視野制限環境下での追跡問題における強化学習の学習効率向上法

山森　一人 a)・吉田　雅也 b)・相川　勝 c)

Learning Efficiency Improvement of Reinforcement Learning in Restricted View Environment

Kunihito YAMAMORI, Masaya YOSHIDA, Masaru AIKAWA

Abstract

Reinforcement learning is one of the unsupervised machine learning methods. In reinforcement learning, multi-agent learning is usually used. In recent years, some researchers try to utilize drones in search of missing persons, but efficiency is not enough for the real world. In this research, we propose a multiple single role agent system which gives some roles for layered agents. Experimental results showed that our method was able to capture the target with 30% faster steps than the conventional method.

Keywords: Reinforcement learning, Layered agent, Role, Multi agent system

1. はじめに

強化学習(Reinforcement Learning : RL)[1]は教師なし機械学習の一つである。環境モデルの状態を知覚し、過去の経験から選択すべき行動を決定する学習エージェントの学習手段としてRLは用いられる。学習エージェントが選択した行動は環境モデルを遷移させ、遷移した環境モデルは学習エージェントに対して報酬を与える。学習エージェントは、行動の結果得られた報酬をもとに先の選択を評価し、学習を進める。RLの目的は、将来の報酬も踏まえた上で、最も多くの報酬を得る行動の選択方針を学習エージェントが学習することである。RLでは、実社会への応用を目指して、複数の学習エージェントを用いるマルチエージェント学習が用いられる。RLの代表的な問題に追跡問題があり、渡邊ら[2]によりRLの適用が行われている。

近年、行方不明者の捜索活動などでドローンの活用が試みられているが、ドローン1機に操縦者1人を要し、効率の悪さが指摘されている。賀数ら[3]は行方不明者の捜索を追跡問題に見立て、マルチエージェントシステム（Multi Agent System : MAS）を用いる手法を提案した。しかし、MASのエージェントは視野を360°としており、ドローンの多くが搭載しているカメラは全周カメラではないという現実に合致しない。

本研究では、エージェントを階層化し、更にエージェントに役割を付与する新たなMASの形として、マルチプル・シングルロール・エージェントシステム（Multiple Single Role Agent System : MSRAS）を提案する。具体的には、エージェントに指揮、追跡、監視の3つの役割を付与し、指揮役を上位エージェント、追跡役と監視役を下位エージェントとして追跡問題の学習を行う。本研究の目的は、視野制限環境下の追跡問題において、提案手法により役割の異なるエージェントを協調行動させることで学習効率が向上することを示すことである。

2. 強化学習

2.1. RL の構成要素

RL は以下の要素から成り立つ。

- 報酬関数：環境モデルの状態が遷移したときに、環境がエージェントに与える報酬を決定する。
- 価値関数：現在の状態において選択した行動の価値を決定する。
- 環境モデル：エージェントが適応する環境を示す。本研究の場合、追跡問題のフィールドを指す。
- 方策：エージェントが行動を選択する方法を定義し、$\varepsilon - greedy$方策 [4]が知られている。$\varepsilon - greedy$方策では、時刻tでの環境モデルの状態s_tにおいて行動aを選択する価値$Q(s_t, a)$をもとに、式(1)に従って行動a_tが選択される。

a) 情報システム工学科教授
b) 情報システム工学科
c) 宮崎大学工学部教育研究支援技術センター技術職員

$$a_t = \begin{cases} arg\max_{a \in A} Q(s_t, a), & 1.0 - \varepsilon, \\ random, & \varepsilon. \end{cases} \quad (1)$$

ここで、εは[0,1.0]の定数値である。$\varepsilon - greedy$方策では、価値が大きい行動ほど優先的に選択されるが、常に価値が最大の行動を選択すると同じ行動ばかりが選択され、最適な行動の選択の探索が行われなくなる可能性がある。そこで、εの確率でランダムに行動を選択し、探索の範囲を拡げる。またAはエージェントの選択できるすべての行動である。

2.2. 学習の流れ

RL の流れを、Watkins ら [5]が提案した Q 学習を例に説明する。時刻tの環境モデルs_tにおいて、行動aを実行したときの価値関数の更新手順を以下に示す。

STEP1 : 現在の環境モデルの状態s_tを取得し、価値関数を用いて選択できるすべての行動の価値を求める。

STEP2 : STEP1 で得た価値をもとに、方策に従って行動を選択する。

STEP3 : 学習エージェントの行動により環境モデルの状態が遷移し、報酬R_{t+1}が発生する。報酬R_{t+1}は報酬関数によって求める。

STEP4 : 得られた報酬R_{t+1}をもとに、エージェントは式(2)により、選択した行動の価値$Q(s_t, a)$の更新を行う。

$$Q(s_t, a) \leftarrow Q(s_t, a) + \alpha TDerror,$$
$$TDerror = \left[R_{t+1} + \gamma \max_{p \in A} Q(s_{t+1}, p) - Q(s_t, a)\right]. \quad (2)$$

ここで、$TDerror$は予想していた価値と実際に得られた価値の差を表す。αは学習率と呼ばれ(0,1.0]の定数である。一方、γは割引率と呼ばれ[0,1.0]の定数である。

学習エージェントが状態を取得して行動を 1 回選択するまでをステップと呼ぶ。また、初期状態からステップを繰り返し、終了状態に到達するまでをエピソードと呼ぶ。

3. 提案手法

3.1. MARAS

本研究では、視野を制限したエージェントによる追跡問題を取り扱う。追跡問題では、学習エージェントは捜索エージェント（Search Agent : SA）として獲物を追跡し、ターゲットを少ないステップ数で捕獲できる行動の選択方針を学習する。一般に、視野制限のないエージェントを用いた方が早くターゲットを捕獲できることは明白である。そこで本研究では、その差を埋めるために MSRAS を提案する。MSRAS は、エージェントに役割を付与した階層型強化学習の呼称である。

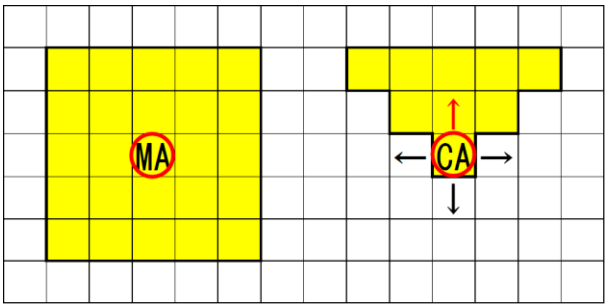

図 1.追跡役・監視役 SA の視野.

本研究では以下の条件による追跡問題を用いて提案手法を評価する。

- 50 × 50の正方形のフィールドを想定する。
- ターゲット 3 体と SA15 体を、重複しないランダムな位置に初期配置する。
- ターゲットと SA は 1 ステップで上下左右方向へ 1 マス移動、その場に停止の 5 種類の行動を選択できる。
- SA は各ステップで行動を選択し、ターゲットは 5 ステップごとに行動を選択する。
- SA には、自身の向いている方向から図 1 内の状態を知覚する視野を持つ。図 1 で、MA は監視役 SA、CA は追跡役 SA を表し、詳しくは 3.2 節で説明する。
- SA は、直前の移動方向を向いているものとする。図 1 の CA では、矢印で移動可能な方向を示し、赤矢印は SA の向きを示す。
- 1 体の SA がターゲットを見つけると、他の SA にターゲットの座標が伝達される。
- SA の視野内に障害物がある場合、障害物の向こう側は視野として考慮しない。

3.2. SA の役割

SA に付与する役割として、以下の 3 種類を設定する。

- 指揮役：
  1 体のみこの役割を設定する。環境内には存在せず、下位層の SA からの情報の統合、下位層の SA に対する情報伝達、下位層の SA の役割配分を決定する役割を持つ。
- 追跡役：
  環境内に複数存在し、ターゲットを追跡する。ターゲットを発見すると指揮役 SA に報告する役割も持つ。
- 監視役:

環境内に複数存在し、移動せず、ターゲットを監視する。追跡役 SA よりも広い視野を持つ。ターゲットを発見すると指揮役 SA に報告する役割を持つ。

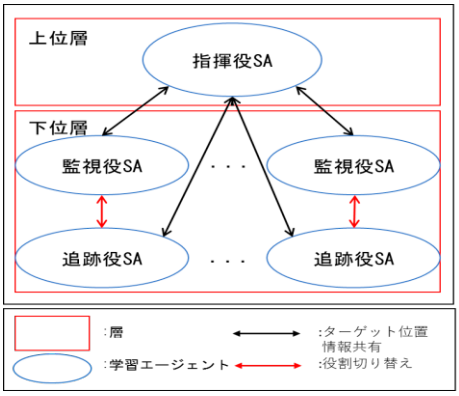

図 2.提案手法の学習モデル.

図 3.上位層の指揮役 SA が考慮する環境の状態.

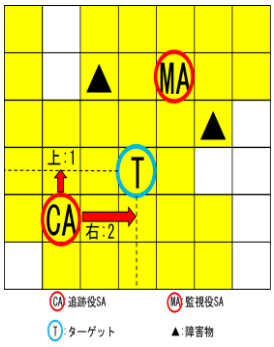

図 4.下位層の SA が考慮する環境の状態.

提案する学習モデルを図 2 に示す。上位層の SA は役割配分を、下位層の CA は停止を含む実行すべき行動を、MA はターゲットをできるだけ多く発見できる場所をそれぞれ学習する。上位層の SA は現在の状態から下位層の SA の現在の役割を評価し、役割を再分配するかどうかを選択する。また、下位層の SA に視野外のターゲットの位置を伝える。下位層の SA はターゲットを捕獲するために各役割の SA が実行すべき行動を選択する。上位層の SA、下位層の SA 共に Q 学習によって学習を行う。Q 学習の方策には ε–*greedy* 方策を用いる。

3.3. 上位層の学習

上位層の指揮役 SA は、下位層の SA に設定した役割配分で、全ターゲットを捕獲するのにかかったステップ数をもとに、選択すべき役割配分を学習する。上位層の指揮役 SA が考慮する環境の状態の例を図 3 に示す。図 3 では、下位層の SA の現在の視野を黄色のマスで表しており、指揮役 SA は黄色のマスのみを考慮する。上位層に対する報酬関数の設計を式 (3) に示す。

$$R_{t+1}^{g} = \begin{cases} \lambda_1^g, & (\text{ターゲット捕獲時}), \\ \lambda_2^g, & (\text{最短ステップ更新時}), \\ \lambda_3^g, & (\text{全ターゲット捕獲時}), \\ \mu_4^g, & (\text{ステップ上限到達時}), \\ \mu_5^g, & (otherwise). \end{cases} \tag{3}$$

ここで、$\lambda_1^g, \lambda_2^g, \lambda_3^g$は正の報酬、$\mu_4^g, \mu_5^g$は負の報酬、$g$は指揮役であることを表す。

3.4. 下位層の学習

下位層の SA は、現在の位置からターゲットの位置へ向かうときの距離をもとに行動を選択する。ターゲットまでの距離は、上下左右のそれぞれの方向に対する座標の差の絶対値として定義する。下位層の SA が考慮する環境の状態の例を図 4 に示す。図 4 の場合、SA とターゲットの上下方向、左右方向の距離は、それぞれ上 1、右 2 となる。下位層の SA の報酬関数の設計を式 (4)、及び式(5)に示す。

- 追跡役

$$\mathrm{r}_{t+1}^{c} = \begin{cases} \lambda_1^c, & (\textit{ターゲット発見時}), \\ \lambda_2^c, & (\textit{ターゲット捕獲時}), \\ \mu_3^c, & (otherwise). \end{cases} \tag{4}$$

- 監視役

$$\mathrm{r}_{t+1}^{m} = \begin{cases} \lambda_1^m, & (\textit{ターゲット発見時}), \\ \mu_2^m, & (otherwise). \end{cases} \tag{5}$$

ここで、$\lambda_1^c, \lambda_2^c, \lambda_1^m$は正の報酬、$\mu_3^c, \mu_2^m$は負の報酬、$c$は追跡役、$m$は監視役を表す。

4. 提案手法の評価

4.1. 学習環境

評価実験は、3.1 節で説明した環境を用いた追跡問題で行う。実験時の学習パラメータは、追跡問題でよく用いられる値として $\alpha = 0.4$、$\gamma = 0.7$、$\varepsilon = 0.1$ にそれぞれ設定する。また、各階層の学習で用いる報酬の値を表 1 に示す。表 1 中の a はステップ上限、b は発見したターゲットの数、t はステップ数を表す。これらの報酬は予備実験を行って定めた。

エピソード開始時にターゲットと SA をランダムに配置する。SA2 体がターゲットを対面 2 方向で挟んだ時、捕獲が完了したものとする。以上の行程を全ターゲット分行うか、36,000 ステップ経過するまで試行を続ける。1 エピソード 36,000 ステップの学習を 300 エピソード繰り返す。

4.2. 実験結果

図5に、CAが全ターゲットを捕獲するまでのステップ数の推移を示す。手法1は視野制限を行った従来手法、手法2は視野制限を行わない従来手法を示している。ランダムウォークは、学習を行わなかった場合での捕獲にかかるステップ数を示している。図5から、100エピソードくらいまで学習が急激に進み、250エピソード付近でほぼ学習が完了していることが分かる。提案手法は、視野制限を行ったCAのみの従来手法に比べ、捕獲に必要なステップ数が約8,000 ステップ減少している。以上から、エージェントに役割を付与することにより、従来手法より早くターゲットを捕らえることができ、学習効率を向上できたといえる。

表.1 実験時の各報酬値

報酬	正の報酬	負の報酬
r^{c}_{t+1}	$\lambda^{c}_{1}=5$	$\mu^{c}_{3}=-0.05$
	$\lambda^{c}_{2}=50$	
r^{m}_{t+1}	$\lambda^{m}_{1}=5\times a$	$\mu^{m}_{2}=-0.05$
R^{g}_{t+1}	$\lambda^{g}_{1}=5$	$\mu^{g}_{4}=-10$
	$\lambda^{g}_{2}=\frac{10\times b}{t}$	$\mu^{g}_{5}=-0.05$
	$\lambda^{g}_{3}=100$	

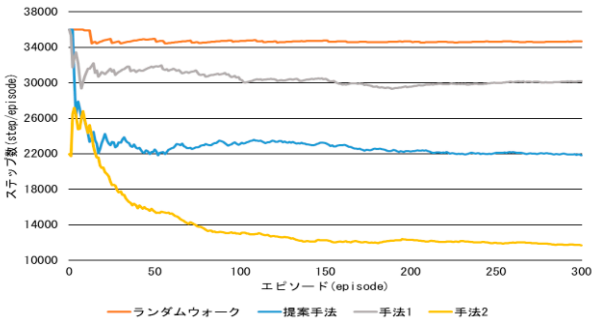

図 5. 捕獲にかかるステップ数の推移.

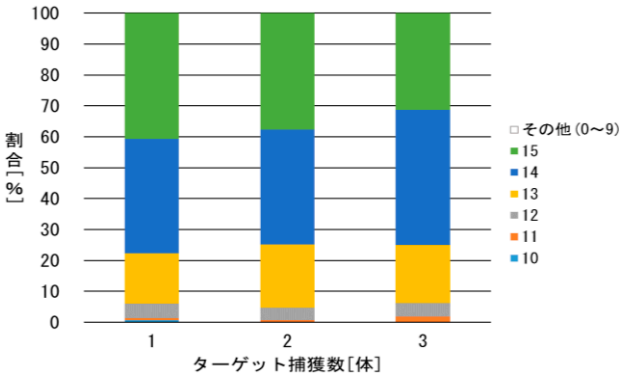

図 6. 学習後の上位層の選択の傾向.

図6の100%積み上げグラフは、全エピソード終了までに上位層のSAが何体のCAを配分したかを示す。横軸はターゲットの捕獲数である。図6から、ターゲットの捕獲数が増加するにつれて、追跡役を少なく、監視役を多く選択する傾向にあることが分かる。これは、監視役の方が追跡役より視野が広いため、少しでもターゲットを発見する確率を上げるよう指揮役 SA が学習を行った結果と考えられる。つまり、指揮役 SA はターゲットに合わせた役割配分を学習できているといえる。

5. おわりに

本研究では、視野が制限された現実的な環境でも効率よくターゲットを捕獲するために、エージェントに役割を付与し、エージェント同士の協調行動の視点から行動を選択する MSRAS を提案した。具体的には、エージェントに指揮、追跡、監視の3役のいずれかを設定し、階層化を行う。

追跡問題により、視野制限を行った従来手法と、ターゲットの捕獲にかかるステップ数の比較を行った。実験の結果、視野制限を行った従来手法よりターゲットを約8,000 ステップ早く捕獲できることを示した。ターゲットの捕獲数が増加するにつれて、監視役が多く選択されていることから、指揮役 SA がターゲットに合わせた役割の配分を行っていることを実験により示した。

今後の課題として、ターゲットが視野外時の探索手段がランダムウォークのみなので、効率的に探索するアルゴリズムを導入すること、エージェントに与える報酬の値をヒューリスティックに定めたので、報酬の値も学習するようなアルゴリズムの導入をすることなどが挙げられる。

参考文献

1) R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, MIT press (1998).
2) 渡邊俊彦, 和田竜也, “マルチエージェント追跡 問題のための相対座標系に基づく階層型モジュ ラー強化学習”, バイオメディカル・ファジィ・シ ステム学会誌, Vol. 12, No. 2, pp. 65–74 (2010).
3) 賀数元春, 深海悟, “障害物で視界が限定されたマルチエージェント追跡問題に関する研究”, 第 78 回全国大会講演論文集, 第 2016 巻, pp. 381–382 (2016).
4) 高玉圭樹, マルチエージェント学習: 相互作用の謎に迫る, コロナ社 (2003).
5) C. J. Watkins and P. Dayan, “Q-learning”, Machine learning, Vol. 8, No. 3-4, pp. 279–292 (1992).