コントローラ最適化における QD アルゴリズムの比較

清水川　七海　(指導教員：オベル加藤ナタナエル)

1　はじめに

Quality Diversity アルゴリズム（以下 QD アルゴリズム）は従来の単目的最適化方法の代わりとして考案されたアルゴリズムである。良質な複数の異なる解に注目することで発散と収束の間の適正なバランスをとることができる。空間に離散化された格子を設定し各格子に最も優れた個体を保有させることで良質な複数の異なる解を一度に保有することができ、局所的最適解に陥ることを防げるという利点がある。QD アルゴリズムの性能を測る指標はいくつか定義されてきたが [1]、まだアルゴリズムを自動で評価するようなフレームワークに欠けている。本研究では OpenAI gym を用いた自動ベンチマークプラットフォームを提示する。

1.1　QD アルゴリズムについて

QD アルゴリズムは多くの高品質で多様な解を求めることを主旨とした進化的アルゴリズムである。 [2] [3] 従来のアルゴリズムと違い、主に特徴空間や行動空間で動くことやできるだけで全ての特徴空間を埋めようとすることが特徴である。中でも主流として扱われているのが MAP-Elites アルゴリズムだ。このアルゴリズムは多様な特徴点ごとに優秀な結果を複数保持することで局所的最適解に陥ることを防ぐことができる。また本研究では AURORA (AUtonomous RObots Realising their Abilities)[4] と呼ばれる QD アルゴリズムを使用する。AURORA の特徴は、関連する行動ディスクリプタをどのように定義するかを自動的に学習しながら、多様で高品質な行動のコンテナを学習できることである。自身で行動ディスクリプタを学習することでユーザーが手動で定義するよりも有意義な解が得られると考えられる。AURORA 内部の動きは図 1 に示す。このアルゴリズムの中では QD の段階とエンコーダーの更新段階の 2 つの動きが交互に作用している。

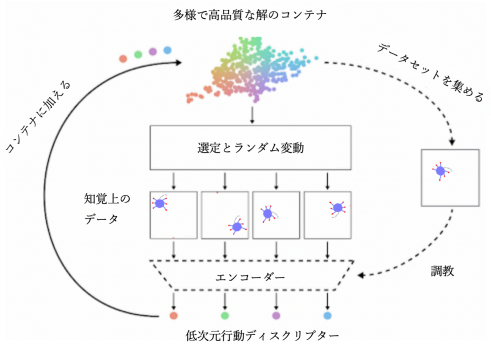

図　1:　AURORA アルゴリズムの内部動作　実線矢印が QD 段階、破線矢印がエンコーダーの更新段階を示す。

1.2　実験環境

実験では強化学習のシミュレーション用プラットフォームである OpenAI Gym[5] を使用する。中でも本研究では Classic Control に含まれる Mountain Car, Cartpole, Pendulum, Acrobot の 4 つの環境で実験を行った。それぞれの実験環境の環境設定は表 1 に示す。またこの時、状態空間が入力値、行動空間が出力値を表す。

	行動空間	**状態空間**
MountainCar	Discrete(3)	(2,)
Cartpole	Discrete(2)	(4,)
Pendulum	Box(-2.0, 2.0, (1,), float32)	(3,)
Acrobot	Discrete(3)	(6,)
	状態最高値	**状態最低値**
MountainCar	[0.6 0.07]	[-1.2 -0.07]
Cartpole	[4.8 inf 0.42 inf]	[-4.8 -inf -0.42 -inf]
Pendulum	[1. 1. 8.]	[-1. -1. -8.]
Acrobot	[1. 1. 1. 1. 12.57 28.27]	[-1. -1. -1. -1. -12.57 -28.27]

表 1: OpenAI Gym ベンチマークの環境設定

2　研究方法

本研究では 4 つのベンチマークを同時に AURORA で探索する。また類似するネットワーク次元が生成できるような異なる形のレイヤーを用いて実験を行い結果に差異が生じるかを実験する。ニューラルネットワークとは入力を線形変換する処理単位であり入力層、中間層、出力層から成り立つ。実験ではこの中間層の形を変更する。表 2 では中間層が二重構造の [10,10] である時と三重構造の [10,6,7] である時に生成されるネットワークの次元を表示している。

実験はそれぞれ 5 回ずつ実行されバジェットは 10000 に設定されている。本研究は QD アルゴリズムを実行するフレームワークである QDpy[6] を用いており、使用したコードは全て Github[7] から入手可能である。

	[10,10]	**[10,6,7]**
MountainCar	173	169
Cartpole	182	181
Pendulum	213	209
Acrobot	161	164

表 2: レイヤーサイズとネットワーク次元

3　結果

図 3 と図 4 はそれぞれレイヤーサイズが [10,10] と [10,6,7] によって出力された結果の一部である。右端のヒートマップではより明るい色のものが高品質な解で

あることを示している。両図において、ほぼ全てが暗い点で覆われている Pendulum は解の品質が悪く、一方で分布のほとんどを明るい色が占めている Acrobot は解の品質が高いことを表している。

また比較として図 2 は Map-Elites による Mountain-Car の結果を例示する。左上が特徴空間を距離と速度に設定しレイヤー [10,10] で測定したもの、右上が [100] で測定したものである。また下の図は二重構造のレイターの大きさを変えた時の性能の平均値を表している。

AUROARA の性質上、行動ディスクリプタは自ら学習して定義されるのでそれぞれ解の分布が異なっていることがわかる。ただほぼ同次元のネットワークにおいてレイヤーの構造を変形したことによる特徴的な差異などは観測することができなかった。

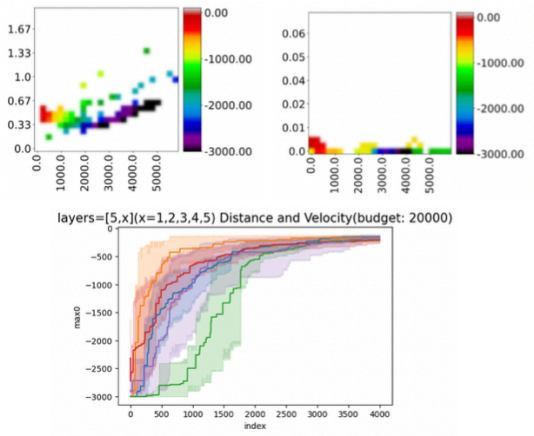

図 2: MAP-Elites アルゴリズムによる結果

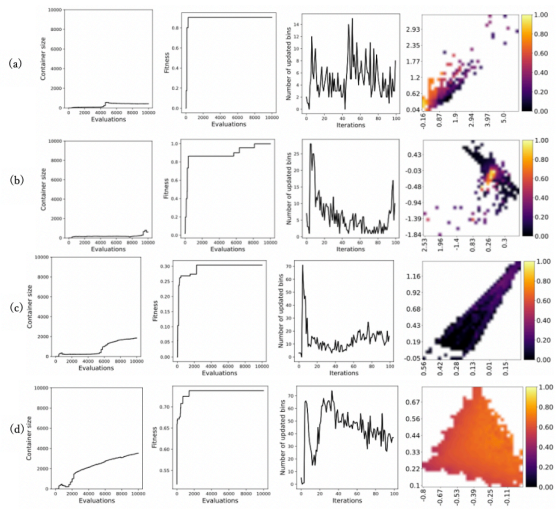

図 3: レイヤーサイズ [10,10] の結果 (a) Mountain Car (b) Cartpole (c) Pendulum (d) Acrobot

4 まとめと今後の課題

本実験では AURORA を用いて OpenAI Gym の機能を応用した自動ベンチマークプラットフォームを提案した。ただ今回は OpenAI Gym の中でも Classic

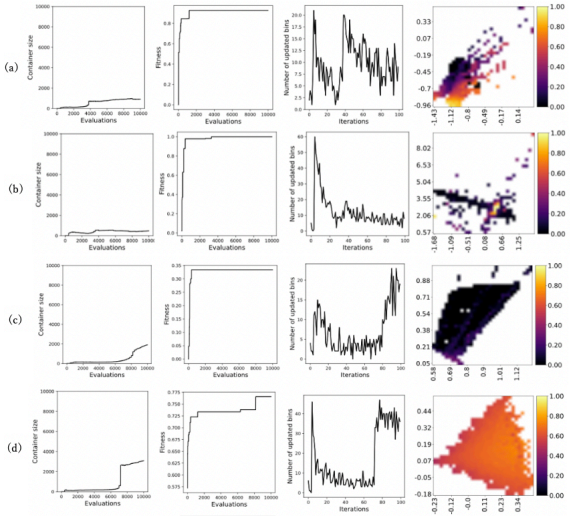

図 4: レイヤーサイズ [10,6,7] の結果 (a) Mountain Car (b) Cartpole (c) Pendulum (d) Acrobot

Control の属している限られたベンチマークのみの応用となった。Classic Control のベンチマークは比較的に一般的な手法で解きやすいとされているものが多い。

今後の課題としては同じ OpenAI Gym でも Atari 環境 [8] などより複雑なベンチマークにも対応できるようなプラットフォームの作成が挙げられる。また最終的にはどのようなベンチマークが与えられても自動で有意義な行動ディスクリプタを生成できるような機能を作成することが目標である。

参考文献

[1] Antoine Cully and Yiannis Demiris. Quality and diversity optimization: A unifying modular framework. 2017.

[2] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. 2015.

[3] Joel Lehman and Kenneth O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. 2011.

[4] Luca Grillotti and Antoine Cully. Unsupervised Behaviour Discovery with Quality-Diversity Optimisation. 2021.

[5] Openai gym. `https://www.gymlibrary.dev`.

[6] Leo Cazenille. Qdpy. `https://gitlab.com/leo.cazenille/qdpy`.

[7] Nanami Shimizugawa. Github. `https://github.com/shimizugawa/aigym`.

[8] Openai gym, atari environments. `https://www.gymlibrary.dev/environments/atari/`.