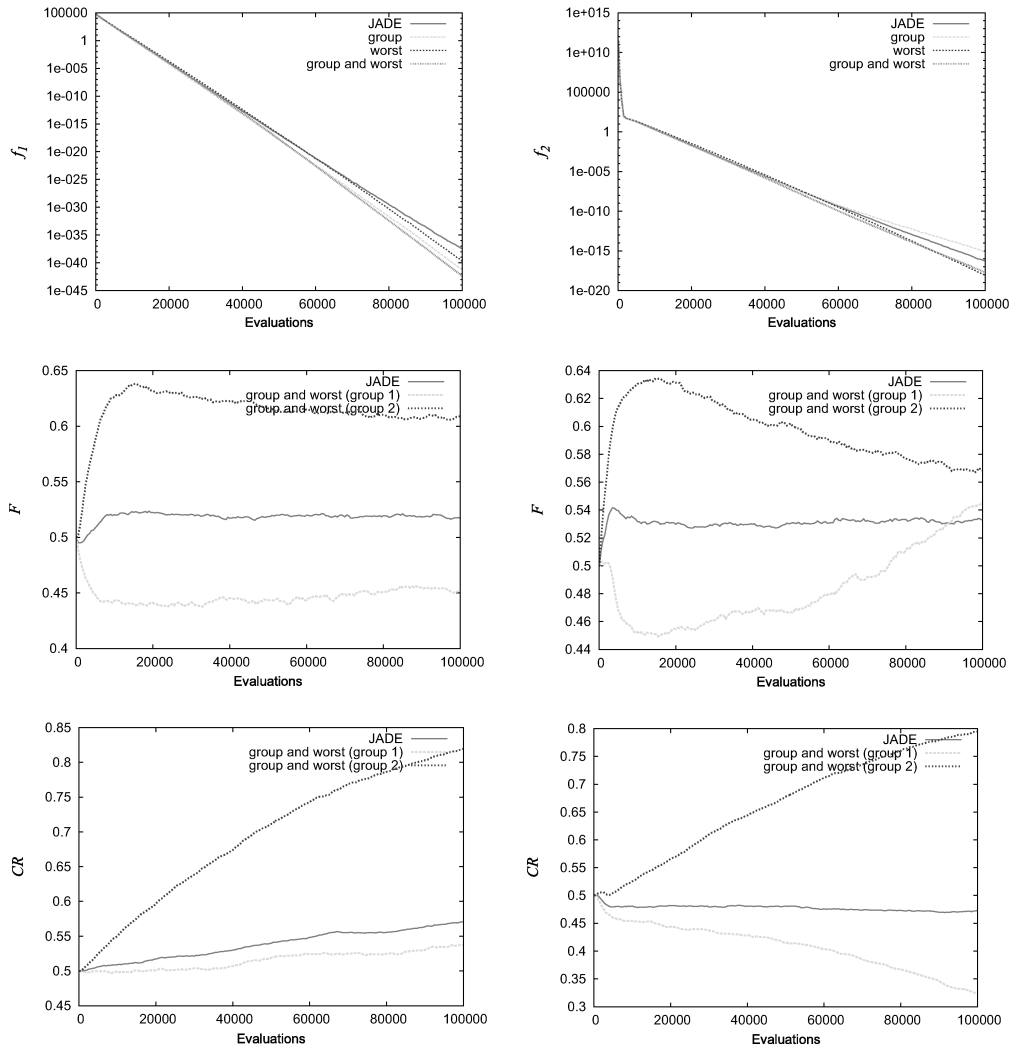


図 3: f_1 と f_2 における関数値, μ_F^k , μ_{CR}^k の変化

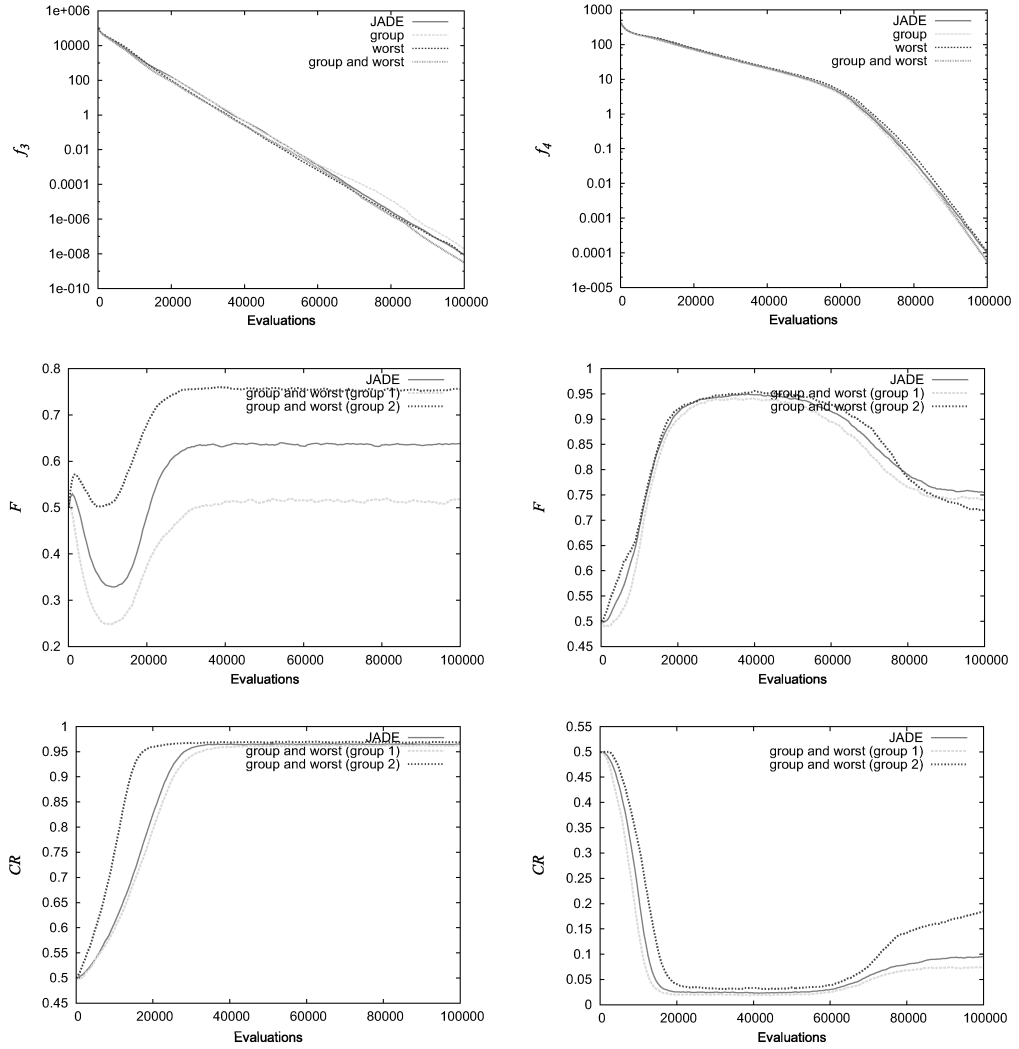


図 4: f_3 と f_4 における関数値, μ_F^t , μ_{CR}^t の変化

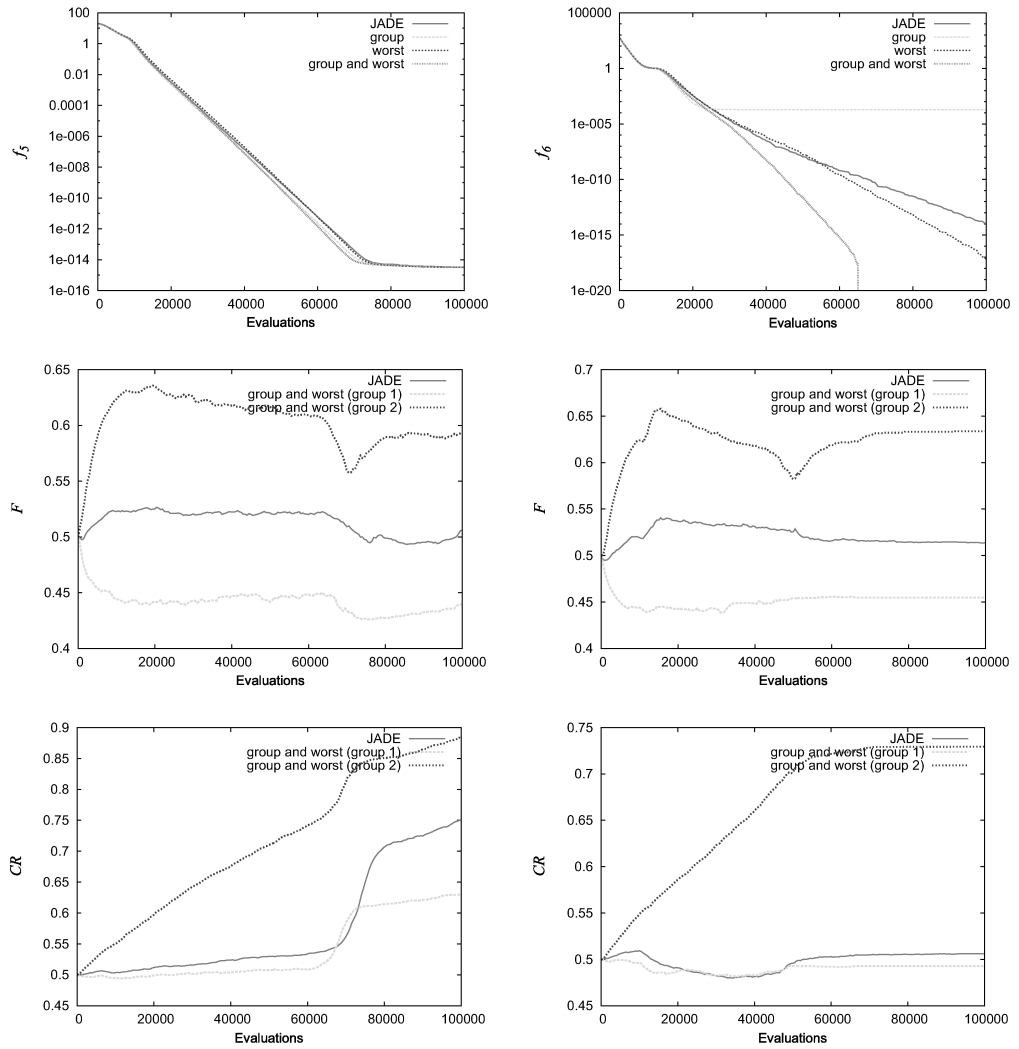


図 5: f_5 と f_6 における関数値, μ_F^k , μ_{CR}^k の変化

- [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 Computing, **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] R. Tanabe and A. Fukunaga: “Success-history based parameter adaptation for differential evolution”, Proc. of the 2013 IEEE Congress on Evolutionary Computation, pp. 71–78 (2013).
- [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), pp. 618–623 (2014).
- [19] 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).
- [20] 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).