令 和 3 年 度

修　士　論　文

学習データの共有による
マルチエージェントシステム学習手法の改良

電気システム制御プログラム　　社会情報学研究室
M203613　　劉　啓

指導教員

教　授　　西崎　一郎
准教授　　林田　智弘
助　教　　関崎　真也

令和４年２月１４日

広島大学大学院先進理工系科学研究科

あらまし

マルチエージェントシステム（MAS：Multi-Agent System）では，複数のエージェントが自律的に行動するため，マルコフ決定過程で記述することが難しくなる．一般的に，シングルエージェントシステム（SAS：Single-Agent System）でエージェントと環境の相互作用がマルコフ決定過程で記述することができるが，これを学習における不確実性と呼ばれる．不確実性を回避する方法として，深層強化学習アルゴリズムやエージェント間で学習データを共有する手法が有効である．本論文は MAS の不確実性を回避するため，Actor-Critic アルゴリズムを基礎としてエージェント間での学習データの共有による MAS の学習手法を提案する．提案手法では，MAS の各エージェントが他のエージェントが持つ全ての経験データではなく，一部の経験データを用いて学習することにより，MAS の学習効率の向上を目的とする．本論文では，様々な環境による 3 種類の学習データの共有アーキテクチャを提案した．複雑さが違う複数の迷路環境を用いたシミュレーション実験を実施し，提案手法の有効性を証明した．さらに，エージェント間での学習データを共有しない手法と全ての学習データを共有する手法の 2 つの手法との比較実験の結果により，学習データを一部共有する提案手法の成功率が高いと判明した．

目 次

第 1 章

はじめに

コンピュータ上の仮想空間において，行動主体となるエージェントが複数存在するシステムはマルチエージェントシステム（Multi-Agent System，MAS）と呼ばれる．MAS におけるエージェントの学習はエージェントが 1 体だけ存在するシングルエージェントシステム（Single-Agent System，SAS）における学習より複雑となることが従来の研究で示されている [1]．すなわち，SAS では 1 体のエージェントが環境との相互作用があり，多くの SAS で仮定されるマルコフ決定過程により記述可能である．一方で，MAS ではすべてのエージェントが環境に影響を与えることで，マルコフ決定過程が成立しない場合が一般的であり，学習は SAS の学習より不安定になりやすい．この MAS 学習の不安定性を MAS の不確実性と呼び，これを回避するための手法が数多く提案されている [2]．SAS の学習手法である深層強化学習（Deep Reinforcement Learning，DRL）を基礎として MAS への拡張手法であるマルチエージェント深層強化学習手法（Multi-Aagent Deep Reinforcement Learning，MADRL）[3] や MAS 内の全てのエージェントの経験データをうまく利用してシステムの学習アーキテクチャを構築する方法 [4] が注目を浴びている．

MAS の学習におけるシステムの不確実性を回避するため，MADRL が効果的であることが示されている [3]．深層強化学習手法である A3C(Asynchronous Advantage Actor-Critic) アルゴリズム [5] を SAS から MAS へ拡張して実際の自動運転システムの学習問題を解決した研究が発表された [6]．車輌をエージェントとして自動運転に適切な学習を考えると，複数のエージェントが存在する現実的な交通状況において MAS の学習をしなければならない．常に不規則的に変化している環境の中，Bacchiani *et al.*[6] は MADRL 手法を用いることで，MAS の不確実性を回避し，自動運転の車両に正しい行動を学習させた．MADRL の他，MAS 内の全てのエージェントが学習データを共有するような学習手法も MAS の不確実性の回避に有用であることが示されている [2]．Lowe *et al.*[4] は MAS 内の全てのエージェントが学習データを共有するような学習手法を提案した．提案された学習データの共有手法では，他のエージェントにより共有された学習データに基づいて，そのエージェントが選択する行動を推測するような学習方法が構築された．推測した最善な行動の価値を用いてエージェント自身の行動を改善し，他のエージェントが取る行動を把握できるように学習することによって MAS 学習の不確実性を回避する．

Lowe *et al.*[4] は，MAS におけるエージェント間の学習データはすべて共有するとされており，他のエージェントの経験データも参照することで，様々な状況にお

ける方策を効率的に学習することが可能であると考えられる．一方で，全てのエージェントが全ての情報を共有することで，不要な情報も参照することとなり，必ずしも学習効率が向上するとは限らない．

本論文では，エージェントの学習データの一部を共有する Actor-Critic アルゴリズム [7] を使用して MAS におけるエージェント学習効率を向上する学習手法の開発を目指す．エージェントの間の記録した情報データの一部を共有することにより，単一のエージェントより多い環境の情報を得ると同時に，全部データ共有による学習効率の低下も避けられる．本論文では，複数のエージェントが存在する迷路環境でのシミュレーション実験により，提案手法の有効性について検討する．

本論文の構成は次の通り，第 2 章で強化学習とマルチエージェントシステムを紹介する．第 3 章で，提案手法である 3 つの学習データの共有アーキテクチャを詳しく述べる．第 4 章で，シミュレーション実験を実施し，3 つの学習データの共有アーキテクチャの有効性を示す．主に，複雑さが違う迷路環境において，複数のエージェントが共に指定地点に到達する問題を考えている．最後の第 5 章で，本論文の成果と今後の課題を述べる．

第 2 章

強化学習とマルチエージェントシステム

2.1. 強化学習

マルコフ決定過程 (MDP：Markov Decision Process) は，離散時間で定義された状態遷移が確率的に生じる動的システムの確率モデルであり，将来の状態の条件付き確率分布が現在の状態のみ依存する特性であるマルコフ性を満たす数学モデルである．MDP は状態の集合 S と状態遷移確率行列 P によって定義される．具体的には，時刻 t の状態 s_t は状態遷移確率行列 P に基づいて次状態 s_{t+1} に遷移する．

強化学習 (RL：Reinforcement Learning)[8] とは，システムにおけるエージェントが環境と繰り返しの相互作用によってタスクを実行できるような手法である．図 2.1 のように，RL における 1 体のエージェントと環境の相互作用を示している．RL は時刻 t の状態 s_t，行動 a_t，行動選択方策 $\pi(a_t|s_t)$，報酬 r_t および割引率 γ という要素によって定義される．行動選択方策を求めるため，目的関数を $\sum_{t=0}^{T} \gamma^t r_{t+1}$ と定義する．ここで，終端時刻を T とする．目的関数を最大化するような行動選択方策 π を学習するのが目的である．そして，複数のエージェントが存在する場合，時刻 t におけるエージェント i（i=1, 2, ..., N）が観測できる環境状態を s_t^i，状態集合を $\mathcal{S}$，行動を a_t^i，行動集合を $\mathcal{A}$，行動選択方策を π とする．各エージェントは観測状態に基づいて行動を選択し，時刻 t における全てのエージェントの行動の組を $A_t^{joint} = (a_t^1, a_t^2, \ldots, a_t^N)$ とする．環境は A_t^{joint} に基づいて，報酬 r_t と次の状態 s_{t+1}^i を決定する．

MDP が RL におけるエージェントと環境の相互作用を記述する数理モデルとして使われることが現状である．RL では，MDP の方策 π と状態遷移確率を結合させることで状態遷移確率行列 P としている．RL の手法として，行動選択機能と行動評価機能を分離する Actor-Critic アルゴリズムが使用されている．Actor-Critic アルゴリズムを使用する場合，エージェントは行動選択機能によって行動を選択し，行動評価機能によって選択した行動を評価しつつ学習する．学習中，現時点で推定する状態価値と学習の目標値である次の時点の価値の差分である TD 誤差 (Temporal Difference Error) が使用されている．本論文では RL を使用し，マルチエージェントシステムの学習手法を構築する．

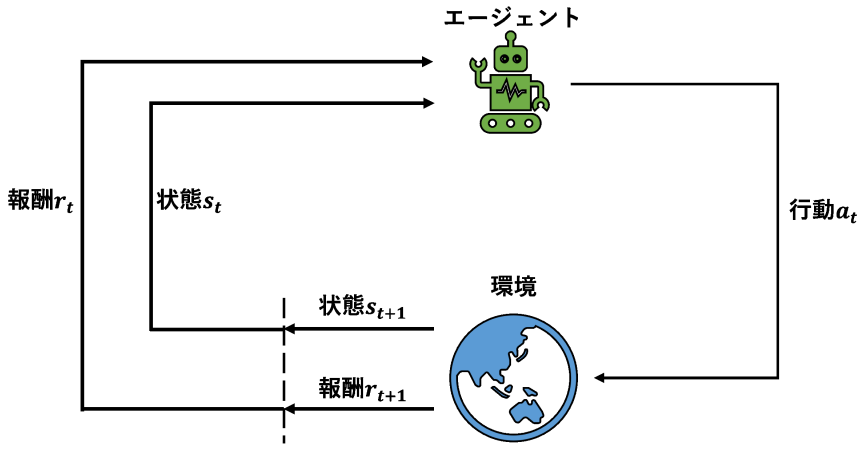

図 2.1: 強化学習

2.2. Actor-Critic

Actor-Critic アルゴリズムは，図 2.2 のように，エージェントの行動選択に対応する Actor とエージェントの行動評価に対応する Critic の 2 つから構成される方策勾配系のアルゴリズムである．環境の観測状態 s に基づいて，エージェントは行動選択器 Actor によって行動 a を選択し，環境と相互作用をする．報酬 r と行動評価器 Critic である価値関数の値によって TD 誤差を算出する．TD 誤差を使用して Actor と Critic の損失関数を計算し，損失関数を使用してパラメータに関する勾配を算出してパラメータを更新するように学習する．すなわち，行動評価器によって選択した行動を評価するような学習アーキテクチャとなる．行動評価器 Critic の学習によって，より正確に行動を評価できるようになり，正しい評価値に基づいて行動選択器をより正確に学習させる．行動価値関数を着目点として学習を展開する Q 学習や Sarsa と違い，方策勾配法は方策自体を着目点として学習する．特に，連続行動空間を扱う問題の場合，方策勾配法がよく使用されている．行動選択器 Actor はエージェントの方策 $\pi(a|s,\theta)$ を，行動評価器 Critic は状態価値関数 $V_\omega(s)$ を近似的にモデル化したものである．エージェントは $V_\omega(s)$ の最大化を目的として学習する．

終端時刻を T，報酬関数を $R(a_t,\ s_t)$ として，時刻 t の方策 π に基づく目的関数 $\mathcal{J}(\theta)$ を式 (2.1) とする．方策 $\pi(a|s)$ のパラメータ θ に関する $\mathcal{J}(\theta)$ の勾配は式 (2.2) になる．目的関数を最大化することは，式（2.3）で与えられる損失関数を最小化することとなる．

式 (2.2) と方策勾配定理 [7] に基づき，実際の Actor，Critic の損失関数を式 (2.4), (2.5) とする．式 (2.6) は TD 誤差であり，報酬関数に代用する．γ は割引率である．

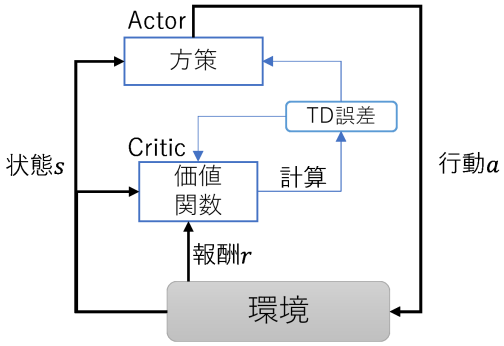

図 2.2: Actor-Critic の概念図

Actor の損失関数を時刻 t から終端時刻 T までの収益の期待値の逆値とする.

$$\mathcal{J}(\theta) = \sum_{a_t} \pi(a_t|s_t,\theta) R(a_t,s_t) \tag{2.1}$$

$$\begin{aligned}
\nabla_\theta \mathcal{J}(\theta) &= \nabla_\theta \sum_{a_t} \pi(a_t|s_t,\theta) R(a_t,s_t) \\
&= \sum_{a_t} \nabla_\theta \pi(a_t|s_t,\theta) R(a_t,s_t) \\
&= \sum_{a_t} \frac{\pi(a_t|s_t,\theta)}{\pi(a_t|s_t,\theta)} \nabla_\theta \pi(a_t|s_t,\theta) R(a_t,s_t) \\
&= \sum_{a_t} \pi(a_t|s_t,\theta) \frac{\nabla_\theta \pi(a_t|s_t,\theta)}{\pi(a_t|s_t,\theta)} R(a_t,s_t) \\
&= 1 \times \nabla_\theta \log \pi(a_t|s_t,\theta) R(a_t,s_t) \\
&= \nabla_\theta \log \pi(a_t|s_t,\theta) R(a_t,s_t)
\end{aligned} \tag{2.2}$$

$$\mathcal{L}_{actor}(\theta) = -\mathcal{J}(\theta) = -\log \pi(a_t|s_t,\theta) R(a_t,s_t) \tag{2.3}$$

$$\mathcal{L}_{actor}(\theta) = -\frac{1}{T} \sum_{t=1}^{T-1} \log \pi(a_t|s_t,\theta) \delta_{t+1}(\omega) \tag{2.4}$$

$$\mathcal{L}_{critic}(\omega) = \sum_{t=1}^{T-1} |\delta_{t+1}(\omega)|^2 \tag{2.5}$$

$$\delta_{t+1}(\omega) = r_t + \gamma V_\omega(s_{t+1}) - V_\omega(s_t) \tag{2.6}$$

式 (2.7)，(2.8) によりエピソード毎に Actor と Critic の重みを更新する．ここで，α は学習率である．

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \mathcal{L}_{actor}(\theta) \tag{2.7}$$

$$\omega \leftarrow \omega + \alpha \nabla_{\omega} \mathcal{L}_{critic}(\omega) \tag{2.8}$$

2.3. 深層強化学習（DRL）

深層強化学習（DRL：Deep Reinforcement Learning）とは，深層学習（Deep Learning）と強化学習の組み合わせた手法である [9]．主に，深層学習によって観測状態となる画像の特徴の抽出や学習対象となる行動方策のモデル化などを実現し，強化学習によってタスクを実現できるように学習する．深層学習において，多層のニューラルネットワークを積み重ねる深層ニューラルネットワーク（DNN：Deep Neural Network）とその学習の技術体系が存在する．特に，DNN である畳み込みニューラルネットワーク (CNN：Convolutional Neural Network) と再帰型ニューラルネットワーク (RNN：Recurrent Neural Network) が有効なモデルとして応用されている．CNN は，画像などのデータから，畳み込み演算により画像が持つ空間情報を次の層に渡す操作を繰り返して，画像の特徴量を抽出する方法である．抽出した特徴量を分類問題に適用すれば，画像分類問題を予測するモデルを学習できる．RNN は，現在の層の情報だけではなく，前の層の入力も考慮した学習モデルであり，言語の自動翻訳や文書生成などに応用されている．これらの DNN が深層強化学習へ拡張される．

強化学習において，環境の状態をモデル化する必要があるため，学習する前に状態を離散化するなどのデータの処理をする必要がある．現実的な問題を考えると，例えば，自動運動のような路面状況の画像を環境の状態として使用する場合，画像データを強化学習によって処理することは難しい．一方で，特徴量を抽出するなどの画像データに対する処理能力を持つ深層学習は，状態と行動の相互作用が連続するような最適化問題を解決する強化学習の機能を所持していない．深層学習と強化学習を結合し，両方の長所を発揮する方法が DRL である．DRL では，深層学習を用いて環境の探知を行って学習データを獲得し，獲得した学習データを強化学習によって制御方策を学習する．さらに，強化学習において，複雑な問題を解決するための方策や環境の報酬関数などをモデル化する関数近似器が DNN を適用できる．DNN を用いて，方策や報酬関数などに代用し，より正確的に問題を解決できる．

DRL の 1 つの手法として，A3C(Asynchronous Advantage Actor-Critic) アルゴリズム [5] が提案されている．A3C アルゴリズムは，1 体のエージェントが複数の複写体を生成し，各複写体を複数の環境の複写体で学習させ，それぞれの学習結果を収

集し，1 つのすべての複写体が共有する Actor-Critic の学習に反映するような学習アーキテクチャである．すべての複写体が共有する Actor-Critic は Actor である方策 $\pi(a|s;\theta)$ のパラメータ θ と Critic である価値関数 $V(s;\omega)$ のパラメータ ω を持ち，global shared network（GSN）と呼ばれる．エージェントの複写体 i（i=1, 2, ..., N）は GSN のパラメータ θ と ω をコピーした Actor-Critic ネットワークである thread special network(TSN) を用いて環境の複写体と相互作用する．各複写体の TSN のパラメータは Actor である方策 $\pi(a|s;\theta_i')$ のパラメータ θ_i' と Critic である価値関数 $V(s;\omega_i')$ のパラメータ ω_i' になる．1 体のエージェントの複写体と 1 つの環境の複写体を 1 組とし，1 単位のスレッドとする．A3C は同時に複数のスレッドの相互作用をして GSN のパラメータ θ と ω を更新する．すなわち，i 番目のスレッドの相互作用が終端状態に到達するか一定の時間ステップに到達するかとすると，TSN のパラメータ θ_i'，ω_i' に関する目的関数の勾配更新量 $\nabla_{\theta_i}\mathcal{L}_{actor}(\theta_i)$ と $\nabla_{\omega_i}\mathcal{L}_{critic}(\omega_i)$ を計算し，それを GSN に送信する．GSN のパラメータ θ，ω を式 (2.7) と式 (2.8) を用いて更新する．2.2 節で述べた手順で，目的関数の勾配更新量を計算する．各スレッドは他のスレッドの相互作用に影響を与えない．一定の時間ステップに到達すると，TSN は GSN のパラメータ θ，ω をコピーし，i 番目の TSN パラメータ θ_i'，ω_i' とする．複写体の数が 4 の時の A3C のアーキテクチャを図 2.3 に示す．

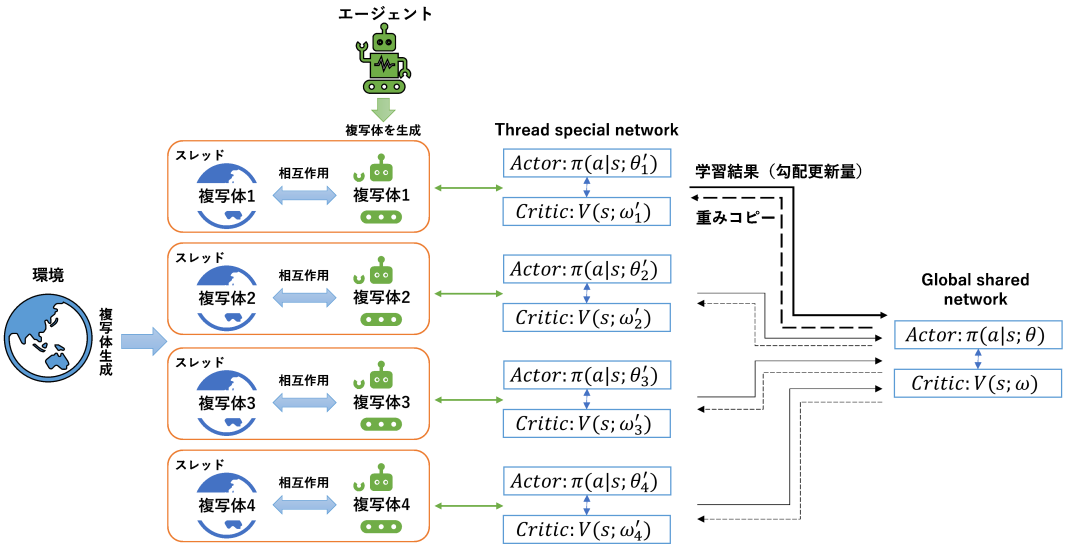

図 2.3: A3C の概念図

2.4. マルチエージェントシステム（MAS）

MAS とは，複数のエージェントが環境と相互作用するシステムである．エージェントが独立して環境と相互作用するため，環境への影響は 1 体のエージェントだけが存在する SAS に比べて複雑であると言える．MAS の学習手法の選定については，Ovalle *et al.*[10] は DRL を用いた SAS から MAS へ拡張した学習手法を提案している．主に，深層学習のニューラルネットワークによって行動価値関数をモデル化する深層 Q 学習手法を SAS から MAS へ拡張する手法を提案した．一方で，Foerster *et al.*[11] はすべてのエージェントがアクセス可能な価値関数を共有するような手法を提案している．しかし，各エージェントが価値関数を更新する時に自身の経験データのみを用いることで，局所的な情報に基づく関数値の振動現象などにより，適切な学習が難しい場合があることがあった．すなわち，MAS には，他のエージェントの行動の影響による環境変化における不確実性が存在するうえ，エージェントの経験データが不十分であるために適切な学習が難しい場合がある．この問題を回避するため，MAS における複数のエージェントの間で経歴データを共有する手法が提案されている [12]．ここで，経験データとは状態，行動，報酬からなる情報であり，エージェント間の情報共有を導入することで，SAS から拡張した手法を MAS の学習手法として適切に適用することができることを示している．

2.5. MAS におけるエージェント間の情報共有

エージェント間の情報共有を基礎として DRL を拡張して MAS の学習手法として用いることを考える．DRL である A3C をそのまま MAS へ拡張すると，MAS では複数のエージェントが存在するため，すべてのエージェントが複写体を生成する場合，複写体が膨大な数の必要となり，数多くの複写体から収集した経験データを蓄積することが学習効率の低下につながる．膨大な経験データによるシステムの計算時間や計算機の能力なども学習効率に影響を与える．

先行研究 [4] では，N 体エージェントが存在する MAS において，エージェント i（i=1, 2, ..., N）は自身に関する環境の状態 s^i に基づいて方策 π_i により行動 a^i を選択して環境と相互作用をする．MAS の不確実性を回避し，エージェントと環境の相互作用を把握するため，すべてのエージェントの情報を共有するような学習手法を提案した．エージェント i（i=1, 2, ..., N）はパラメータ θ_i を持つ $\pi(a^i|s^i;\theta_i)$ を方策として環境と相互作用する．現時刻のすべてのエージェントの環境の状態から結成する環境の状態のベクトルを $S = (s^1, s^2, \ldots, s^N)$ とする．$Q_i(S, a^1, a^2, \ldots, \pi(a^i|s^i;\theta_i), \ldots, a^N)$ は S とすべてのエージェントの行動を入力とする行動価値関数である．次時刻のすべてのエージェントの環境の状態のベクトルを $S' = (s^{1,\prime}, s^{2,\prime}, \ldots, s^{N,\prime})$，エージェント i の次の行動を $a^{i,\prime}$，エージェント i が他のエージェント j の次の行動を予測する方策を行動予測器 $\hat{\mu}_i^j(s^j)$ とする．エージェント i が $(N-1)$ 個行動予測器を用いて

他の $(N-1)$ 体エージェントの次の行動 $a^{j,\prime}(j \neq i)$ を予測する．行動価値関数と式 (2.9) で示す目標値 y の最小二乗誤差を損失関数とし，式 (2.10) で示す．

パラメータ ϕ_i^j を持つ行動予測器 $\hat{\mu}_i^j(s_j)$ の損失関数を式 (2.11) で示す．λ は割引率，式（2.12）のように，$H(\hat{\mu}_i^j)$ は行動予測器の方策エントロピーである．エージェント i は他のエージェント j の経験データを用いて損失関数 (2.11) を最小化するようにパラメータ ϕ_i^j を方策勾配法で更新して行動予測器を学習させる．そして，他のエージェントの予測行動を用いて損失関数 (2.10) を最小化するように方策のパラメータ θ_i を方策勾配法で更新する．すべてのエージェント経験データは $[S, S', a^1, \ldots, a^N, r^1, \ldots, r^N]$ として記録する．r^i はエージェント i の報酬である．

$$y = r^i + \gamma Q_i(S', a^{1,\prime}, a^{2,\prime}, \ldots, a^{i,\prime}, \ldots, a^{N,\prime}) \tag{2.9}$$

$$\mathcal{L}(\theta_i) = [Q_i(S, a^1, a^2, \ldots, \pi(a^i|s^i;\theta_i), \ldots, a^N) - y]^2 \tag{2.10}$$

$$\mathcal{L}(\phi_i^j) = -[\log \hat{\mu}_i^j(s^j) + \lambda H(\hat{\mu}_i^j)] \tag{2.11}$$

$$H(\hat{\mu}_i^j) = -\sum_{a^j} \hat{\mu}_i^j(s^j) \log \hat{\mu}_i^j(s^j) \tag{2.12}$$

計算時間や計算機の能力などの現実的な問題によるエージェント間の経験データの伝達に関する制限が存在する場合，A3C のようにすべてのエージェントの経験データを共有する方法の適用は難しくなる．先行研究 [4] のようにエージェント i が自身の方策以外に他のエージェントの行動推測器も学習する必要があり，計算時間の増加によって学習効率の低下が起こる．本論文では，すべての経験データの共有による学習効率低下の状況を回避するため，すべてのエージェントの経験データを共有することではなく，一部の経験データを共有する学習手法を提案する．先行研究 [4] のようにすべてのエージェントの経験データをまとめて記録することではなく，式 (2.13) のようにエージェント毎に記録する．k エピソード目のエージェント i の状態 $s^{i,k}$，報酬 $r^{i,k}$，行動 $a^{i,k}$ が含まれる 1 単位の経験データを式 (2.13 で示す．

$$Exp^{i,k} = [\{s_1^{i,k}, a_1^{i,k}, r_1^{i,k}\}, \{s_2^{i,k}, a_2^{i,k}, r_2^{i,k}\}, \ldots, \{s_{T-1}^{i,k}, a_{T-1}^{i,k}, r_{T-1}^{i,k}\}, s_T^{i,k}] \tag{2.13}$$

第 3 章

学習データ共有を考慮した MAS の構築

先行研究 [4] では，MAS 内のエージェントは他のエージェントのすべての経験データを使用して行動予測器を学習し，他のエージェントの選択する可能性が一番高い行動を予測する．エージェントは各予測行動と自身の行動を組として，行動価値関数を学習する．学習した行動価値関数を方策学習の際の評価値とする．すなわち，現在の観測状態で予測した他のエージェントが取る行動と自身の行動の組に基づき，行動価値関数を最大化するような学習アーキテクチャを構築した．しかし，エージェントが自身の行動選択器である方策を学習する以外，他の各エージェントに対応する行動予測器も学習する必要があるため，MAS 内のエージェントの数の増加によって学習すべき行動予測器の数が増える．N 体エージェントが存在する場合，行動予測器の数は $N \times (N-1)$ となる．このため，エージェントの数が大きくなると，行動予測器の学習に必要な他のエージェントの経験データの量が増え，全体的に学習効率が低下する傾向がある．

本論文では，他のエージェントが共有する経験データの量を減らことで学習効率を向上する手法を提案する．先行研究 [4] のように他のエージェントの行動予測器を学習するのではなく，エージェント間の経験データの一部を共有しつつエージェント自身の方策に注目して学習するため，本論文の学習アーキテクチャは学習中の計算負担を軽減して学習効率を向上することができる．エージェントの経験データは，環境との相互作用によって記録するデータとなり，式 (2.13) のように記述する．エージェントの学習データとは，Actor-Critic のパラメータに関する損失関数の勾配更新量の計算に使うデータである．エージェントの共有データとは，あるエージェントの経験データから選択せれて別のエージェントの学習データとするデータである．なお，エージェント $i(i = 1, 2, \ldots, N)$ の 1 単位の経験データ Exp^i を式 (2.13) で示し，すべての経験データを集合 D^i とする．例えば，エージェント i が 100 単位の経験データを持つ場合，$D^i = \{Exp^{i,1}, Exp^{i,2}, \ldots, Exp^{i,k}, \ldots, Exp^{i,100}\}$ となる．なお，エージェント i の経験データの総量は $|D^i| = 100$ となる．

経験データの共有率は他のエージェントと経験データを共有する割合を意味する．例えば，経験データ共有率が 25% の場合，図 3.1 のようにエージェント 1 は他のエージェントの保持する経験データのうち，25% の経験データを得る．1 エピソードが終了した後，各エージェントは他のエージェントの共有データと自身の経験データを学習データとして方策を更新する．

先行研究 [4] の手法では，他の各エージェントの行動予測器を適切に学習し，学習

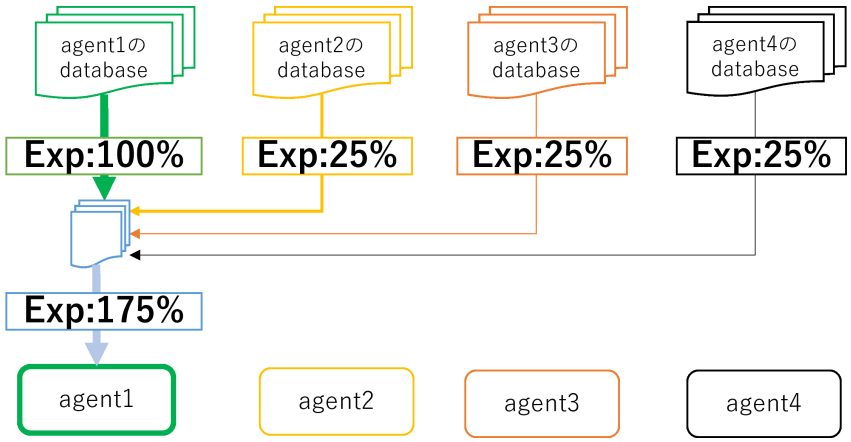

図 3.1: 経験データ共有率 25% のエージェント 1 の学習データ

結果である行動予測の正確率が向上するため，エージェント i の方策を正確に学習することができる．すなわち，MAS の全体の学習を収束させるため，同じ入力に対して各エージェントが持つ別々の行動価値関数が近い行動価値を出力する必要がある．そのため，各エージェントが持つ行動価値関数の目標値 y(式 (2.9)) が近い値に収束するような学習が必要となり，各エージェントが持つ同じエージェントに対する行動予測器の予測正確率が一定の程度で一致しないといけない．例えば，エージェント 1 のエージェント 2 に対する行動予測器とエージェント 3 のエージェント 2 に対する行動予測器の結果が近い値になることが求められる．すべてのエージェントの経験データを共有しつつ学習するため，同じエージェントに対する複数の行動予測器の学習結果が同じ値に収束するまでの学習時間が長くなると考えている．MAS のエージェントの数が増加すれば増加するほど，経験データの量が膨大に必要となる．そのため，同じエージェントに対する複数の行動予測器の学習結果が同じ値に収束する時間が長くなり，常に同じ予測結果を出る確率が低くなり，全体の学習効率が低下する傾向がある．しかし，エージェントの経験データを完全に不参照ではなく，学習に適応な一部のデータを共有すれば，方策改善につながると考えている．他のエージェントが所持する数多くの経験データの中，自身の方策の学習に適応なデータが存在すると想定し，この部分の経験データをうまく識別して共有すれば，先行研究 [4] のようなすべての経験データを共有することにより学習効率が低下する問題を回避できると考えている．

データ共有の方法について，様々な方法が考えられるため，本論文では様々な環境における学習に適応できるように，3 種類の経験データ共有方式を提案する．第一のデータ共有アーキテクチャ（ランダム方式）では，各エージェントは自身の経験データベースからランダムに経験データ共有率 $X_A\%$ の経験データを選択して他のエージェントと共有する．第二のデータ共有アーキテクチャ（固定割合方式）では，

各エージェントは自身の経験データを累積報酬に基づいてランク付けし，ランクの高い順から経験データの $X_A\% \times X_B\%$，低い順から $X_A\% \times (100 - X_B)\%$ のデータを選択して共有する．第三のデータ共有アーキテクチャ（変動割合方式）では，固定割合方式の固定する高い順の割合 X_B を適応的に調整し，共有データを選択する．エージェント i が他のエージェント $j(j \neq i)$ から受け取った共有データを集合 $D^{i,j}$ とし，それぞれの経験データ共有方式において，$D^{i,j}$ を $D^{i,j}_{X_A}$，$D^{i,j}_{X_B}$，$D^{i,j}_{X_C}$ と記述する．

エージェント i が持つ Actor-Critic アルゴリズムのパラメータを θ_i，ω_i とし，Actor と Critic の損失関数は式 (3.1)，(3.2) とする．なお，式 (3.3) は TD 誤差である．学習データは式（2.13）のようにエージェントの 1 エピソードの経験データを 1 単位とし，k 番目の 1 単位の学習データ $Exp^{i,k}$ のパラメータ θ_i と ω_i の損失関数に関する勾配更新量は $\nabla_{\theta_i}\mathcal{L}^k_{actor}(\theta_i)$ と $\nabla_{\omega_i}\mathcal{L}^k_{critic}(\omega_i)$ とする．

$$\mathcal{L}^k_{actor}(\theta_i) = -\frac{1}{T}\sum_{t=1}^{T-1}\log\pi(a^{i,k}_t|s^{i,k}_t,\theta_i)\delta_{t+1}(\omega_i) \tag{3.1}$$

$$\mathcal{L}^k_{critic}(\omega_i) = \sum_{t=1}^{T-1}|\delta_{t+1}(\omega_i)|^2 \tag{3.2}$$

$$\delta_{t+1}(\omega_i) = r^{i,k}_t + \gamma V_{\omega_i}(s^{i,k}_{t+1}) - V_{\omega_i}(s^{i,k}_t) \tag{3.3}$$

3.1. ランダム方式

ランダム方式において，他のエージェントの経験データから共有データの選択と方策のパラメータに関する勾配更新量の計算を果たす機能ブロックをスレッドと定義する．使用するスレッドの数を H とする．すなわち，1 エージェントが H 個スレッドを用いることになる．エージェント $i(i = 1, 2, \ldots, N)$ の 1 つのスレッド $h(h = 1, 2, \ldots, H)$ が他の $(N-1)$ 体のエージェントの経験データからランダムに選択できる共有データを集合 $D^i_h = \cup^N_{j=1,j\neq i} D^{i,j}_{X_A}$ とする．ここで，$|D^{i,j}_{X_A}| = |D^j|_{j\neq i} \times X_A\%$ となる．エージェント i はスレッドが選択した共有データを自分の学習データとするため，1 つのスレッド h が選択できる $|D^i_h|$ 単位の共有データと自身の $|D^i|$ 単位の経験データを含む $|D^i_h + D^i|$ 単位の学習データの損失関数に関する勾配更新量の和 $\nabla^h_{\theta_i}\mathcal{L}_{actor}(\theta_i)$ と $\nabla^h_{\omega_i}\mathcal{L}_{critic}(\omega_i)$ は式 (3.1)–式 (3.5) により計算する．

$$\nabla^h_{\theta_i}\mathcal{L}_{actor}(\theta_i) = \sum_{k\in(D^i\cup D^i_h)}\nabla_{\theta_i}\mathcal{L}^k_{actor}(\theta_i) \tag{3.4}$$

$$\nabla^h_{\omega_i}\mathcal{L}_{critic}(\omega_i) = \sum_{k\in(D^i\cup D^i_h)} \nabla_{\omega_i}\mathcal{L}^k_{critic}(\omega_i) \tag{3.5}$$

エージェント i のすべてのスレッドの計算が完了したあと，式 (3.6)，(3.7) によりパラメータ θ_i，ω_i を更新する．

$$\theta_i \leftarrow \theta_i + \alpha \sum_{h=1}^{H} \nabla^h_{\theta_i}\mathcal{L}_{actor}(\theta_i) \tag{3.6}$$

$$\omega_i \leftarrow \omega_i + \alpha \sum_{h=1}^{H} \nabla^h_{\omega_i}\mathcal{L}_{critic}(\omega_i) \tag{3.7}$$

N 体のエージェントが存在する場合，MAS 内で使用するスレッドの数が $N \times H$ となる．図 3.2 は，エージェントの数 $N = 2$，スレッド数 $H = 4$，$X_A = 25\%$ のランダム方式のエージェント 1 のマルチスレッド機能を示している．ランダム方式において，共有データをランダムに選択するため，1 回だけ共有データを選択すると，必ず学習に最適なデータを選択できるとは限らない．他のエージェントの経験データから複数のスレッドにより複数回データを選択して共有すると，学習に良好な影響を与えるデータを選択する確率が高くなると考えられる．複数のスレッドを同時に使用するマルチスレッド機能の利用により，計算時間を削減し，学習効率の向上が期待される．

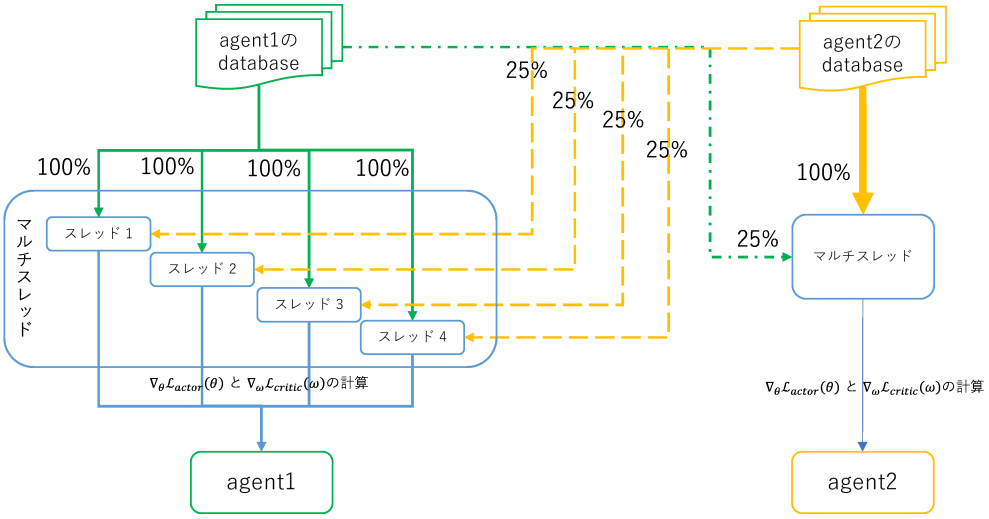

図 3.2: ランダム方式のマルチスレッド機能

3.2. 固定割合方式

固定割合方式では，各エージェントの経験データを累積報酬においてランク付けをする．式 (2.13) で示すエージェント $i(i = 1, 2, \ldots, N)$ の k 番目の 1 単位の経験データ $Exp^{i,k}$ の累積報酬 $r_{total^{i,k}}$ を $(r_1^{i,k} + r_2^{i,k} + \ldots + r_{T-1}^{i,k})$ により計算する．経験データ共有率を X_A，高い順の割合を X_B とする場合，共有データはランクの高い順から $X_A\% \times X_B\%$，低い順から $X_A\% \times (100 - X_B)\%$ の経験データにより構成する．例えば，経験データ共有率 X_A を 25，X_B を 80 とする場合，ランク付けのデータベースの上位から $X_A\% \times X_B\% = 25\% \times 80\% = 20\%$，下位から $X_A\% \times (100 - X_B)\% = 25\% \times 20\% = 5\%$ を共有データとして選択することとなる．図 3.3 はエージェントの数 $N = 2$ の場合の固定割合方式の共有データの選択方法を示している．

エージェント $i(i = 1, 2, \ldots, N)$ が持つランク付け経験データを集合 D_{rank}^i とする場合，エージェント $i(i = 1, 2, \ldots, N)$ が他の $(N - 1)$ 体のエージェントのランク付け経験データの高い順から選択できる共有データを集合 $D_{high}^i = \cup_{j=1, j\neq i}^{N} D_{X_B}^{i,j}$，低い順から選択できる共有データを集合 $D_{low}^i = \cup_{j=1, j\neq i}^{N} D_{(100-X_B)}^{i,j}$ とする．ここで，$|D_{X_B}^{i,j}| = |D^j|_{j\neq i} \times X_A\% \times X_B\%$，$|D_{(100-X_B)}^{i,j}| = |D^j|_{j\neq i} \times X_A\% \times (100 - X_B)\%$ となる．$|D_{high}^i \cup D_{low}^i|$ 単位の共有データを学習データとし，その損失関数に関する勾配更新量の和 $\nabla_{\theta_i}^B \mathcal{L}_{actor}(\theta_i)$ と $\nabla_{\omega_i}^B \mathcal{L}_{critic}(\omega_i)$ を式 (3.1)–(3.3)，(3.8)，(3.9) により計算する．

$$\nabla_{\theta_i}^B \mathcal{L}_{actor}(\theta_i) = \sum_{k \in (D_{high}^i \cup D_{low}^i)} \nabla_{\theta_i} \mathcal{L}_{actor}^k(\theta_i) \tag{3.8}$$

$$\nabla_{\omega_i}^B \mathcal{L}_{critic}(\omega_i) = \sum_{k \in (D_{high}^i \cup D_{low}^i)} \nabla_{\omega_i} \mathcal{L}_{critic}^k(\omega_i) \tag{3.9}$$

エージェント i は自身の $|D_{rank}^i|$ 単位の経験データも含めた損失関数に関する勾配更新量を用いて式 (3.10)，(3.11) に基づいてパラメータ θ_i，ω_i を更新する．

$$\theta_i \leftarrow \theta_i + \alpha\Big(\sum_{k \in D_{rank}^i} \nabla_{\theta_i} \mathcal{L}_{actor}^k(\theta_i) + \nabla_{\theta_i}^B \mathcal{L}_{actor}(\theta_i)\Big) \tag{3.10}$$

$$\omega_i \leftarrow \omega_i + \alpha\Big(\sum_{k \in D_{rank}^i} \nabla_{\omega_i} \mathcal{L}_{critic}^k(\omega_i) + \nabla_{\omega_i}^B \mathcal{L}_{critic}(\omega_i)\Big) \tag{3.11}$$

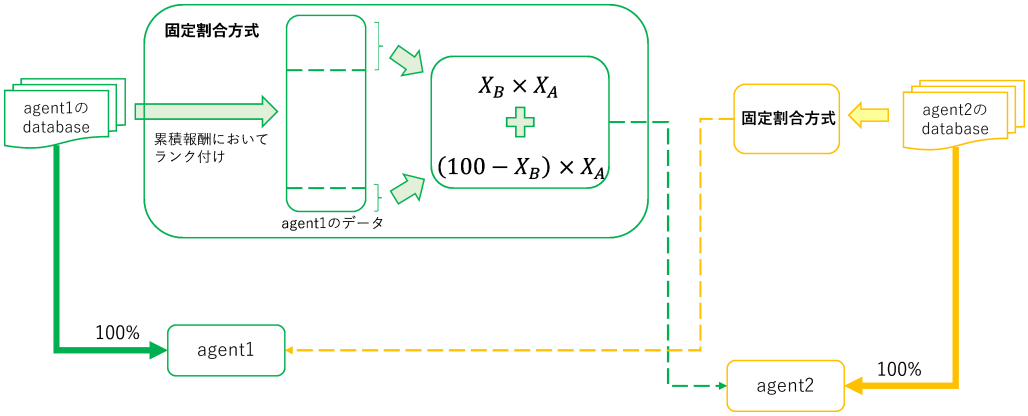

図 3.3: 固定割合方式のデータ選択方法

3.3. 変動割合方式

変動割合方式では，固定割合方式の高い順の選択割合 X_B を調整しつつ，共有データを選択する．変動割合方式で，X_B を X_C と記述し，X_C と低い順から選択する割合 $(100 - X_C)$ をペアとしてデータ選択パターン $P(X_C, 100 - X_C)$ と記述する．最初のデータ選択パターンによる影響を検証するため，一定数のエピソードを用いて $P(X_C, 100 - X_C)$ を変更せずに実行する．$P(X_C, 100 - X_C)$ を変更しないエピソードの数を初期エピソード $E_{initial}$ とする．高い順から選択する割合 X_C を変更しないため，変動割合方式の初期エピソードにおける学習効果は固定割合方式の学習効果と一致すると考えられる．$E_{initial}$ の調整により固定割合方式を用いるエピソードの数を調整して学習を行うことになるが，簡単な MAS であれば，固定割合方式が十分な学習効果を得られると考えられる．一方で，複雑な MAS では，固定割合方式は必ずしも学習に適応な経験データを選択できることと限らない．エージェント数の増加により不適切な共有データが選択されることで，学習効率が低下するような場合もある．本論文では，このような問題を回避するため，変動割合方式のように $P(X_C, 100 - X_C)$ の X_C を適応的に調整するようなデータ共有アーキテクチャを構築する．変動割合方式が X_C を学習状況により調整しつつ，様々な MAS の学習に適応な経験データを共有できるため，固定割合方式の学習効率低下の問題を解決できると考えている．すなわち，エージェント数が多い MAS では，学習に適応なデータが上位にランク付けされるとは限らないので，X_C を適応的に調整することで，学習に必要な経験データを適切に共有されることが期待できる．初期の $E_{initial}$ エピソードを実行したあと，X_C を適応的に調整する．図 3.4 に，横軸をエピソード数，縦軸を X_C として，変動割合方式におけるパラメータ X_C の調整例を示す．

変動割合方式において，X_C を調整するためのエピソードの数を調整エピソード E_{modify} とし，E_{modify} の 1 単位を E_{adjust} エピソードとする．$E_{adjust} = 20$ のとき，1

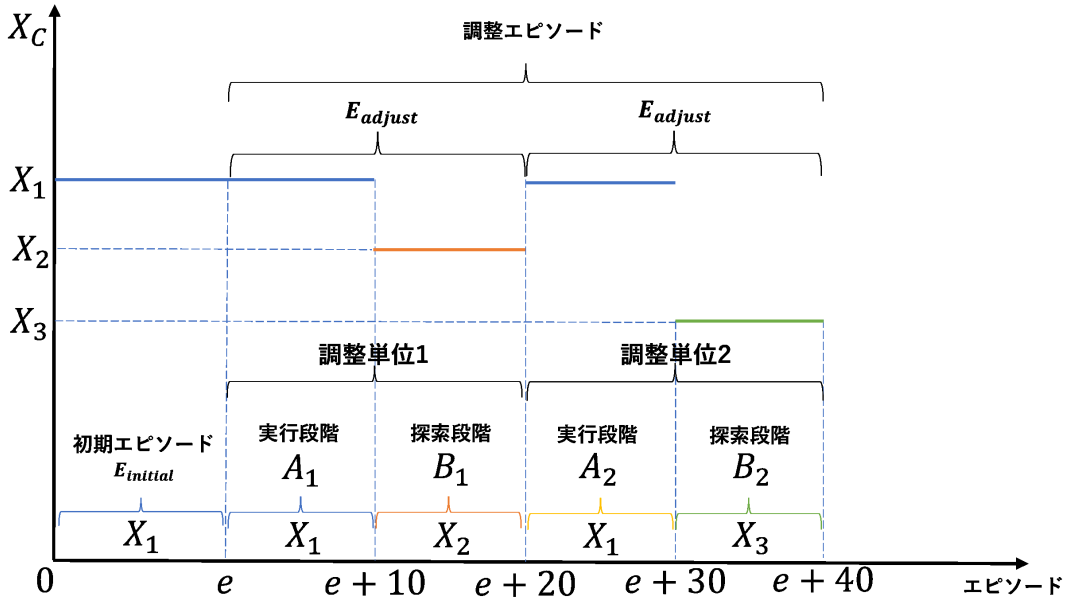

図 3.4: 変動割合方式の X_C の調整例

つの調整単位の前半の 10 エピソードを実行段階 A，後半の 10 エピソードを探索段階 B とする．調整単位 L として，その前半の実行段階 A_L において，直前の調整単位 $(L-1)$ の結果により X_C を調整する．調整単位 $(L-1)$ の探索段階 B_{L-1} の報酬，調整単位 $(L-1)$ の実行段階 A_{L-1} の報酬，調整単位 $(L-2)$ の探索段階 B_{L-2} の報酬を R^B_{L-1}，R^A_{L-1}，R^B_{L-2} とする．$(R^A_{L-1} - R^B_{L-2}) > (R^B_{L-1} - R^A_{L-1})$ のとき，調整単位 L の実行段階 A_L は A_{L-1} の X_C を採用し，逆の場合は B_{L-1} の X_C を採用するような調整ルールを用いて X_C を調整する．調整単位 L の後半の探索段階 B_L において，直前の調整単位 $(L-1)$ の探索段階 B_{L-1} の X_C を 5% 減らし，$X_C = X_C - 0.05$ として共有データを選択する．

図 3.5 に変動割合方式のフローチャートを示す．図 3.4 に示される調整例では，初期エピソード $E_{initial} = e$，調整エピソードの 1 単位 $E_{adjust} = 20$ となる．$E_{initial}$ 内のエピソードの X_C を X_1 とし，データ選択パターン $P(X_C, 100 - X_C)$ で共有データを選択する．$E_{initial}$ を実行した後，X_C を調整するため，調整エピソード E_{modify} を 2 つの調整単位に分ける．調整単位 1 の前半の 10 エピソードの実行段階 A_1 の X_C を $X_C = X_1$，後半の 10 エピソードの探索段階 B_1 の X_C を $X_C = X_2$ とする．調整単位 2 の実行段階 A_2 と探索段階 B_2 の X_C を調整する．$E_{initial}$ の最後の 10 エピソードの平均報酬を計算し，$R_{initial}$ とする．実行段階 A_1 の平均報酬を R^A_1 とする．同じように，探索段階 B_1 平均報酬を R^B_2 とする．実行段階 A_1 は最初の調整単位の実行段階であるため，直近のエピソードは初期エピソードとなる．A_1 の平均報酬の増量は初期エピソードの最後の 10 エピソードの平均報酬を使用して計算する．A_1 の平均報

酬の増量 $(R_1^A - R_{initial})$ と B_1 の平均報酬の増量 $(R_1^B - R_1^A)$ の大小関係により次の調整単位 2 の実行段階 A_2 の X_C を調整する．図 3.4 に示される調整例のように，A_1 の平均報酬の増量の方が多いため，A_2 の X_C を A_1 の $X_C = X_1$ に調整した．そして，調整単位 2 の探索段階 B_2 の X_C を B_1 の X_C より 5% 減らした値 $X_3 = X_2 - 0.05$ とした．

調整単位の前半と後半のエピソードの X_C を調整しつつ，共有データを選択する．k 番目のエピソードの時，X_C を X_C^k，経験データの共有率を X_A^k，エージェント $i(i = 1, 2, \ldots, N)$ が持つランク付け経験データを集合 D_i^k とする．固定割合方式と同じように累積報酬を指標として経験データをランク付けする．そして，k 番目のエピソードの時，エージェント i が他の $(N-1)$ のエージェントのランク付け経験データの高い順から選択できる共有データを集合 $D_{high}^{i,k} = \cup_{j=1,j\neq i}^{N} D_{X_C}^{i,j,k}$，低い順から選択できる共有データを集合 $D_{low}^{i,k} = \cup_{j=1,j\neq i}^{N} D_{(100-X_C)}^{i,j,k}$ とする．ここで，$|D_{X_C}^{i,j,k}| = |D_j^k|_{j\neq i} \times X_A^k\% \times X_C^k\%$，$|D_{(100-X_C)}^{i,j,k}| = |D_j^k|_{j\neq i} \times X_A^k\% \times (100 - X_C^k)\%$ となる．$|D_{high}^{i,k} \cup D_{low}^{i,k}|$ 単位の共有データを学習データとし，その損失関数に関する勾配更新量の和 $\nabla_{\theta_i}^C \mathcal{L}_{actor}(\theta_i)$ と $\nabla_{\omega_i}^C \mathcal{L}_{critic}(\omega_i)$ は式 (3.1)–(3.3)，(3.12)，(3.13) により計算する．

$$\nabla_{\theta_i}^C \mathcal{L}_{actor}(\theta_i) = \sum_{k\in(D_{high}^{i,k}\cup D_{low}^{i,k})} \nabla_{\theta_i} \mathcal{L}_{actor}^k(\theta_i) \tag{3.12}$$

$$\nabla_{\omega_i}^C \mathcal{L}_{critic}(\omega_i) = \sum_{k\in(D_{high}^{i,k}\cup D_{low}^{i,k})} \nabla_{\omega_i} \mathcal{L}_{critic}^k(\omega_i) \tag{3.13}$$

エージェント i は自身の経験データも含めた損失関数に関する勾配更新量を用いて式 (3.14)，(3.15) に基づいてパラメータ θ_i，ω_i を更新する．

$$\theta_i \leftarrow \theta_i + \alpha(\sum_{k\in D_i^k} \nabla_{\theta_i} \mathcal{L}_{actor}^k(\theta_i) + \nabla_{\theta_i}^C \mathcal{L}_{actor}(\theta_i)) \tag{3.14}$$

$$\omega_i \leftarrow \omega_i + \alpha(\sum_{k\in D_i^k} \nabla_{\omega_i} \mathcal{L}_{critic}^k(\omega_i) + \nabla_{\omega_i}^C \mathcal{L}_{critic}(\omega_i)) \tag{3.15}$$

X_C を調整する調整エピソード以後から最後の学習エピソードまでの X_C を調整しないエピソードの数を非調整エピソード E_{no_adjust}，総学習エピソードを E_{total} とする．X_C の計算手順を図 3.5 およびアルゴリズム 1 で示している．

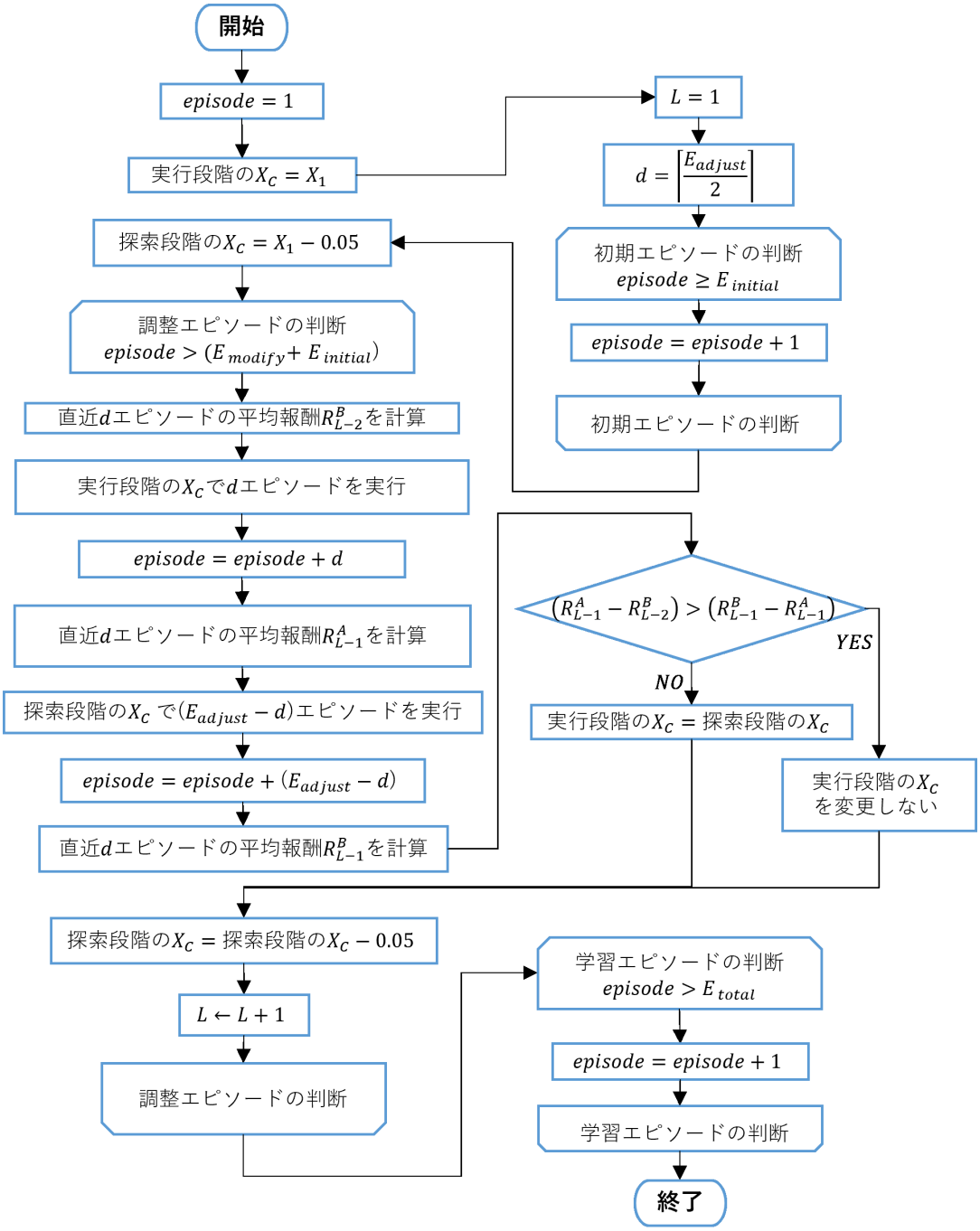

図 3.5: 変動割合方式の X_C の調整フローチャート

Algorithm 1 すべてのエピソードの X_C の X_C^k の計算

総学習エピソード：E_{total}，初期エピソードの数：$E_{initial}$，非調整エピソードの数：E_{no_adjust}，調整単位のエピソード数：$E_{adjust} = 20$，実行段階 A の X_C の初期値：X_1，探索段階 B の X_C の初期値：X_2，実行段階の標識：FH = TRUE，調整単位の番号：$L = 1$，直近 10 エピソードの平均報酬：R_{L-1}^B，直近 20 から 11 までのエピソードの平均報酬：R_{L-1}^A，直近 30 から 21 までのエピソードの平均報酬：R_{L-2}^B，$k = 1$
for $k \leqq E_{total}$ **do**
  if $k \leqq E_{initial}$ or $k > (E_{total} - E_{no\_adjust})$ **then**
    (エピソード k は調整エピソード以外のエピソード)
    $X_C^k \leftarrow X_1$
  else
    (エピソード k は調整エピソード)
    $k \leftarrow k - E_{initial}$
    if $(k-1)$ % $E_{adjust}/2 = 0$ **then**
      (エピソード k は実行段階や探索段階の最初のエピソード)
      R_{L-2}^B，R_{L-1}^A，R_{L-1}^B を計算する；$L \leftarrow L + 1$
      if FH = TRUE **then**
        (調整単位 L の前半：実行段階)
        FH $\leftarrow$ FALSE
        if $(R_{L-1}^A - R_{L-2}^B) > (R_{L-1}^B - R_{L-1}^A)$ **then**
          (前の調整単位の実行段階の平均報酬の増量の方が多い)
          $X_C^k \leftarrow X_1$
        else
          (前の調整単位の探索段階の平均報酬の増量の方が多い)
          $X_1 \leftarrow X_2$；$X_C^k \leftarrow X_1$；$X_2 \leftarrow X_2 - 0.05$
        end if
      else
        (調整単位 L の後半：探索段階)
        $X_C^k \leftarrow X_2$；FH $\leftarrow$ TRUE
      end if
    else
      (エピソード k は実行段階や探索段階の最初のエピソードではない)
      (X_C^k を調整しない)
    end if
  end if
  $k \leftarrow k + 1$
end for

第 4 章

シミュレーション実験

4.1. 実験設定

本論文では，エージェント間で一部の情報を共有することによる学習効率向上の効果を検証するため，提案手法の 3 種類のデータ共有アーキテクチャを用いて複数のエージェントが存在する迷路環境における宝物探し問題を用いた数値実験を行う．図 4.1–4.3 に，本実験で用いる 3 種類の迷路環境を示す．迷路環境 1 は 7×7，迷路環境 2 は 7×14 ，迷路環境 3 は 11×14 の 2 次元グリッド環境である．エージェントの目的は，迷路環境において，規定ステップ数以内で全ての宝物の場所に到達することである．なお，図 4.2 に示される A，B については 4.2 節で述べる．

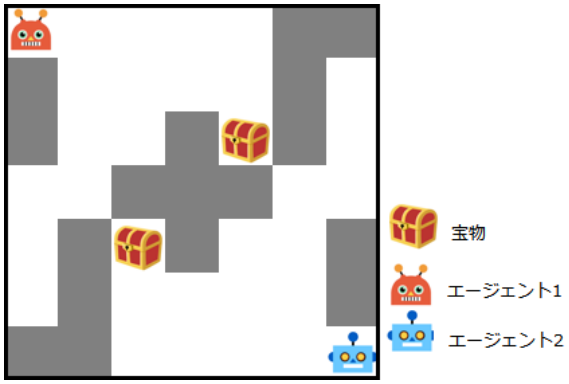

図 4.1: 迷路環境 1

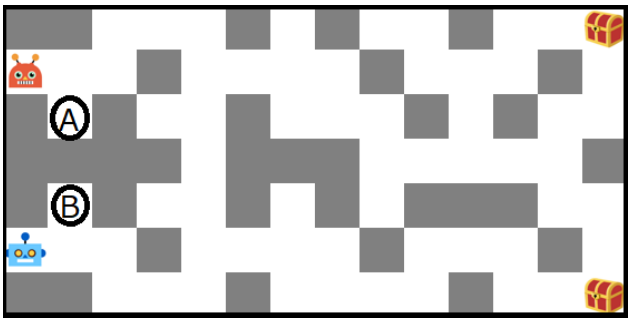

図 4.2: 迷路環境 2

図 4.3: 迷路環境 3

各エージェントが観測できる状態 s はエージェントの上，下，左，右の方向に配置されるものを識別できるものとする．具体的には，通路を 1，壁を 0，宝物を 2，罠を 3，他のエージェントを 4 とする 4 次元ベクトルとして周囲の環境を認識する．行動集合を $\mathcal{A} \equiv \{$ 上，下，左，右 $\}$ とする．迷路環境 1，2 の報酬設定を表 4.1，4.2 に示す．エピソード開始後の早いステップで，エージェントを通路を選択させるように通路の報酬を多めに設定する．迷路環境 3 の報酬設定は次節で述べる．

表 4.1: 各状態の報酬

1 つ目宝物	2 つ目宝物	壁	衝突	罠
10	20	-0.1	-1	-1

表 4.2: 1 ステップごとに得る報酬

ステップ数の上限に対する割合	報酬
0 以上 1/4 未満	0.04
1/4 以上 1/2 未満	-0.01
1/2 以上	-0.04

式 (3.3) の割引率を γ=0.9，式 (3.6)，(3.7) などのパラメータ θ と ω の更新式の学

習率を α=0.001 として，行動選択には ϵ-greedy 法を使用する．ここで，ϵ の初期値を ϵ=0.1 とする．学習の初期，様々な方策を試すために，ϵ を大きく設定し，その後の微調整で ϵ を小さくする．表 4.3 のように，エピソード数によって ϵ=0.01 まで段階的に小さくする．

表 4.3: ϵ の変化

総エピソードの割合	ϵ
0 以上 1/2 未満	0.1
1/2 以上 3/4 未満	0.05
3/4 以上	0.01

規定ステップ数に到達するか，全ての宝物を回収することが各エピソードの終了条件として，迷路環境 1，2，3 のステップ数の上限をそれぞれ 40，100，100 とする．

4.2. 実験結果

4.2.1. ランダム方式の実験結果

ランダム方式で，マルチスレッド機能の使用するスレッド数 $H = 4$，1 試行のエピソード数を 100 とする．簡単な MAS である迷路環境 1 と複雑な MAS である迷路環境 2 で，ランダム方式の学習効果をシミュレーションする．ランダム方式の学習の効果とすべての経験データを共有する学習の効果，経験データを共有しない学習の効果を比較するため，経験データ共有率 X_A を 100%，75%，50%，25%，0% として，それぞれ 10 試行のシミュレーションを行う．100 エピソードのうち，1 つ以上の宝物を回収したエピソード数と 2 つの宝物を回収したエピソード数について 10 試行の箱ひげ図を図 4.4，4.5 に示す．10 試行の実験結果の最大値，75% 値，中央値，25% 値，最小値，平均値と平均線を箱ひげ図に示す．横軸を経験データ共有率 X_A，縦軸を宝物の回収成功率として，図中の数値を表 4.4，4.5 に示す．

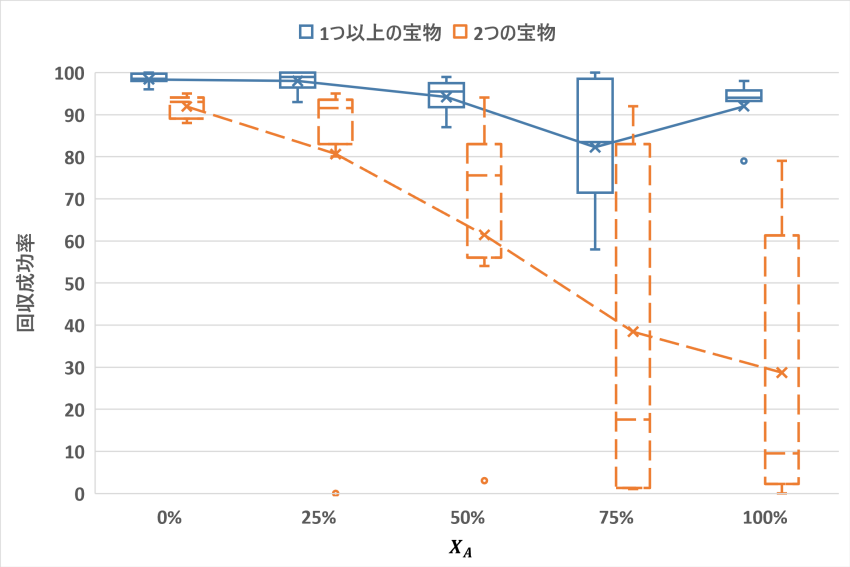

図 4.4: ランダム方式の回収成功率（迷路環境 1)

表 4.4: ランダム方式の回収成功率（迷路環境 1)

X_A	宝物	最大値	75%値	中央値	25%値	最小値	平均値
0%	1 つ以上	100.00	99.75	98.50	98.00	96.00	98.40
	2 つ	95.00	94.00	93.00	89.00	88.00	91.90
25%	1 つ以上	100.00	100.00	99.00	96.50	93.00	98.00
	2 つ	95.00	93.50	91.50	83.00	0.00	80.60
50%	1 つ以上	99.00	97.50	95.50	91.75	87.00	94.20
	2 つ	94.00	83.00	75.50	56.00	0.00	61.40
75%	1 つ以上	100.00	98.50	83.50	71.50	58.00	82.30
	2 つ	92.00	83.00	17.50	1.25	1.00	38.40
100%	1 つ以上	98.00	95.75	94.00	93.25	79.00	92.00
	2 つ	79.00	61.25	9.50	2.25	0.00	28.70

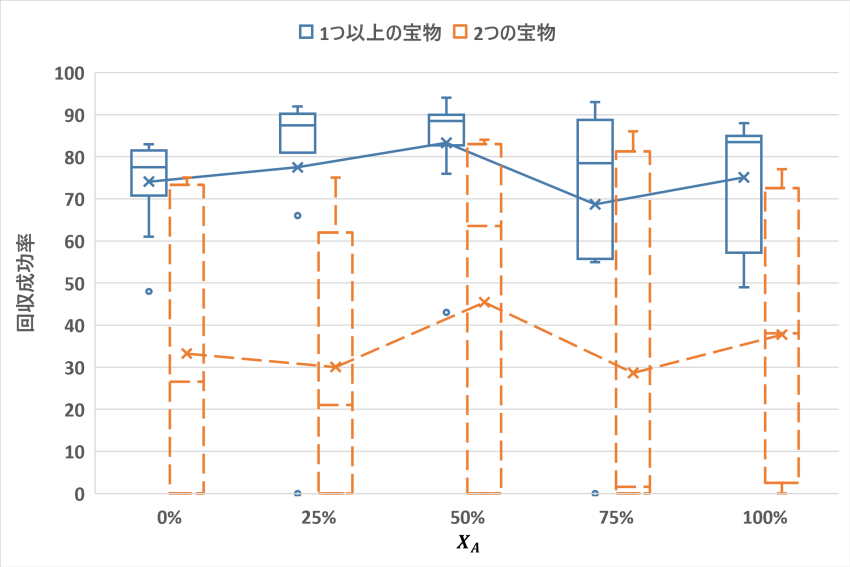

図 4.5: ランダム方式の回収成功率（迷路環境 2)

表 4.5: ランダム方式の回収成功率（迷路環境 2)

X_A	宝物	最大値	75%値	中央値	25%値	最小値	平均値
0%	1 つ以上	83.00	80.75	77.50	74.50	48.00	74.10
	2 つ	75.00	69.00	26.50	0.00	0.00	33.20
25%	1 つ以上	92.00	89.75	87.50	86.00	0.00	77.50
	2 つ	75.00	56.25	21.00	0.00	0.00	30.00
50%	1 つ以上	94.00	89.00	88.50	85.50	43.00	83.30
	2 つ	84.00	81.25	63.50	0.25	0.00	45.40
75%	1 つ以上	93.00	87.75	78.50	57.00	0.00	68.70
	2 つ	86.00	67.75	1.50	0.25	0.00	28.60
100%	1 つ以上	88.00	84.75	83.50	63.50	49.00	75.10
	2 つ	77.00	69.50	38.00	5.25	0.00	37.70

図 4.4，表 4.4 から，迷路環境 1 では X_A が上がっても，回収成功率が改善しないことがわかる．一方で，図 4.5，表 4.5 から，迷路環境 2 では，X_A が 0% の場合に比べて，X_A の増加により，エージェントが 2 つの宝物の回収に成功する頻度が増加していることがわかる．特に X_A を 50% とした場合に最も成功率が高い．さらに，X_A を大きくしても (100%) 必ずにも成功率が改善されるわけではないことがわかる．

迷路環境 1 は簡単な迷路環境であり，2 体のエージェントの最適な行動が互いに大

きく異なる．具体的には，スタート地点から見てエージェント 1 は右下，エージェント 2 は左上に宝物があるため，エージェントがデータを共有しても回収成功率が上がらなかったと考えられる．一方で，複雑な迷路環境であり，各エージェントの最適な方策が部分的に類似するような迷路環境 2 では，十分なデータを共有することで，MAS における学習に対して良好な影響を与えるデータを共有することができ，2 つの宝物の回収成功率が改善したと考えられる．

4.2.2. 固定割合方式の実験結果

固定割合方式の学習効果を検証するため，迷路環境 2 に罠を追加した問題を用いた実験を実施する．どちらかのエージェントが罠のあるセルに到達すると，そのエピソードは終了し，各エージェントを負の報酬（-1）を得るものとする．ここでは，罠の場所を図 4.2 の A とした場合（A）と，A，B とした場合（AB）の 2 通りで実験する．すなわち，図 4.6，4.7 のような問題を考える．

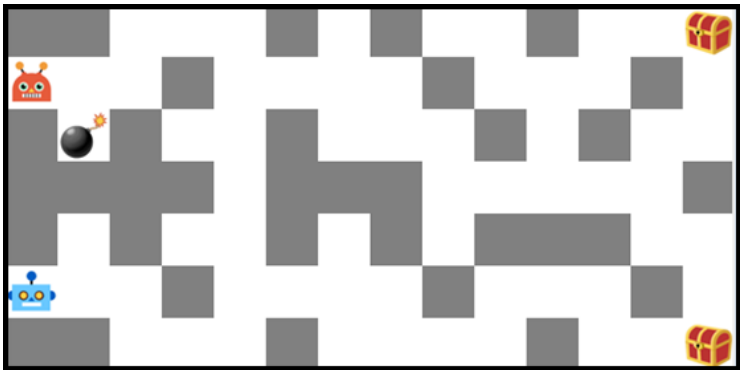

図 4.6: 迷路環境 2(A)

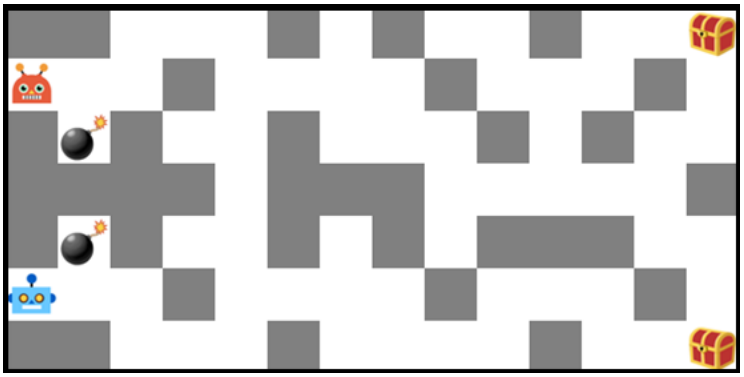

図 4.7: 迷路環境 2(AB)

ここでは，経験データ共有率 X_A を 25% として，各データを累積報酬においてランク付けする．$X_B = 100\%$，99%，95%，80%，60%，50%，40%，20%，0% とした 9 種類の実験とランダム方式の 10 種類の実験を行う．1 試行のエピソード数を 100 とし，罠のある迷路環境 $2(A)$，$2(AB)$ で，それぞれシミュレーション実験を 10 試行行う．図 4.8，4.9 に実験結果を示す．横軸を X_B，縦軸を宝物の回収成功率と罠にはまった割合とする．実験結果の各数値を表 4.6–4.9 に示す．

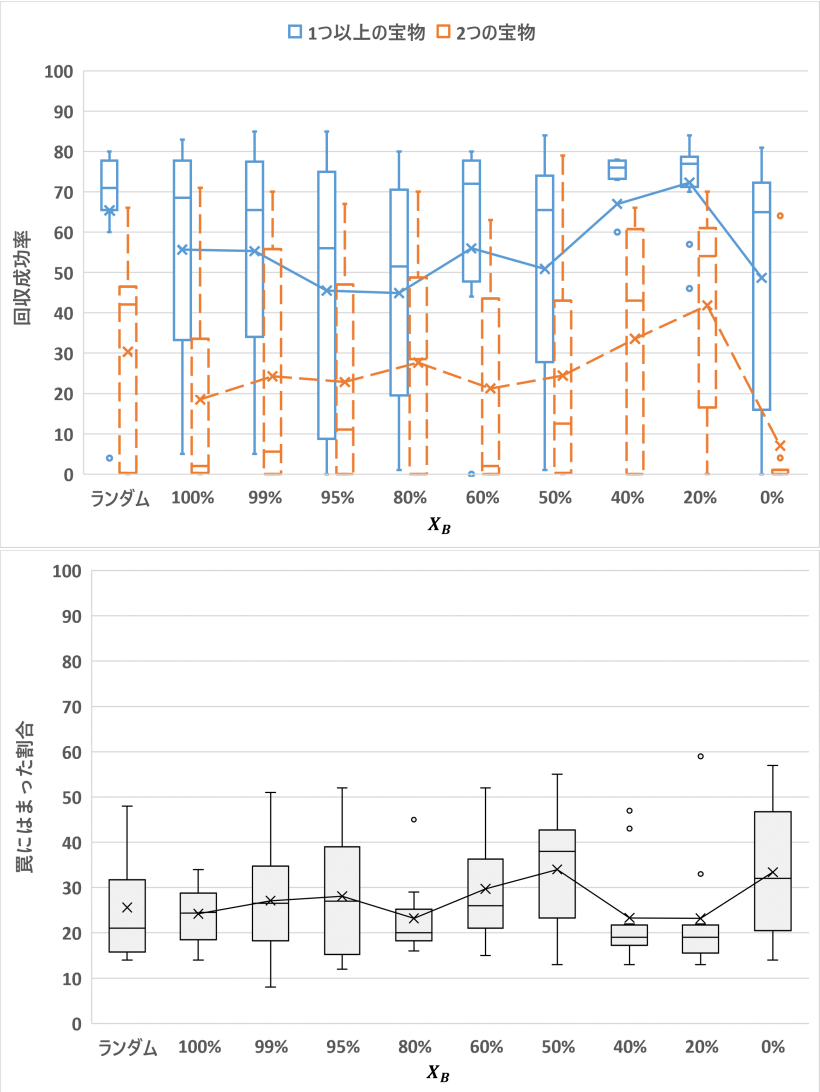

図 4.8: 固定割合方式（迷路環境 $2(A)$）

表 4.6: 固定割合方式（迷路環境 2(A)) の宝物回収率

	宝物	最大値	75%値	中央値	25%値	最小値	平均値
ランダム	1つ以上	80.00	77.75	71.00	65.50	4.00	65.40
	2つ	66.00	46.50	42.00	0.25	0.00	30.30
100%	1つ以上	83.00	77.75	68.50	33.25	5.00	55.70
	2つ	71.00	33.50	2.00	0.25	0.00	18.50
99%	1つ以上	85.00	77.50	65.50	34.00	5.00	55.30
	2つ	70.00	55.75	5.50	0.00	0.00	24.20
95%	1つ以上	85.00	75.00	56.00	8.75	0.00	45.50
	2つ	67.00	47.00	11.00	0.00	0.00	22.80
80%	1つ以上	80.00	70.50	51.50	19.50	1.00	44.90
	2つ	70.00	48.75	28.50	0.00	0.00	27.60
60%	1つ以上	80.00	77.75	72.00	47.75	0.00	56.00
	2つ	63.00	43.50	2.00	0.00	0.00	21.20
50%	1つ以上	84.00	74.00	65.50	27.75	1.00	50.90
	2つ	79.00	43.00	12.50	0.25	0.00	24.40
40%	1つ以上	78.00	77.75	76.00	73.25	0.00	67.00
	2つ	66.00	60.75	43.00	0.00	0.00	33.50
20%	1つ以上	84.00	78.75	77.00	71.25	46.00	72.30
	2つ	70.00	61.00	54.00	16.50	0.00	41.80
0%	1つ以上	81.00	72.25	65.00	16.00	0.00	48.70
	2つ	64.00	1.00	0.00	0.00	0.00	7.00

表 4.7: 固定割合方式（迷路環境 $2(A)$) の罠にはまった割合

	最大値	75%値	中央値	25%値	最小値	平均値
ランダム	48.00	31.75	21.00	15.75	14.00	25.60
100%	34.00	28.75	24.50	18.50	14.00	24.20
99%	51.00	34.75	26.50	18.25	8.00	27.10
95%	52.00	39.00	27.00	15.25	12.00	28.10
80%	45.00	25.25	20.00	18.25	16.00	23.20
60%	52.00	36.25	26.00	21.00	15.00	29.70
50%	55.00	42.75	38.00	23.25	13.00	34.00
40%	47.00	21.75	19.00	17.25	13.00	23.30
20%	59.00	21.75	19.00	15.50	13.00	23.20
0%	57.00	46.75	32.00	20.50	14.00	33.40

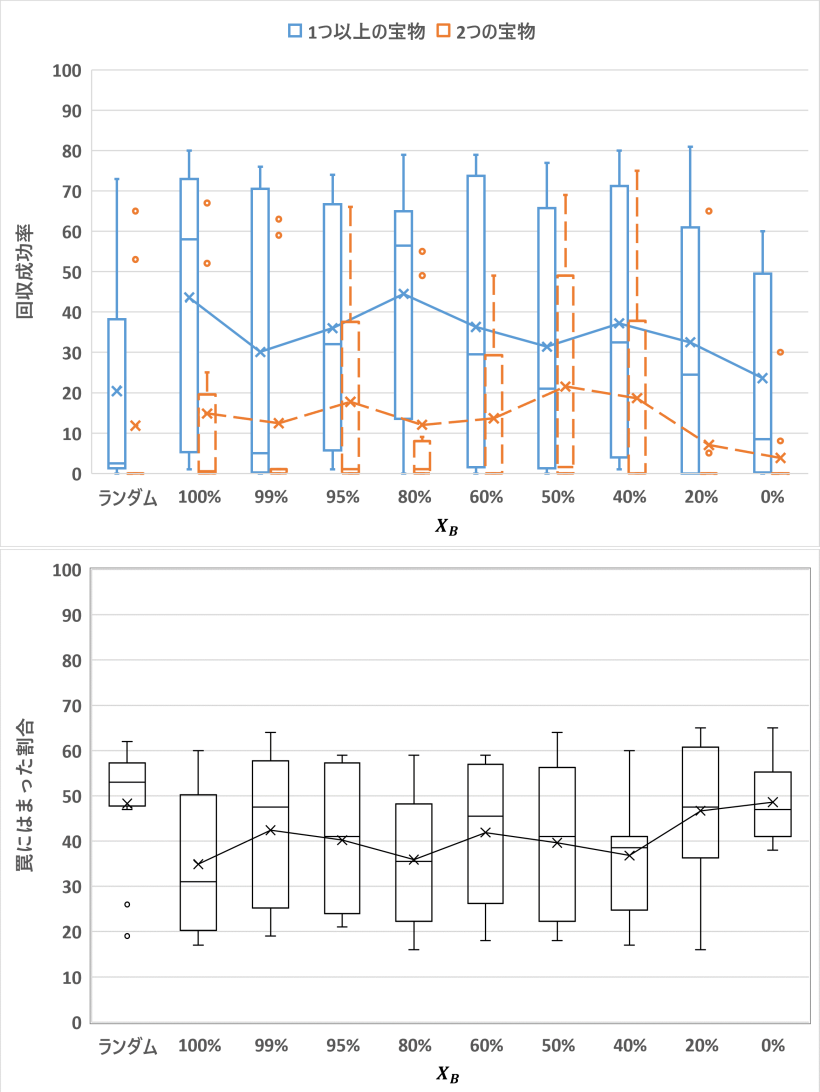

図 4.9: 固定割合方式（迷路環境 2(AB))

表 4.8: 固定割合方式（迷路環境 2(AB)) の宝物回収率

		最大値	75%値	中央値	25%値	最小値	平均値
ランダム	1つ以上	73.00	38.25	2.50	1.25	0.00	20.40
	2つ	65.00	0.00	0.00	0.00	0.00	11.80
100%	1つ以上	80.00	73.00	58.00	5.25	1.00	43.60
	2つ	67.00	19.50	0.50	0.00	0.00	14.80
99%	1つ以上	76.00	70.50	5.00	0.25	0.00	30.10
	2つ	63.00	1.00	0.00	0.00	0.00	12.40
95%	1つ以上	74.00	66.75	32.00	5.75	1.00	36.00
	2つ	66.00	37.50	1.00	0.00	0.00	17.70
80%	1つ以上	79.00	65.00	56.50	13.50	0.00	44.50
	2つ	55.00	8.00	1.00	0.00	0.00	12.00
60%	1つ以上	79.00	73.75	29.50	1.50	0.00	36.30
	2つ	49.00	29.25	0.00	0.00	0.00	13.60
50%	1つ以上	77.00	65.75	21.00	1.25	0.00	31.40
	2つ	69.00	49.00	1.50	0.00	0.00	21.50
40%	1つ以上	80.00	71.25	32.50	4.00	1.00	37.20
	2つ	75.00	37.75	0.00	0.00	0.00	18.60
20%	1つ以上	81.00	61.00	24.50	0.00	0.00	32.50
	2つ	65.00	0.00	0.00	0.00	0.00	7.00
0%	1つ以上	60.00	49.50	8.50	0.25	0.00	23.60
	2つ	30.00	0.00	0.00	0.00	0.00	3.80

表 4.9: 固定割合方式（迷路環境 2(AB)) の罠にはまった割合

	最大値	75%値	中央値	25%値	最小値	平均値
ランダム	62.00	57.25	53.00	47.75	19.00	48.30
100%	60.00	50.25	31.00	20.25	17.00	34.90
99%	64.00	57.75	47.50	25.25	19.00	42.40
95%	59.00	57.25	41.00	24.00	21.00	40.20
80%	59.00	48.25	35.50	22.25	16.00	35.90
60%	59.00	57.00	45.50	26.25	18.00	41.90
50%	64.00	56.25	41.00	22.25	18.00	39.60
40%	60.00	41.00	38.50	24.75	17.00	36.80
20%	65.00	60.75	47.50	36.25	16.00	46.70
0%	65.00	55.25	47.00	41.00	38.00	48.60

図 4.8，表 4.6，4.7 から，$X_B = 20\%$ のとき迷路環境 2（A）において 2 つの宝物の回収成功率が最も高く，罠への到達率が最も低いことから，最良の結果が得られていると言える．図 4.9，表 4.8，4.9 から，迷路環境 2（AB）において $X_B = 50\%$ のとき，最良の結果が得られた．高い順の割合 X_B により累積報酬が高いデータを選択し，低い順の割合 $(100 - X_B)$ により累積報酬が低いデータを選択するため，学習データは成功体験と失敗体験の両方を含める．X_B の調整により学習データのバランスが変更する．罠があり，学習が難しい迷路環境 $2A$，$2AB$ では，成功体験だけではなく失敗体験を多く共有することで，MAS における全体の学習効率の向上が成功したと考えられる．両迷路環境では，成功経験だけまだは失敗経験だけの学習効果が良くないと考えられる．そして，罠がある迷路環境 2（A）と 2（AB）の複雑な MAS の学習において，固定割合方式はランダム方式より学習効率を向上することが判明した．

4.2.3. 変動割合方式の実験結果

変動割合方式の学習効果を検証するため，迷路環境 1，2，2(A)，2(AB) で実験を実施する．迷路環境 1，2 の X_C の初期値を 100%，迷路環境 2(A)，2(AB) の X_C の初期値を 50% とする．変動割合方式の学習効果をすべての経験データを共有する学習の効果，経験データを共有しない学習の効果と比較するため，経験データ共有率 X_A を 100%，75%，50%，25%，0% として，それぞれ 10 試行のシミュレーションを行う．各迷路環境の 1 試行のエピソード数，初期エピソード数 $E_{initial}$，調整エピソード数 E_{modify}，調整単位のエピソード数 E_{adjust} を表 4.10 とし，総エピソードと毎 50 エピソードの 2 つの宝物の回収成功率，1 つ以上の宝物の回収成功率，罠にはまった割合の 3 つの指標を観察する．

表 4.10: 各迷路環境の実験設定

迷路環境	X_C の初期値 (%)	1 試行のエピソード数	$E_{initial}$	E_{modify}	E_{adjust}
1	100	500	25	275	20
2	100	500	25	275	20
$2(A)$	50	1000	25	275	20
$2(AB)$	50	1000	25	275	20

図 4.10–4.13 に各迷路環境の実験結果を示す．実験結果の各数値を表 4.11–4.16 に示す．各迷路環境の実験結果を分析すると，エージェント間で経験データの一部を共有することは全体を共有することより学習結果が良い．具体的には，罠がない迷路環境 1 で，経験データ共有率 X_A が 0% の場合の学習効果が一番良い．経験データ共有率 X_A の上昇とともに 2 つの宝物の回収成功率が大幅に減少している．罠がない迷路環境 1 は簡単な迷路環境であるため，エージェント間で経験データを全部共有すると，学習に不適切なデータを共有することとなる．迷路環境 1 より複雑な迷路環境 2，$2(A)$，$2(AB)$ で，それぞれの一番良い学習効果を得た X_A が 50%，75%，25% である．罠がある複雑な MAS である迷路環境 $2(A)$，$2(AB)$ の X_C の初期値を 50% とするため，早期の学習段階において失敗の経験を共有できてエージェントを適切に学習させることが考えられる．すなわち，変動割合方式で，X_C を学習状況に応じて調整できるため，学習に適応な経験データを共有できると言える．

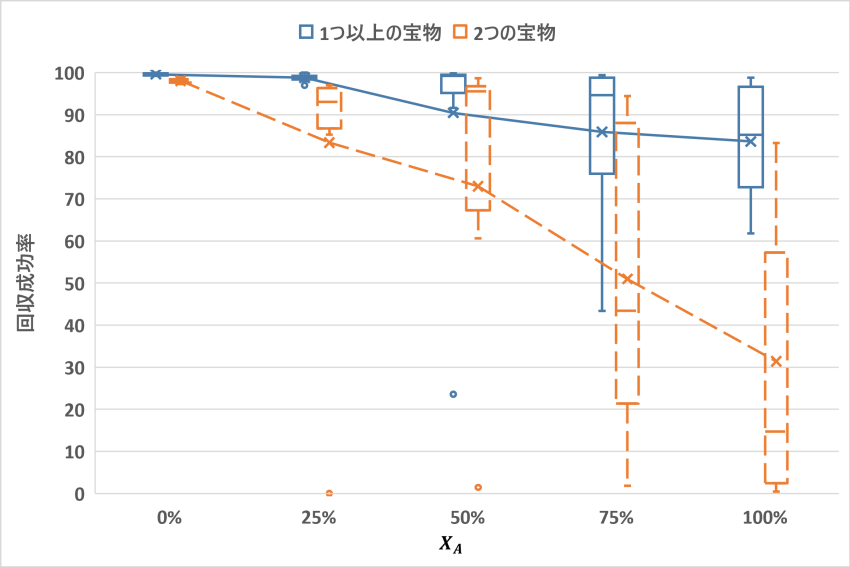

図 4.10: 変動割合方式の回収成功率（迷路環境 1)

表 4.11: 変動割合方式の回収成功率（迷路環境 1)

X_A	宝物	最大値	75%値	中央値	25%値	最小値	平均値
0%	1 つ以上	100.00	99.80	99.70	99.30	99.20	99.60
	2 つ	98.80	98.40	98.00	97.60	97.20	98.02
25%	1 つ以上	100.00	99.20	99.00	98.40	97.00	98.78
	2 つ	97.00	96.30	93.00	86.65	0.00	83.30
50%	1 つ以上	99.80	99.50	99.20	95.15	23.60	90.44
	2 つ	98.60	96.75	95.50	67.25	0.20	72.96
75%	1 つ以上	99.40	98.75	94.60	75.95	43.40	85.90
	2 つ	94.40	88.00	43.40	21.30	1.80	50.92
100%	1 つ以上	98.80	96.65	85.20	72.75	61.80	83.62
	2 つ	83.20	57.20	14.70	2.40	0.40	31.38

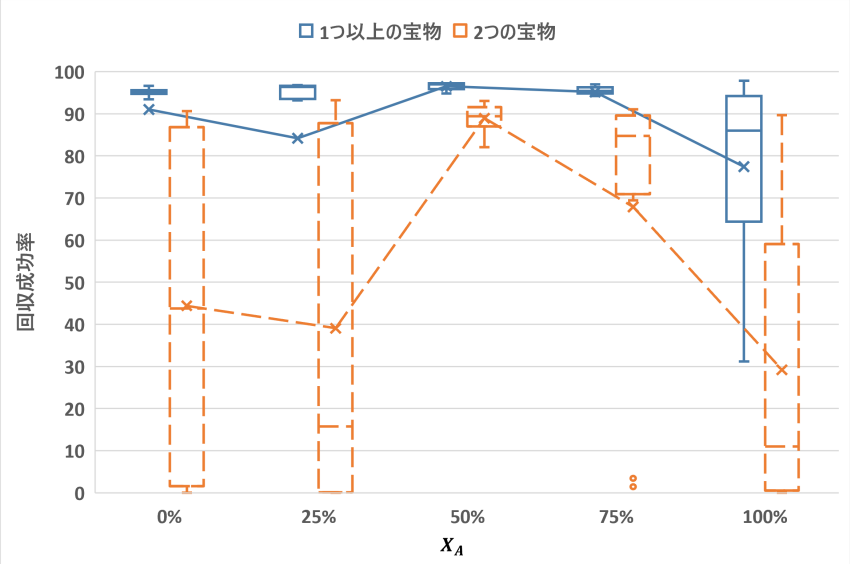

図 4.11: 変動割合方式の回収成功率（迷路環境 2)

表 4.12: 変動割合方式の回収成功率（迷路環境 2)

X_A	宝物	最大値	75%値	中央値	25%値	最小値	平均値
0%	1 つ以上	97.00	95.60	95.10	94.70	52.00	91.00
	2 つ	90.60	86.75	43.70	1.55	0.00	44.36
25%	1 つ以上	96.80	96.60	96.40	93.55	0.40	84.16
	2 つ	93.20	87.75	15.70	0.10	0.00	39.02
50%	1 つ以上	97.20	97.20	97.00	95.85	94.80	96.52
	2 つ	93.00	91.50	89.40	86.95	82.00	88.90
75%	1 つ以上	97.00	96.30	95.40	94.80	91.20	95.16
	2 つ	91.00	89.55	84.70	70.90	1.40	67.86
100%	1 つ以上	97.80	94.20	86.00	64.35	31.20	77.44
	2 つ	89.60	59.05	11.00	0.50	0.00	29.16

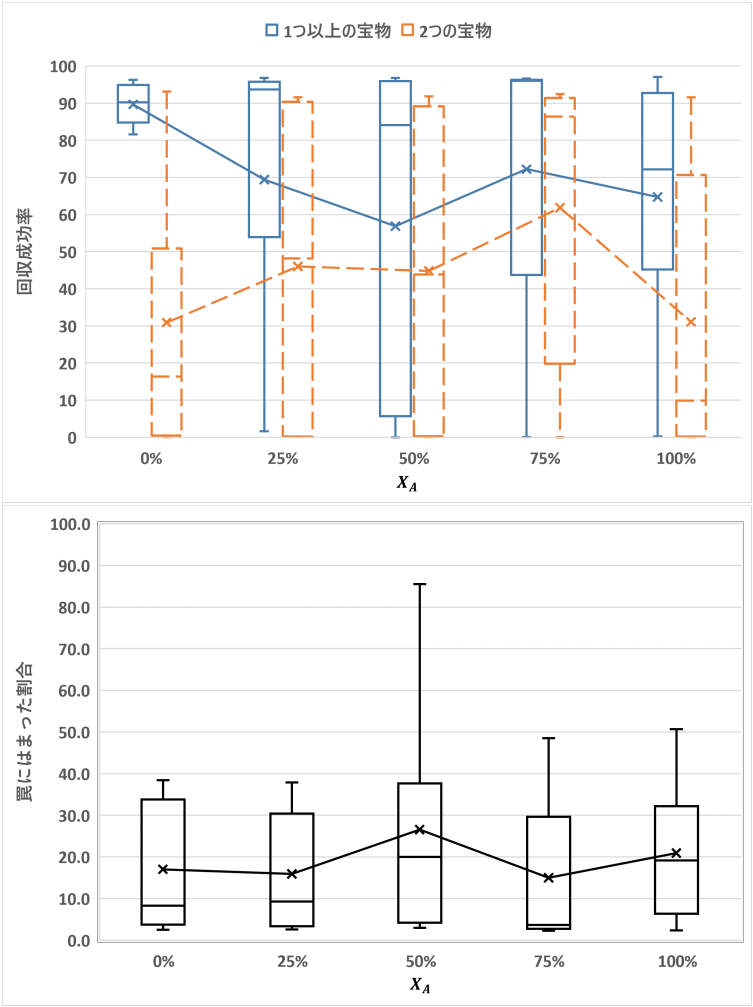

図 4.12: 変動割合方式（迷路環境 2(A)）

表 4.13: 変動割合方式（迷路環境 2(A)) の回収成功率

X_A	宝物	最大値	75%値	中央値	25%値	最小値	平均値
0%	1 つ以上	96.30	94.88	90.20	84.78	81.60	89.67
	2 つ	93.10	50.85	16.35	0.38	0.00	30.87
25%	1 つ以上	96.80	95.78	93.70	53.95	1.60	69.38
	2 つ	91.50	90.30	48.10	0.15	0.10	45.97
50%	1 つ以上	96.80	95.98	84.10	5.65	0.00	56.88
	2 つ	91.80	89.10	43.85	0.28	0.00	44.80
75%	1 つ以上	96.60	96.30	96.00	43.73	0.10	72.22
	2 つ	92.40	91.33	86.30	19.75	0.00	61.81
100%	1 つ以上	97.10	92.78	72.15	45.23	0.20	64.72
	2 つ	91.50	70.58	9.80	0.15	0.00	31.05

表 4.14: 変動割合方式（迷路環境 2(A)) の罠にはまった割合

X_A	最大値	75%値	中央値	25%値	最小値	平均値
0%	38.40	33.83	8.30	3.78	2.50	17.04
25%	37.90	30.40	9.30	3.40	2.60	15.93
50%	85.50	37.68	20.00	4.23	3.00	26.56
75%	48.50	29.68	3.65	2.73	2.30	15.00
100%	50.70	32.20	19.15	6.38	2.40	20.94

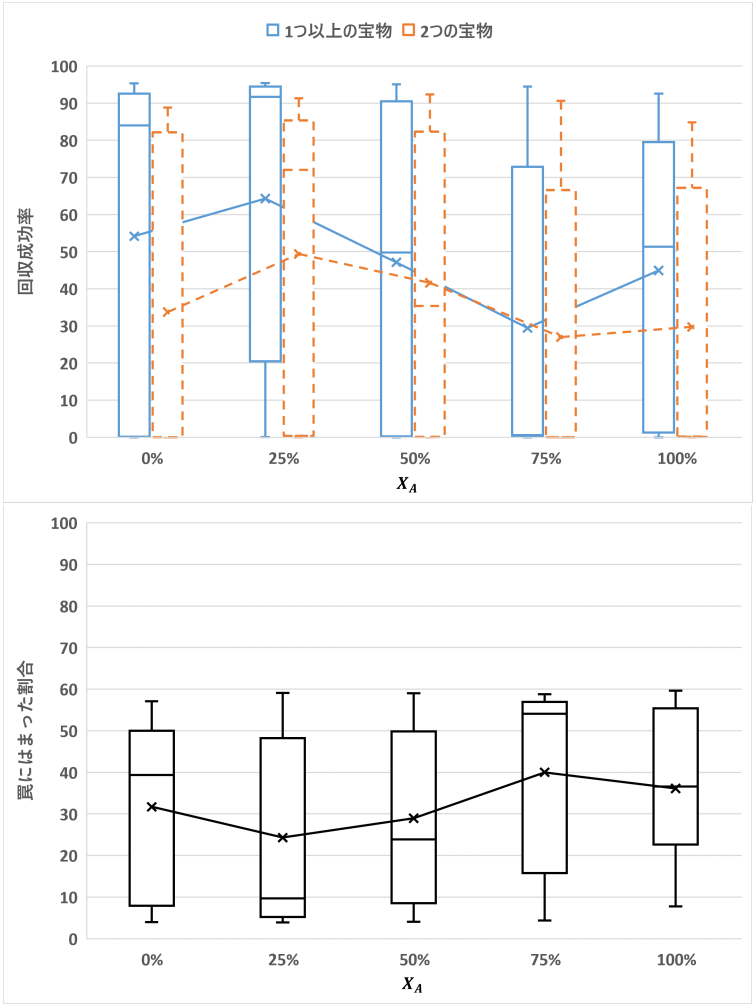

図 4.13: 変動割合方式（迷路環境 2(AB))

表 4.15: 変動割合方式（迷路環境 2(AB)) の回収成功率

X_A	宝物	最大値	75%値	中央値	25%値	最小値	平均値
0%	1 つ以上	95.30	92.60	84.00	0.18	0.00	54.15
	2 つ	88.80	82.13	0.00	0.00	0.00	33.71
25%	1 つ以上	95.40	94.48	91.70	20.43	0.10	64.29
	2 つ	91.30	85.30	71.95	0.33	0.00	49.36
50%	1 つ以上	95.10	90.50	49.80	0.23	0.00	47.12
	2 つ	92.30	82.25	35.35	0.03	0.00	41.56
75%	1 つ以上	94.50	72.88	0.55	0.13	0.00	29.43
	2 つ	90.60	66.53	0.00	0.00	0.00	26.95
100%	1 つ以上	92.60	79.55	51.30	1.28	0.00	44.94
	2 つ	84.80	67.13	0.15	0.03	0.00	29.69

表 4.16: 変動割合方式（迷路環境 2(AB)) の罠にはまった割合

X_A	最大値	75%値	中央値	25%値	最小値	平均値
0%	57.10	49.98	39.35	7.93	4.00	31.71
25%	59.10	48.20	9.70	5.23	3.90	24.31
50%	59.00	49.88	23.90	8.53	4.10	28.95
75%	58.80	56.95	54.05	15.80	4.40	40.01
100%	59.60	55.38	36.55	22.65	7.80	36.11

表 4.21，4.22 は迷路環境 1 の実験結果，表 4.23，4.24 は迷路環境 2 の実験結果，表 4.25–4.27 は迷路環境 2(A) の実験結果，表 4.28–4.30 は迷路環境 2(AB) の実験結果を示している．各表は総エピソードと毎 50 エピソードの 2 つの宝物の回収成功率，1 つ以上の宝物の回収成功率，罠にはまった割合 (罠がある迷路環境) の指標を示す．表 4.27，4.30 で示す結果を分析すると，罠がある複雑な迷路環境である迷路環境 2(A) と 2(AB) の罠にはまった割合の値はエピソード 751〜800 の段階から激減することが分かった．具体的には，表 4.27 で示すように，迷路環境 2(A) の一番良い学習効果を得た $X_A = 75\%$ の罠にはまった割合がエピソード 701〜750 の段階の 12.40% から次のエピソード 751〜800 の段階の 6.00% に半分以上減らした．同じのように，表 4.30 の結果から，迷路環境 2(AB) の一番良い学習効果を得た $X_A = 25\%$ の罠にはまった割合が 20.40% から 7.20% まで減少した．本論文では，行動選択に ϵ-greedy 法を使用するため，表 4.3 のように総エピソード数 1000 の 3/4 である 750 に到達すると，$\epsilon = 0.01$ となってエージェントが一番高い行動価値を得る行動を選択

する確率 $(1-\epsilon)$ が高くなり，適応な行動を選択できて罠を避けたことが考えられる．

提案手法により複数のエージェントがうまく学習して罠を避けて宝物に到達できることを検証するため，迷路環境 $2(AB)$ より大規模な迷路環境で変動割合方式を用いてシミュレーション実験を行う必要がある．図 4.3 で示すように，エージェント，罠，宝物の数がそれぞれ 1 を増え，より大規模な迷路環境である迷路環境 3 で，経験データ共有率 X_A を 25%，高い順の割合 X_C の初期値を 50%，初期エピソード数 $E_{initial}$ を 25，調整エピソード数 E_{modify} を 275，調整単位のエピソード数 E_{adjust} を 20 とし，1 試行 1000 エピソードで 10 試行のシミュレーション実験を行う．宝物の報酬は 1 つ目で $(+10)$，2 つ目で $(+15)$，3 つ目で $(+20)$ とする．総エピソードと毎 50 エピソードの 3 つの宝物，2 つ以上の宝物，1 つ以上の宝物の回収成功率，罠にはまった割合の 4 つの指標を観察する．図 4.14 と表 4.17–4.20 に示す実験結果を分析すると，総エピソードの後半の罠にはまった割合が前半より低くなり，学習に適応な経験データを共有できて正確に行動を選択できたと考えられる．

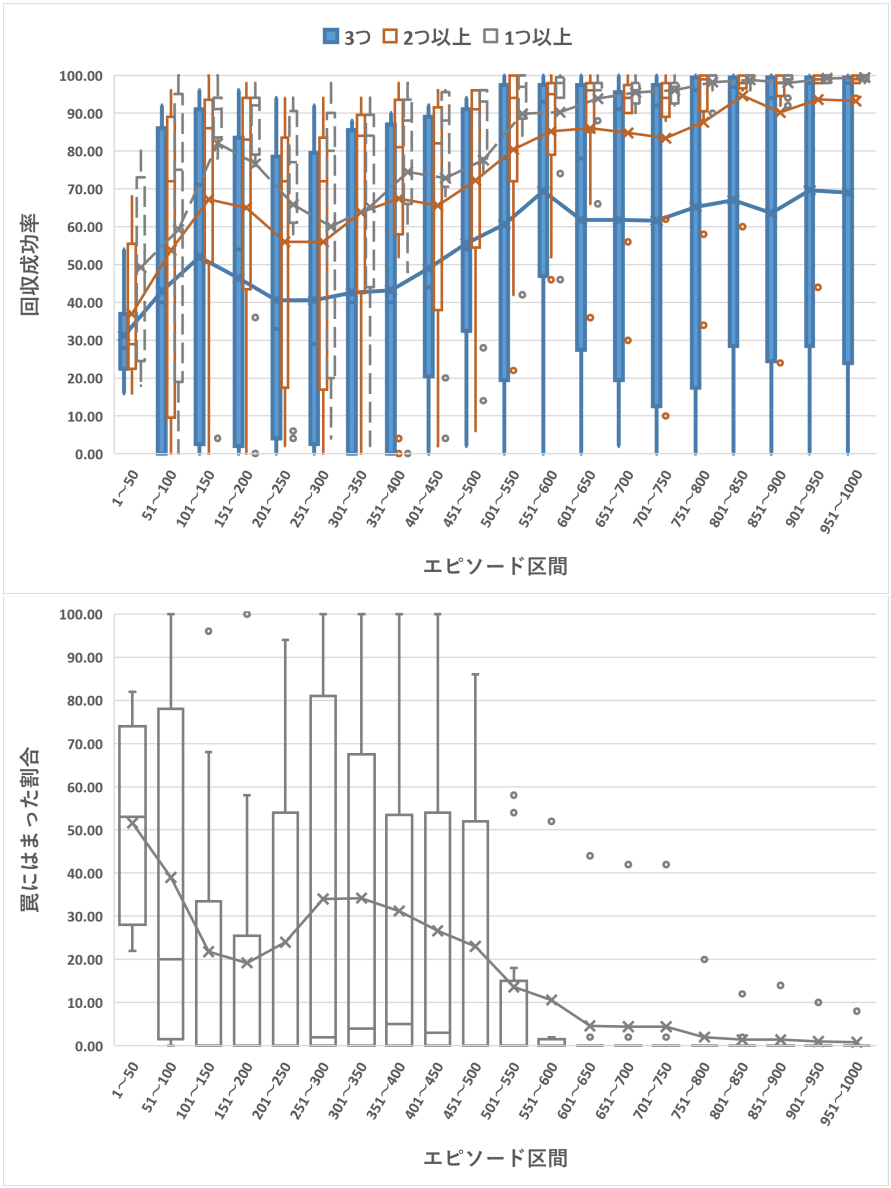

図 4.14: 変動割合方式 (迷路環境 3)

表 4.17: 変動割合方式 (迷路環境 3) の 3 つの宝物の回収成功率

エピソード区間	最大値	75%値	中央値	25%値	最小値	平均値
1〜50	54.00	37.00	28.00	22.50	16.00	31.20
51〜100	92.00	86.00	40.00	0.00	0.00	43.00
101〜150	96.00	91.00	71.00	2.50	0.00	52.00
151〜200	96.00	83.50	54.00	2.00	0.00	46.40
201〜250	94.00	78.50	33.00	4.00	0.00	40.60
251〜300	92.00	79.50	29.00	2.50	0.00	40.60
301〜350	88.00	85.50	40.00	0.00	0.00	42.60
351〜400	90.00	87.00	40.00	0.00	0.00	43.20
401〜450	92.00	89.00	44.00	20.50	0.00	49.00
451〜500	94.00	91.00	54.00	32.50	2.00	55.60
501〜550	100.00	97.50	78.00	19.50	0.00	60.60
551〜600	100.00	97.50	93.00	47.00	0.00	69.40
601〜650	100.00	97.50	78.00	27.50	0.00	61.80
651〜700	100.00	95.50	91.00	19.50	2.00	61.80
701〜750	100.00	97.50	92.00	12.50	0.00	61.60
751〜800	100.00	99.50	96.00	17.50	0.00	65.20
801〜850	100.00	99.50	97.00	28.50	0.00	67.00
851〜900	100.00	99.50	94.00	24.50	0.00	63.60
901〜950	100.00	99.50	98.00	28.50	0.00	69.60
951〜1000	100.00	99.50	98.00	24.00	0.00	69.00

表 4.18: 変動割合方式 (迷路環境 3) の 2 つ以上の宝物の回収成功率

エピソード区間	最大値	75%値	中央値	25%値	最小値	平均値
1〜50	68.00	55.50	29.00	22.50	16.00	37.00
51〜100	96.00	89.00	72.00	9.50	0.00	53.80
101〜150	100.00	93.50	86.00	50.50	0.00	67.20
151〜200	98.00	94.00	83.00	43.50	0.00	65.00
201〜250	94.00	83.50	72.00	17.50	2.00	56.00
251〜300	94.00	83.50	72.00	17.00	0.00	56.00
301〜350	94.00	89.50	84.00	42.50	0.00	63.80
351〜400	98.00	93.50	81.00	58.00	0.00	67.40
401〜450	96.00	91.50	82.00	38.00	2.00	65.60
451〜500	96.00	96.00	91.00	54.50	6.00	72.20
501〜550	100.00	100.00	94.00	72.00	22.00	80.40
551〜600	100.00	98.00	95.00	79.00	46.00	85.20
601〜650	100.00	98.00	96.00	85.00	36.00	86.00
651〜700	100.00	97.50	94.00	90.00	30.00	84.80
701〜750	100.00	98.00	94.00	89.00	10.00	83.40
751〜800	100.00	100.00	99.00	90.50	34.00	87.60
801〜850	100.00	100.00	98.00	96.50	60.00	94.60
851〜900	100.00	100.00	98.00	94.50	24.00	90.20
901〜950	100.00	100.00	99.00	98.00	44.00	93.60
951〜1000	100.00	100.00	99.00	98.00	44.00	93.20

表 4.19: 変動割合方式 (迷路環境 3) の 1 つ以上の宝物の回収成功率

エピソード区間	最大値	75%値	中央値	25%値	最小値	平均値
1〜50	80.00	73.00	49.00	24.50	18.00	49.20
51〜100	100.00	95.00	75.00	19.00	0.00	59.20
101〜150	100.00	94.00	91.00	83.50	4.00	82.00
151〜200	98.00	94.00	92.00	79.00	0.00	76.60
201〜250	94.00	90.50	77.00	61.00	4.00	65.80
251〜300	98.00	90.00	80.00	20.00	4.00	60.00
301〜350	94.00	89.50	84.00	44.00	2.00	65.20
351〜400	98.00	93.50	88.00	66.00	0.00	74.40
401〜450	96.00	95.50	88.00	70.50	4.00	72.80
451〜500	96.00	96.00	93.00	78.00	14.00	77.60
501〜550	100.00	100.00	97.00	88.50	42.00	89.80
551〜600	100.00	99.50	98.00	94.00	46.00	90.20
601〜650	100.00	98.00	98.00	96.00	66.00	93.80
651〜700	100.00	98.00	96.00	92.50	90.00	95.40
701〜750	100.00	98.00	97.00	92.50	92.00	96.00
751〜800	100.00	100.00	100.00	98.00	90.00	98.20
801〜850	100.00	100.00	99.00	98.00	96.00	98.80
851〜900	100.00	100.00	99.00	98.00	92.00	98.00
901〜950	100.00	100.00	100.00	98.00	98.00	99.20
951〜1000	100.00	100.00	100.00	98.50	98.00	99.40

表 4.20: 変動割合方式 (迷路環境 3) の罠にはまった割合

エピソード区間	最大値	75%値	中央値	25%値	最小値	平均値
1〜50	82.00	74.00	53.00	28.00	22.00	51.60
51〜100	100.00	78.00	20.00	1.50	0.00	39.00
101〜150	96.00	33.50	0.00	0.00	0.00	21.80
151〜200	100.00	25.50	0.00	0.00	0.00	19.20
201〜250	94.00	54.00	0.00	0.00	0.00	24.00
251〜300	100.00	81.00	2.00	0.00	0.00	34.00
301〜350	100.00	67.50	4.00	0.00	0.00	34.20
351〜400	100.00	53.50	5.00	0.00	0.00	31.20
401〜450	100.00	54.00	3.00	0.00	0.00	26.60
451〜500	86.00	52.00	0.00	0.00	0.00	23.00
501〜550	58.00	15.00	0.00	0.00	0.00	13.60
551〜600	52.00	1.50	0.00	0.00	0.00	10.60
601〜650	44.00	0.00	0.00	0.00	0.00	4.60
651〜700	42.00	0.00	0.00	0.00	0.00	4.40
701〜750	42.00	0.00	0.00	0.00	0.00	4.40
751〜800	20.00	0.00	0.00	0.00	0.00	2.00
801〜850	12.00	0.00	0.00	0.00	0.00	1.40
851〜900	14.00	0.00	0.00	0.00	0.00	1.40
901〜950	10.00	0.00	0.00	0.00	0.00	1.00
951〜1000	8.00	0.00	0.00	0.00	0.00	0.80

本論文では，提案手法の 3 種類の経験データ共有アーキテクチャを用いて複数のエージェントが存在する迷路環境における宝物探し問題を用いた数値実験を行い，エージェント間で一部の情報を共有することによる学習効率向上の効果を得た．具体的には，簡単な MAS である迷路環境 1，2 で，エージェント間での経験データを共有しない学習やランダム方式で一部の経験データを共有する学習が良い学習効果が得られる．複雑な MAS である迷路環境 2(A)，2(AB) で，累積報酬などの指標により分類された成功体験と失敗体験の両方を共有する固定割合方式と変動割合方式が適用できると考えられる．より大規模な MAS である迷路環境 3，変動割合方式により学習に適応な経験データを共有できると言える．

表 4.21: 迷路環境 1 の 2 つの宝物の回収成功率

$X_A(\%)$	0	25	50	75	100
total	98.02%	83.30%	72.96%	50.92%	31.38%
1〜50	80.60%	46.80%	50.60%	31.60%	17.80%
51〜100	100.00%	73.40%	61.20%	42.60%	21.20%
101〜150	99.80%	86.20%	72.20%	52.60%	17.20%
151〜200	100.00%	88.60%	70.60%	53.00%	23.60%
201〜250	99.80%	89.80%	76.80%	55.80%	34.40%
251〜300	100.00%	89.60%	78.40%	55.80%	44.60%
301〜350	100.00%	90.00%	79.80%	67.80%	39.40%
351〜400	100.00%	89.80%	80.00%	51.20%	35.00%
401〜450	100.00%	89.60%	79.80%	49.20%	39.20%
451〜500	100.00%	89.20%	80.20%	49.60%	41.40%

表 4.22: 迷路環境 1 の 1 つ以上の宝物の回収成功率

$X_A(\%)$	0	25	50	75	100
total	99.60%	98.78%	90.44%	85.90%	83.62%
1〜50	96.00%	89.80%	86.40%	86.00%	64.60%
51〜100	100.00%	98.20%	94.60%	88.20%	72.80%
101〜150	100.00%	99.80%	94.60%	96.00%	91.40%
151〜200	100.00%	100.00%	88.60%	92.80%	83.60%
201〜250	100.00%	100.00%	90.20%	90.00%	85.60%
251〜300	100.00%	100.00%	90.00%	84.60%	94.60%
301〜350	100.00%	100.00%	90.00%	85.00%	97.20%
351〜400	100.00%	100.00%	90.00%	78.00%	83.80%
401〜450	100.00%	100.00%	90.00%	75.80%	81.00%
451〜500	100.00%	100.00%	90.00%	82.60%	81.60%

表 4.23: 迷路環境 2 の 2 つの宝物の回収成功率

$X_A(\%)$	0	25	50	75	100
total	44.36%	39.02%	88.90%	67.86%	29.16%
1〜50	10.00%	13.40%	35.20%	15.80%	11.00%
51〜100	35.40%	39.00%	81.20%	54.20%	26.40%
101〜150	50.40%	39.60%	93.80%	70.60%	29.40%
151〜200	49.20%	36.40%	95.00%	66.20%	29.80%
201〜250	50.60%	39.00%	94.60%	76.40%	31.60%
251〜300	50.40%	45.20%	97.60%	77.40%	32.40%
301〜350	49.60%	48.20%	96.60%	79.00%	30.40%
351〜400	50.40%	47.80%	98.80%	79.40%	37.00%
401〜450	49.80%	41.00%	99.40%	79.60%	29.40%
451〜500	47.80%	40.60%	96.80%	80.00%	34.20%

表 4.24: 迷路環境 2 の 1 つ以上の宝物の回収成功率

$X_A(\%)$	0	25	50	75	100
total	91.00%	84.16%	96.52%	95.16%	77.44%
1〜50	58.00%	51.20%	65.20%	59.20%	44.60%
51〜100	98.60%	83.80%	100.00%	97.40%	83.20%
101〜150	93.00%	89.80%	100.00%	98.80%	92.00%
151〜200	92.00%	89.80%	100.00%	99.60%	91.40%
201〜250	94.00%	90.00%	100.00%	99.80%	82.80%
251〜300	97.40%	89.20%	100.00%	99.00%	78.40%
301〜350	90.20%	89.60%	100.00%	98.00%	83.80%
351〜400	93.80%	89.40%	100.00%	100.00%	75.80%
401〜450	99.60%	87.00%	100.00%	99.80%	69.20%
451〜500	93.40%	81.80%	100.00%	100.00%	73.20%

表 4.25: 迷路環境 2(A) の 2 つの宝物の回収成功率

X_A(%)	0	25	50	75	100
total	30.87%	45.97%	44.80%	61.81%	31.05%
1〜50	4.00%	8.20%	11.40%	15.00%	9.40%
51〜100	18.80%	34.00%	37.40%	47.20%	26.60%
101〜150	20.40%	37.40%	42.20%	51.40%	27.20%
151〜200	23.00%	45.40%	40.40%	59.80%	28.80%
201〜250	21.60%	45.40%	47.20%	64.80%	25.60%
251〜300	23.40%	50.20%	46.60%	60.80%	27.20%
301〜350	22.80%	55.60%	45.60%	60.20%	31.60%
351〜400	29.40%	55.00%	46.00%	65.00%	32.60%
401〜450	30.60%	51.00%	47.20%	63.80%	31.80%
451〜500	32.60%	47.20%	45.60%	65.00%	35.00%
501〜550	22.80%	48.80%	48.80%	66.80%	37.00%
551〜600	27.40%	48.60%	48.40%	66.00%	35.40%
601〜650	33.60%	48.40%	48.60%	67.60%	29.60%
651〜700	34.40%	48.00%	46.60%	68.00%	29.00%
701〜750	36.80%	48.20%	48.80%	67.80%	29.20%
751〜800	43.00%	49.40%	49.60%	69.60%	29.60%
801〜850	45.00%	49.40%	49.40%	69.60%	35.60%
851〜900	49.40%	49.60%	49.00%	69.60%	40.00%
901〜950	50.00%	49.60%	48.60%	69.40%	40.00%
951〜1000	48.40%	50.00%	48.60%	68.80%	39.80%

表 4.26: 迷路環境 2(A) の 1 つ以上の宝物の回収成功率

X_A(%)	0	25	50	75	100
total	89.67%	69.38%	56.88%	72.22%	64.72%
1〜50	39.20%	38.60%	32.80%	37.20%	38.00%
51〜100	85.40%	63.80%	54.80%	65.40%	64.80%
101〜150	88.80%	64.60%	56.20%	71.00%	69.20%
151〜200	91.20%	66.00%	53.80%	73.00%	62.80%
201〜250	89.20%	70.60%	58.20%	75.20%	56.60%
251〜300	86.00%	69.80%	57.40%	77.40%	59.20%
301〜350	90.20%	68.40%	57.80%	77.60%	64.20%
351〜400	90.60%	68.00%	57.40%	79.00%	64.20%
401〜450	91.00%	69.00%	58.60%	75.00%	62.20%
451〜500	90.80%	72.00%	56.60%	77.00%	66.80%
501〜550	86.60%	74.20%	58.40%	75.60%	63.40%
551〜600	89.40%	70.60%	61.00%	76.40%	63.20%
601〜650	94.80%	75.20%	58.80%	75.20%	62.00%
651〜700	95.40%	76.60%	58.80%	75.60%	59.00%
701〜750	89.20%	77.00%	59.00%	74.60%	73.20%
751〜800	99.60%	75.80%	59.40%	72.00%	71.20%
801〜850	98.80%	70.20%	60.00%	72.00%	72.60%
851〜900	99.40%	76.00%	59.20%	71.40%	70.00%
901〜950	98.40%	71.00%	59.80%	71.80%	71.40%
951〜1000	99.40%	70.20%	59.60%	72.00%	80.40%

表 4.27: 迷路環境 2(A) の罠にはまった割合

X_A(%)	0	25	50	75	100
total	17.04%	15.93%	26.56%	15.00%	20.94%
1〜50	43.40%	44.60%	49.60%	43.00%	49.00%
51〜100	22.20%	26.80%	30.80%	23.20%	30.00%
101〜150	21.20%	27.20%	29.80%	23.60%	22.00%
151〜200	19.60%	25.60%	30.80%	19.40%	29.40%
201〜250	23.20%	26.00%	32.60%	18.60%	36.60%
251〜300	26.80%	23.80%	31.60%	19.40%	33.60%
301〜350	20.80%	17.20%	30.40%	18.00%	27.20%
351〜400	20.60%	22.20%	32.80%	17.40%	31.20%
401〜450	20.60%	19.40%	30.20%	19.80%	30.60%
451〜500	22.40%	17.80%	33.40%	18.40%	31.00%
501〜550	18.00%	10.80%	27.20%	12.20%	18.20%
551〜600	20.00%	11.20%	20.80%	11.20%	15.60%
601〜650	14.00%	12.00%	26.40%	12.80%	17.20%
651〜700	17.00%	11.60%	24.40%	11.00%	19.40%
701〜750	13.20%	10.20%	26.40%	12.40%	11.20%
751〜800	3.00%	3.80%	16.00%	6.00%	4.60%
801〜850	4.80%	2.80%	14.20%	3.80%	3.20%
851〜900	3.00%	2.20%	15.00%	2.40%	3.00%
901〜950	3.20%	1.80%	14.00%	4.00%	2.80%
951〜1000	3.80%	1.60%	14.80%	3.40%	3.00%

表 4.28: 迷路環境 2(AB) の 2 つの宝物の回収成功率

X_A(%)	0	25	50	75	100
total	33.71%	49.36%	41.56%	26.95%	29.69%
1〜50	1.60%	10.60%	9.20%	8.00%	3.60%
51〜100	6.40%	28.20%	22.80%	23.60%	20.00%
101〜150	21.40%	36.80%	35.40%	27.20%	22.80%
151〜200	33.80%	42.20%	38.60%	27.00%	23.40%
201〜250	34.20%	44.00%	40.00%	26.40%	18.40%
251〜300	37.20%	42.20%	34.60%	25.80%	18.60%
301〜350	37.40%	53.40%	39.00%	28.00%	24.00%
351〜400	36.60%	55.00%	38.80%	28.00%	26.20%
401〜450	36.40%	56.20%	45.00%	27.40%	31.00%
451〜500	38.60%	55.40%	45.80%	27.20%	32.80%
501〜550	38.80%	56.80%	46.00%	28.00%	36.60%
551〜600	38.60%	55.40%	45.20%	29.00%	34.00%
601〜650	38.00%	52.00%	48.80%	28.40%	36.00%
651〜700	38.80%	54.20%	48.20%	29.20%	34.40%
701〜750	38.60%	54.00%	47.00%	28.20%	35.40%
751〜800	39.80%	58.60%	50.00%	29.60%	38.80%
801〜850	39.60%	59.80%	49.40%	30.00%	39.20%
851〜900	39.60%	59.40%	48.20%	29.80%	38.80%
901〜950	39.40%	54.00%	49.60%	28.80%	40.00%
951〜1000	39.40%	59.00%	49.60%	29.40%	39.80%

表 4.29: 迷路環境 2(AB) の 1 つ以上の宝物の回収成功率

X_A(%)	0	25	50	75	100
total	54.15%	64.29%	47.12%	29.43%	44.94%
1〜50	21.60%	32.20%	22.80%	19.00%	21.20%
51〜100	50.60%	56.20%	40.00%	29.60%	36.20%
101〜150	52.00%	60.00%	44.20%	31.80%	41.80%
151〜200	54.20%	63.60%	44.40%	31.40%	43.20%
201〜250	53.20%	65.00%	45.60%	29.80%	36.60%
251〜300	55.00%	65.40%	41.20%	28.60%	40.20%
301〜350	54.60%	65.40%	47.80%	30.00%	39.60%
351〜400	53.60%	65.20%	47.80%	29.00%	39.80%
401〜450	54.40%	65.80%	48.80%	28.40%	48.20%
451〜500	56.00%	65.20%	49.20%	30.00%	48.00%
501〜550	56.00%	66.60%	54.00%	31.40%	46.80%
551〜600	56.40%	67.00%	52.00%	30.80%	51.60%
601〜650	55.40%	65.40%	52.80%	29.20%	51.40%
651〜700	57.60%	67.40%	51.00%	30.00%	50.40%
701〜750	56.00%	67.40%	48.60%	30.60%	51.80%
751〜800	59.40%	69.60%	52.20%	30.00%	51.20%
801〜850	59.20%	69.60%	50.00%	30.00%	50.40%
851〜900	59.20%	69.60%	50.00%	30.00%	49.40%
901〜950	59.20%	69.80%	50.20%	29.40%	50.40%
951〜1000	59.40%	69.40%	49.80%	29.60%	50.60%

表 4.30: 迷路環境 2(AB) の罠にはまった割合

X_A(%)	0	25	50	75	100
total	31.71%	24.31%	28.95%	40.01%	36.11%
1〜50	60.20%	51.20%	57.00%	59.80%	60.20%
51〜100	47.80%	42.00%	46.80%	57.00%	53.20%
101〜150	46.40%	35.60%	42.20%	55.60%	49.60%
151〜200	42.20%	33.60%	41.60%	53.40%	47.60%
201〜250	42.00%	30.20%	39.60%	54.80%	56.20%
251〜300	39.80%	29.00%	40.60%	53.20%	55.40%
301〜350	39.40%	30.00%	36.60%	51.60%	55.40%
351〜400	39.20%	30.80%	38.40%	52.40%	54.60%
401〜450	41.20%	26.80%	33.80%	51.00%	42.60%
451〜500	41.80%	28.20%	35.40%	51.60%	40.40%
501〜550	31.60%	21.80%	27.80%	39.80%	34.20%
551〜600	30.20%	22.60%	26.40%	39.60%	33.20%
601〜650	28.60%	25.60%	24.60%	39.00%	30.20%
651〜700	30.60%	22.80%	27.00%	41.40%	32.00%
701〜750	30.80%	20.40%	28.00%	41.80%	31.80%
751〜800	9.80%	7.20%	7.60%	11.60%	9.40%
801〜850	9.00%	7.00%	5.60%	10.00%	9.00%
851〜900	5.80%	6.40%	7.00%	10.00%	10.60%
901〜950	9.80%	7.40%	6.20%	12.00%	7.60%
951〜1000	8.00%	7.60%	6.80%	14.60%	9.00%

第 5 章

おわりに

本論文では，MAS のエージェント間経験データを共有する手法を改善した提案手法はすべてのデータではなく，一部を共有する学習手法である．シミュレーション実験により，データを部分的に共有することで，学習効率が向上することが確認できた．さらに，共有されるデータをランダムに選択するランダム方式，得られた報酬に基づいてランク付けデータの高い順から一定の比率で共有されるデータを選択する固定割合方式，固定割合方式の高い順の比率を学習状況により調整する変動割合方式の 3 種類の経験データ共有アーキテクチャにより，様々な複雑さが違う MAS における学習効率の改善することを示した．

今後の課題として，3 種類の経験データ共有アーキテクチャの組み合わせにより最適化する手法の構築や，現実的な問題の適用が考えられる．

謝辞

本論文作成の全過程を通じて終始理解ある御教授，御指導，御鞭撻を賜りました，西崎 一郎教授，林田 智弘准教授，関崎 真也助教に厚く御礼申し上げます.

また，2 年間を通じて，親切な御助力を頂きました社会情報学研究室の皆様に深く御礼申し上げます.

参考文献

[1] L. Busoniu, R. Babuska, B. Deschutter, “Multi-agent reinforcement learning:An overview,” *Chapter 7 in Innovations in Multi-Agent Systems and Applications – 1(D. Srinivasan and L.C. Jain, eds.), vol.310 of Studies in Computational Intelligence, Berlin, Germany: Springer*, pp. 183-221, 2010.

[2] K. Zhang, Z. Yang, T. Başar, “Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms”, *In: Vamvoudakis K.G., Wan Y., Lewis F.L., Cansever D. (eds) Handbook of Reinforcement Learning and Control. Studies in Systems, Decision and Control*, 325. Springer, Cham, 2021.

[3] T.T. Nguyen, N.D. Nguyen, S Nahavandi, “Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications”, *IEEE Trans. Cybernetics*, 50, pp. 3826–3839, 2020.

[4] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, I. Mordatch, “Multi-agent Actor-Critic for mixed cooperative-competitive environments”, *In Advances in Neural Information Processing Systems*, pp. 6379–6390, 2017.

[5] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T. Harley, T.P. Lillicrap, D. Silver, K. Kavukcuoglu, “Asynchronous Methods for Deep Reinforcement Learning”, *In Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pp 1928–1937, 2016.

[6] G. Bacchiani, D. Molinari, M. Patander, “Microscopic traffic simulation by cooperative multi-agent deep reinforcement learning”, *In Proceedings. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019)*, 2019, IFAAMAS.

[7] R.S. Sutton and A.G. Barto. *Reinforcement learning: An introduction*, MIT press, 2018.

[8] 牧野 貴樹, これからの強化学習, 森北出版, 2016.

[9] 伊藤 多一, *Python* 深層強化学習入門, 翔詠社, 2019.

[10] Ovalle, Alvaro. “Deep Reinforcement Learning Variants of Multi-Agent Learn-

ing Algorithms. ” (2016).

[11] Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S., “Counterfactual Multi-Agent Policy Gradients”. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

[12] Hernandez-Leal, P., Kartal, B. & Taylor, M.E. A survey and critique of multiagent deep reinforcement learning. *Auton Agent Multi-Agent Syst* 33, 750–797 (2019).