

分割と協調を導入した差分進化アルゴリズム

丹羽貴敏[†] 伊原滉也^{†‡} 加藤昇平^{†‡‡}

[†]名古屋工業大学 大学院工学研究科 情報工学専攻

[‡]名古屋工業大学 情報科学フロンティア研究院

^{††}名古屋工業大学 大学院工学研究科 工学専攻 情報工学系プログラム

1 はじめに

共進化 (Cooperative Coevolution, CC) [1] とは問題を複数の部分問題に分割し部分問題を個別に進化計算 (Evolutionary Computation, EC) を用いて最適化することで全体を最適化する手法である. CCの流れを以下に示す. 加えて図1に分割数が4の場合の例を示す.

- (1) 問題を複数の部分問題へと分割する.
- (2) 部分問題に対応する個体に対して EC によって進化させる.
- (3) 個体を組合せて適応度を計算し, 個体の適応度を割り当てる.
- (4) 終了条件を満たすまで (2), (3) を繰り返す.

(2) の EC に差分進化を用いたものを Cooperative Coevolutionary Differential Evolution (CCDE) と呼ぶ. (3) における個体の組合せ方と適応度の割り当て方を Collaboration Model と呼ぶ. 初期の CC では今までで最も適応度が優れていた個体の組合せと組合せて個体の適応度を求める. そのため変数間に依存関係のある目的関数を適切に分割できない場合, 探索に悪影響を及ぼす. しかし事前に変数の依存関係を分析し, 目的関数を分割するのは困難である. そこで本稿では変数間に依存関係のある関数において有効な個体の評価方法を提案する.

2 評価方法を切り替える共進化型差分進化

2.1 グループワークを用いた Collaboration Model[2]

グループワークではグループに与えられた仕事を分割し, メンバーがそれぞれ分担して行う. メンバーは単独で評価されるのではなくグループの成果と個人の貢献度から総合的に決定される. この評価方法を Collaboration Model に用いる. 初めに各領域から個体を1つずつ取り出しグループを作成する. 1個体は1グループにのみ参加し, 複数のグループを掛け持つことはない. 1領域内の個体数を N_s とすると一度のグループ

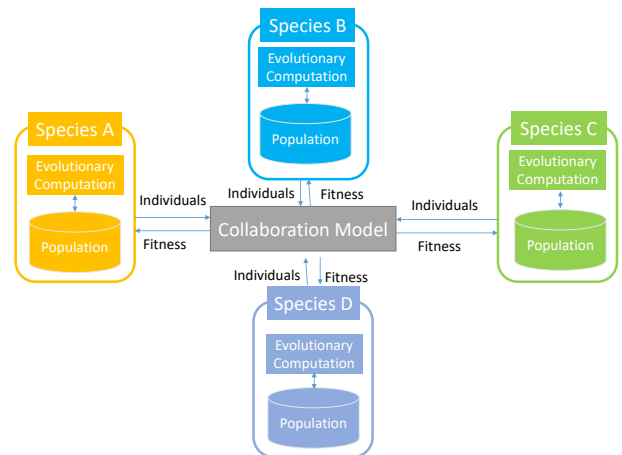


図1: Cooperative Coevolution (分割数4の場合)

ワーク編成で N_s 個のグループが作成される. グループの適応度を求め, グループに属したすべての個体に対して適応度を割り当てる.

2.2 Group Working Differential Evolution

まず CC の Collaboration Model にグループワークの評価方法を用いた Group Working Differential Evolution (GWDE) を提案する. GWDE の流れを Algorithm 1 に示す. ステップ 10 では個体をランダムに組合せグループを生成する. ステップ 12 ではグループの適応度をグループに属するすべての個体に対して同一の適応度を与える. また複数の Collaboration Model を使用して個体評価を行う Cooperative Coevolutionary Differential Evolution Switching Collaboration Model (CCDESCM) を提案する. CCDESCM では始めに従来の CC の評価方法で個体評価を行い, その後数世代ごとに従来の CC の評価方法とグループワークの評価方法を切り替えて個体を評価する.

3 性能比較実験

3.1 実験設定

GWDE, CCDESCM, CCDE の探索性能を比較する. ベンチマーク関数は CEC2013 の 28 関数 [3] を用いる. 以下にベンチマーク関数の構成を示す.

- 単峰性関数 $f_1 \sim f_5$
- 多峰性関数 $f_6 \sim f_{20}$
- 合成関数 $f_{21} \sim f_{28}$

28 関数のうち, f_1, f_5, f_{11}, f_{22} の 4 関数は分離可能な関数であり, 残りの 24 関数は分離不可能な関数である. ベンチマークの次元数は 20 次元で実験を行う. 決定変数の分割数は 2, 4, 5, 10, 20 分割で行い, 最良

Differential Evolution Introducing Problem Decomposition and Cooperation

Takatoshi NIWA[†], Koya IHARA[†], Shohei KATO^{†‡‡}

[†]Dept. of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

[‡]Frontier Research Institute for Information Science, Nagoya Institute of Technology

^{††}Computer Science Program, Dept. of Engineering, Graduate School of Engineering, Nagoya Institute of Technology

Algorithm 1 Group Working Differential Evolution

```

1:  $s \cdots$  the number of division
2:
3:  $gen = 0$ 
4: divide the decision variable vector
5: for each species  $s$  do
6:    $Pop_s(gen)$  =randomly initialized population
7: end for
8: while termination condition not met do
9:    $gen = gen + 1$ 
10:  create groups by combining individuals
11:  calculate group fitness
12:  assign fitness to all individuals belonging to the
    group
13:  for each species  $s$  do
14:    Apply the mutation, crossover, and selection
    operation of DE
15:  end for
16: end while

```

の探索性能となった分割数で比較する。最適解 x^* の目的関数値 $f(x^*)$ と探索中に求めた最良解 x^{best} の目的関数値 $f(x^{best})$ の誤差値 $|f(x^*) - f(x^{best})|$ が 1.0×10^{-8} 以下となった場合最適解到達とし最適解到達または関数評価回数が 2.0×10^6 以上になった場合に探索を終了する。個体数 $N_p = 20$ 、スケールングファクター $F = 0.5$ 、交叉率 $CR = 0.8$ とする。各手法ごと 30 試行し、ウィルコクソンの順位和検定によって有意差を検証する。有意水準 $P = 0.05$ とする。CCDESCM における Collaboration Model を変更する周期を 50 世代とする。

3.2 実験結果

表 1 に各手法の最適解との誤差値を示す。GWDE と CCDE の比較で優位であったものを*, CCDE と CCDESCM の比較で優位であったものを°, GWDE と CCDESCM の比較で優位であったものを●で示す。GWDE の方が優位であった関数は 13 関数, CCDE が優位であった関数は 13 関数, 有意差が確認されなかった関数は 2 関数であった。関数によって有効な手法が異なる結果になった。一方 CCDESCM と CCDE を比較すると CCDESCM の方が優位であった関数は 9 関数, CCDE が優位であった関数は 1 関数, 有意差が確認されなかった関数は 18 関数であった。このことから従来の評価方法のみで探索するのではなくグループワークの評価方法と切り替えることが探索に有効であることが確認された。CCDESCM と GWDE を比較すると CCDESCM の方が優位であった関数は 13 関数, GWDE が優位であった関数は 12 関数, 有意差が確認されなかった関数は 3 関数であった。GWDE が優位であった 12 関数は GWDE と CCDE の比較の際に GWDE が優位であった関数と同一であった。このことからグループワークの評価方法のみまたは従来の評価方法とグループワークの評価方法を併用すること

表 1: 最適解との誤差値

	GWDE	CCDE	CCDESCM
f_1	0.00.E+00*•	0.00.E+00	0.00.E+00
f_2	2.09.E+04*	3.66.E+04	2.76.E+04
f_3	3.58.E+05*•	4.26.E+07	1.39.E+07
f_4	4.30.E-05*•	1.03.E+03	2.59.E+01°
f_5	1.00.E-07*•	1.00.E-07°	1.00.E-07
f_6	2.15.E-04*•	2.03.E+01	1.26.E+01
f_7	1.16.E-01*•	1.72.E+01	1.24.E+01°
f_8	2.06.E+01*•	2.07.E+01	2.07.E+01°
f_9	4.86.E+00*•	1.01.E+01	8.45.E+00°
f_{10}	1.04.E-01*•	4.04.E+00	3.52.E+00°
f_{11}	4.26.E+00	0.00.E+00*	1.00.E-07•
f_{12}	7.52.E+01	3.94.E+01*	3.06.E+01°•
f_{13}	7.63.E+01	7.20.E+01*	6.33.E+01°•
f_{14}	3.39.E+02	1.56.E+00*	1.06.E+00•
f_{15}	3.55.E+03	1.97.E+03*	2.10.E+03•
f_{16}	1.40.E+00	1.18.E+00*	1.15.E+00•
f_{17}	4.10.E+01	2.03.E+01*	2.03.E+01•
f_{18}	3.11.E+01	2.02.E+01*	2.02.E+01•
f_{19}	1.79.E+00	6.35.E-01*	6.60.E-01•
f_{20}	5.72.E+00*•	6.23.E+00	5.97.E+00
f_{21}	3.13.E+02	3.21.E+02	3.13.E+02
f_{22}	4.74.E+02	1.15.E+02*	1.65.E+02•
f_{23}	3.94.E+03	2.79.E+03*	2.63.E+03•
f_{24}	2.41.E+02	2.29.E+02*	2.24.E+02°•
f_{25}	2.50.E+02	2.31.E+02*	2.27.E+02•
f_{26}	2.00.E+02*•	2.68.E+02	2.18.E+02°
f_{27}	4.33.E+02*•	5.51.E+02	5.37.E+02
f_{28}	3.58.E+02	6.67.E+02	7.46.E+02

で探索性能が向上できる。

4 おわりに

本稿では CCDE の Collaboration Model にグループワークを用いた GWDE を提案した。ベンチマーク関数を用いた性能評価実験を行い、探索性能の向上を確認した。本実験では 20 次元にて実験を行ったが今後は 1000 次元などの高次元における提案手法の有効性を検証していく。

参考文献

- [1] Potter, M. A. and De Jong, K. A.: A Cooperative Coevolutionary Approach to Function Optimization, in *International Conference on Parallel Problem Solving from Nature*, pp. 249–257Springer (1994).
- [2] Niwa, T., Ihara, K. and Kato, S.: A group work inspired generation alternation model of real-coded GA, in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 302–303, Springer (2019).
- [3] Liang, J., Qu, B., Suganthan, P. and Hernández-Díaz, A. G.: Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization, *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, Vol. 201212, pp. 3–18 (2013).