

Map-Elites アルゴリズムを用いた GE による解探索 Solution Search by GE using Map-Elites Algorithm

田村 謙次[†]
Kenji Tamura

1. はじめに

一般的な進化計算は適用問題における適応度の優劣に応じて個体選択を行い解探索を行うが、大域的な探索と局所的な探索とのバランスが探索性能に寄与することから、それを実現するためのさまざまな手法が提案されてきた。そのような背景において、適応度だけではなく解の新規性、すなわち探索された解の履歴から逸脱した解や履歴を元に生成される解を予測し、その予測解からの逸脱した個体を選択することにより、探索領域を広めながら解探索を行う Novelty Search や Surprise Search などが提案され、盛んに研究されている。

MAP-Elites は、NS+LC や MOLE に代表される Illumination Algorithm の一手法で、分割された特徴空間上における最適解を保持することにより多様性の維持と解探索を行うことができる手法で、単目的の進化計算と異なり適応度が低い個体でも次世代に生存することが可能で、その個体が飛び石となり、新たな領域を探索することが可能で近年注目を集めている。

本研究では、GE における進化的操作を MAP-Elites で行うことにより、従来 GE よりも効果的な解探索が行えることを報告する。

2. 研究背景

2.1 Map-Elites

MAP-Elites(Multi-dimensional Archive of Phenotypic Elites)[1, 2]は、Novelty Search + Local Competition (NS+LC)[3]や Multi-Objective Landscape Exploration algorithm (MOLE)[4]に代表される Illumination Algorithm の一手法で、解の性能と着目する特徴の間のトレードオフを含めて、特徴空間の各領域の解を探索することで多様性を維持しながら解空間における最高性能の解を見つけようとするアルゴリズムであり、MAP-Elites は NS+LC や MOLE と比較してアルゴリズムがシンプルであることや実験的な比較においては高い性能が得られることが報告されている。また、NS は必要な計算量やメモリ量が多く、MOLE は解探索が不安定であることなどの問題が指摘されている。

進化計算はビルディングブロックやステップングストーン[6]を利用することにより効果的な解探索が行われているとされるが、文献[5]では関連性のある多数の飛び石を同時に探索する方が、個々の飛び石に到達するための方法としてははるかに優れている可能性があることや、ある特徴空間内の解をより遠くの特徴空間の解に変異させることで、その特徴空間の解を生成する可能性が高くなる場合があることが指摘されている。この現象はゴール・スイッチング[5]と呼ばれ、各特徴空間で高パフォーマンスの解を個別に探

```

procedure MAP-ELITES ALGORITHM (SIMPLE, DEFAULT VERSION)
(P ← ∅, X ← ∅)
for iter = 1 → I do
  if iter < G then
    x' ← random solution()
  else
    x ← random selection(X)
    x' ← random variation(x)
    b' ← feature descriptor(x')
    p' ← performance(x')
    if P(b') = ∅ or P(b') < p' then
      P(b') ← p'
      X(b') ← x'
return feature-performance map (P and X)

```

図 1 Map-Elites アルゴリズム [2]

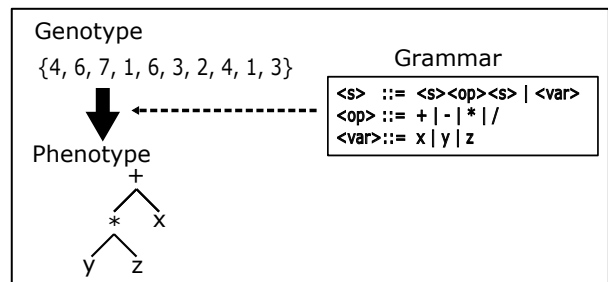


図 2 GE における表現型への変換

索するよりも高いパフォーマンスの解を生み出すこと報告されている。

Map-Elites は図 1 のように、一般的な進化計算のようにシンプルなアルゴリズムで、個体生成や遺伝的操作(交叉、突然変異)、適応度計算が行われ、個体の特徴に応じた空間における他の個体との適用度の優劣による選択が行われる。適用例としてはロボット生成[1]において、移動速度を適応度、特徴を高さ、重量、消費エネルギーとしたものなどがあり、特徴空間において用いる特徴には、適用問題に関するものや解表現に関するものが考えられる。

2.2 GE

GE(Grammatical Evolution) [7]は、進化計算の一つである遺伝的プログラミング(GP)から着想を得た手法で、図 2 のように線形の遺伝子型(Genotype)と、それを構造を持つ表現型(Phenotype)に変換する文法(Grammar)からなり、整数値で表現された可変長の遺伝子型は、非終端記号・終端記号・変換規則・開始記号からなるBNF(Backus Naur Form)で表現された文法規則により、表現型に変換される。これらの規則は生命におけるDNAが特定のルールに基づいてタンパク質を生成することに例えられる。

GE は、遺伝的アルゴリズムと同様な方法で、線形の遺伝子列に交叉や突然変異などの遺伝的操作を行い、GP のように構造を持つ表現型を生成することが出来る手法であ

[†] 中央学院大学 Chuo Gakuin University

る。GE は、デザインの生成や最適化問題などさまざまな問題に利用され、GP と同等の性能があることが確認されている。

3. 提案手法

本研究では GE に MAP-Elites を用いた手法を提案する。文献[1, 2]では、適用問題毎に異なる特徴空間を用いているが、本研究においては適用問題に直接は依存しない GE の特徴である遺伝子長、利用された遺伝子列長、木の深さをを用いた。また、文献[1]において良質な個体の多くは特徴空間上のすぐ近くの個体ではなく、中程度の距離にいる個体から生成されると報告されていることから、リーグ選択による解選択を導入した。GE や GP(Genetic Programming)ではトーナメント選択が用いられることが多いが、リーグ選択ではランダムに選択した解個体群における各個体間の特徴空間上での距離を計算し、その中から解を選択するものとする。

4. 実験環境・結果

適用問題には最小化問題である String-match 問題と Vladislavleva4 を用いて、GE のパラメータは個体数 500、世代 500、トーナメントサイズ 4 とし、リーグ選択におけるリーグサイズはある程度の大きさを持つことが統計的には効果的であると考えられるが、MAP-Elites は個体評価を同一の特徴を持つ個体間で行うことにより、その計算量を抑制することが優位性の一つであるため、本研究ではそのサイズを 3 とし、各個体間の特徴空間における距離はユークリッド距離を用いて、3 個体間距離の中から 2 番目に該当する 2 個体を選択するものとし、その特徴には GE における遺伝子長、利用された遺伝子列長、木の深さの組み合わせを用いた。

図 3 は String-match でのトーナメント選択の結果を示しており、図中の G は遺伝子長(genome length)、U は利用された遺伝子列長(used codons)、D は木の深さ(depth)を表し、G&U は G と U が特徴として用いられていることを表している。提案手法は GE と比較して良好な結果となり、提案手法内の比較においてはリーグ選択よりもトーナメント選択の方が優位な結果となった。

Vladislavleva4 では、トーナメント選択における 2 要素の特徴では GE に劣る結果となったが、トーナメント選択における 3 要素およびリーグ選択では良好な結果となった。

5. まとめ

本研究では、GE における進化的操作を MAP-Elites で行う手法を導入し、適用問題における特徴空間の次元や選択手法にも影響を受けるが、適用問題の特徴ではなく GE の特徴を用いることで、従来 GE と同等以上の効果的な解探索が行えることを示した。

今後の課題としては、他の性質の異なる問題への適用や特徴空間に関する解析、特徴空間に問題依存の特徴を加えることにより探索性能にどのような影響を与えるかの解析、解探索に有効な特徴の自動的な選択などが挙げられる。

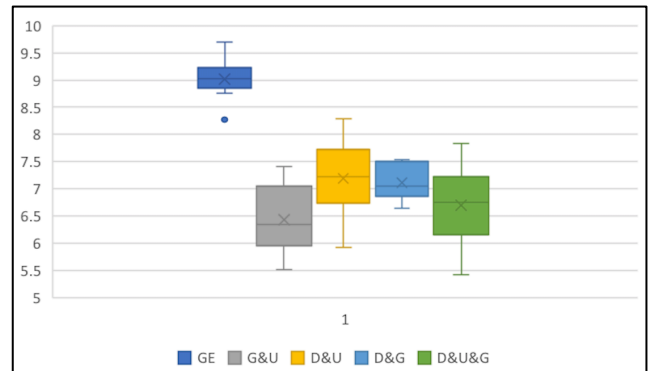


図 3 String-match (Tournament)

参考文献

- [1] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like natural animals", arXiv, 1407.3501 (2015).
- [2] Jean-Baptiste Mouret and Jeff Clune, "Illuminating search spaces by mapping elites", arXiv, 1504.04909, (2015).
- [3] J. Lehman and K. O. Stanley, "Evolving a diversity of virtual creatures through novelty search and local competition", In Proceedings of the 13th annual conference on Genetic and evolutionary computation, pp.211–218 (2011).
- [4] J. Clune, J.-B. Mouret, and H. Lipson. The evolutionary origins of modularity. Proceedings of the Royal Society B, 280(20122863), 2013.
- [5] A. Nguyen, J. Yosinski, and J. Clune, "Innovation engines: Automated creativity and improved stochastic optimization via deep learning", In Proceedings of the Genetic and Evolutionary Computation Conference (2015).
- [6] S. Forrest, M. Mitchell, "Relative Building-Block Fitness and the Building-Block Hypothesis", Foundations of Genetic Algorithms, Vol.2, pp.109-126 (1993).
- [7] O'Neill, Ryan, "Grammatical Evolution", Kluwer Academic Publishers (2003).