

種を用いた Fitness Sharing Differential Evolution の提案

Proposal of Fitness Sharing Differential Evolution Using Species.

芦田 高志†
Takashi Ashida

ラック ターウォンマツ†
Ruck Thawonmas

1. はじめに

当研究室ではゲーム用 AI の最適化を行っている。ゲーム用 AI の最適化においては大きく 2 つの問題点が存在する。1 つは”評価に時間がかかる”と言う点。もう 1 つは”評価が安定しない”という点である。どちらにも共通する理由として、近年のゲームにおいては同じ行動をしたとしても結果が同一であると言うことがほとんど起こり得ないという点があげられる。その最大の原因はプレイヤーの操作するキャラクター以外のキャラクター (NPC) などにランダムな動きが含まれているからである。再現性が低いために、正確な評価をするためには複数回数の試行が必要であり、また一度の試行に関してもゲームのプレイ時間が必要であるために時間がかかる。

本論文ではこの 2 つの問題点のうち、前者について解決を試みる。個体の数を減らせば評価回数が減り、最適化にかかる時間を短縮することにつながると考える。そのため、少ない個体数でも従来と同程度の世代数で最適化することのできる手法を提案する。

2. Differential Evolution

Differential Evolution (DE) は, Storn & Price により提案された, 進化的戦略の一つである [1]。DE は非線形問題, 微分不可能な問題, 非凸問題, 多峰性問題などのさまざまな最適化問題に適応され, それらに対し高速で頑強なアルゴリズムであることが示されている。基本的な DE のアルゴリズムを図 1 に示す。

1. 探索空間内にランダムで初期個体を生成し, 初期集団を構築する。
2. 各個体を親とし, 集団から重複しないよう 3 個の個体を選択する。
3. 1 つを基本ベクトルとなる親 a とし, 残りの個体で差分ベクトル $(b - c)$ を求める。
4. 差分ベクトルにスケールファクター F を乗算し, 基本ベクトルに加える。 $a + F(b - c)$
5. 4. のベクトルと親を交叉させ子を生成する。
6. 親よりも子の評価が高ければ親を子で置換する。
7. 2. から 6. を繰り返し, 終了条件を満たすと終了する。

図 1. DE のアルゴリズム

DE における交差には exp 交差を利用している。exp 交差は各遺伝子の交差確率が交差を経る毎に指数関数的に下がっていく交差法である。exp 交差のアルゴリズムを図 2 に示す。 r を遺伝子毎に生成するので, 指数関数的に下がることとなる。

1. 遺伝子を一つランダムで選択する。
2. 乱数 r を生成し, 交差確率 CR と比較し $r < CR$ の時に遺伝子を交差する。
3. 遺伝子を一つずらす。
4. 2. から 3. を繰り返し, 途中で $r > CR$ となるか, 全ての遺伝子が交差されたら終了する。

図 2. exp 交差のアルゴリズム

3. Fitness Sharing Differential Evolution

Neuro Evolution of Augmenting Topologies (NEAT) [2] で用いられている種 の概念を参考とし, 評価の共有を行っている。NEAT は Neural Network (NN) の進化手法である。NEAT では遺伝子長は可変長である。しかし, DE は遺伝子長が固定であるので, NEAT で使われている概念をそのまま利用することはできない。そのため, DE でも利用することができるように変更している。種 の決定と評価値の共有を DE に加えた Fitness Sharing Differential Evolution を提案する。

3.1 種 の決定

探索空間内で一定距離以内の個体群を 1 つの種としてあつかう。各世代において, 前世代の個体群の種をベースとしたユークリッド距離を用いて各個体の種を決定する。また, 初期世代では前世代を参考にすることができないので 1 個体をランダムで選択し, その個体を中心とした種と考え, 残りの個体の種を決定する。

ユークリッド距離を用いて種を決定する際には 閾値 を用いる。判別したい個体とある種に属する全ての個体との距離が全て閾値以下の場合に判別個体はその種に属すると判断する。もし, どの種にも属さなかった場合, 新たな種として判断する。

3.2 共有評価値

通常, 新たな個体ができたと際にその個体の評価が低いと淘汰される。しかし, その個体が探索されていない領域の個体であったりした場合, 一時的に低い評価がでたとしても, その後の進化により評価が高くなるという可能性がある。そういった個体の可能性を守るために種 の保護という概念を導入する。これは各個体の評価値を属する種 の個体数に反比例となるように減少させることで, 規模の小さな種に属する個体の評価が高くなるようにする。減少させた評価値を共有評価値と呼ぶ。各個体の共有評価値には式 (1) を用いる。

$$f_i' = \frac{f_i}{Pop(Specie(i))} \quad (1)$$

ここで $Pop(a)$ は種 a に属する個体数を, $Specie(n)$ は個体 n

† 立命館大学大学院, 修士生. Ritsumeikan University graduate school master student.

‡ 立命館大学知能情報学科, 教授. Professor Ritsumeikan

の属する種を示す.つまり,共有評価値は元の評価値を個体の属する種の個体数で割った値となっている.

3.3 Fitness Sharing Differential Evolution

Fitness Sharing Differential Evolution (FSDE) は従来の DE に種の決定と評価値の共有とを加えた最適化問題解決手法である.基本的なアルゴリズムは従来の DE と同様である.異なるのは,各個体に種を設定し,個体が生成された際に種が決定され,属する種の規模に応じ共有評価値が設定されることである.親個体と子個体を比較する際に共有評価値が用いられ,新しい種,規模の小さな種に属する個体が生成された場合にそちらの個体が優先され,同一の種に属する場合はより評価の良い個体が選択されるといったものになっている.基本的な FSDE のアルゴリズムを図 3. に示す.

1. 探索空間内にランダムで初期個体を生成し, 初期集団を構築し,各個体の種の決定,共有評価値の計算を行う.
2. 各個体を親とし, 集団から重複しないよう 3 個の個体を選択する.
3. 1 つを基本ベクトルとなる親 a とし, 残りの個体で差分ベクトル($b - c$)を求める.
4. 差分ベクトルにスケールファクター F を乗算し, 基本ベクトルに加える. $a + F(b - c)$
5. 4. で生成されたベクトルと親を交叉し子を生成する.
6. 生成した子の種を決定し,共有評価値を計算する.
7. 親よりも子の共有評価値を比較し,評価の良い個体を選択する.
8. 2.から 6.を繰り返し,終了条件を満たすと終了する.

図 3. FSDE のアルゴリズム

4. 実験方法

Sphere 関数,Rosen Brock 関数,Rastrigin 関数の解を求める実験を行い,DE と FSDE の性能比較を行う.各関数の式は以下のとおりである.式の後に表示している範囲は各関数におけるベクトルの取りうる範囲である.

• Sphere

$$f(x) = \sum_{i=0}^{D-1} x_i^2, \text{ with } -3.14 \leq x \leq 3.14$$

• Rosen Brock (RB)

$$f(x) = \sum_{i=0}^{D-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$$

with $-2.048 \leq x \leq 2.048$

• Rastrigin

$$f(x) = 10D + \sum_{i=0}^{D-1} [x_i^2 - 10 \cos(2\pi x_i)]$$

with $-5.12 \leq x \leq 5.12$

上記関数の 10 次元問題を利用する.個体数 5,10,30 で世代数 300 の実験を 50 回ずつ試行する.それぞれで共通しているパラメータはすべて同じ値を用いている.それぞれの実験において個体数以外のパラメータ (交差確率 $CR=0.75$, スケールファクター $F=0.7$) は統一しており,FSDE における種の決定に利用する閾値も全て同じ値で実行している.

5. 実験結果・考察

個体数 5 での DE,FSDE での各関数の解を求めたときの 50 試行の中央値,最良値を表 1 に示す.同様に個体数 10,30 での結果を表 2,3 に示す.

表 1 個体数 5 の時の結果

	FSDE		DE	
	中央値	最良値	中央値	最良値
Sphere	7.76E-2	1.09E-6	3.79E-1	2.35E-6
RB	3.83E-0	4.45E-1	1.18E+1	7.93E-2
Rastrigin	4.83E-0	2.89E-0	1.30E+1	2.02E+1

表 2 個体数 10 の時の結果

	FSDE		DE	
	中央値	最良値	中央値	最良値
Sphere	5.18E-8	6.85E-10	5.09E-7	2.33E-8
RB	5.97E-1	3.43E-1	5.35E-0	1.77E-0
Rastrigin	1.69E-0	1.30E-0	1.00E+1	1.44E+1

表 3 個体数 30 の時の結果

	FSDE		DE	
	中央値	最良値	中央値	最良値
Sphere	2.54E-8	5.33E-9	8.56E-7	2.02E-7
RB	1.35E-1	6.87E-2	3.87E-0	2.93E-0
Rastrigin	5.43E-1	4.27E-1	1.59E+1	1.15E+1

これらの結果から,個体数にかかわらず FSDE は DE よりも良い結果を得やすいといえる.個体数 5 の時の Rosen Brock 関数においてのみ DE の結果が優れているが,中央値では FSDE の結果が優れている.ここに表記はしていないが 50 試行の結果を見たところ,個体数 5 の DE での結果はばらつきが激しい. FSDE の結果にもばらつきはあるものの,中央値が低いために,全体的には良い結果が得られている回数が DE よりも多かった.

以上のことより,本論文での提案手法である Fitness Sharing Differential Evolution は従来手法の Differential Evolution よりも優れた結果を得ることができ,個体数を減らした際も従来よりも良い結果を得ることができると言える.

参考文献

- [1] Storn, R.; Price, K. (1997). "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces". Journal of Global Optimization 11: 341–359.
- [2] Stanley, K. O., and Miikkulainen, R. (2001). "Evolving neural networks through augmenting topologies." Technical Report AI2001-290, Department of Computer Sciences, The University of Texas at Austin.
- [3] X. Li. (2005) "Efficient differential evolution using speciation for multimodal function optimization." GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, 873–880.
- [4] 芦田高志, Ruck Thawonmas (2010) "Ms. Pac-Man 自動制御プログラムの最適化における進化戦略の拡張" ゲーム学会第 9 回全国大会論文集, 21-24.