

最良個体を用いた適応的差分進化による連続最適化問題の解法 An Algorithm for Continuous Optimization Problem using Adaptive Differential Evolution with the Best Particle

岩崎 丈徳[†] 穴田 一[†]
Takenori Iwasaki Hajime Anada

1. はじめに

ニューラルネットワークの学習やエネルギーシステムの構築、新幹線の形状設計など、多くの工学分野や産業分野の問題が連続最適化問題を解くことに帰着される。しかし、高次元や複雑な曲面、微分不可能な問題などほとんどの問題が現実的な時間内で厳密解を求めることが困難になってしまう。そのような問題に対する最適化アルゴリズムにメタヒューリスティクスと呼ばれる手法があり、近年研究が盛んに行われている。メタヒューリスティクスとは、最適性の保証はないが、多くの問題に適用できる汎用性を持ち、現実的な時間内で高速かつ高精度に解を求めることができる近似解法である。メタヒューリスティクスには、生物の進化や群れ行動を模倣した進化計算 (Evolutionary Computation, EC) と呼ばれる近似解法の枠組みが存在する。その EC の 1 つに差分進化 (Differential Evolution, DE) [1] と呼ばれる手法があるが、DE の探索性能はハイパラメータの設定に依存してしまう。そのため、パラメータ調整を適応化した適応的差分進化が提案されているが、集団内の良い情報を有効活用しないまま収束してしまう。

そこで本研究では、適応的差分進化の中でも J. Zhang らによって提案された JADE[2] と呼ばれる代表的な手法をベースとして、変異ベクトル生成時に差分ベクトルにより最良個体を変異させた適応的差分進化を提案し、その有効性を確認した。

2. 差分進化 (DE)

差分進化 (Differential Evolution, DE) [1] とは、1997 年に R. Storn らによって提案された EC で、解を解空間上の個体で表現する。ターゲットベクトルと呼ばれる現世代の各個体に対して基本ベクトルと呼ばれる基準となる親個体を個体群からランダムに選択する。それらの基本ベクトルをランダムに選択した個体との差分ベクトルにより変異させることで変異ベクトルを生成し、その変異ベクトルと現世代の個体 (ターゲットベクトル) を交叉させることでトライアルベクトルと呼ばれる子個体を生成する。最後にトライアルベクトルと現世代の個体の評価値を比較し、トライアルベクトルが現世代の個体より良ければ、次世代の個体へと更新する。以下に DE の基本的な手順を示す。

STEP1 初期個体群の生成

n 次元の実数値ベクトルで表される初期個体 $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ をランダムに m 個生成する。具体的には、個体 x_i の j 次元目の決定変数 x_{ij} を一様乱数により生成する。

STEP2 変異ベクトルの生成 (突然変異)

ターゲットベクトルと呼ばれる現世代の個体 x_i において、基本ベクトルと呼ばれる個体 x_{r1} を個体群からランダムに選択する。次に差分ベクトルの生成のために個体群からランダムに個体 x_{r2}, x_{r3} を選択する。ただし、 $x_i, x_{r1}, x_{r2}, x_{r3}$ は互いに異なるように選択する ($x_i \neq x_{r1} \neq x_{r2} \neq x_{r3}$)。基本ベクトル x_{r1} と差分ベクトル $x_{r2} - x_{r3}$ に基づき変異ベクトル v_i を以下の式で生成する。

$$v_i = x_{r1} + F(x_{r2} - x_{r3}) \quad (1)$$

ここで、 F はスケーリングファクタで $[0, 1]$ の値をとり、局所的な探索と大域的な探索のバランスを調整するパラメータである。

STEP3 トライアルベクトルの生成 (交叉)

STEP2 で生成した変異ベクトル v_i と現世代の個体 x_i を一様交叉し、トライアルベクトルと呼ばれる子個体 u_i を生成する。ここで用いる交叉方法として、binomial 交叉を用いる。以下に binomial 交叉の式を示す。

$$u_{ij} = \begin{cases} v_{ij}, & \text{if } \text{rand}(0, 1) \leq CR \text{ or } j = j_{\text{rand}} \\ x_{ij}, & \text{otherwise} \end{cases} \quad (2)$$

ここで、 CR は交叉率で $[0, 1]$ の値をとり、変異ベクトルの影響度合いを調整するパラメータである。また、 j_{rand} は $1 \sim n$ の範囲内で生成した整数乱数で、変異ベクトルの j_{rand} 次元目の値は必ず交叉させる。

STEP4 トライアルベクトルとの比較 (生存選択)

STEP3 で生成したトライアルベクトル u_i と現世代の個体 x_i の評価値を比較し、トライアルベクトルが現世代の個体より良ければトライアルベクトルを次世代の個体として置き換える。以下がその更新式である。

$$x_i(t+1) = \begin{cases} u_i(t), & \text{if } f(u_i) < f(x_i) \\ x_i(t), & \text{otherwise} \end{cases} \quad (3)$$

以上の STEP2~STEP4 を定められた世代数繰り返す。

ここで、本研究のベースとなる JADE[2] という手法は、基本的な DE のアルゴリズムから 3 つの変更点がある。以下にその変更点を示す。

変更点①: F, CR を確率分布に基づき生成

スケーリングファクタ F と交叉率 CR を各個体 x_i に対して、以下の確率分布に基づき生成する。

$$F_i \sim \text{Cauchy}(\mu_F, \sigma_F) \quad (4)$$

$$CR_i \sim \text{Norm}(\mu_{CR}, \sigma_{CR}) \quad (5)$$

[†] 東京都市大学大学院
Tokyo City University Graduate School

ここで, F_i, CR_i は個体*i*のスケーリングファクタと交叉率, μ_F, μ_{CR} はスケーリングファクタと交叉率の平均値, σ_F, σ_{CR} はスケーリングファクタと交叉率の標準偏差である.

変更点②: μ_F, μ_{CR} の更新

上記の変更点①における μ_F, μ_{CR} を以下の更新式で世代毎に更新する.

$$\mu_F(t+1) = (1-c)\mu_F(t) + cS_{F^2}/S_F \quad (6)$$

$$\mu_{CR}(t+1) = (1-c)\mu_{CR}(t) + cS_{CR}/S_N \quad (7)$$

ここで, S_F, S_{F^2}, S_{CR} はそれぞれ式(3)における更新成功時の F_i, F_i^2, CR_i の和, S_N は更新に成功した回数, c は[0,1]の学習率である.

変更点③: 変異ベクトルの生成方法

変異ベクトル v_i の生成方法を式(1)から以下に示す式に変更する.

$$v_i = x_i + F_i(x_{pbest} - x_i) + F_i(x_{r2} - x_{r3}) \quad (8)$$

ここで, x_{pbest} は上位*p*%からランダムに選択した個体である.

3. 提案手法

J. Zhang らによって提案された JADE [2]の課題として, 集団内の良い情報を有効活用しないまま収束してしまう.

そこで本研究では, 変異ベクトル生成時に差分ベクトルにより最良個体を変異させた JADE を提案した. 以下に提案手法のアルゴリズムの手順を示す.

STEP1 初期個体群の生成と初期設定

n 次元の実数値ベクトルで表される初期個体 $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ をランダムに*m*個生成する. ここで, 各個体*i*のこれまでの最良解 $pbest_i$ に現在の解 x_i を設定し, それらの最良解 $gbest$ を $pbest_i$ の中から決定する.

STEP2 変異ベクトルの生成 (突然変異)

ターゲットベクトル x_i において, 差分ベクトル生成のためにランダムに個体 $x_{r1} \sim x_{r6}$ を互いに異なるように選択する. 選択した個体 $x_{r1} \sim x_{r6}$ に基づき 3つの差分ベクトルを生成し, $x_i, pbest_i, gbest$ を以下のような式で変異させ, $x'_i, pbest'_i, gbest'$ を生成する.

$$x'_i = x_i + F_i(x_{r1} - x_{r2}) \quad (9)$$

$$pbest'_i = pbest_i + F_i(x_{r3} - x_{r4}) \quad (10)$$

$$gbest' = gbest + F_i(x_{r5} - x_{r6}) \quad (11)$$

ここで, F_i は個体*i*のスケーリングファクタで式(4)のようにコーシー分布で生成される. そして, $x'_i, pbest'_i, gbest'$ に基づき以下のような式で変異ベクトル v_i を生成する.

$$v_i = F_i \times x'_i + F_i \times pbest'_i + F_i \times gbest' \quad (12)$$

STEP3 トライアルベクトルの生成 (交叉)

STEP2 で生成した変異ベクトル v_i と現世代の個体 x_i を一樣交叉し, トライアルベクトルと呼ばれる子個体 u_i を生成する. 交叉方法として, 式(2)で示した binomial 交叉を用いた. ここで, 交叉率 CR は式(5)のように各個体 x_i に対して, 正規分布で生成される.

STEP4 トライアルベクトルとの比較 (生存選択)

STEP3 で生成したトライアルベクトル u_i と現世代の個体 x_i の評価値を比較し, トライアルベクトルが現世代の個体より良ければトライアルベクトルを次世代の個体として式(3)のように置き換える.

STEP5 μ_F, μ_{CR} の更新

μ_F, μ_{CR} を式(6), (7)によって更新する.

STEP6 $pbest_i, gbest$ の更新

全個体の $pbest_i, gbest$ を更新する.

以上の STEP2~STEP6 を定められた世代数繰り返す.

4. 評価実験

本研究では, 提案手法の有効性を評価するために [3]で定義されているベンチマーク関数 (F_1 : 単峰性, F_2, F_3 : 多峰性, F_4 : 変数間依存性, F_5 : 悪スケール性) を用いて既存手法との性能比較を実施した. 提案手法のパラメータは予備実験により最も良かった $\mu_F(0) = 0.5, \mu_{CR}(0) = 0.5, c = 0.1$ と設定した. 終了世代数は 1000 step とし, 各手法を 50 回試行平均した際の既存手法との性能比較を表 1 に示す.

表 1 既存手法との性能比較

Function	Dim	JADE		Proposed	
		ave	std	ave	std
F_1	100	2.28e-02	3.54e-02	8.00e-121	5.59e-120
	500	2.30e+02	3.72e+01	1.85e-104	1.05e-103
	1000	9.02e+02	1.08e+02	1.44e-96	9.89e-96
F_2	100	7.00e+00	1.75e+00	0.00e+00	0.00e+00
	500	1.56e+01	5.60e-01	0.00e+00	0.00e+00
	1000	1.69e+01	3.69e-01	0.00e+00	0.00e+00
F_3	100	1.62e+02	1.38e+01	0.00e+00	0.00e+00
	500	3.23e+03	2.82e+02	0.00e+00	0.00e+00
	1000	7.26e+03	1.04e+03	0.00e+00	0.00e+00
F_4	100	4.08e+02	2.45e+02	9.80e+01	6.12e-01
	500	5.58e+04	1.38e+04	4.98e+02	2.27e-01
	1000	2.72e+05	4.79e+04	9.98e+02	1.38e-01
F_5	100	1.99e+02	8.36e+02	6.64e-109	2.79e-108
	500	1.16e+06	1.89e+05	1.81e-97	1.26e-96
	1000	5.75e+06	6.53e+05	2.11e-98	1.10e-97

上記の結果より, 提案手法は既存手法より解精度が向上したことが確認でき, 単峰性, 多峰性, 変数間依存性, 悪スケール性の全ての関数に対して高精度な解を求めることができた. 特に, F_2, F_3 の多峰性を持つ関数に対しては厳密解を求めることができ, 次元数が 100, 500, 1000 と高くなっても精度が保持されており, 多峰性の関数においては高次元にも対応できる.

参考文献

- [1] R. Storn, K. Price, "Differential Evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces", J. Global Optimiz., Vol.11, No.4, pp.341-359 (1997).
- [2] J. Zhang, A. C. Sanderson, "JADE: Adaptive Differential Evolution With Optional External Archive", IEEE Tran. Evol. Comput., Vol.13, No.5, pp.945-958 (2009).
- [3] 小林重信, "実数値 GA のフロンティア", 人工知能学会誌, Vol.24, No.1, pp.147-162 (2009).