各目的の Critic のうち最大 TD-error を用いて Actor を更新する多目的強化学習

Multi-Objective Reinforcement Learning Algorithm Updating Actor by Using Maximum TD-error Extracted from Some Critics

○正　長峰大智 (東洋大)　正　山田和明（東洋大）

Taichi NAGAMINE, Toyo University, s16A01500932@toyo.jp
Kazuaki YAMADA, Toyo University

Multi-agent system (MAS) is constructed by many autonomous agents. Conflicts occur in MAS because of complex interactions among many agents. An agent needs to carry out a task and to avoid conflicts at same time. That is, each agent has to achieve the contradicting purposes. Therefore, this paper proposes a new approach by using multi-objective reinforcement learning as decision making system of an agents. We investigate the efficiency of the proposed approach through a simulation experiment that two agents pass each other in the narrow path.

Key Words: Multi-agent system, Multi-objective reinforcement learning, Max-Min Actor-Critic

1　緒言

マルチエージェントシステム（Multi-Agent System: MAS）[1] は，中央集権的な管理機構を持たず，多数の自律エージェントが環境や近傍のエージェントとの相互作用を通してシステム全体の秩序を形成するという特徴を持つ．そのため，MAS はシステム内外の環境変化に対して頑健であるとされている．しかし，多数のエージェントが相互作用するためエージェント間に複雑なダイナミクスが発生する．そのため，設計者が予めエージェントが遭遇するすべての状況を想定し，適切な行動をエージェントに組込むことは極めて困難である．この課題に対し，各エージェントの意思決定機構として強化学習 [2] を用い，協調行動や競合回避行動を学習させるマルチエージェント強化学習（Multi-Agent Reinforcement Learning: MARL）[3] が注目されている．

強化学習の枠組みでは，学習エージェントは，観測した状態に対して行動を実行し，その評価として環境から得られる報酬を頼りに，目的を達成するために必要な状態-行動間の関係を試行錯誤的に学習する．そのため，シングルエージェントの学習において，設計者は，エージェントが目的を達成したときにのみスカラ型の報酬を与えるよう報酬系を設計すればよい．一方，MAS の学習に従来の強化学習をそのまま適用する場合，設計者は，両方のエージェントが目的を達成したときのみ報酬を与えるよう報酬系を設計することになる．しかし，MAS では多くの場合，各エージェントの目的とシステム全体の目的は競合する．そのため，個々のエージェントが各自の目的を達成しても，システム全体の目的が達成できないといった問題が発生する．

従来の強化学習に対し，複数の目的を設定して各目的を満足するよう学習させる多目的強化学習（Multi-Objective Reinforcement Learning: MORL）[4] が提案されている．MORL は，従来の強化学習が目的を達成したときにスカラ型の報酬を与えるのに対し，目的ごとに異なる報酬を設定したベクトル型の報酬を与える．また，MORL は，目的ごとに報酬を設計するため，MAS 環境において各エージェントの目的とシステム全体の目的が競合する場合でも円滑に学習することが期待できる．

そこで我々の研究グループでは，多目的強化学習の一種である Max-Min Actor-Critic (MMAC)[5] をマルチエージェントシステムに適用し，その有効性を検証してきた [6, 7, 8]．MMAC は，環境から与えられる報酬の種類だけ評価器（Critic）を用意し，ある状態 s において最小の状態価値関数 $V(s)$ を出力する評価器を選択し，その TD 誤差を用いて行動器（Actor）の政策 π を改善する．すなわち，MMAC は最小の状態価値 $V(s)$ を持つ状態 s の改善に主眼を置いている．そのため，正の報酬を得る状態より負の報酬を得る状態の改善を優先する傾向があった．そこで本稿では，MMAC の枠組みを用い，複数の評価器のうち最大の TD 誤差により行動器の政策 π を改善する新しい多目的強化学習を提案する．すなわち，提案手法は，期待報酬と実際の報酬の誤差が最大の状態 s，すなわち，報酬の予測精度が最も低い状態 s の改善に主眼を置いている．そのため，学習初期では負の報酬を頼りに学習し，学習が進展した学習後期では正の報酬を頼りに学習することが期待できる．

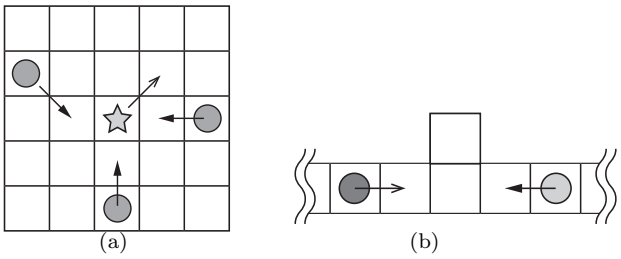


Fig.1 Multi-agent problem.

本稿では，次章において多目的強化学習を MAS に適用する利点について述べ，3 章では提案手法のアルゴリズムについて詳述する．4 章において，従来手法と提案手法の学習の特徴を計算機実験により検証する．そして，5 章において本稿のまとめと今後の課題について述べる．

2　多目的強化学習を MAS に用いる利点

強化学習は，環境から得られる報酬を最大化する行動規則を学習する．そのため，強化学習をマルチエージェントシステムに適用する場合，個々のエージェントの最適化とシステム全体の最適化が必ずしも一致しないという問題が発生する．例えば，協調問題の一つである追跡問題の場合（Fig.1(a)），複数のエージェントが協力して獲物を包囲し，捕獲する必要がある．しかし，獲物を捕獲したエージェントのみに報酬を与えた場合，獲物を追い込むエージェントの学習が進まないという問題が発生する．一方，競合回避問題の一つである狭路すれ違い問題の場合（Fig.1(b) に示す），1 章で述べた通り，各エージェントの目的とシステム全体の目的が一致しないという問題が発生する．

従来研究では，この問題を解決するために，次のアプローチが提案されている．追跡問題において，保知ら [9] は，エージェント群が獲物を捕獲したとき，システム全体に与えられる報酬を各エージェントの貢献度に併せて分配する方法を提案している．また，張ら [10] は，獲物を捕獲したとき，各エージェントに与えられる報酬とシステム全体に与えられる報酬を，各エージェントの貢献度にあわせて分配する方法を提案している．一方，狭路すれ違い問題において，市川ら [11] は，エージェント間の学習進度が

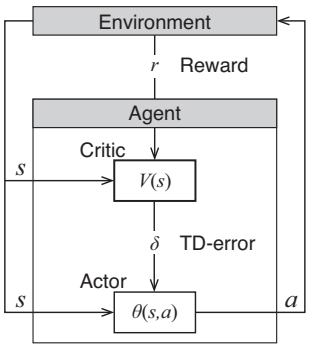


Fig.2 Actor-Critic.

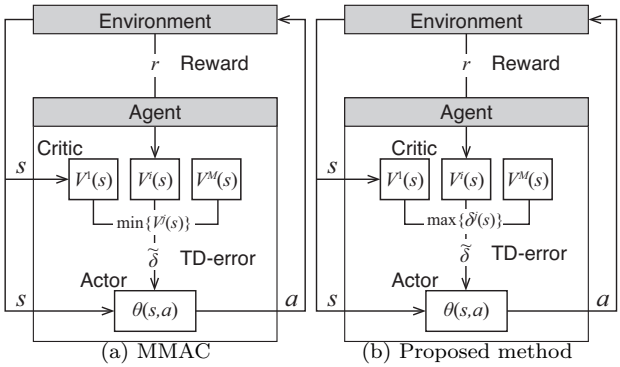


Fig.3 MMAC and proposed method.

揃うよう各エージェントの学習パラメータ（学習率と割引率）を調整することで，競合回避行動を獲得している．この手法では，個々のエージェントが自らの目的を達成したときのみ報酬を与える．しかし，各エージェントの学習進度を互いに共有し，どちらか一方の学習が収束し，他方の目的達成を阻害するとき，学習パラメータを調整することで，システム全体の目的を達成する確率を高めている．

このように，従来の強化学習をマルチエージェントシステムに適用する場合，個々のエージェントの最適化とシステム全体の最適化のバランスを取る仕組みが必要となる．その理由は，従来の強化学習が，個々のエージェントの報酬 r_l，あるいは，システム全体の報酬 r_g，または，その両方の報酬を一つの価値関数で管理することに起因する．例えば，エージェントが r_l または r_g のどちらか一方を受け取った場合，エージェントは与えられた報酬を足し合わせて一つの価値関数で管理する．そのため，エージェントは，与えられた報酬が個々のエージェントの行動を強化する報酬なのか，あるいは，システム全体の目的を達成する行動を強化する報酬なのか判断できない．また，両方の報酬を受け取った場合，報酬同士が打ち消し合う恐れがある．そこで本稿では，多目的強化学習の一種である Max-Min Actor-Critic (MMAC)[5] を改良した新しい多目的強化学習をマルチエージェントシステムに適用する．

3 多目的強化学習

本節では，従来の強化学習である Actor-Critic (AC)，AC を複数の目的が扱えるように拡張した Max-Min Actor-Critic (MMAC)[5]，そして，MMAC を改良した提案手法の概要を説明する．

3.1 Actor-Critic (AC)

Actor-Critic は，Fig.2 に示すように行動器（Actor）と評価器（Critic）から構成されている．行動器は，エージェントが観測した状態 s_t において行動 a_t を実行する方策 $\pi(s_t, a_t)$ の行動価値関数 $\theta(s_t, a_t)$ に基づいて実行する行動を確率的に選択する．評価器は，環境から与えられる報酬を頼りに状態 s_t における状態価値関数 $V_t(s_t)$ を推定する．そして，状態価値の TD 誤差によって，実行した方策の行動価値を更新する．評価器と行動器の学習は次のように行われる．

まず，評価器は，時刻 t における状態 s_t の状態価値関数 $V_t(s_t)$ と環境から与えられる報酬 r から TD 誤差 δ_t を次式により求める．

$$\delta_t = r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t) \tag{1}$$

評価器の状態価値関数 $V_t(s_t)$ は，学習率を $\alpha_c (0 < \alpha_c < 1)$ とすると，次式により更新される．

$$V_{t+1}(s_t) = V_t(s_t) + \alpha_c \delta_t \tag{2}$$

行動器の行動価値関数 $\theta_t(s_t, a_t)$ は，状態 s_t で方策 $\pi(s_t, a_t)$ に従い行動 a_t を選択し，次状態 s_{t+1} に遷移したとき，学習率を $\alpha_a (0 < \alpha_a < 1)$ とすると，次式により更新される．

$$\theta_{t+1}(s_t, a_t) = \theta_t(s_t, a_t) + \alpha_a \delta_t \tag{3}$$

3.2 Max-Min Actor-Critic (MMAC)

MMAC は，複数の種類の報酬をベクトル型で受けることにより多目的な学習ができる．Fig.3(a) に示すように MMAC の評価器は，環境から与えられる複数の種類の報酬 $r = (r_1, \ldots, r_M)^T$ の数だけ，状態価値関数 $V = (v_1, \ldots, v_M)^T$ を持つ．MMAC は，報酬 $r = (r_1, \cdots r_M)^T$ が与えられると，評価器 i における状態価値の TD 誤差を次式により求める．

$$\delta^i = r_t^i + \gamma_i V_t^i(s_{t+1}) - V_t^i(s_t), \quad i = 1, \ldots, M \tag{4}$$

評価器の各状態価値関数は，学習率を α_c とすると，次式により更新される．

$$V_{t+1}^i(s_t) = V_t^i(s_t) + \alpha_c \delta^i \tag{5}$$

行動器では，まず，拡張 Max-Min 法を用いてベクトルの TD 誤差を下記の (6) 式によりスカラー化した $\tilde{\delta}$ を求める．そして，学習率を α_a とすると，行動価値を (7) 式により更新する．

$$\begin{aligned} \tilde{\delta} &= \delta^k, \\ k &= \arg\min_i \{V^i\} \end{aligned} \tag{6}$$

$$\theta_{t+1}(s_t, a_t) = \theta_t(s_t, a_t) + \alpha_a \tilde{\delta} \tag{7}$$

ただし，ζ は $[0, 1]$ の正の定数である．なお，$V^i + \zeta\delta^i$ を最小にする i が複数存在する場合はその中から等確率でランダムに選択する．

3.3 提案手法

提案手法は，MMAC の枠組みを用い，行動器の更新に用いる TD 誤差を下記の通り求める点が異なる．

$$\begin{aligned} \tilde{\delta} &= \delta^k, \\ k &= \arg\max_i \{\delta^i\} \end{aligned} \tag{8}$$

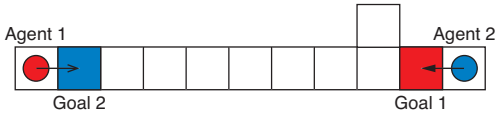
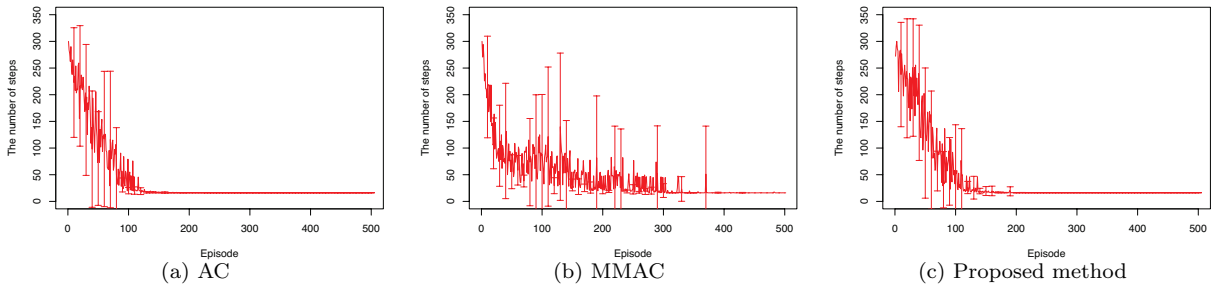
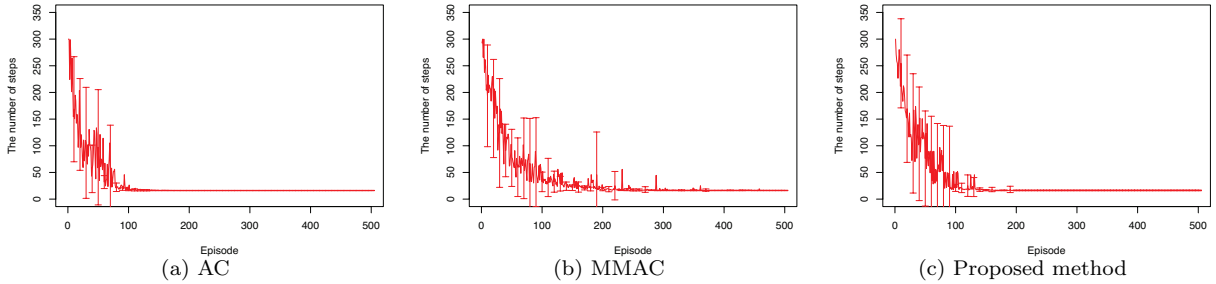


Fig.4 Simulation setting.



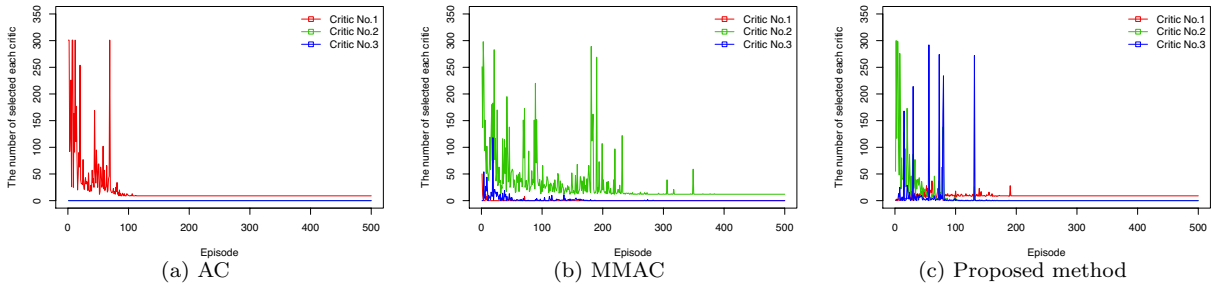
(a) AC (b) MMAC (c) Proposed method

Fig.5 Learning results ($r_{nogoal} = -1.0$)



(a) AC (b) MMAC (c) Proposed method

Fig.6 Learning results ($r_{nogoal} = -5.0$).



(a) AC (b) MMAC (c) Proposed method

Fig.7 The number of selected times of each critic ($r_{nogoal} = -5.0$).

4 計算機実験

4.1 実験設定

本実験では，AC，MMAC，提案手法の学習プロセスを分析するために，Fig.4 に示す狭路すれ違い問題に適用する．各エージェントの目的は，スタートから同じ番号のゴールに到達することである．しかし，各エージェントが利己的に行動すると互いにゴールに到達できない．実験では，エージェントが 300 ステップ行動するか，両方のエージェントがゴールしたとき，エピソードを更新する．そして，500 エピソードを 1 試行とし，10 試行繰り返す．従来研究 [11] では，エージェントがゴールしたとき正の報酬 r_{goal} を与え，それ以外のとき負の報酬 r_{nom} を与える．本実験では，さらに 300 ステップ行動しても両方のエージェントがゴールできなかったとき負の報酬 r_{nogoal} を与える．各エピソードにおいて，ゴールに到達したエージェントは，エピソードが更新されるまでゴールに留まり，行動選択や状態価値，行動価値の更新はしない．ただし，両方のエージェントが 300 ステップ以内にゴールできなかった場合，エピソード更新時に双方のエージェントに r_{nogoal} を与える．エージェントは，入力として自分と相手のエージェントの座標を知覚できる．また，行動として上下左右と停止のいずれかを選択して実行する．

実行する行動方策は Soft-max 法 [1] により選択し，逆温度パラメータを 0.01 とする．行動器と評価器の学習率をそれぞれ $\alpha_a = 0.1$，$\alpha_c = 0.1$ とし，割引率を $\gamma = 0.9$ とする．また，状態価値 V と行動価値 θ の初期値をそれぞれ $V_0 = 1.0$，$\theta_0 = 1.0$ とする．実験では，エージェントがゴールすると $r_{goal} = 5.0$ を与え，それ以外の行動を実行したとき $r_{nom} = -0.01$ を与える．そして，両方のエージェントが 300 ステップ以内にゴールできなかったときに与える報酬 r_{nogoal} を-1.0，-5.0 と変化させた場合の学習結果を比較する．

4.2 学習結果

AC，MMAC，提案手法において，両方のエージェントがゴールするまでに要したステップ数の平均と標準偏差を図 5 と 6 に示す．なお，図 5 は，両方のエージェントがゴールできなかったとき与える報酬を $r_{nogoal} = -1.0$ としたときの結果であり，図 6 は，$r_{nogoal} = -5.0$ とした場合の結果である．上記の実験結果から MMAC の学習の収束が他の手法より遅いことがわかる．また，報酬 r_{nogoal} の値が-1.0 よりも-5.0 のときの方が学習の収束が早いことがわかる．

次に，AC，MMAC，提案手法において，行動器（Actor）を更新するために用いた各評価器（Critic）の使用回数を図 7 に示す．なお，図中の Critic 1（赤線）は，ゴールした各エージェントに与えられる報酬 $r_{goal} = 5.0$ を扱う評価器，Critic 2（黄緑線）は，ゴール以外の状態で与えられる報酬 $r_{nom} = -0.01$ を扱う評価器，Critic 3（青線）は，両方のエージェントがゴールできなかったとき与えられる報酬 r_{nogoal} を扱う評価器である．MMAC は，最小の状態価値関数 $V(s)$ を出力する評価器（Critic）の TD

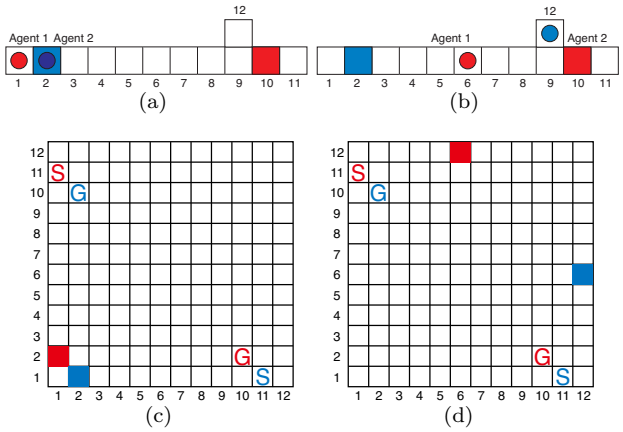


Fig.8 The relationship between agent position and state-value function.

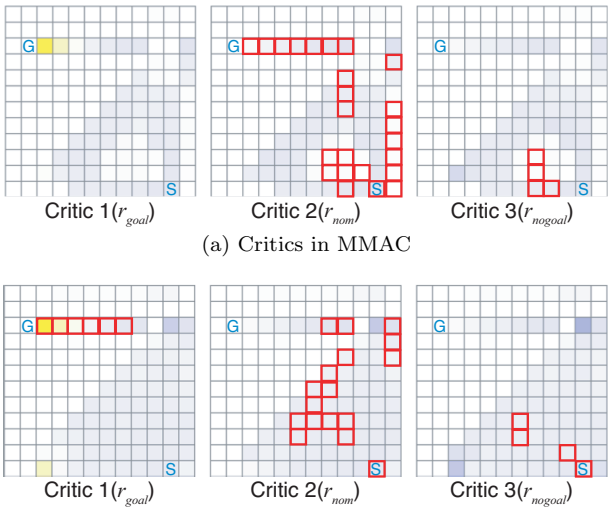


Fig.9 The extracted critics in order to update an Actor.

誤差を用いて Actor の政策を更新する．そのため，図 7(b) を見ると Critic 2 と 3 を用いて学習し，Critic 1 はほぼ使われていない．一方，提案手法は，期待報酬と実際の報酬の差が大きい，すなわち，報酬の予測精度が高くない状態 s の状態価値関数 $V(s)$ を出力する評価器（Critic）の TD 誤差を用いて行動器（Actor）の更新を行う．そのため，図 7(c) を見ると，学習初期は Critic 2 と 3 を多用し，ゴール周辺では，Critic 1 を用いて学習していることがわかる．

4.3 評価器内の状態価値関数

本節では，MMAC と提案手法の評価器内の状態価値関数の学習プロセスを分析する．まず，シミュレーション上におけるエージェント間の関係を評価器内の状態価値関数にマッピングする．本研究では，図 8(a) のように各セルに 1〜12 までの数字を割り振り，図 8(c) のように，例えば赤エージェントは，自分はセル 1 におり，相手エージェントがセル 2 にいるため，図 8(c) の赤色のセルのところにマッピングされる．また，図 8(b) に示すように，エージェントが脇道にいるときは，図 8(d) のようにマッピングされる．また，図 8(c) と図 8(d) の S と G は，それぞれスタートとゴール地点を表す．

次に，図 9(a) に MMAC の学習途中の状態価値関数の様子を示す．図 9(a) から MMAC が Critic 2 と Critic 3 を用いて学習していることがわかる．一方，図 9(b) から提案手法がゴールから遠い状態では，Critic 2 と Critic 3 を用いて学習し，ゴール付近では Critic 1 を用いて学習していることがわかる．したがって，MMAC は，最小の状態価値関数を出力する状態 s の TD 誤差を用いて学習するため，学習に時間が掛かることがわかる．一方，提案手法は報酬の予測精度が悪い状態の状態価値関数を用いて学習するため，MMAC に比べて学習の収束が改善されている．

5 おわりに

本稿では，多目的強化学習の一つである Max-Min Actor-Critic (MMAC) の枠組みを用い，行動器（Actor）の政策を更新するとき，最大の TD 誤差を出力する Critic を選択し，選択された Critic の TD 誤差を用いて Actor を更新する学習法を提案した．Actor-Critic，MMAC，提案手法を狭路すれ違い問題（提案手法を各エージェントの目的とシステム全体の目的が異なる競合問題）に適用し，分析した．その結果，提案手法は MMAC より学習の収束が早くなることが判明した．．今後の課題として，多数のエージェントからなるマルチエージェント環境において，多目的強化学習により協調行動や競合回避行動が獲得できるか検証する予定である．

謝辞

本研究の一部は，JSPS 科研費 18K11554 の助成を受けたものです．

参考文献

[1] 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)

[2] R. S. Sutton and A. G. Barto, Reinforcement Learning: An introduction, A Bradford Book (1998)

[3] Erfu Yang and Dongbing Gu, A survey on multiagent reinforcement learning towards multi-robot systems, IEEE 2005 Symposium on Computational Intelligence and Games, CIG'05. IEEE, 292/299 (2005)

[4] Chunming Liu, Xin Xu and Dewen Hu, Multiobjective Reinforcement Learning: A Comprehensive Overview, IEEE Transactions on Systems, Man, and Cybernetics: Systems, (**45**)-3, 385/398 (2015)

[5] 上岡拓未, 内部英治, 銅谷賢治, Max-Min Actor-Critic による複数報酬課題の強化学習, 電子情報通信学会論文誌. D, 情報・システム, (**90**)-9, 2510/2521 (2007)

[6] 西田公己, 山田和明, 多目的強化学習のマルチエージェントシステムへの適用, 第 59 回システム工学部会研究会資料 20/23 (2018)

[7] 山田和明, マルチエージェントシステムにおける多目的強化学習による競合回避行動の獲得, 日本機械学会 ロボティクス・メカトロニクス講演会 2018, CD-ROM, 2P2-F15, 北九州 (2018)

[8] 長峰大智, 山田和明, マルチエージェントシステムへの多目的強化学習の適用 - 学習プロセスの分析 -, 計測自動制御学会システム・情報部門学術講演会 2018 (SSI2018), SS06-06 (2018)

[9] 保知良暢, 松井藤五郎, 犬塚信博, 世木博久, マルチエージェント強化学習における報酬発生条件に基づく貢献度判別と報酬分配, 人工知能学会全国大会論文集, (**16**), 2D3-02 (2002)

[10] 張坤, 前田陽一郎, 高橋泰岳, 貢献度評価に基づくマルチエージェント強化学習の報酬分配, 日本知能情報ファジィ学会ファジィシステムシンポジウム講演論文集, (**26**), 246/251 (2010)

[11] 市川嘉裕, 高玉圭樹, 学習進度に基づくマルチエージェント Q 学習における競合回避, 計測自動制御学会論文集, (**48**)-11, 764/772 (2012)