

# 未知環境に適応可能なプログラムの構築手法の提案

嶋 崇人<sup>†</sup>

<sup>†</sup> 横浜国立大学 理工学部

長尾 智晴<sup>††</sup>

<sup>††</sup> 横浜国立大学 大学院環境情報研究院

## 1. はじめに

近年、人手で構築困難なアルゴリズムを自動的に最適化する自動プログラミング手法が幅広い分野で活躍している。中でも、木構造で表現されたプログラムを交叉・突然変異などの遺伝オペレータにより最適化を行う遺伝的プログラミング(Genetic Programming; GP)[1]は強力な自動プログラミング手法として発展を遂げてきた。しかし従来の自動プログラミング手法では、学習環境と異なる環境下で一定以上の適応度が得られる個体を生成することは非常に難しい。そこで本稿では、未知環境にも適応可能なプログラムの構築手法について提案する。

## 2. 提案手法

本手法では、傾向の異なる複数のプログラムをモジュールとし、入力情報に応じて使用するモジュールを切り替えることで幅広い環境に適応できるプログラムの構築を目的としている。図 1 に提案手法の構造を示す。本手法はモジュール群とスイッチャの 2 つの層から構成される。

各モジュールは、性質ごとに分けられた学習環境に対しプログラムを最適化することで生成する。これによりそれぞれ性質の異なったモジュールが獲得できる。モジュールの生成には、グラフ構造状プログラムの自動生成手法である Graph Structured Program Evolution(GRAPE)[2]を用いる。一方スイッチャは各モジュールを制御する役割を担う。スイッチャは各モジュールほど複雑なプログラムを記述する必要はなく、モジュールの切替方法のみ組み込めばよい。

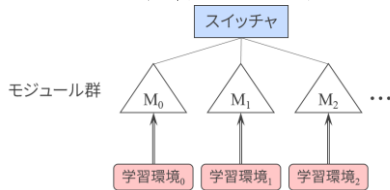


図 1. 提案手法の構造

## 3. 実験設定

本稿では部屋の全探索問題に対して実験を行った。図 2 に実験に用いた部屋の例を示す。個体の適応度は、部屋の面積 $S_{room}$ ・探索済みの面積 $S$ ・部屋を 9 割探索するまでにかかったステップ数 $step_{0.9}$ を用い次式によって算出する。なお、本実験では $\alpha = 10$ とした。

$$fitness = \begin{cases} \frac{S}{S_{room}} + \alpha \cdot (step_{0.9})^{-1} & (\text{if } \frac{S}{S_{room}} \geq 0.9) \\ \frac{S}{S_{room}} & (\text{otherwise}) \end{cases}$$

部屋のクラスは、広い部屋・通路・物体の多い部屋・仕切りの多い部屋の 4 種類とし、これと同じ数のモジュールを生

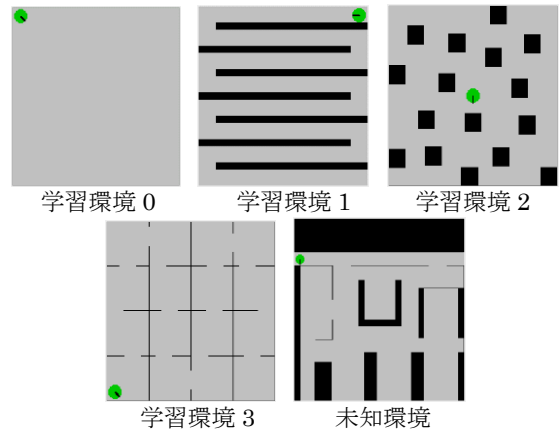


図 2. 実験に用いた部屋の例

表 1. 探索率の比較(5 試行中)

	平均 (標準偏差)	最良
提案手法	<b>0.8400</b> (0.0590)	<b>0.9309</b>
単純 GRAPE	0.4082 (0.2084)	0.6907
ランダムサーチ	<u>0.7239</u> (0.0980)	<u>0.8909</u>

成した。比較対象は単純 GRAPE とランダムサーチとした。

## 4. 実験結果

実験結果として、実験で用いた各手法の集塵率を表 1 に示す。単純 GRAPE の場合行動規則が 1 種類のみであるため行動の種類に幅がなく、未知環境での適応度が大幅に下がってしまう。一方提案手法の場合は、現在使っているモジュールが有効ではないことがわかると他のモジュールに切り替え、有効に探索するものを使い続けることで未知環境でも良好な適応度が得られた。

## 5. まとめ

本稿では、プログラムのモジュール化により未知環境に適応するプログラムの構築手法について提案した。また実験により、複数のモジュールを適応的に切り替えることで、未知の環境でも一定以上の性能が得られることが確認できた。今後は、実験で行ったものと異なる最適化手法の導入や、他領域の問題への適用を考えていく。

## 参考文献

- [1] Koza, John R. "Genetic programming: on the programming of computers by means of natural selection. Vol. 1." MIT press, 1992.
- [2] Shirakawa, Shinichi, Shintaro Ogino, and Tomoharu Nagao. "Graph structured program evolution." Proceedings of the 9th annual conference on Genetic and evolutionary computation. ACM, 2007.