Hybrid Reward Architecture を用いたリアルタイムな意思決定の改善

藤村 悠太朗[1,a)]　金子 知適[2,3,b)]

概要：強化学習を行うエージェントにおいて，対象とする問題が大規模かつ複雑な状態を持つほど，学習が難しくなるという関係にある．Hybrid Reward Architecture (HRA) は，状態が構造を持つドメインにおいて，その構造を利用して状態行動空間を分割し効率的に学習を進めるフレームワークである．本研究では，この HRA を GVG-AI のテストセットに適用することで，空間の分割が自明ではない問題での HRA の性能を調査した．その結果，比較的構造が明確な対象を選んだにもかかわらず，報酬に遅れが発生する，報酬が発生する間隔が疎である，分割が完全に独立でない， など困難な性質を持つドメインでは，HRA 単独では十分な学習が難しいことが分かった．しかし，学習を容易にするような擬似報酬を用意することにより，性能が向上することを確認した．

Improvement of Real-Time Decision Making by Hybrid Reward Architecture

Yutaro Fujimura[1,a)]　Tomoyuki Kaneko[2,3,b)]

Abstract: Reinforcement learning in large and complex domains is a challenging research problem. In Hybrid Reward Architecture (HRA), a reward function is decomposed in advance to enhance learning in such domains, and then value functions are separately learned for decomposed reward functions. In this paper, we apply HRA into a game in GVG-AI test sets to evaluate the performance of HRA in domains where decomposition of the reward is not straightforward. The results indicated that HRA need more enhancements to play GVG-AI games decently, though we found that a pseudo reward worked well with HRA.

1. はじめに

人工知能の研究において，リアルタイムな意思決定を行うことは重要な課題の 1 つである．ビデオゲームであれば 1/60 秒ごとという非常に短い時間の間に判断をする必要があり，実世界において活動するロボットに組み込まれる人工知能も，短時間で意思決定を下せることが必要とされる場合がある．

また，環境のモデルが未知であったり複雑である場合にも，エージェントが環境と相互作用する中で自律的に学習をすすめる強化学習が近年注目を集めている．典型的なアルゴリズムに Q 学習があり，Q 学習と深層ニューラルネットを組み合わせた Deep Q-Network (DQN) が，Atari 2600 のゲームの一部において人間を上回るスコアを記録した [1]．しかし，盤面の取る状態が複雑になる問題に対しては，学習が遅く安定しないという課題が存在する．

van Seijen らは，Hybrid Reward Architecture (HRA) と呼ばれる，DQN では学習が遅くなる問題に対して，問題固有の知識を用いて状態行動空間を分割し効率的に学習を進める手法を提案した [2]．Atari 2600 のゲームの 1 つである Ms. Pacman は DQN では学習することが難しかった

[1] 東京大学教養学部学際科学科
Department of Interdisciplinary Sciences, College of Arts and Sciences, The University of Tokyo
[2] 東京大学大学院情報学環
Interfaculty Initiative in Information Studies, the University of Tokyo
[3] 国立研究開発法人科学技術振興機構さきがけ
JST, PRESTO
[a)] yut-mak874@g.ecc.u-tokyo.ac.jp
[b)] kaneko@acm.org

が [1]，HRA を適用したエージェントが DQN より早く学習を進めることに成功している．

Ms. Pacman ではそれぞれのエサが完全に独立した報酬と見なすことが可能で，人間による空間の分割が容易であるという性質があり，それを HRA は利用している．しかし，一般のゲームではそれぞれの要素が互いに独立でなく，空間の分割が自明でないものが考えられる．

本研究では，Ms. Pacman よりゲームの構造が複雑で，空間の分割が自明でないようなゲームにおける，HRA の性能を調査した．

2. 強化学習

2.1 モデル

状態集合 $\mathcal{S}$, 行動集合 $\mathcal{A}$, 環境報酬関数:$R_{env} : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$, 遷移確率関数 $P : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [0,1]$ を持つ, マルコフ決定過程を考える．時間 t ごとに，エージェントは状態 $s_t \in \mathcal{S}$ を観測し, 行動 $a_t \in \mathcal{A}$ をとる．そして, 遷移確率関数 $P(s_t, a_t)$ から導かれた状態 s_{t+1} を観測し, 報酬 $r_t = R_{env}(s_t, a_t)$ を得る．その動作はどの行動をとるかを選択するかを表現した確率分布 $\pi : S \times A \to [0,1]$ として定義され, これを方策 (policy) と呼ぶ. エージェントの目標は, 期待される報酬を最大化するような方策を見つけることであり, 期待報酬を $G_t := \sum_{i=0}^{\infty} \gamma^i r_{t+i}$ とする．ここで, $\gamma \in [0,1]$ は割引率であり, 即時的な報酬と将来の報酬とのバランスをとるための数値である．

方策 π において, 状態 s において行動 a をとることの価値を行動価値関数と言い, これを期待報酬と一致する．つまり,

$$Q^{\pi}(s,a) = \mathbb{E}[G_t | s_t = s, a_t = a, \pi] \tag{1}$$

これにより, 最適な行動価値関数は $Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$ と求められ, Q^* の行動価値関数に対して貪欲に行動を選べば, 最適な方策 π^* を得られる．

2.2 Hybrid Reward Architecture

通常の Q 学習では, 行動価値関数を以下のように更新する.[3]

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[y - Q(s_t,a_t)] \tag{2}$$

$$y = r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) \tag{3}$$

Hybrid Reward Architecture [2] では, 環境報酬関数 R_{env} を n 個の報酬関数に w_i で重み付けをして分割することを考える, つまり,

$$R_{env}(s,a) = \sum_{k=1}^{n} w_k R_k(s,a) \tag{4}$$

この分割された報酬関数それぞれに対して, 分割された強化学習のエージェントを訓練する．それぞれのエージェントが持つ行動価値関数を $Q_i(s,a)$ として,

$$Q_k(s_t,a_t) \leftarrow Q_k(s_t,a_t) + \alpha[y_k - Q_k(s_t,a_t)] \tag{5}$$

$$y_k = r_{k,t+1} + \gamma \max_a Q_k(s_{t+1}, a_{t+1}) \tag{6}$$

と計算でき, すると,

$$Q_{HRA}(s,a) = \sum_{k=1}^{n} w_k Q_k(s,a) \tag{7}$$

が計算できる．最適行動価値関数 Q^*_{HRA} は, $Q^*_1, \ldots, Q^*_n$ によって,

$$Q^*_{HRA}(s,a) := \sum_{k=1}^{n} w_k Q^*_k(s,a) \tag{8}$$

と定義される．一般には, Q^*_{HRA} は Q^*_{env} とは一致しないが, Q 値の学習に expected Sarsa を用いる, つまり,

$$y_k = r_{t+1} + \gamma \sum_a \pi(a|s_{t+1}) Q_k(s_{t+1}, a) \tag{9}$$

を用いて $Q^{\pi}_1, \ldots, Q^{\pi}_n$ を更新することで, $Q^{\pi}_{HRA}(s,a) := \sum_{k=1}^{n} w_k Q^{\pi}_k(s,a)$ が Q^{π}_{env} と一致することが示されている．問題固有の知識を用いた報酬関数の分割を行うことで, 通常の Q 学習より行動価値関数の収束が早いことが [2] で示されている．

3. GVG-AI Framework

GVG-AI Framework は，1 つの特定のゲームだけでなく，複数のゲームを攻略できる汎用的なゲーム AI の作成を支援するために開発されたフレームワークである．[4]

GVG-AI Framework では，Video Game Description Language (VGDL) によってゲームが記述されている．[5] VGDL は 2D ビデオゲームに特化した言語であり，簡単なテキストによってゲームのルールやマップを記述できるため，人間にとって作成や変更が容易である点が特徴である．

GVG-AI Competition はこのフレームワークを用いて GVG-AI の汎用エージェントの能力を競うコンテストである．第一回が 2014 年の IEEE CIG に始まり，その後も 1 年ごとにコンテストが開かれている．[*1] メインのコンテストは，Planning track で，エージェントが初見のゲームをプレイして性能を競う．エージェントには，ゲームの環境をやりとりするインタフェースに加えて，内部で環境のシミュレーションをするための Forward model が与えられるため，ある程度の探索をしてから意思決定を行うことが出来る．新たなコンテストとして，2017 年に Learning track が新設された．こちらのコンテストでは Forward model が与えられない代わりに，エージェントはゲームをプレイして性能評価を受ける直前に，一定の制限時間の中で事前に同じゲームを繰り返しプレイすることで学習を進め，最終

[*1] http://www.gvgai.net/

的に良いスコアを獲得するための準備が許されている．こちらでは，エージェント作成者の目標は，どのようなゲームでも短時間で学習出来るようなエージェントを作成することである．

Planning track では，ドメイン固有の知識を必要としないモンテカルロ木探索やその応用アルゴリズムによって，与えられた Forward model を利用して探索する手法が有効であることが示されている．[4], [6] 一方で，Learning track では，Foward model が利用できないため，エージェントが実際に行動して学習する必要があるため，Q 学習のような強化学習の手法が有効であると考えられる．しかし，複雑なゲームの場合は状態行動空間が膨大になり，学習の効率が非常に悪くなることが考えられる．

4. 提案手法

本研究では, HRA を適用したエージェントを作成し，GVG-AI のテストセットの 1 つである Fireman でその性能を評価する．また，同時に，問題固有の知識を用いた擬似報酬を HRA のエージェントに与えることで，性能の向上が見込めるかを調査する．

4.1 Learning Track の仕様

GVG-AI Learning Track では，以下の手順でエージェントを評価する．

- Training Phase
  Phase1 Level 0, 1, 2 をこの順番で 1 回ずつプレイする
  Phase2 Level 0, 1, 2 からエージェントが自由にレベルを選択してプレイをする．制限時間である 5 分を経過するまで，繰り返しゲームをプレイする．
- Validation Phase
  – Level 3, 4 を交互に 10 回ずつプレイする．

4.2 Fireman の概要

Fireman は次のようなルールのゲームである．

- ゲームの概要
  マップ上に炎，木箱，消火栓が複数配置されている．消防士であるキャラクタを操作して，マップ上に存在する炎をすべて消火すると勝利となり，炎と接触してダメージを受けて死亡したり，炎が全て消火できないまま規定ターン数を超過すると敗北となる．炎は木箱に確率的に引火するため，なるべく早く消火する必要がある．
- 操作キャラクタ
  1 ターンに (LEFT, RIGHT, UP, DOWN, USE) のいずれかの行動を 1 つ選択する．消火栓に接触すると一定量の水を取得する．水を保有している場合は USE を選択するとキャラクタの向いている方向に一直線に飛ぶ水弾が発射される．水は最大水弾 10 発分だけ保持できる．一定回数以上炎と接触すると死亡する．
- 点数の獲得
  水弾が炎と接触するとその時点で +2 点を得る．炎が木箱に引火するとその時点で −1 点を得る．エージェントは獲得した点数を報酬と見なす．

先行研究 [2] で扱われた Ms. Pacman より困難である理由として，報酬が発生する要素同士が独立ではないことが挙げられる．例えば，炎が木箱に引火するというルールがあるため，マップに炎と木箱が 1 つずつ配置され隣接しているマップを，炎が 1 つのマップと箱が 1 つのマップに分割して考えることがどの程度適切かは自明ではない．

4.3 報酬関数の分割の手法

HRA を構成する上で，報酬を式 (4) のように分割する方法がいくつか考えられるが，本研究では点数を発生させる要素ごとに分割した．Fireman では点数が発生する対象は木箱と炎のみであるので，それぞれの木箱及び炎ごとに Q 値を計算する．先行研究 [2] ではニューラルネットワークを用いて $Q_k(s,a)$ を近似しているが，本研究ではハッシュ表に持つことで実装をした．

また，式 (2) では盤面の状態 s_t をそのまま用いて学習を行うが，盤面の状態をそのまま用いると状態空間が依然として膨大になるため，盤面の状態 s_t からキャラクタの状態のみを切り出した状態 $s_{t,chara}$ を代わりに用いることで，状態の圧縮を行う．

4.4 擬似報酬を与えるエージェント

このゲームにおいては，正の点数を得ることができるのは，水を発射することができる時，つまり水を持っている時に限られる．そこで，エージェントが水を持っていない状態で，エージェントからユーグリッド距離で最も近い消火栓の距離が，エージェントの行動によって小さくなった場合に，擬似報酬を与えるエージェントを作成し，本研究では HRA+と呼称する．擬似報酬 r' は, 上記の条件を満たす時に，盤面に存在する全ての k に対して，式 (6) を

$$y_k = r_{k,t+1} + r' + \gamma \max_a Q_k(s_{t+1}, a_{t+1}) \qquad (10)$$

と変更することで与える．

また，HRA,HRA+に共通して，負の報酬を避けようとして自殺することを防止するため，エピソード途中で死亡した場合に，擬似報酬 $r'' = -1000.0$ を式 (10) と同様に与える．

5. 実験

式 (5),(6) で示した Q 学習のパラメータは $\alpha = 0.1, \gamma = 0.95$ とし，式 (7) で示した分割した Q 値を結合する重み w_k の値は定数 $w_k = 1$ とした．HRA+に式 (10) で与える擬似報酬は $r = 1.0$ とした．

表 1　行動回数の比較
Table 1　A comparison of action counts

言語	行動回数
Python	10320
Java	78489

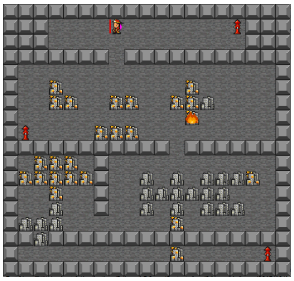

図 1　配布されているマップ
Fig. 1　Default map

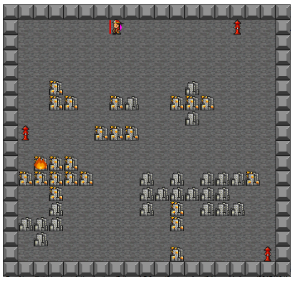

図 2　図 1 を変更したマップ
Fig. 2　The modified map of Fig. 1

5.1　実験環境

GVG-AI の Learning Track には，Java 版と Python 版の 2 つの API が存在する．本研究では，実装の容易さから Python による実装を行った．すべての行動をランダムに行うエージェントが Training Phase2 の 5 分間に行動を選択した回数を，Java 実装と Python 実装で比較した結果を**表 1** に示す．表 1 からわかるように，同じアルゴリズムのエージェントでも，実装する言語の差異によって結果が異なる可能性がある．

5.2　使用マップについて

GVG-AI で配布されている Fireman のマップでは，正の点数を得るための行動の組み合わせが限られており，1500 ターン以内に正の点数を得ることが難しかったため，**図 1**,**2** のように配布マップの四方以外の壁を床に変更したマップで学習を行った．

5.3　各エージェントの Fireman 環境での獲得点数

GVG-AI Learning Track の仕様に従い，それぞれのエージェントに対して，Training Phase で学習を行い，Validation Phase で評価する．Validation Phase の 20 エピソードの点数について，100 回実験を行った時のヒストグラムを**図 3** に示す．また，ゲームの性質上，短いターン数で死亡すると点数が 0.0 に近くなりやすいことを考慮して，あるエピソードで獲得した点数 p に対して，その評価値 p' を $p' = \max(0, p)$ と定め，評価値の平均に関するデータを**表 2** に示す．表 2 の右の項目は Validation Phase の 20 エピソード中に $p' > 0$ となるエピソードが平均していくつあったかを示すものである．random と HRA の間にはあまり差異がなく，Fireman では HRA の性能が落ちることがわかる．一方，HRA+は HRA に比べて性能が向上し，

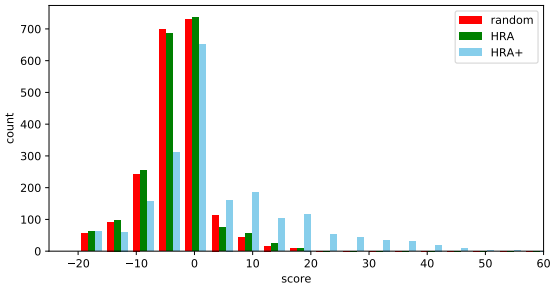

図 3　点数のヒストグラム
Fig. 3　A histogram of scores in validation phase

表 2　各エージェントの評価値
Table 2　Evaluation scores of three agents

	平均評価値 p'	$p' > 0$ の平均エピソード数
random	0.675	2.39
HRA	0.689	2.16
HRA+	5.997	8.42

擬似報酬が点数の獲得において有効であったことが確認できたが，人間が容易に正の点数を取ることができるゲームであることを考えると，十分な性能とは言えないだろう．

6.　おわりに

本研究では，HRA のエージェントを状態行動空間の分割が自明でないゲームの 1 つである Fireman の環境で性能を評価した．空間の分割が自明でない中で，ゲームの性質を利用した報酬関数の分割を行ったが，HRA のみでは点数の獲得に有効ではなかった．しかし，擬似報酬と組み合わせることによって性能が向上することが確認できた．

今後の課題としては，式 (4) の分割を人間が行うのではなく，自動的に学習することが挙げられる．

謝辞　この研究の一部は，JSPS 科研費 16H02927 と JST さきがけの支援を受けています。

参考文献

[1] Mnih, V. et al.: Human-level control through deep reinforcement learning, *Nature*, Vol. 518, No. 7540, pp. 529–533 (2015).
[2] Seijen, H. V. et al.: Hybrid Reward Architecture for Reinforcement Learning (2017).
[3] Sutton, R. S. and Barto, A. G.: *Reinforcement Learning* (1998).
[4] Perez-Liebana, D. et al.: The 2014 General Video Game Playing Competition, *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 8, No. 3, pp. 229–243 (2016).
[5] Schaul, T.: A video game description language for model-based or interactive learning, *IEEE Conference on Computatonal Intelligence and Games, CIG* (2013).
[6] Soemers et al.: Enhancements for real-time Monte-Carlo Tree Search in General Video Game Playing, *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8 (2016).