修士論文

マルチエージェントシステムを用いた荷物搬送問題における
強化学習の行動特性の検証

Verification of action selection of reinforcement learning
in the luggage transportation problem using a multi-agent system

---

報告者

学籍番号：1265047
氏名：橋本 大輔

指導教員

星野 孝総 准教授

令和6年 2 月 19日

高知工科大学大学院工学研究科
基盤工学専攻電子・光工学コース

目 次

第1章　序論

1.1.　研究背景

近年，機械学習 (Machine Learning) を用いたロボットやシステムが生活の一部で身近に活用されるようになった．通販サイトでは倉庫から荷物発送をするまでのプロセスに機械学習を取り入れることにより，荷物発送までを自動化する企業や，コロナ禍後の非接触や人員削減を目的としたファミレスでの自動配膳ロボットの普及など，様々な分野で応用されている [1–3]．機械学習の始まりは 1950 年代のニューラルネットワークやチェッカー・プログラムとされている [4]．ここから人工知能のブームが始まり，現在は第 3 次人工知能ブームとして深層学習が精力的に研究されている [5]．ここで機械学習とは，基本的に何らかの損失関数を設定し，その損失の期待値を最小化することを目的とする学習法であり，その計算や処理とされている [6].

機械学習には一般に，教師あり学習，教師なし学習，半教師あり学習，強化学習などの学習方法に分類される．また，これらの学習法の中にもニューラルネットワークを用いた深層学習，遺伝的アルゴリズムを用いた進化計算など人間・自然界からアイディアを得た学習法が存在する．特に，深層学習は医療分野や画像処理の分野で多く用いられ，近年の産業発展に大きくかかわっている [7].

教師あり学習とは，人間があらかじめ正解データ・訓練データを分類し，それを学習させる．教師あり学習の目的は，入力に対して出力する分類ラベルの訓練データとの損失関数を最小化することである．教師あり学習は，画像処理における物体検出など，現在では実際に日常生活に用いられることが増加した．しかし，学習を行う場合にはデータセットを作成する必要があり，データの分類ラベル付けを人が行わなければならない問題がある．

教師なし学習とは，教師あり学習と異なり入力に対して分類ラベルが付いておらず，入力データの分布や組み合わせを推定することが目的の機械学習である．このため，教師なし学習はデータの低次元圧縮やクラスタリングなどに用いられる．

半教師あり学習とは，教師あり学習と教師なし学習の中間となる学習法であり，一部のデータのみラベル付けし，分類ラベル付きデータと分類ラベルなしデータを同時に学習させるものである．教師あり学習と比較して，分類ラベル付けされていないデータを用いることで推定のモデル性能を向上させることができる [8].

強化学習は，対象に対する知識が未知であり，対象に対してはたらきかけを行った場合に観測環境が変化した際，最適なはたらきかけを見つけ出す学習である [9,10]．最も古典的な強化学習では，周囲にはたらきかける行動の主体をエージェント，はたらきかけられた場合に変化する対象のことを環境として扱う．強化学習では，エージェントが自律して環境にはたらきかけ，設定した目標に達した場合に報酬が与えられる．このため，画像認識やニューラルネットワークなどの教師あり学習に用いられる正解データを必要としな

い．人間が正解データを与える必要がないため，プログラムの設計者にとって解が自明でない場合であってもタスク達成することができる．このため強化学習は，囲碁や将棋などの場面ごとでの正解が存在しないゲームなどで，報酬を最大化させることで人間を打ち破ることができる [11]．また，未知な問題設定に関しても有効性があるため，災害時の経路探索問題や荷物搬送問題 [12,13]，実ロボットへの応用などが研究されている [14–16]．特に，荷物搬送問題などでは，複数体のエージェントを用いるマルチエージェントシステムを採用することで，学習の効率が向上する．

マルチエージェントシステムとは，複数体の独立したエージェント群が，タスク達成のために行動と学習を行うシステムである．一般に，マルチエージェントシステムでは複数のエージェントを用いることにより，シングルエージェントと比較して学習の高速化と並列処理によるタスクの効率化が期待される．マルチエージェントシステムの応用例として，荷物搬送問題や Multi-agent pickup and delivery(MAPD) 問題がある [17,18]．これら問題は，複数のエージェントを用いて荷物搬送を自動化させるものである．各エージェントのタスクは，荷物搬入口や資材置き場などから荷物を受け取り，移動可能なマスやノード間を伝い荷物搬出口や荷物が必要な場所へと搬送するものである．この技術は，近年発達しているドローン運搬や倉庫での荷物搬送の自動化に期待される技術である．しかし，エージェントが通ることによる環境の変化や，エージェント数の単純な増加によるリソース競合により，シングルエージェントと比較して効率が低下してしまう場合がある．特に，デッドロックというエージェント同士の相互進路妨害が発生した場合，エージェントの内部状態だけでは解決が困難であり，全体のタスク効率が大幅に低下する．このため，マルチエージェントシステムには各エージェントの協調行動が不可欠である [19]．

1.2. 研究目的

本研究では，マルチエージェントシステムを用いた荷物搬送問題において，強化学習を導入することによる特性の検証とそれに伴う学習の効率化を目的とする．一般に，マルチエージェントシステムを用いた強化学習の場合，それぞれのエージェントは通信やデータの集計を行い，それらのデータを用いて学習を行う [20]．特に，環境のすべてが知覚できる全知覚や，状態に対する行動の期待報酬が現在の状態と行動にのみ依存するマルコフ決定過程では，強化学習の結果が最適方策に収束することが知られている [21]．しかし，導入コストの問題や通信の行えない環境での学習など，エージェントにとって学習に用いることのできる情報が少ないほど学習を収束させることが困難となる．特に，部分観測でそれぞれ独立して学習を行うエージェントは，学習過程を環境に強く依存するため，どのような環境で荷物搬送が可能であるかを確認する必要がある．このため本研究では，部分観測環境下のエージェントが環境に対してどのように学習行い，どのように振る舞うかを検証する．

また，部分観測環境下でのマルチエージェントシステムを用いた強化学習の検証において，本研究では従来研究で用いられる行動選択手法に加え 2 種類の行動選択手法を比較する．1 つ目は，一般に用いられる行動選択手法である ϵ-greedy 方策と Softmax 方策を組み合わせた手法である．確率 ϵ でランダム探索を行う ϵ-greedy 方策によりあらかじめ環境探索を行い，ある程度探索が完了したのちに ϵ-greedy 方策よりも平均ステップ数の少ない Softmax 方策を用いることにより，Softmax 方策のみでは方策の収束が困難であった環

境に対する特性を検証する．2 つ目は，相対ベクトルを導入する手法である．清本らの研究 [22] では，部分観測環境下の強化学習では，初期位置を基準とした位置ベクトルを導入することにより最適行動を獲得できるとしている．そこで本研究では，数ステップ前と現在位置との相対的なベクトルを用いることにより，最適行動の獲得が向上すると考える．清本らの手法では，固定された初期位置を使う必要があり，学習した場所と別の初期位置の場合，すべてを再学習しなければならない．また，部分観測の特性よりも位置ベクトルへの依存度が強いため，汎用性が低い．対して数ステップ前の位置を用いた相対ベクトルであれば，これらの課題を解決できると考える．

本研究では，これら手法と各種実験環境の実験・比較することにより，学習が行える環境の推定及びエージェントの行動特性について検証する．

本論文は全 6 章で構成されている．第 1 章では，本研究における背景と目的を述べた．第 2 章では，本研究で用いる基礎知識として，強化学習のアルゴリズムとマルチエージェントシステムについて述べる．第 3 章では，荷物搬送問題における環境別の学習変化について実験結果を述べ，この実験結果をもとに第 4 章で相対ベクトルを導入した手法の概要および実験結果について述べる．第 5 章では，実験条件である停止行動及び第 4 章におけるエージェントの利他的行動に対する考察を行う．最後に，第 6 章で本研究のまとめについて述べている．その後，今後の研究課題と展望について述べる．

第2章　強化学習

2.1.　強化学習

強化学習とは，行動決定の主体となるエージェントと環境の相互作用を学習することにより，環境に対して適切な行動を選択できるようになる機械学習の一種である．本論文では，エージェントが行う学習とは行動選択時の環境に対するはたらきかけの種類と定義する．環境に対して行った行動が望ましい場合，報酬が与えられる．この報酬は環境から与えられ，エージェントはどの行動が環境に対してより高い報酬を得られるかを探索する．このため，エージェントを用いる目的としては，得られる報酬の総和を最大化する行動の探索である．強化学習の利点として，エージェントが自律的な行動を起こして報酬を得られるように学習するため，教師信号のような正解データを必要としない点がある．このため，未知な環境や問題に用いることが可能である．

本節では，強化学習の概要と仕組みについて述べる．

2.1.1.　強化学習の概要

強化学習を構成する要因は，主に図 2.1 に示すようにエージェント，環境，エージェントと環境間の相互作用である．エージェントが環境の中で試行錯誤的に環境の知覚と行動を繰り返し，報酬を受け取る．報酬は目標とした状態になると環境からエージェントに与えられる．このとき，エージェントは状態 $s(s \in S$: S は知覚可能な状態の集合) を観測 $o(o \in O$: O は知覚可能な観測の集合) として知覚する．エージェントは，状態 s において実行できる行動群 A の中から 1 つを選び実行する．状態 s における行動 $a(a \in A)$ をルール (s, a) と呼び，ルールを選ぶ判断基準を方策 (Policy) と呼ぶ．このルールを実行することで，環境から報酬 r を得る．強化学習では，エージェントがこの報酬を受け取るための行動を選択する根拠となる状態価値関数 (State-value Function) と，それに伴う報酬関数 (Reward Function) が用いられる．報酬は試行錯誤により最終的に目標を達成した際に与えられることが多い．このため，ある時点でのエージェントが選択した行動の良し悪しの判定には時間的な遅れが存在する．このことを遅延報酬という．

2.1.2.　強化学習の仕組み

強化学習の仕組みとしては，エージェントと環境のやりとりである．強化学習の仕組みを図 2.1 に示す．

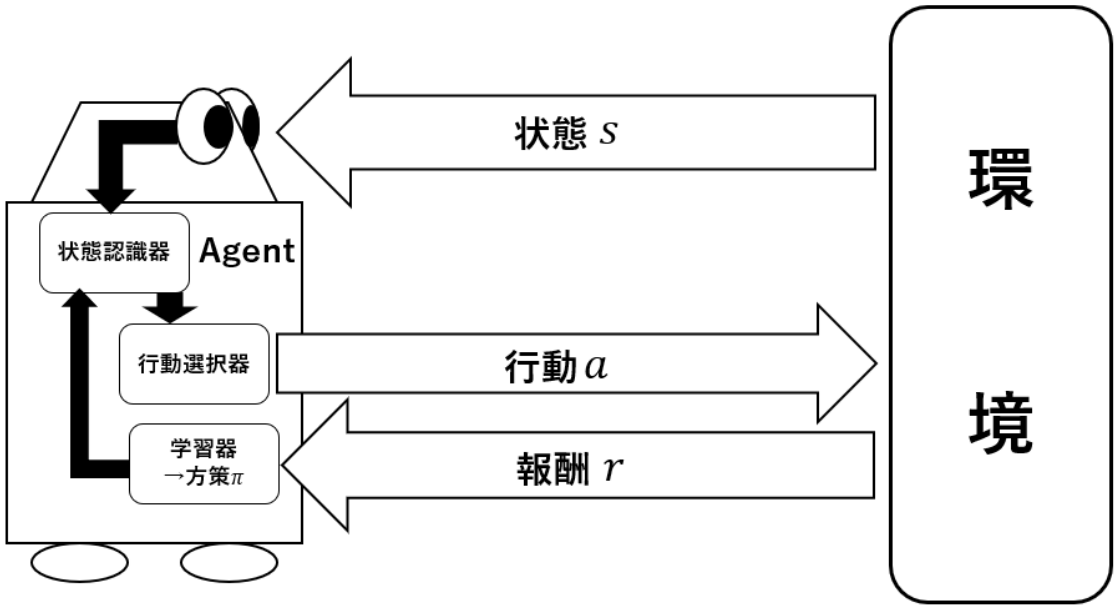

図 2.1: 強化学習の概要

強化学習の流れを以下に示す.

1. エージェントはある時刻 t において，環境の状態観測 s_t に基づき意思決定を行い，行動 a を環境に出力する．意思決定とは状態認識器で生成された候補行動集合の中から，行動選択器で行動を選択することである.

2. エージェントの行動 a により，環境は s_{t+1} へ遷移し，遷移に応じた報酬 r_t が環境からエージェントに与えられる．環境から何も受け取らない場合は，報酬 $r_t = 0$ を受け取ったとする．エージェントは，受け取った報酬から学習器で方策 π の評価を更新する.

3. 時刻 t を $t + 1$ に遷移させる.

以上のプロセスを学習が終わるまで繰り返す．報酬から方策の評価を更新するため，方策が改善され，報酬関数を最大化できる．このため強化学習とは，報酬関数を最大化する方策を構築するアルゴリズムである.

以下に強化学習の主要な構成要素である方策，状態価値関数，報酬関数について述べる.

方策

方策 $\pi(s, a)$ は，ある時点のエージェントに対して，行動に関する基準を表す．$\pi(s, a)$ は，状態 s において行動 a を選択する確率を表す．方策は一般に，すべての $s \in S$,$a \in A$ に対して $\pi(s, a) > 0$ である.

強化学習に用いられる方策の例として，式 (2.1) で表される ϵ-greedy 方策や，式 (2.2) で表される Softmax 方策がある [23].

$$\pi(s,a) = \begin{cases} 1-\epsilon+\frac{\epsilon}{|A(s)|} & (a = \underset{a\in A}{\text{argmax}} Q(s,a)) \\ \frac{\epsilon}{|A(s)|} & (\text{otherwise}) \end{cases} \tag{2.1}$$

$$\pi(s,a) = \frac{exp(Q(s,a)/T)}{\sum_{b\in A} exp(Q(s,b)/T)} \tag{2.2}$$

ϵ-greedy 方策は，図 2.2 に示すように確率 $\epsilon(0 \leq \epsilon \leq 1)$ で行動をランダムに選択し，$1-\epsilon$ で最も有望と思われる行動を選択する手法である．$|A(s)|$ は行動選択の候補行動数である．常に行動価値の最も高い行動を選択する greedy 方策と比較して，行動の収束までより多くの試行回数を必要とするが，確率 ϵ で探索を行うため最適な結果に収束しやすい．

Softmax 方策は，図 2.3 に示すように行動価値を等級付けした関数によって行動確率を変化させるものである [24]．このため，行動価値の最も高い行動に最も高い選択確率が与えられる．ここで，T は温度パラメータと呼ばれる正の定数である [25]．温度定数 T が小さいと有望と思われる行動の選択確率が大きくなり，T が大きくなるにつれてランダムな選択確率へと近づく．ここで $Q(s,a)$ はルールに対する有効性を表す値である．

複数の方策が提案される理由は，強化学習に知識利用と探索のジレンマが存在するからである．知識を利用すればするほど，探索が広く行われず，探索を広く行えば，知識があまり利用されないトレードオフの関係が両者には成り立っている [26]．強化学習では，環境や方策で値の取り方が大きく変わることから，エージェントと環境の両方から適する方策を選択することが重要である．

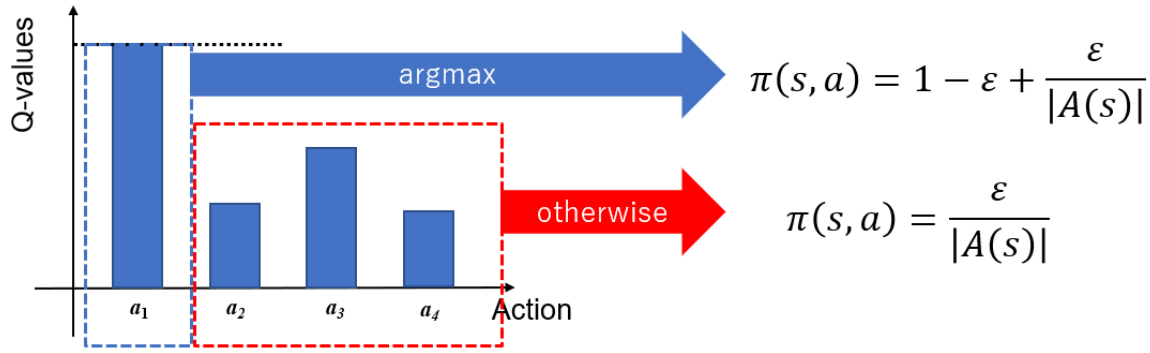

図 2.2: ϵ-greedy 方策の概要図

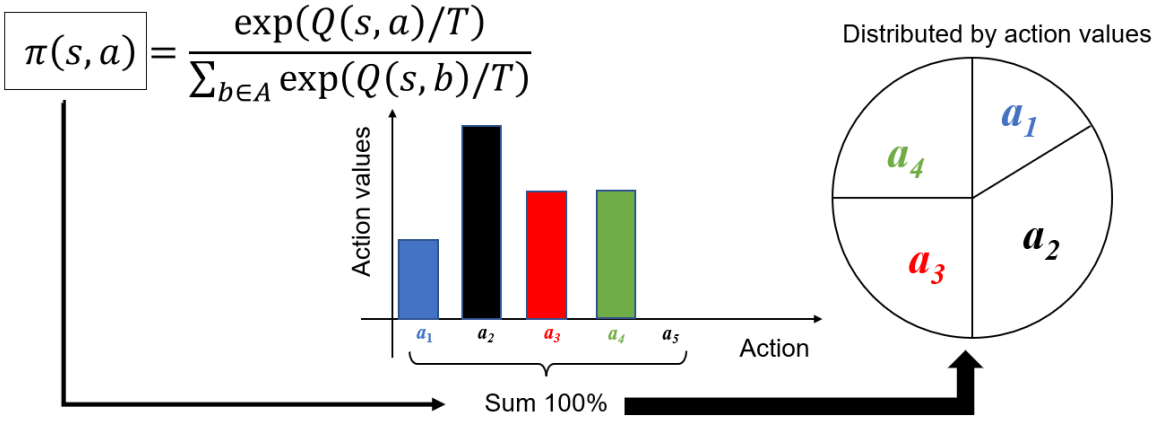

図 2.3: Softmax 方策の概要図

報酬関数

報酬 r は，強化学習において環境から受け取るエージェントの目標である [27]．報酬関数は，エージェントがある方策 π に従った際の状態価値関数が，将来エピソード終了までに獲得できる報酬の総和を定義する．ここで，エピソードとは初期状態から報酬までのルール系列，もしくは報酬から次の報酬までのルール系列間のことを指す．この報酬の総和を期待報酬 (Expected Reward) と呼ぶ．ある時間ステップ t 以降に獲得した報酬の系列を $r_{t+1}, r_{t+2}, r_{t+3}, \cdots$ と表すと，期待報酬 R_t は式 (2.3) で表される．

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \cdots + r_T = \sum_{k=1}^{T-t} r_{t+k} \tag{2.3}$$

ここで，T はエピソードの終了ステップである．式 (2.3) は，時間ステップ t から将来獲得できる報酬の単純和で表されるが，エピソードがループするような終端状態のない無限期間時系列においては，最終ステップ $T = \infty$ で期待報酬を最大化する場合，期待報酬が無限の値をとりうる．このため，将来の報酬を割り引いて期待報酬を求める割引期待報酬 (Discounting Expected Reward) という概念も存在する．割引期待報酬 R_t は，式 (2.3) から割引率 γ というパラメータを用いた式 2.4 で表される．

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots + \gamma^{T-1} r_T = \sum_{k=1}^{T-t} \gamma^{k-1} r_{t+k} \tag{2.4}$$

割引率 γ は, 将来の報酬が現在どれだけの価値があるかを表すパラメータであり, $0 \leq \gamma \leq 1$ である．$\gamma = 1$ のとき，式 (2.4) は式 (2.3) と等価となり，将来の報酬を現在の報酬と同等に評価する．また，$\gamma = 0$ の場合は $R_t = r_{t+1}$ となり，エージェントは即時報酬を受け取る行動を選択する．

状態価値関数

状態価値関数とは，環境の状態から実数値への写像であり，その状態の望ましさを表す関数である．望ましさという概念は，その状態から将来の見積もりの期待報酬であると定

義される [28]．すなわち，状態価値関数の値が大きければ，将来受け取れる期待報酬が大きいことを表す．エージェントが将来受け取れる報酬は，エージェント自身が今後どのような行動を行うかに依存するため，状態価値関数は，エージェントの方策に対して定義される．エージェントがある方策 π に従って行動するとき，方策 π のもとでの期待値を E_π とし，環境の状態 $s \in S$ に対する状態価値関数を $V^\pi(s)$ で表すと，$V^\pi(s)$ は式 (2.5) で定義される．ここで E_π は方策 π のもとでの期待値を表す．

$$V^\pi(s) = E_\pi\{R_t|s_t = s\} \simeq E_\pi\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s\} \tag{2.5}$$

また，状態価値関数に対して，エージェントが方策 π のもと状態 s に対して行動 a を行うことの価値を，行動価値関数 (Action-value Function) と呼ぶ．行動価値関数を Q^π で表すと，Q^π は式 (2.6) で定義される．状態価値関数と行動価値関数の概要を図 2.4 に示す．

$$Q^\pi(s, a) = E_\pi\{R_t|s_t = s, a_t = a\} \simeq E_\pi\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s, a_t = a\} \tag{2.6}$$

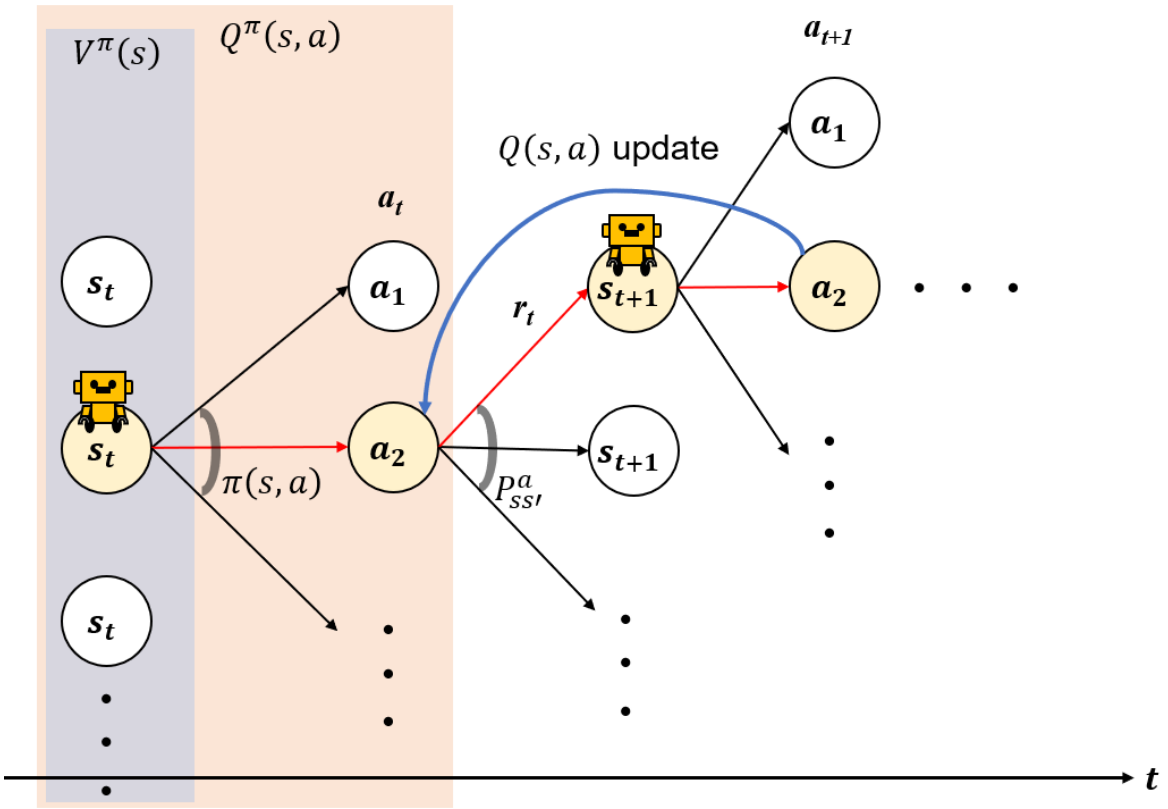

図 2.4: 状態価値関数と行動価値関数の概要

状態 s に対して，2 つの方策 π と π' があるとする．方策 π の状態価値関数 $V^\pi(s)$ がすべての状態において $V^\pi(s) \geq V^{\pi'}(s)$ が成り立つならば，方策 π は方策 π' よりも最適に近い方策となる．このとき，すべての方策の中で最も優れている方策を最適方策 (Optimal Policy) と定義する．最適方策は，最適状態価値関数 (Optimal State-value Function) と呼

ばれる同一の状態価値関数を持っている．最適状態価値関数 V^* は，式 (2.7) で定義される．

$$V^*(s) = \max_{a \in A} V^\pi(s) \tag{2.7}$$

また，最適方策は最適行動価値関数 (Optimal Action-state Function) と呼ばれる同一の行動価値関数も持っている．最適行動価値関数 $Q^*(s,a)$ は式 (2.8) で定義される．

$$Q^*(s,a) = \max_{a \in A} Q^\pi(s,a) \tag{2.8}$$

この式 (2.8) で，$Q^*(s,a)$ は状態 s において行動 a をとった際の報酬と，その後も最適状態価値関数を最大化する行動をとったときに得られる期待報酬との和に等しいため，式 (2.9) のように表すことができる．

$$Q^*(s,a) = r + \gamma \max_{a \in A} Q^\pi(s,a) \tag{2.9}$$

2.1.3. 環境モデル

強化学習では，エージェントが環境のモデルに陽を持つモデルベースの手法とモデルを使用せずに方策を求めるモデルフリーな手法が存在する．環境の状況，すなわち遷移関数と報酬関数がわかっている場合であれば，モデルベースの手法で価値を推定し方策を求めることができる．モデルベースの代表的な手法として，価値反復や方策反復の反復解法で方策を求める動的計画法 (DP 法)，遷移サンプルを用いて方策を求める TD 法，DP 法における状態の価値を後続状態の価値から得られる報酬価値を利用して学習を行うブートストラップ手法に TD 法のサンプリング手法を組み合わせる TD 学習が存在する．しかし，環境が未知である場合では遷移関数と報酬関数が不明であり，モデルを用いた価値反復が困難である．このため，未知な環境に対してはモデルフリーな手法が用いられる．モデルフリーの代表的な手法として，Sarsa や Q-Learning などがある．

2.1.4. 環境同定型強化学習

環境同定型強化学習は，最適な学習結果を得られる手法である．しかし，最適な方策を求めるためには，環境の正確な同定が必要である．このため，経験強化型に比べ多くの試行回数が必要となる．また，環境が動的に変化すると，再び環境を同定しなくてはならないため，ロバスト性が低い．以下にこの環境同定型強化学習の代表的な学習法である Sarsa と Q-Learning について述べる．

Sarsa

Sarsa とは，試行錯誤の経験で式 (2.10) で表されるベルマン方程式を解くアルゴリズムである．なお，式 (2.10) はある方策 π のもとで状態価値関数に関するベルマン方程式である．Sarsa は，時刻 t における状態 S_t で行動 A_t を選択した結果，報酬 R_t と次状態 S_{t+1} を観測した場合，次状態で選択する予定の行動 A_{t+1} をもとに，式 (2.11) によって更新する学習アルゴリズムである．Sarsa は行動選択の方策を用いて価値関数 $Q(S_t, A_t)$ の更新

を行うため，方策オンの手法と呼ばれる．行動選択に応じて価値関数の推定を行うため，学習中の性能が向上する傾向にある．

$$V^{\pi}(s) = \sum_{a \in A(s)} \pi(a|s) \sum_{s' \in S} P(s'|s,a)(r(s,a,s') + \gamma V^{\pi}(s')) \tag{2.10}$$

$$Q(S_t, A_t) \leftarrow (1-\alpha)Q(S_t, A_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})) \tag{2.11}$$

Q-Learning

Q-Learning では，状態 s と行動 a のルールに対して，Q 値と呼ばれるそのルールの有効価値を表す値を持つ．Q 値はスカラー量である．この Q 値を式 (2.9) における $Q^*(s,a)$ の推定値とし，逐次更新する．このように，Q 値を最適状態価値関数に近づける手法である．時刻 t において状態が s_t であり，行動 a_t を行った結果，状態 s_{t+1} に遷移したとすると，Q 値は式 (2.12) のように更新される．

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a_{t+1})] \tag{2.12}$$

式 (2.12) において，γ は割引率，α は学習率である．割引率 γ は，2.1.2 で述べたように $0 \leq \gamma \leq 1$ の実数値をとり，将来の報酬を割り引いて現在に分配する関数である．割引率が低いと即時報酬を受け取りやすい行動を学習する．学習率 α は，状態価値関数の更新度合いを表すパラメータであり，$0 < \alpha \leq 1$ の実数値をとる．学習率の値が大きいと一般に学習速度が速くなるが，状態価値関数の変化が多くなるため，最適な値に収束しない場合がある．一方，学習率の値が小さいと学習速度は遅くなるが，最適な値に収束しやすくなる [29]．また，r_{t+1} は行動の結果受け取った報酬であり，$\max_{a \in A} Q(s_{t+1}, a_{t+1})$ は次状態で選択可能な行動による Q 値の最大値を表す．

また，Q-Learning は，問題領域がマルコフ決定過程 (Markov Decision Processes: MDP) 下において，すべての状態が十分にサンプルされると仮定したとき，Q 値が最適な値に収束することが Watkins らの研究 [21] で証明されている．

Q-Learning は，行動選択に用いる方策と価値関数の更新に用いる方策が異なるため，方策オフの学習と呼ばれる．また，学習に環境モデルを用いないモデルフリーな学習である．

2.1.5. 経験強化型強化学習

経験強化型強化学習は，学習の立ち上がりが早く，学習に要する試行回数が少ない手法である．また，環境が動的に変化した際に影響を受けるのが，変化した部分を含むエピソードだけのため，ロバスト性を有する．しかし，学習が最適な結果に収束する保証がない．以下にこの経験強化型強化学習の代表的な学習法である Profit Sharing について述べる．

Profit Sharing

Profit Sharing(報酬割り当て法) は，遺伝アルゴリズム (Genetic Algorithm: GA) を併用する分類器での信用割り当ての方法として提案された [30]．現在では GA のみでなく，強化学習でも用いられている．Profit Sharing は，学習の立ち上がりが早く，後述する不完全知覚状態である部分観測マルコフ決定過程に対しても有効であることが示されている [31].

Profit Sharing とは，報酬に至るまでのエピソードにおいて，状態 s と行動 a のルール系列を記憶し，報酬 r が得られた際にルールを一括して強化する学習方法である．ルール系列は，式 (2.13) を用いて強化する．

$$w(s_i, a_i) \leftarrow w(s_i, a_i) + f(r, i) \tag{2.13}$$

式 (2.13) における $w(s_i, a_i)$ はエピソード系列上の i 番目のルールの重み，f は強化関数である．強化関数は，報酬獲得時点から i ステップ前の強化値である．

ここで，あるエピソードにおいて，同一状態が二回以上存在し，その状態に対して異なるルールを選択している場合，その間のルール系列を迂回系列と呼ぶ．迂回系列は，図 2.5 のように表される．

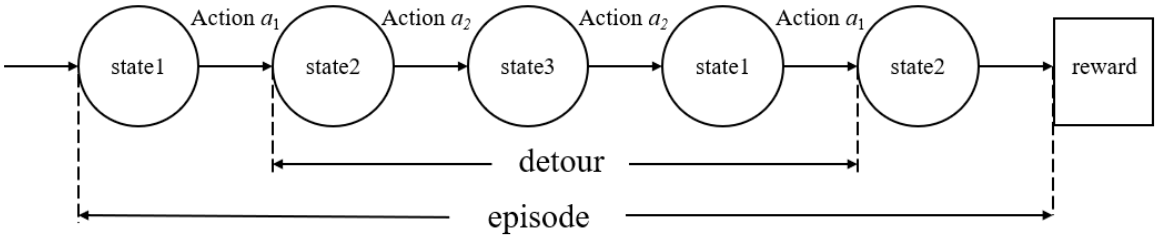

図 2.5: 迂回系列を含むエピソード

現在までのすべてのエピソードで，常に迂回 (detour) 系列上にあるルールを無効ルール，それ以外のルールを有効ルールという．無効ルールと有効ルールが競合するならば，明らかに無効ルールを強化すべきではない．このため，無効ルールの抑制定理 [27] に従い，無効ルールを抑制しながら，有効ルールの強化を行う必要がある．これを満たす条件を式 (2.14) に示す．

$$\forall i = 1, 2, \ldots, W. \quad L \sum_{j=i}^{W} f_j < f_{j-1} \tag{2.14}$$

ここで，W はエピソードの最大長，L は同一状態下に存在する有効ルールの最大数である．これを満たす強化関数として，式 (2.15) に示すような，公比が $\frac{1}{\text{行動の種類}}$ の等比減少関数が考えられる [32].

$$f(r, j) = \frac{1}{S} f(r, j-1), \quad j = 1, 2, \ldots, W-1. \tag{2.15}$$

ここで，S は報酬割引率であり，$S \geq L + 1$ とする．また，式 (2.15) で表される等比減少関数のモデルを図 2.6 に示す．

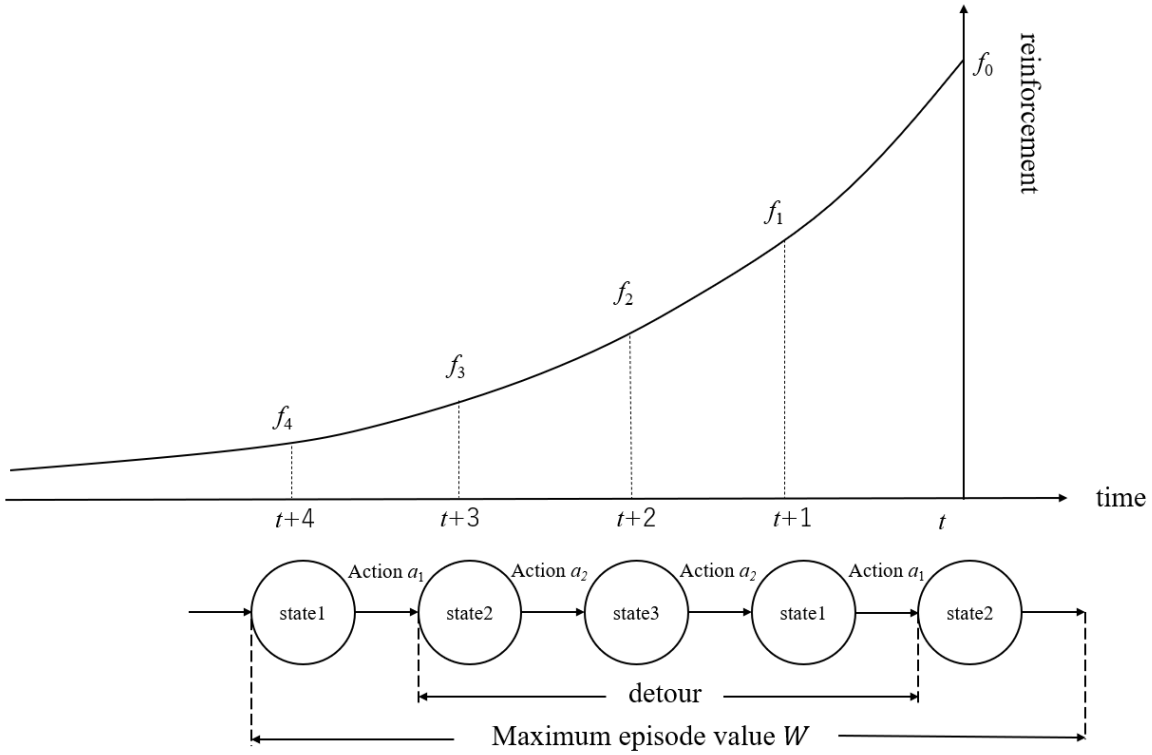

図 2.6: 無効ルールを抑制した強化関数

2.1.6. Actor-Critic 法

Actor-Critic 法は，現在の状態行動価値をもとにより多くの報酬を得るように方策を更新する Actor と，現在の方策に対する状値行動価値を推定する Critic という 2 つの学習器で学習を進める強化学習法である．方策において，非線形関数の関数近似を用いることで，連続行動空間を直接取り扱うことが可能である [33].

2.2. マルチエージェントシステム

マルチエージェントシステムとは，自律した個々のエージェントが多数集まり，相互に依存し合っているシステムのことを指す [28]．単一エージェントシステムとは違い，あるエージェントの意思決定は他のエージェントに影響され，同時に他のエージェントの意思決定に影響を与える．このため，処理がシングルエージェントシステムに比べ複雑化する．しかし，マルチエージェントシステムは適応能力に優れ，並列処理が得意という利点が存在する．また，マルチエージェントシステムは，他技術であるニューラルネットワーク，進化的計算などに用いることが可能である [34]．本節ではマルチエージェントシステムにおけるエージェントの定義とマルチエージェントシステムの利点について述べる．

2.2.1. エージェントの定義

エージェントという用語は，様々な分野で用いられているが，機械学習の分野におけるエージェントは，自身の置かれた環境を知覚し，環境に基づいて行動を行うことによって，環境に対して影響を与えることのできる自律的主体である [28, 35]．このとき，エージェントは自律性を保持しているため，外部からからの指示ではなく，自らの判断で問題処理を行う．エージェントの特徴としては以下の 5 つが挙げられる [36].

- 自律性: 外部から直接的な影響を受けず，エージェント自身の判断で問題処理を行う
- 社会性: 他のエージェントと相互作用を行う
- 反応性: エージェント自身が置かれた環境を認識し，環境変化に対応する
- 自発性: 問題処理のために能動的に行動を行う
- 知性: 問題処理や学習を行うための仕組みや知識を持つ

また，ネットワーク空間のエージェントにおいては，ネットワーク上のサーバ間を移動する行動性も保持している．

本研究におけるエージェントとは，自身の置かれた周囲環境を認識し，問題解決または目標達成のために能動的に行動をする処理体と定義する．

2.2.2. マルチエージェントシステムの利点と欠点

マルチエージェントシステムに期待される利点は，システムの適応能力の向上，エージェント同士の相互作用による並列処理に伴う効率化などが挙げられる [37–39].

適応能力に関しては，各エージェントが自律して行動し学習しているため，目標の変更や問題規模の拡大と縮小に対応することができる．また，あるエージェントが故障した際にも，再び環境を学習することで，最適な行動を学習することができる．並列性に関しては，各エージェントが自律しているため，エージェント同士は並列的かつ非同期的に動作する．このため，システム全体の処理効率の向上を見込むことができる．しかし，エージェント数を大幅に増加させると，次元の呪いとエージェント間の相互作用の指数関数的な増大により学習が困難となるため，エージェントの総数は適切な数である必要がある [36]．ここで次元の呪いとは，全知覚のような制限ない状態での最適解を得るためには，知覚範囲に依存して指数関数的に多くなる組み合わせの中から解を見つけなければならなくなるという指数依存性である [40].

また，協調性に関して荷物搬送問題を例に挙げると，あるエージェントの意思決定は他のエージェントに影響され，同時に他のエージェントの意思決定に影響を与える．この結果，各々のエージェントが互いに避け合いながら最適な行動を選択し，協調性を獲得することができる [41].

2.2.3. 協調行動

マルチエージェントシステムでは，対象とするタスクに対して各エージェントが統制のとれた振る舞いを行う必要がある [42]．この統制がとれた振る舞いは協調行動と呼ばれ，人間社会や自然界でも観測される [43–45]．マルチエージェントシステムの問題処理では，各エージェントの協調性が重要であり，協調行動を行えない場合に効率低下やデッドロックの発生などの問題が起こりえる．しかし，協調行動のために必要な条件は，エージェントの置かれた環境や能力，タスクの質と量に応じて変化するため，あらかじめ想定することは困難である．

2.2.4. 協調行動における利他的行動

マルチエージェントシステムのタスクを達成するためには，各エージェントが最適な行動を行う必要があるが，各エージェント間の相互作用やタスク環境の状態によりシングルエージェントと比較してかえって効率が低下する場合がある．これは，マルチエージェントであっても各エージェントの報酬を最大化させる利己的行動での学習を行い，全体の利益を最大化する学習を行っていないためである．一方人間社会での行動では，ボランティアなど個人の利益とならない利他的な行動が全体としての利益となる場合がある [46]．このため，マルチエージェントシステムではあるエージェントが他のエージェントの利益となる利他的行動をとる場合が全体の利益が最大化することがある．このため，全体としての協調行動が他のエージェントの利他的行動となることがある．

2.2.5. 利他的行動の一例

本研究で現れる利他的行動の一例を図 2.7 に示す．各エージェントの行動目的は荷物の搬送であり，荷物を搬入口から受け取り，それを搬出口まで搬送するタスクである．このタスクでは，すべてのエージェントが荷物搬入口や搬出口，赤く囲った混雑しやすいエリアに集まってしまうと，相互進路妨害を起こし学習がストップしてしまうと考えられる．このため，図 2.7 のような環境では，エージェント 1 やエージェント 2 のように他のエージェントのタスク終了を待つような利他的行動が協調行動となる．

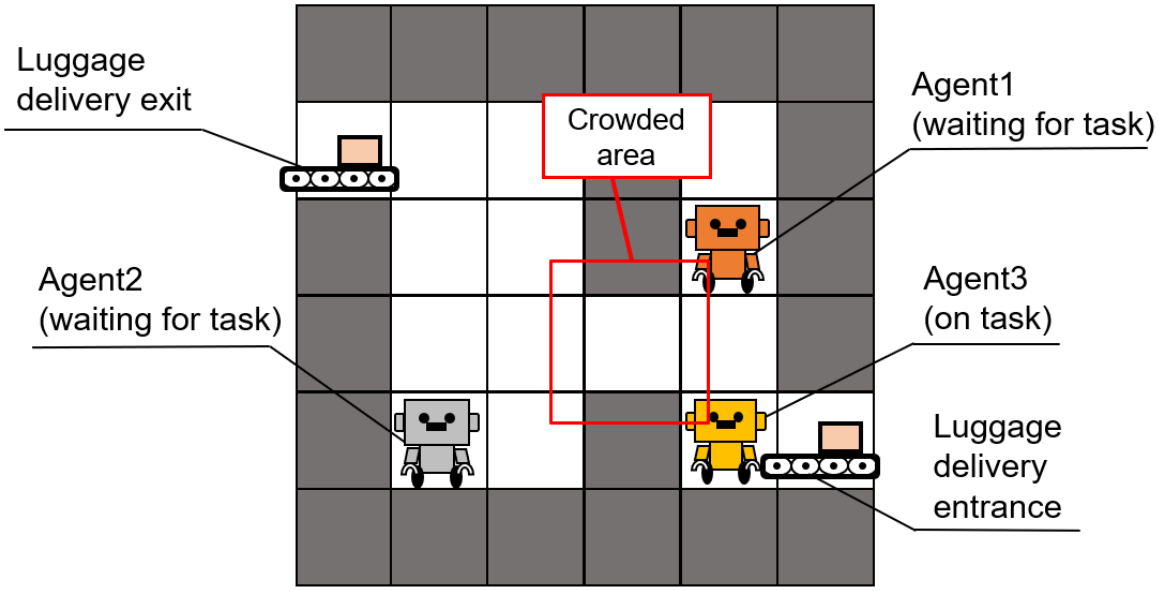

図 2.7: 協調行動の一例

2.3. 部分観測マルコフ過程

強化学習はこれまで，環境がマルコフ決定過程としてモデル化できる仮定のもと，研究が行われてきた [47, 48]．マルコフ決定過程は，状態遷移がマルコフ性を満たすものである．マルコフ性とは，過程の将来状態の条件付き確率分布が，現在状態のみに依存し，過去のいかなる状態にも依存しない特性を持つことである．

しかし，現実世界は非マルコフ環境であり，強化学習が対象としている未知の環境も非マルコフ環境であることが多い．また，エージェント自身の知覚が不完全知覚，すなわち環境全体を一度に観測できない場合もマルコフ性が保証されていない．この場合，すべての状態を観測するマルコフ決定過程では，膨大な数を観測しなければならないため，学習には適さない．このため，部分観測マルコフ決定過程 (Partially Observable Markov Decision Process: POMDP) での学習が不可欠である [49]．本節では，マルコフ決定過程と部分観測マルコフ決定過程について述べる．

2.3.1. マルコフ決定過程 (MDP)

(S, A, P, R) で定義される環境をマルコフ決定過程 (Markov Decision Processes: MDP) という．特に強化学習において，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最も基本となるものがマルコフ決定過程である．マルコフ決定過程ではエージェントが状態 $s \in S$ において行動 $a \in A$ を実行したとき，確率 $P^a_{ss'} = P\{s_{t+1} = s' | s_t = s, a_t = a\}$ に従って状態 s' へ遷移する．この遷移する確率を状態遷移確率という．遷移確率が状態 s と行動 a にのみ依存する場合に，この状態遷移はマルコフ性を持つ．このとき，エージェントの状態 s における行動 a に対して確率的に与えられる報酬を即時報酬といい，その期待報酬 R は，現在の状態と行動にのみ依存する．

2.3.2. 部分観測マルコフ決定過程 (POMDP)

環境がマルコフ決定過程であっても，エージェントが不完全知覚である場合，状態遷移確率 $P^a_{ss'}$ が特定できない．このような環境を，部分観測マルコフ決定過程 (Partially Observable Markov Decision Process: POMDP) という [50].

部分観測マルコフ決定過程は, (S, A, P, R, Ω, O) で定義される．T は $T(s, a, s') = P(s'|s, a)$ で与えられる状態遷移確率を記述する関数であり，$R(s, a)$ はエージェントに与えられる報酬の期待値を記述する関数である．Ω はエージェントの観測を要素にもつ有限の集合であり，$O(s', a, o) = P(o|a, s')$ はエージェントの観測を記述する関数である．POMDP 環境では，エージェントは観測による行動選択，状態遷移，報酬受け取りを繰り返すことにより，各状態において将来的に得られる報酬の和が最大になるような方策を目指す．

POMDP の学習において観測が不完全であり，最適行動が状態 s_1, s_2 で異なる場合であっても，これらを同一とみなすときに生じる問題を，不完全知覚問題という．不完全知覚問題では，エージェントが観測を行う場合には同一の観測状態であるが，環境全体を観測すると異なる状態となるため，報酬の振り方に異常が生じるエイリアス問題などがある [22,51]．エイリアス問題の例を図 2.8 に示す．エージェントは，初期状態 S から上下左右に 1 マスずつ移動が可能であり，ゴール状態 G に到達することで，報酬を得る．エージェントは不完全知覚であり，上下左右の周囲 1 マスしか観測できないとする．この部分観測でのエージェントは，状態 1a と状態 1b を同じ状態と認識してしまう．状態 1a では左方向，状態 1b では右方向の行動が求められるが，エージェントからは同じ状態で観測しているため，報酬の割り振りと行動価値関数に異常が生じ，学習が混乱する．

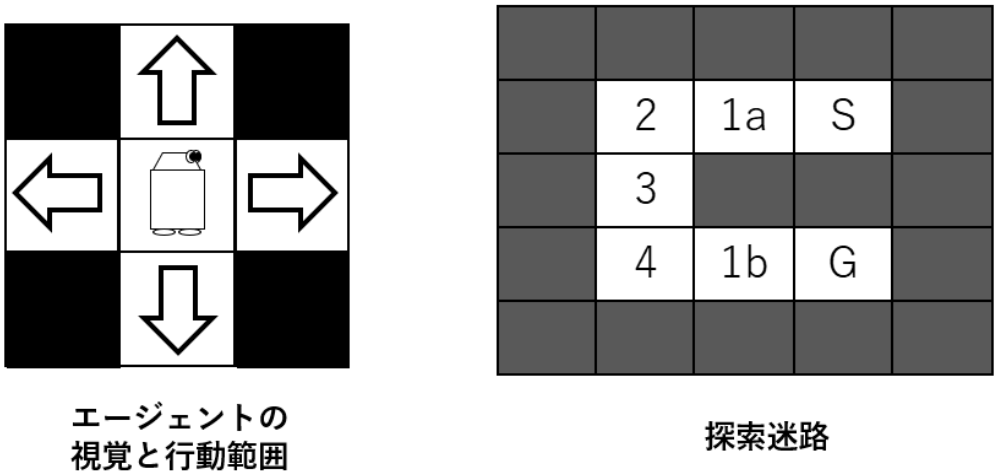

図 2.8: エイリアス問題の一例

2.3.3. 信念状態 (Belief state)

POMDP の学習では，信念状態 (Belief state) という状態が用いられる．信念状態とは，観測できる環境のどの状態にいるかを表す確率を並べてつくる状態である．信念状態では，状態空間上に定義された確率分布として，$b(s) = P(s|h)$ と表せる．ここで，h はその時点までの観測や行動の履歴である．すなわち，信念状態とは $\forall_s \in S$ に対して $b(s) \in [0, 1]$ であり，$\Sigma_{s \in S} b(s) = 1$ を満たし，$(|S| - 1)$ 次元単体上の 1 点としてあらわすことができる．

第3章　荷物搬送問題における環境別の学習変化

3.1.　荷物搬送問題の概要

マルチエージェントシステムの代表的な問題として知られる荷物搬送問題は，エージェントが荷物を荷物搬入口から受け取り，荷物を荷物搬出口へと運搬することをタスクとする問題である [52]．この問題は，図 3.1 に示すように倉庫を模した格子空間内に荷物を受け取る搬入口，荷物を運び出す搬出口が存在する．各エージェントは，他のエージェントと相互作用を及ぼし合いながら経路探索を行い，学習を行う．荷物搬送問題では，エージェントは荷物搬出を行えた場合に報酬が与えられ，より高い報酬を学習することを目標としている．荷物搬送問題における高い報酬とは，割引率によって割り引かれる報酬が少ないルール系列であるため，荷物搬送のより短い経路といえる．本研究では，荷物搬送問題に対し強化学習を用いて実験を行う．

荷物搬送問題は荷物搬送を行えた場合に報酬が与えられるため，荷物搬出量が報酬総和と等しくなる．すなわち，強化学習の性能としての報酬評価が荷物搬出量となる．しかし，エージェントの数や学習手法，エージェントの視界などの実験条件により学習結果が変化するため，すべての環境で正しく学習を行えるようなエージェントの設計は困難である．

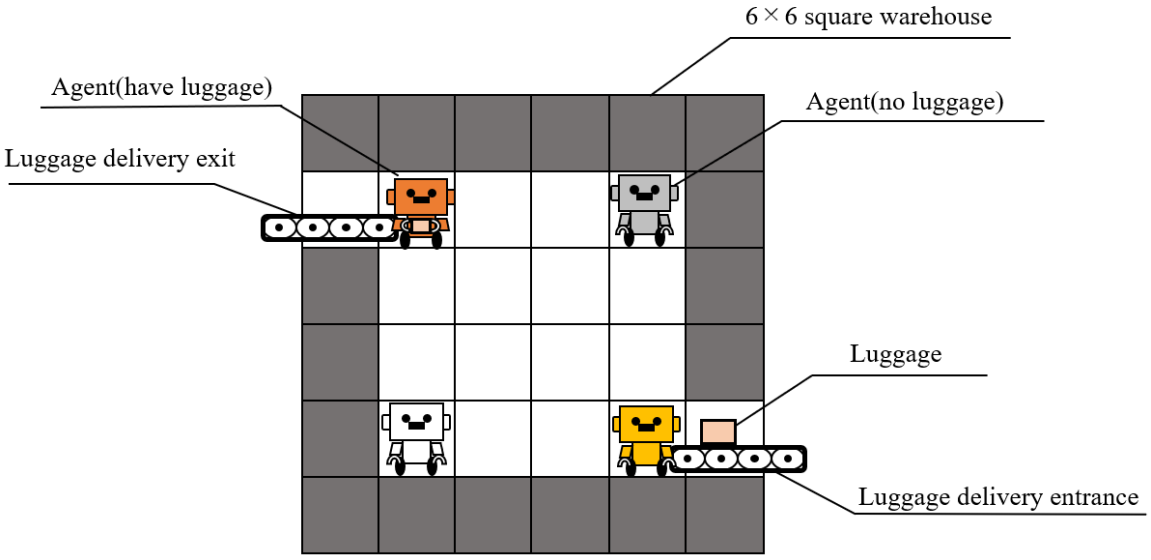

図 3.1: 荷物搬送問題の概要図 (実験環境 1)

3.2. 荷物搬送問題における環境別の評価実験

マルチエージェントシステムを用いた荷物搬送問題では，エージェントの数や環境に依存して学習状況が変化する．学習状況が変化すると学習進度が変化するため，それまで学習が行えていた環境に対して異なる行動を行う場合がある．特に，エージェントの数が多いほど荷物搬送効率は向上するが，相互進路妨害が発生する場合がある．この相互進路妨害は，エージェントの数や環境だけでなく，行動選択や視界範囲にも依存するため，環境の特性を調査する必要がある．このため，本章では荷物搬送問題における進路妨害の発生するエージェントの数と環境の調査およびエージェントの行動特性の検証を目的とする．

行動特性に影響を与える行動選択は，候補行動集合として上下左右などのマス目間の遷移行動 [18,23]，自転角度を用いた進行がある [53,54]．本研究では上下左右の遷移に加え，停止行動を導入することによる行動特性の検証を行う．停止行動を導入することにより，待機行動の学習や狭い環境に対する協調行動獲得に期待できる．エージェントの視界範囲は，2.2.2 で述べたように，エージェントの数と視界範囲が増加することで学習の組み合わせが爆発的に増加するため，視界範囲を環境に対して部分観測となる範囲で学習を行う. また，第 2 章で述べた行動選択手法およびそれらを組み合わせた手法をそれぞれ用いることで，行動選択手法が学習過程に与える影響について検証する．

3.2.1. 組み合わせ方策

行動選択手法はそれぞれに特徴があり，また環境に依存するため，どの手法が最も優れているかは決めることができない．ϵ-greedy 方策は確率 ϵ が 1.0 に近い場合最短経路を見つけやすいが，確率 ϵ でランダムな行動を選択するため，エージェントの行動は収束しない．しかし，確率 ϵ が小さくなるにつれて最も価値の高い行動が貪欲に選択される．そのため，同一行動を選択し続け，探索行動を選択しなくなる．Softmax 方策は，エージェントの行動は収束するが，局所解に陥りやすい．両者を比較した実験では，ϵ-greedy 方策が平均ステップ数が多いが収束ステップ数が少なく，Softmax 方策が平均ステップ数が少ない反面収束ステップ数が多い [55]．本論文では Q-PSPLearning [56] 学習下の ϵ-greedy 方策と Softmax 方策における搬送効率を検証するとともに，2 種類の手法の利点を組み合わせるため，ϵ-greedy 方策と Softmax 方策を組み合わせた手法を提案する．本章では，この行動選択手法を以下 “組み合わせ方策” と記す．

組み合わせ方策は，局所解に陥りづらく収束する行動選択手法を設計するため，始めに ϵ-greedy 方策のランダム行動を用いて広く探索を行うと同時に，環境の同定を行う．その後，Softmax 方策へ切り替えて学習を行う．本手法は，以上の手順を踏むことで Softmax 方策よりも広く探索した上で協調行動を獲得し，最適経路を見つけやすくすることを目標とする．しかし，ϵ-greedy 方策から Softmax 方策への切り替えは任意のタイミングであるため，環境との相互作用とエージェントの学習速度から適切に設定しなければならない．

3.2.2. 実験手法

マルチエージェントシステムを用いた強化学習には，経験強化型強化学習である Profit Sharing が用いられてきた [31,57]．しかし，マルチエージェントシステムではすべての

エージェントが最適協調行動を選択しなければ最適解に収束しないため，効率の低下が課題であった．一方で，マルチエージェントシステムの処理は単一エージェントシステムと比較して環境が複雑化しているため，環境同定型の Q-Learning では学習の速度が極端に遅くなる．特に，環境をすべて観測しながら学習を行う全知覚での学習では，最適解に収束しやすいが環境同定型ではすべての状態を観測するのに膨大な時間がかかるため，学習の高速化が課題であった．この課題は，部分観測の導入によってある程度の改善が見込めるが，2.3.2 で述べたエイリアス問題などが生じる場合がある．これら課題解決の一手法として，堀内ら [56] によって経験強化と環境同定を組み合わせた Q-PSPLearning が提案された．Q-PSPLearning は，Q-Learning の報酬獲得時における報酬分配として Profit Sharing Plan の概念を導入したものである．Q-PSPLearning では，Q-Learning と同様に各ルールが Q 値を持っており，各ステップにおいて実行したルール系列をエピソードとして記録する．そして，報酬が得られた際に一括して過去に実行されたルールの Q 値を更新する手法である．このため，Q-PSPLearning は Q-Learning よりも学習速度が速いとされている [56]．Q-PSPLearning の行動と報酬の概要を，図 3.2 に示す．

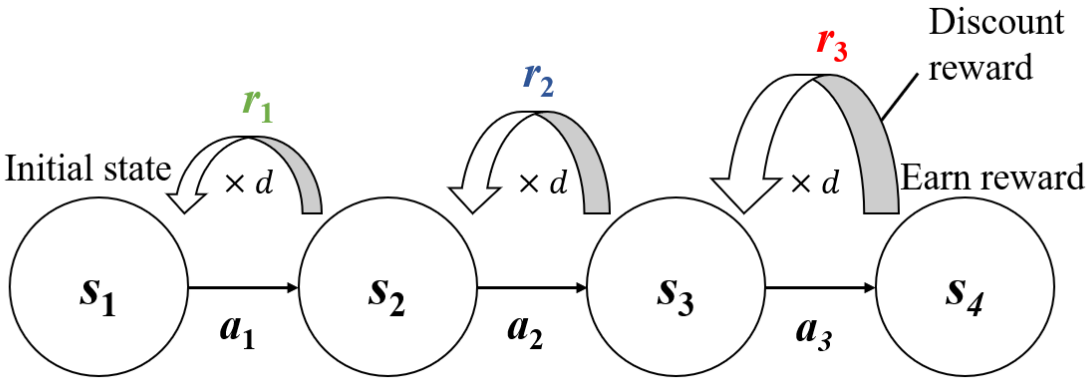

図 3.2: Q-PSPLearning の行動と報酬の概要

堀内らの Q-PSPLearning では，時刻 t において，状態 s_t のもと行動 a_t を実行した結果，状態が s_{t+1} に遷移し，報酬 r が得られたとすると，Q 値は式 (3.1) により更新される．

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[r + \gamma \max_{a \in A} Q(s_{t+1}, a_{t+1})] \tag{3.1}$$

また，実行したルール系列をエピソードとして記録し，0 でない報酬が得られたとき，一括して過去に実行されたルール R_i の Q 値は式 (3.2) に従って更新される．

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha f_i(r) \tag{3.2}$$

ただし，$f_i(r)$ は強化関数であり，エピソードの最後から数えて i ステップ前のルールに分配する報酬の大きさを決める関数である．この強化関数 $f_i(r)$ は，2.1.5 項に示した有効性を保証する条件を考慮し，$f_i(r) = rd^i$ とする．このとき，d は公比であり，$0 < d < 1$ である．この公比は，値が大きい場合には無効ルールも強化し最適な収束をとりにくくなるが，値が小さい場合はルールに分配される報酬が小さいため，収束までの試行回数が多く

なる．また，公比はルールと環境に依存する．ここでの環境とは，エージェントが環境から観測する状態のことを指す．

このQ-PSPLearningの式 (3.1, 3.2) をもとに，本実験では，式 (3.3) を用いて実験を行う．

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a \in A} Q(s_{t+1}, a_{t+1})] \tag{3.3}$$

ここで，r_t は各ステップにおける報酬の値であり，$r_t = rd^i$ である．また，式 (3.3) から得られるエージェントの最大 Q 値は，最大 Q 値を Q' と置くことで以下のように求められる．

$$Q'(s_t, a_t) = (1-\alpha)Q'(s_t, a_t) + \alpha[r_t + \gamma \max_{a \in A} Q'(s_{t+1}, a_{t+1})] \tag{3.4}$$

エージェントが最大 Q 値を得られるまでの学習は試行回数が膨大になる．このとき，$Q'(s_t, a_t)$ は一定の値に収束する．このため，$Q' \simeq Q'(s_t, a_t) \simeq Q'(s_{t+1}, a_{t+1})$ と置くことができる．よってここで，Q' について整理すると，Q' は行動及び状態がルール系列の終端であるため

$$Q' = (1-\alpha)Q' + \alpha(r + \gamma Q') \tag{3.5}$$

と置くことができる．これを整理して，

$$Q' = \frac{r}{1-\gamma} \tag{3.6}$$

と導出できる．このため，エージェントの最大 Q 値は報酬 r と割引率 γ に依存する．

また，ルール条件では，各エージェントの視界環境のみをもとに重み付き候補行動集合を生成し，学習を行う．このルール条件を以下 “環境ルール” と示す．ここで，環境ルールの *if-then* ルール条件式において，エージェントが荷物を持っていない場合の条件を式 (3.7)，エージェントが荷物を持っている場合の条件を式 (3.8) に示す．

$$if \quad \text{環境 and 荷物なし} \quad \text{then} \quad \text{候補行動集合} \tag{3.7}$$

$$if \quad \text{環境 and 荷物あり} \quad \text{then} \quad \text{候補行動集合} \tag{3.8}$$

実験環境においては，エージェントが同知覚による学習の停滞のほか，環境の広さに起因する行動の制約などにより学習が正しく行えない場合がある．前者の条件は，学習前にあらかじめ確認することが可能なことがあるが，後者は実際に学習を行わないと確認することが難しい [36]．特に，人が可能と考える環境設計であっても，実際は学習過程などに問題が発生し，正しく学習が収束しないことがある．このため，本研究では複数の環境で実験を行い，エージェント内部の Q 値などから環境別の学習変化の検証を目的とする．実験を行う環境は，以下に示す 5 通りの実験環境である．

- 実験環境 1: 図 3.1 に示すような，6×6 マスの障害物のない環境
- 実験環境 2: 図 3.3 に示すような，実験環境 1 を通過しづらくした環境
- 実験環境 3: 図 3.4 に示すような，実験環境 2 の上部一マスを通過可能にした環境

- 実験環境 4: 図 3.5 に示すように，実験環境 1 の荷物搬入口と搬出口を狭くした環境
- 実験環境 5: 図 3.6 に示すように，実験環境 4 の荷物搬入口と搬出口の周辺を 1 マス広くした環境

各実験環境において，最短で荷物搬送を行えるステップ数は，実験環境 1 から実験環境 3 が 12 ステップ，実験環境 4 と実験環境 5 が 16 ステップである．また，これらの各実験環境を用いる目的を以下に示す．

- 実験環境 1: 障害物のない正方形の環境下でのエージェントの振る舞い検証
- 実験環境 2: 実験環境 1 に対して，荷物を受け取り搬送するまでのタスク間に 1 体のエージェントのみが通過できる広さの障害物を追加した場合のエージェントの振る舞い検証
- 実験環境 3: 実験環境 2 に対して，各エージェントが上部分と下部分で同時に通過できるようにした際のエージェントの振る舞い検証
- 実験環境 4: 実験環境 1 に対して，荷物搬入口と搬出口を狭くした際のエージェントの振る舞い検証
- 実験環境 5: 実験環境 4 に対して，タスク周辺環境を 1 マス広げた際のエージェントの振る舞い検証

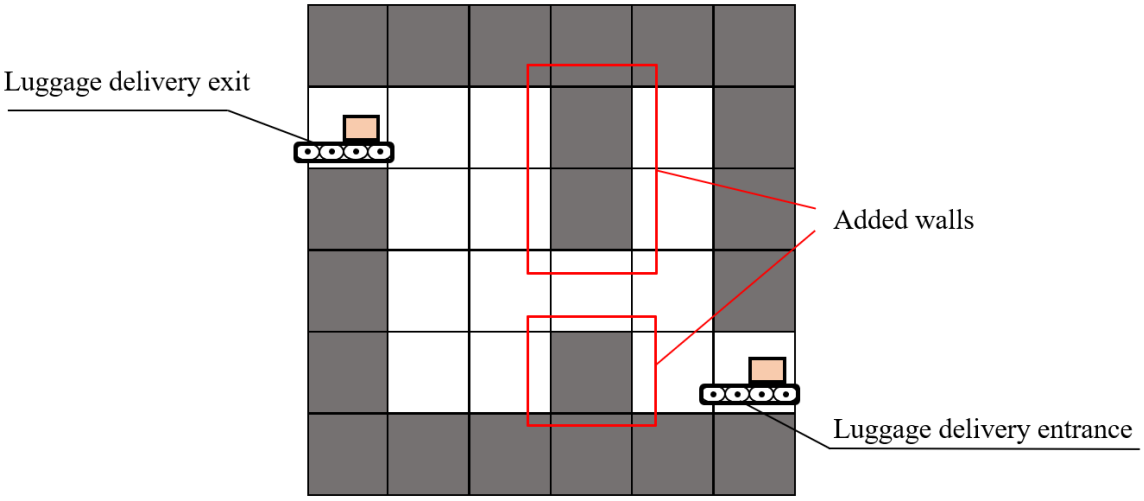

図 3.3: 実験環境 2

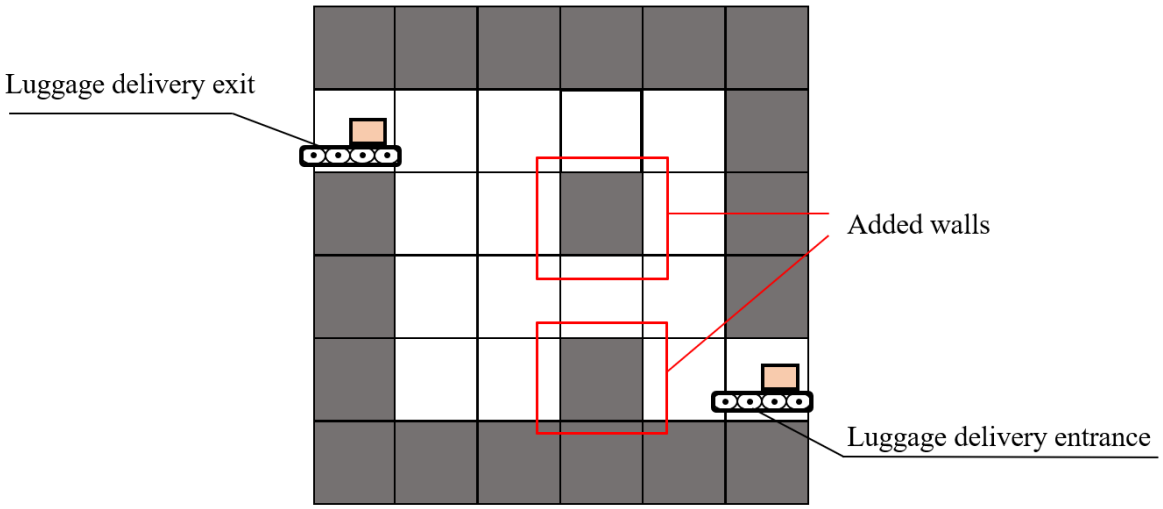

図 3.4: 実験環境 3

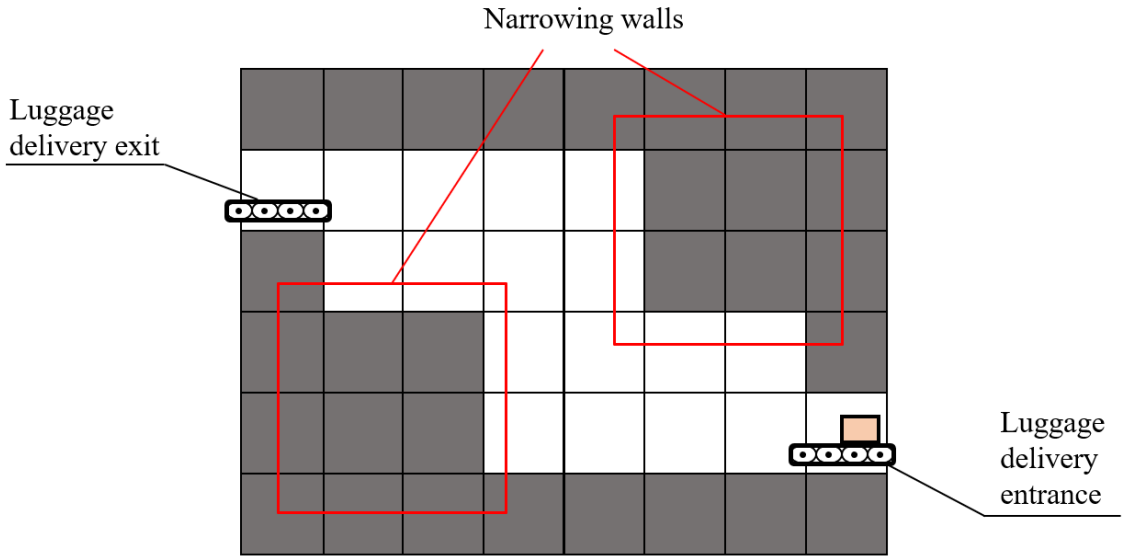

図 3.5: 実験環境 4

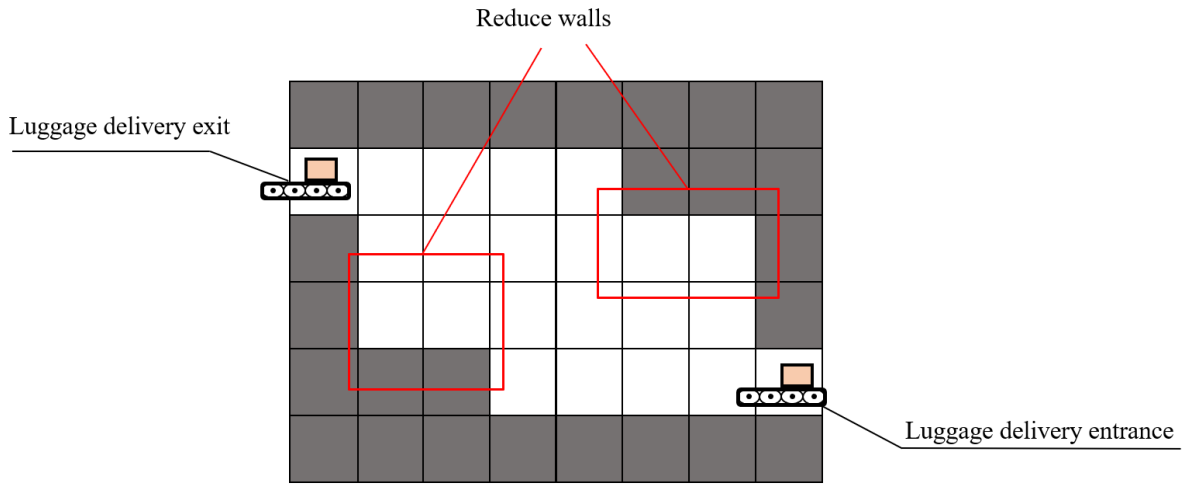

図 3.6: 実験環境 5

3.2.3. 実験内容

従来研究 [3, 19, 53, 58] では，各エージェント間での通信や各エージェントの状態を観測するエージェントなどを用いることで，迷路問題における探索の効率化を図っている．しかしこの手法では，災害現場や未知の環境におけるタスクにおいて，通信環境の構築や全環境の観測に問題がある．このため，本研究ではエージェント間での通信を行わない条件下を想定した実験条件を設定した．また，各エージェントの観測できる視界範囲を環境より小さいものとし，部分観測環境下での学習を行う．また，Q 値の更新式に Q-PSPLearning による式 (3.3) を用いる．ルールの条件はエージェントの視界環境と候補行動集合であり，各エージェントはルールを共有しない．

本実験の環境条件として，図 3.1 に示すような倉庫を模した格子空間の実験環境 1 に加え，図 3.3, 図 3.4，図 3.5，図 3.6 に示す 5 種類の環境で行う．各実験環境マスの外周は壁である．各エージェントは，他のエージェントと壁に衝突しない行動選択を行う．また，エージェントは荷物搬入口から荷物を受け取り，荷物搬出口へ搬送することを目標とし，荷物の搬送が完了した際に報酬を与える．また，エージェントは図 3.7 に示すような周囲 1 マスの視界を持っており，上下左右と停止の 5 種類の行動を選択できる．各エージェントの初期位置は，図 3.1 に示すような，各環境の四隅に配置した．すべてのエージェントが行動を選択し，環境が遷移することを 1 ステップとする．また，荷物の搬送効率は 1000 ステップ試行ごとに対する荷物排出量で判別する．実験環境 1 から実験環境 3 のタスク完了までの最短ステップ数が 12 ステップであるため，1 体あたりの荷物搬出量は理論値の最大は 83.3 回となる．また，実験環境 4 から実験環境 5 の最短ステップ数は 16 であるため，1 体あたりの荷物搬出量は理論値の最大は 62.5 回となる．

実験を行うエージェントの数は，最小を 2 体，最大を 4 体とし，実験中に数を増減させない．また，各エージェントは実験中に故障や初期化されずに学習を行う．また，探索を行わせるために，各エージェントは荷物搬出口と搬入口を知覚不可とし，各エージェントの個体識別を可能とした．個体識別及びルール系列は，エージェントが荷物を持っている場合および持っていない場合で区別して学習を行った．

また，Q 値の更新に用いる式 (3.3) における温度定数 T は $T = 0.52$，報酬 $r_t = rd^i$ における公比 d は式 (2.15) から行動の種類が 5 種類であるため，$d = \frac{1}{5} = 0.20$ で行った．学習率 α と割引率 γ は実験条件がマルチエージェントであり，観測できる環境数の総和が膨大であるため $\alpha = 0.06$，$\gamma = 0.95$ とした．このことから，エージェントが学習に用いる Q 値において，最大 Q 値は式 (3.6) から $\frac{1}{1-0.95} = 20.0$ となる．組み合わせ方策における ϵ-greedy 方策のランダム行動確率 ϵ は 0.1 とし，方策切り替えのタイミングは 50,000 試行ステップ後とした．また，円滑な荷物搬送の学習を行うため，報酬を受け取るまでの連続ステップの制限を 2,000 ステップとした．2,000 ステップ環境が遷移した場合に，1 度も荷物の搬送を行えなかったエージェントは初期位置に戻り，ルールおよび Q 値をリセットせず再び学習を行う．

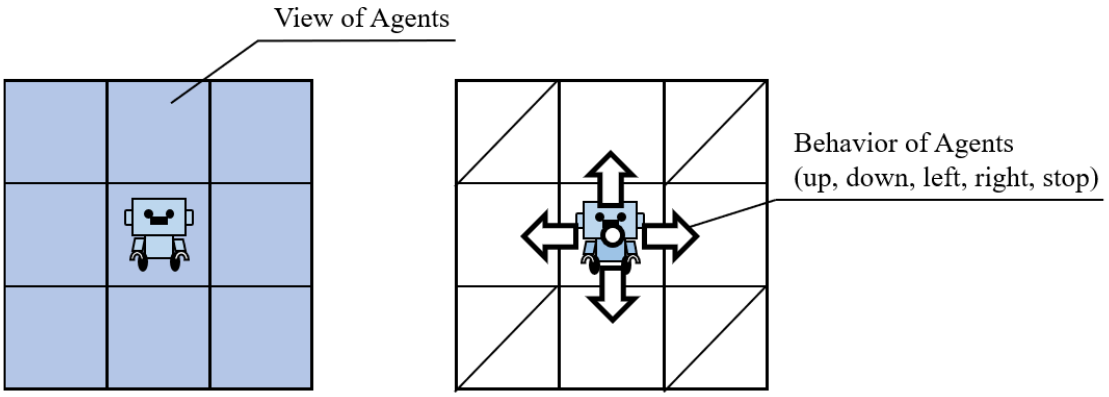

図 3.7: エージェントの視界と行動

表 3.1: Softmax 方策と組み合わせ方策を用いる実験の設定パラメータ

Number of agents	2 to 4
Initial Q value	0.00
Learning rate α	0.06
Discount rate γ	0.95
Planning discount rate d	0.20
Temperature value T	0.52
Reward r	1.0
Total steps	500,000
Limit continuous steps	2000
Number of switching trials	50,000

3.3. 実験結果

以下にエージェント数が 2 体から 4 体の場合における各実験環境の結果を示す．実験結果は，その環境に対して用いた各手法が最も多く発現するパターンを示すものであり，す

べての実験で同様の結果が得られるものではない．本研究では，最も多く発現するパターンからエージェントの学習と振る舞いについて推察する．

3.3.1. エージェントが 2 体での各実験環境の結果

エージェント数が 2 体での実験環境 1 から実験環境 5 において，Softmax 方策での荷物排出量の学習遷移を図 3.8 に示す．図 3.8 において，グラフ縦軸が 1000 ステップごとの荷物搬出量であり，グラフ横軸が試行ステップ数である．実験環境 1 から実験環境 3 の荷物搬出量の理論値は 166.6 回，実験環境 4 から実験環境 5 の荷物搬出量の理論値は 125.0 回となる．

各実験環境において，すべての結果が安定的に荷物搬送が行えているため，学習が正しく行えている．また，試行回数 490,000 回から 500,000 回の 10,000 回間における 1,000 ステップごとの荷物排出量平均と荷物搬出量の標準偏差の結果を表 3.2 に示す．表 3.2 では 10,000 回間の標準偏差がすべての環境で 1.0 以下であり，学習が安定している．各環境の荷物搬出量は，実験環境 2 以外が理論値と等しい値となったため，2 体エージェントでの最適行動を学習できた．実験環境 2 では，1 体のエージェントが荷物を受け取る際，もう 1 体のエージェントが数ステップの間停止行動を選択し，タスク完了を待つため，理論値よりも低い値となった．

学習によるエージェントの振る舞いは，実験環境 1 では図 3.9 に示すように各エージェントが外周に沿って最短経路の周回行動を規則的にとった．実験環境 2 では，1 体のエージェントがもう 1 体のエージェントに追従するような形で荷物搬送を行い，学習が安定的に行えた．実験環境 3 では，実験環境 1 と同様に図 3.9 で示すような最短経路で周回行動を規則的にとった．実験環境 4 と実験環境 5 も同様に，最短経路で周回行動を規則的に行う結果となった．

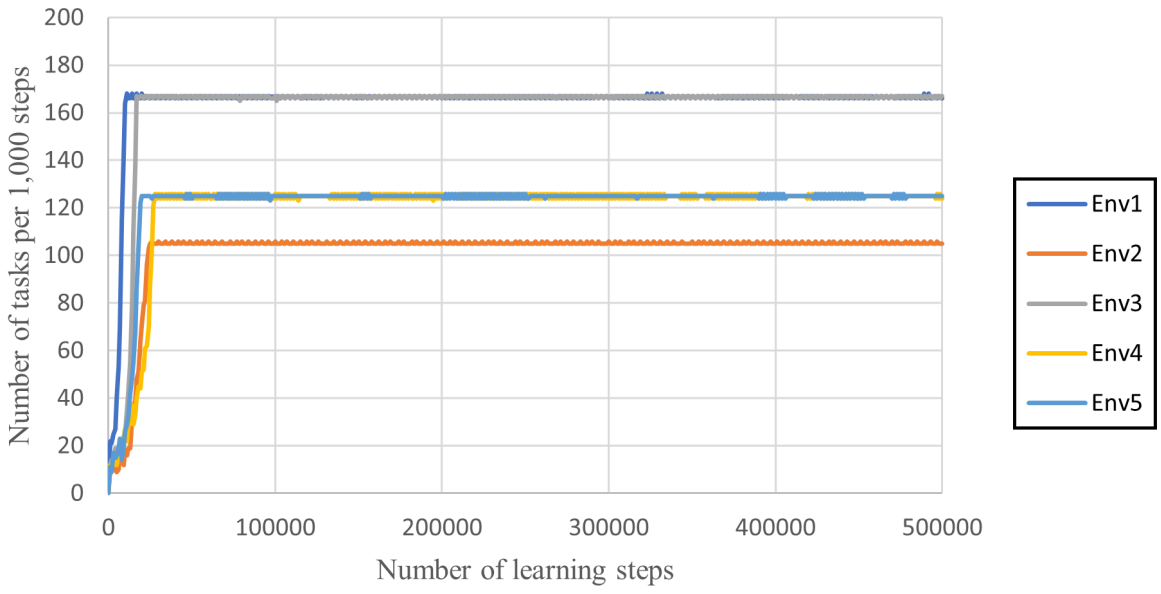

図 3.8: エージェント数が 2 体での Softmax 方策を用いた各実験環境の結果

表 3.2: エージェントが 2 体での各環境別荷物搬出量の平均値と標準偏差

Used policy(Environment)	Theoretical value per 1,000 steps	Average tasks per 1,000 steps	standard deviation
Softmax policy(Env.1)	166.6	166.6	0.699
Softmax policy(Env.2)	166.6	105.2	0.42
Softmax policy(Env.3)	166.6	166.6	0.52
Softmax policy(Env.4)	125.0	124.9	0.74
Softmax policy(Env.5)	125.0	125.0	0.0

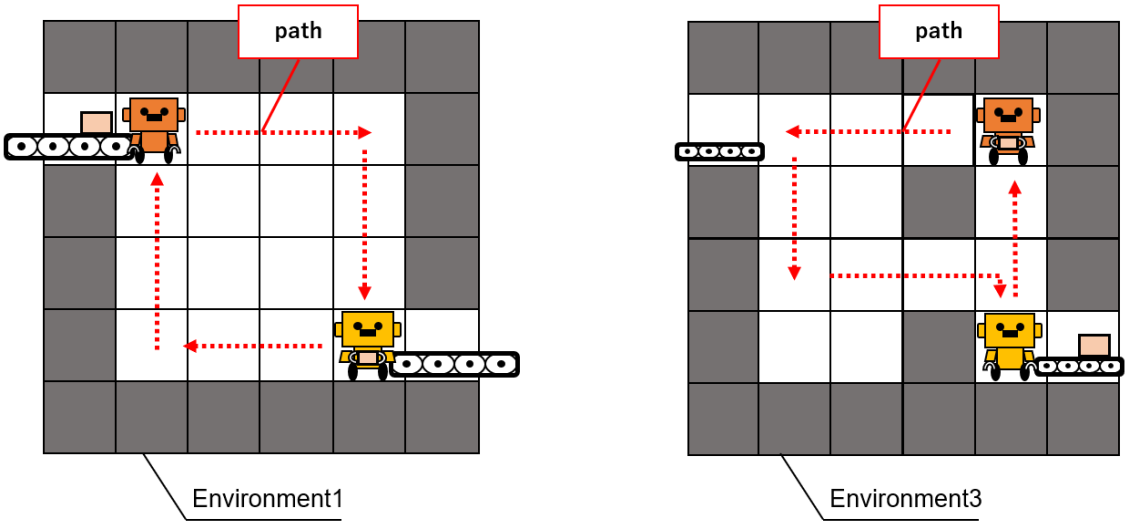

図 3.9: 実験環境 1 および実験環境 3 のエージェントの振る舞い

エージェント数が 2 体での実験環境 1 から実験環境 5 において，組み合わせ方策での荷物排出量の学習遷移を図 3.10 に示す．組み合わせ方策を用いる場合，すべての実験環境において結果は Softmax 方策より安定せず，荷物排出量も理論値より低いものとなった．エージェントの振る舞いに関しては，すべての結果においてエージェントの行動選択は規則的に行われず，荷物を搬送するたび違った行動を選択した．また，荷物排出量平均と荷物搬出量の標準偏差を表 3.3 に示す．荷物排出量はすべての結果が Softmax 方策と比較して減少しており，標準偏差は 2.14 から 3.50 の範囲であり，1000 ステップごとの荷物搬出量にはおよそ 2 4 回程度の差分がある．このため，組み合わせ方策は Softmax 方策と比較して安定的でなく，荷物搬送効率が減少する結果となった．

表 3.3: 組み合わせ方策を用いたエージェントが 2 体での各環境別荷物搬出量の平均値と標準偏差

Used policy(Environment)	Theoretical value per 1,000 steps	Average tasks per 1,000 steps	standard deviation
Integration policy(Env.1)	166.6	107.2	2.97
Integration policy(Env.2)	166.6	93.2	2.14
Integration policy(Env.3)	166.6	113.5	3.50
Integration policy(Env.4)	125.0	60.6	2.55
Integration policy(Env.5)	125.0	75.5	2.32

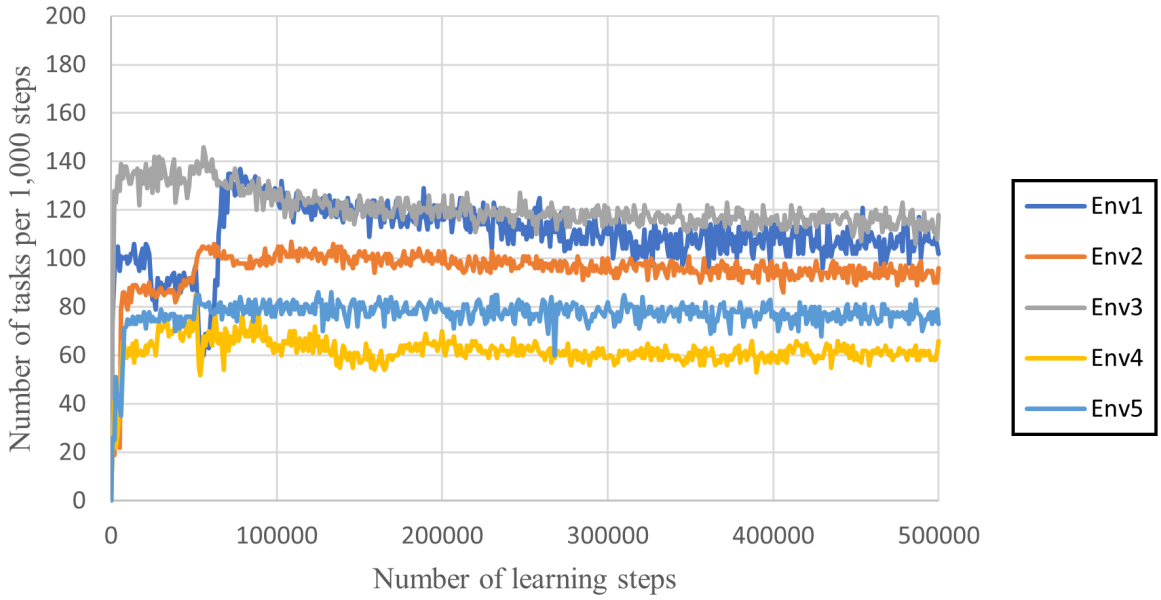

図 3.10: エージェント数が 2 体での組み合わせ方策を用いた各実験環境の結果

3.3.2. エージェントが 3 体での各実験環境の結果

エージェント数が 3 体での実験環境 1 から実験環境 5 において，Softmax 方策での荷物排出量の学習遷移を図 3.11 に示す．図 3.11 において，グラフ縦軸が 1000 ステップごとの荷物搬出量であり，グラフ横軸が試行ステップ数である．実験環境 1 から実験環境 3 の荷物搬出量の理論値は 249.9 回，実験環境 4 から実験環境 5 の荷物搬出量の理論値は 187.5 回となった．実験環境 4 においては，学習が安定する結果と安定しない結果が同程度の頻度で現れたため，図 3.11 には学習が安定した場合の結果を記載している．実験環境 4 において，学習が安定しない結果は図 3.12 に示している．また，試行回数 490,000 回から 500,000 回の 10,000 回間における 1,000 ステップごとの荷物排出量平均と荷物搬出量の標準偏差の結果を表 3.4 に示す．各実験環境において，実験環境 2，実験環境 4 以外はすべて収束している．実験環境 2 では，1000 ステップごとの荷物排出量が最大 100 回程度であるが，最低値が 0 回となっており，相互進路妨害が発生していることが考えられる．また，実験環境 4 において，収束している結果ではエージェントは停止行動や迂回行動を行いながら規則的に行動選択を行った．収束しない結果では，1 体のエージェントが他のエージェントに阻害され，その場に停止もしくは往来し続ける．この 1 体のエージェントが，再び他のエージェントを阻害もしくは環境同定を行えていないルールに遷移するため，行動選択が安定しなかった．

荷物排出量は，実験環境 1 及び実験環境 5 が理論値であり，その他環境は理論値より下回る結果となった．実験環境 2 が低下したのは，エージェント数が 2 体の場合と比較して，待機しなければならないエージェントが増加し，学習環境が複雑化したためと考えられる．実験環境 3 がエージェント 2 体時と比較して理論値より荷物排出量が低下した理由は，エージェント数増加による停止行動および迂回行動の増加である．実験環境 3 は，同時に荷物搬入口に侵入できるエージェントは 1 体のみであり，他のエージェントはその間停止及び迂回しなければ荷物を受け取ることができない．停止行動と迂回行動は最短行動

ではなく，1 体のエージェントがこれら行動を選択すると，他のエージェントにも行動の影響が波及する．このため，実験環境 3 の荷物排出量は理論値よりも低下する．実験環境 4 において，収束している結果ではエージェントは停止行動や迂回行動を行うため，理論値よりも荷物搬出量が低いが安定している．収束しない結果では，停止しているエージェントが他のエージェントを阻害した場合に排出量が低下し，それ以外では残り 2 体のエージェントが荷物排出を続けるため，図 3.12 のようなグラフとなる．

表 3.4: エージェントが 3 体での各環境別荷物搬出量の平均値と標準偏差

Used policy(Environment)	Theoretical value per 1,000 steps	Average tasks per 1,000 steps	standard deviation
Softmax policy(Env.1)	249.9	249.9	0.32
Softmax policy(Env.2)	249.9	63.2	43.69
Softmax policy(Env.3)	249.9	214.2	0.42
Softmax policy(Env.4)	187.5	138.2	1.03
Softmax policy(Env.4:Non-convergence)	187.5	81.6	44.80
Softmax policy(Env.5)	187.5	187.5	0.53

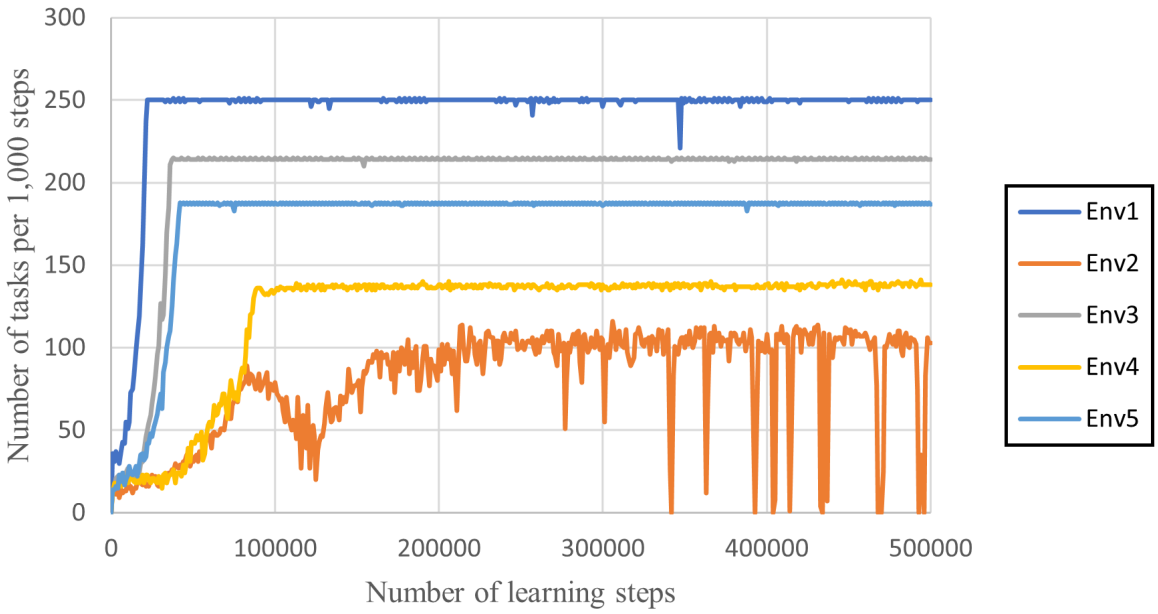

図 3.11: エージェント数が 3 体での Softmax 方策を用いた各実験環境の結果

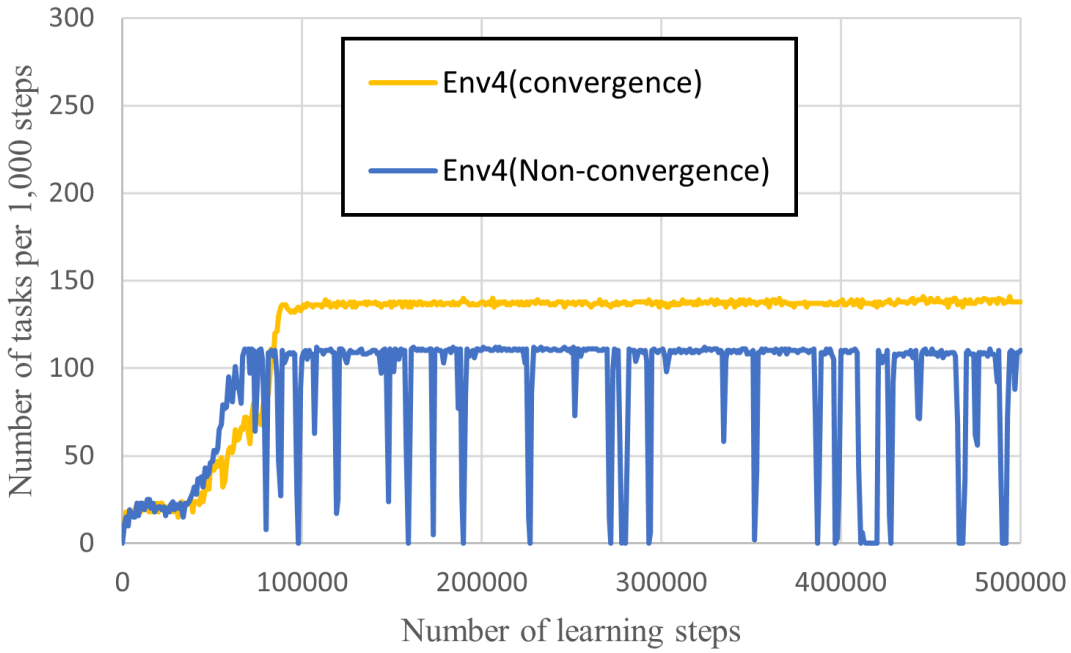

図 3.12: エージェント数が 3 体での Softmax 方策を用いた実験環境 4 の両結果

組み合わせ方策を用いたエージェント数が 3 体での各実験環境の荷物排出量の学習遷移を図 3.13，荷物排出量平均と荷物搬出量の標準偏差を表 3.5 に示す．組み合わせ方策を用いた実験では，エージェント数を 3 体にした場合では実験環境 2 のみ Softmax 方策よりも荷物排出量が多くなり，標準偏差が低くなった．このため，Softmax 方策と比較して実験環境 4 のみ学習が安定化した．

また，実験環境 4 においては，Softmax 方策では半数の結果が学習が安定しないが，組み合わせ方策を用いた結果では過半数が図 3.13 のように収束する結果となり，標準偏差も 0.51 と低い値となった．このため，エージェント数が 3 体における組み合わせ方策は，狭い環境に対して最適ではない安定行動の獲得することができる．

表 3.5: 組み合わせ方策を用いたエージェントが 3 体での各環境別荷物搬出量の平均値と標準偏差

Used policy(Environment)	Theoretical value per 1,000 steps	Average tasks per 1,000 steps	standard deviation
Integration policy(Env.1)	249.9	182.4	2.55
Integration policy(Env.2)	249.9	129.9	0.88
Integration policy(Env.3)	249.9	143.1	2.18
Integration policy(Env.4)	187.5	103.4	0.51
Integration policy(Env.5)	187.5	154.7	0.95

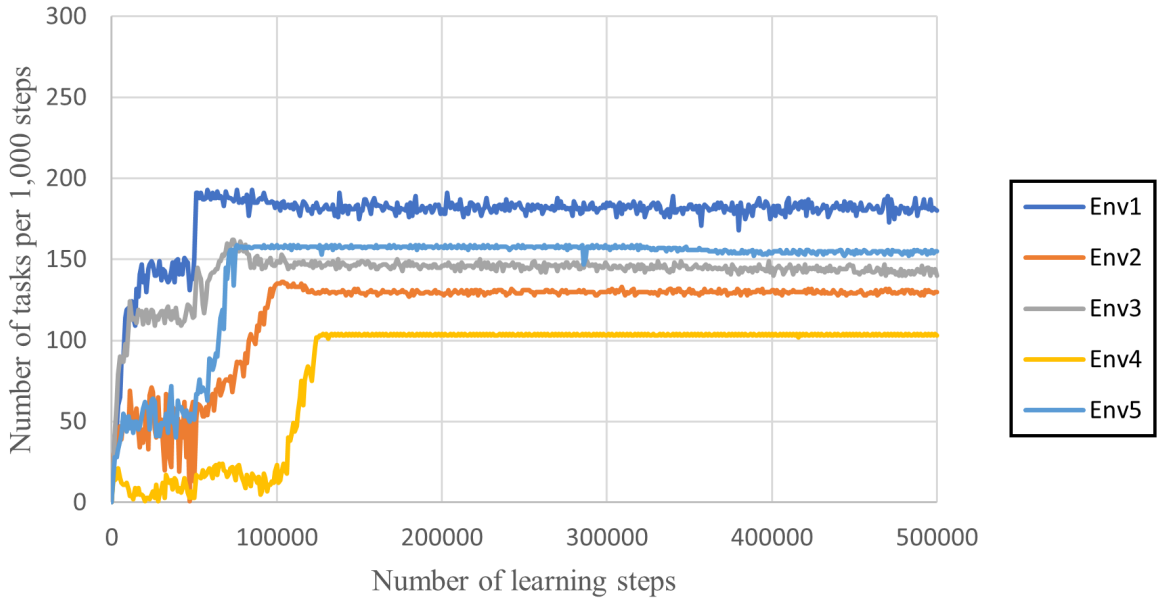

図 3.13: エージェント数が 3 体での組み合わせ方策を用いた各実験環境の結果

3.3.3. エージェントが 4 体での各実験環境の結果

実験環境 1 から実験環境 5 において，Softmax 方策と組み合わせ方策の試行ステップ数が 490,000 回から 500,000 回の 10,000 回間における 1,000 ステップごとの荷物排出量平均と荷物搬出量の標準偏差の結果を表 3.6 に示す．また，荷物排出量の学習遷移を図 3.14 に示す．実験環境 1 から実験環境 3 の荷物搬出量の理論値は 333.3 回，実験環境 4 から実験環境 5 の荷物搬出量の理論値は 250.0 回となる．

実験環境ごとのエージェントの振る舞いに関して，実験環境 1 では規則的に周回する行動をとり荷物搬送を行った．表 3.6 の荷物排出量が理論値であるため，エージェントは最短経路の行動を選択した．このため，図 3.14 に示すように各実験環境の中で最も 1,000 試行ごとの荷物搬出量平均が高くなる．また，環境が他の環境と比較して単純であるため，学習が収束する速度が最も早い．エージェントの振る舞いは，エージェント数が 2 体と 3 体の場合と同様に外周に沿った周回行動を行った．実験環境 2 では，通過可能な一マスでエージェントが互いに干渉し，試行ごとに異なる行動選択を行うことが多い．このため，荷物搬送は規則的に行えなかった．また，荷物排出量も各実験環境の中で最も少なくなるため，学習の進行度が最も遅い．実験環境 3 では，すべてのエージェントが図 3.4 における上部の通路から荷物搬入口へ向かい，その後下部の通路から荷物搬出口へ荷物搬送を行う．外周に沿いながら荷物搬送を行ったが，エージェント数が 3 体時と同様に荷物搬入口へは 1 体のエージェントのみ侵入できるため，実験環境 1 より低い荷物排出量となった．実験環境 4 では，あるエージェントが荷物搬入口で行動不能となる．Softmax 方策では，行動不能のエージェントが他のエージェントを阻害し，学習が正しく収束しなかった．実験環境 5 では，実験環境 4 と異なり他のエージェントを阻害することなく，壁を沿うように周回し周期的な荷物搬送が行えた．

荷物排出量は，実験環境 1 及び実験環境 5 が理論値であり，その他環境は理論値より下回る結果となった．理論値を下回る環境の中では，実験環境 2，実験環境 4 が他のエー

ジェントからの阻害を受けており，荷物排出が安定しなかった．実験環境 3 は環境の構造による停止行動が低下の要因であるため，規則的な荷物排出が行えた．

表 3.6: エージェント数が 4 体での各環境別荷物搬出量の平均値と標準偏差

Used policy(Environment)	Theoretical value per 1,000 steps	Average tasks per 1,000 steps	standard deviation
Softmax policy(Env.1)	333.3	333.3	0.48
Softmax policy(Env.2)	333.3	63.7	13.93
Softmax policy(Env.3)	333.3	235.2	0.42
Softmax policy(Env.4)	250.0	103.4	51.49
Softmax policy(Env.5)	250.0	250.0	0.0

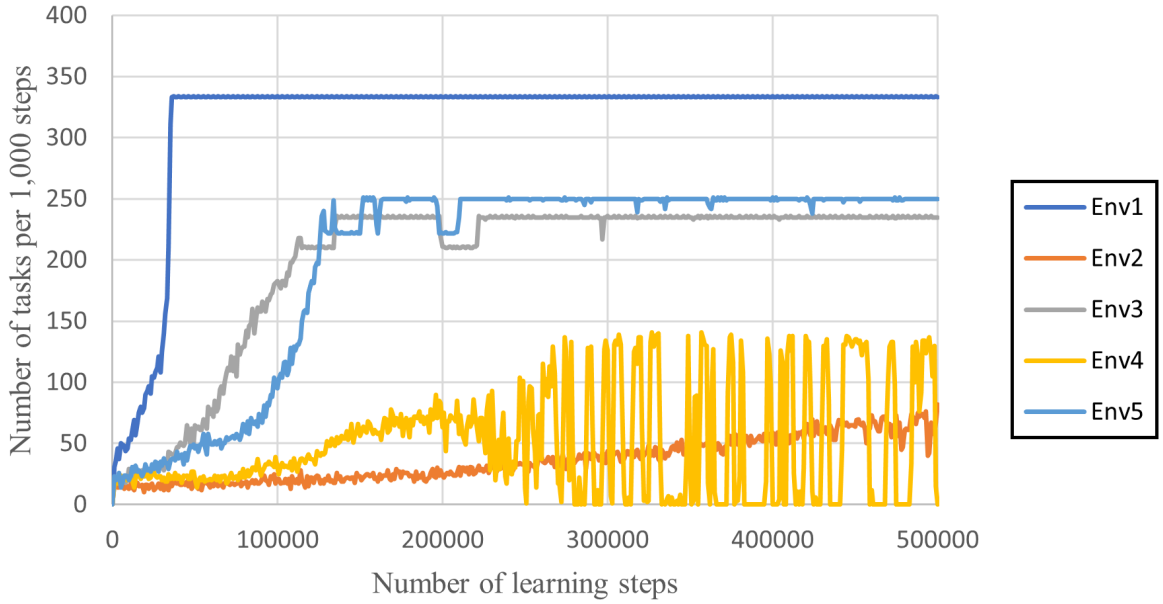

図 3.14: エージェント数が 4 体での Softmax 方策を用いた各実験環境の結果

実験環境 4 におけるエージェントの振る舞いは，図 3.15 に示すようにすべてのエージェントが停止行動を選択し続ける状態と，数体のエージェントが荷物搬送を行う状態の 2 パターンとなる．停止行動を選択し続ける場合，連続上限ステップ数の 2000 ステップまで荷物搬出量が 0 となる．数体のエージェントが荷物搬送を行う場合，1000 ステップごとの荷物搬送量は 140 程度となるが，その後すぐに停止行動のパターンになるため，荷物搬送は安定しない．

3.3.1 で述べた 2 体のエージェントでの実験結果と比較すると，実験環境 1 と実験環境 5 の結果がエージェントの数が 2 体から 4 体に増加させたため，荷物搬出量も 166 回から 333 回，125 回から 250 回の結果となった．一方で，実験環境 2 と実験環境 4 においてはエージェント数が 2 体の場合と比較して荷物搬出量の平均値が減少しており，標準偏差が 10 以上高くなっている．この結果からも実験環境 2 と実験環境 4 は正しく学習が行えていない．

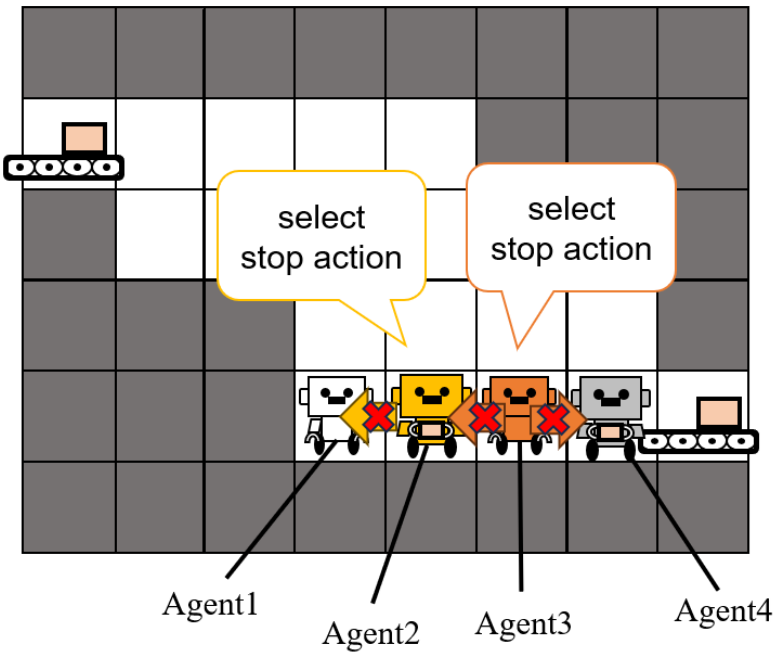

図 3.15: エージェント数が 4 体での各実験環境 4 におけるエージェントの振る舞い

組み合わせ方策を用いたエージェント数が 4 体での各実験環境の荷物排出量の学習遷移を図 3.16，荷物排出量平均と荷物搬出量の標準偏差を表 3.7 に示す．エージェント数が 4 体になった場合の組み合わせ方策を用いた結果では，Softmax 方策と同様に実験環境 2 および実験環境 4 の学習が安定せず，標準偏差の高い値となった．しかし，エージェント数が 3 体における組み合わせ方策と比較して，実験環境 1 と実験環境 5 の荷物搬出量が理論値に近い結果となった．また，実験環境 2 ではエージェント数が 3 体における組み合わせ方策よりも荷物排出量が低下し，標準偏差が増加する結果となった．エージェント数が 4 体であれば，組み合わせ方策でも Softmax 方策と同様に環境の広さに依存した学習を行うため，各エージェントが安定的な振る舞いを行うことができなかったと考えられる．

表 3.7: 組み合わせ方策を用いたエージェントが 4 体での各環境別荷物搬出量の平均値と標準偏差

Used policy(Environment)	Theoretical value per 1,000 steps	Average tasks per 1,000 steps	standard deviation
Integration policy(Env.1)	333.3	306.9	2.47
Integration policy(Env.2)	333.3	56.2	5.03
Integration policy(Env.3)	333.3	199.9	0.316
Integration policy(Env.4)	250.0	12.6	10.6
Integration policy(Env.5)	250.0	249.9	0.74

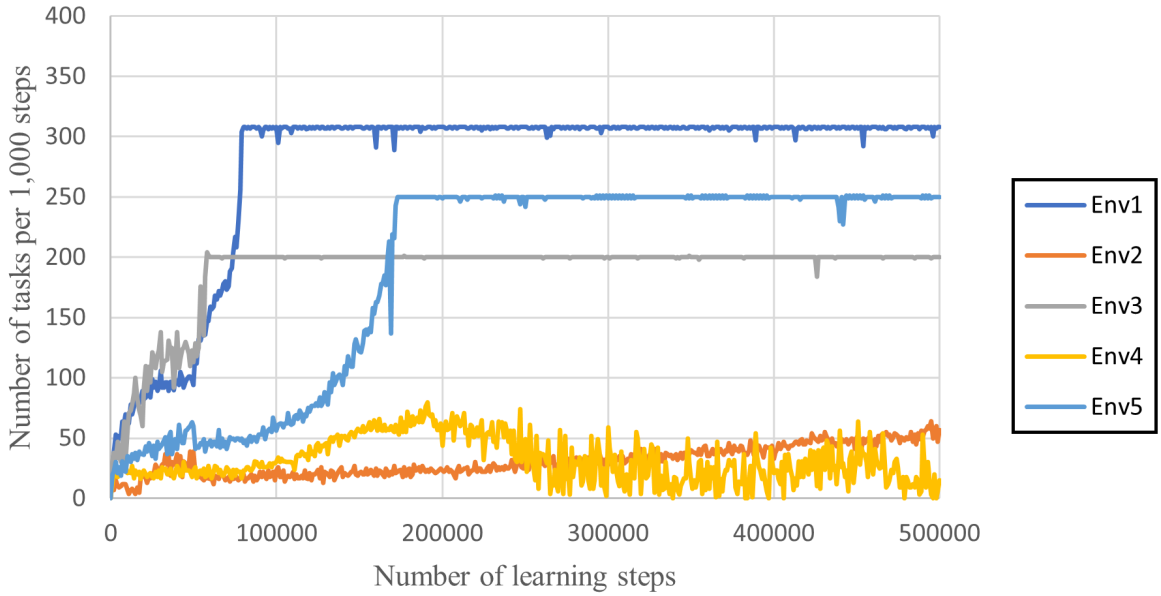

図 3.16: エージェント数が 4 体での組み合わせ方策を用いた各実験環境の結果

以上の結果より，Softmax 方策及び組み合わせ方策を用いる場合，エージェント数が増加するほど実験環境 2 や実験環境 4 のような環境では正しく学習を行うことが困難であることが確認された．

3.4. 考察

まず，エージェント数が 3 体における実験環境 2 と実験環境 4 の振る舞いについて考察する．実験環境 2 における各エージェントの内部 Q 値において，候補行動集合のうち最大 Q 値から 2 番目に高い Q 値の差分を 1000 ステップごとの平均値を図 3.17 に示す．左縦軸が 1000 ステップごとの荷物の搬出量，右縦軸が各エージェントの候補行動集合内の最大 Q 値から 2 番目に高い Q 値の差分を 1000 ステップごとの平均値で表したものである．右縦軸の値が高い場合，候補行動集合の中から最大 Q 値の行動が選択されやすくなる．横軸が試行ステップ数である．以下，グラフ右軸を差分平均と表記する．また，図 3.17 に示す凡例は Luggage transport が左軸参照であり，Agent1 から Agent3 までが右縦軸参照の差分平均である．図 3.17 から横軸 400000 ステップ付近では，荷物排出量が低下するのと同時に Agent2 の差分平均が減少し，Agent3 の差分平均が上昇している．このため，Agent1 または Agent3 がより高い報酬を得られる行動を行う場合，Agent2 がその行動を阻害し膠着状態に陥ることで荷物搬出が行えなくなると考えられる．実験環境 4 における荷物排出量と各エージェントの差分平均のグラフを図 3.18 に示す．実験環境 4 における結果も実験環境 2 と同様に，標準偏差が高く荷物搬送が安定しない．これは，実験環境 4 では Agent3 の差分平均が低く，安定した行動選択を行っていないためである．また，Agent3 の差分平均が上昇するタイミングで荷物排出量が 0 となり，他のエージェントの差分平均が減少しているため，Agent3 が荷物搬送を行おうとした場合に他のエージェントの妨げとなり，環境が膠着状態に陥ることが考えられる．式 (2.2) から Agent3 のように差分平均が 2 程度のとき，約 96%で最大 Q 値の行動が選択されるが，それ残りの 4%で

行動を選択する場合に他のエージェントの妨げとなる行動を選択していると考えられる．この差分平均が小さいことによるエージェントの行動選択の不安定化は，組み合わせ方策のすべての実験環境でも起こりえたと推察される．

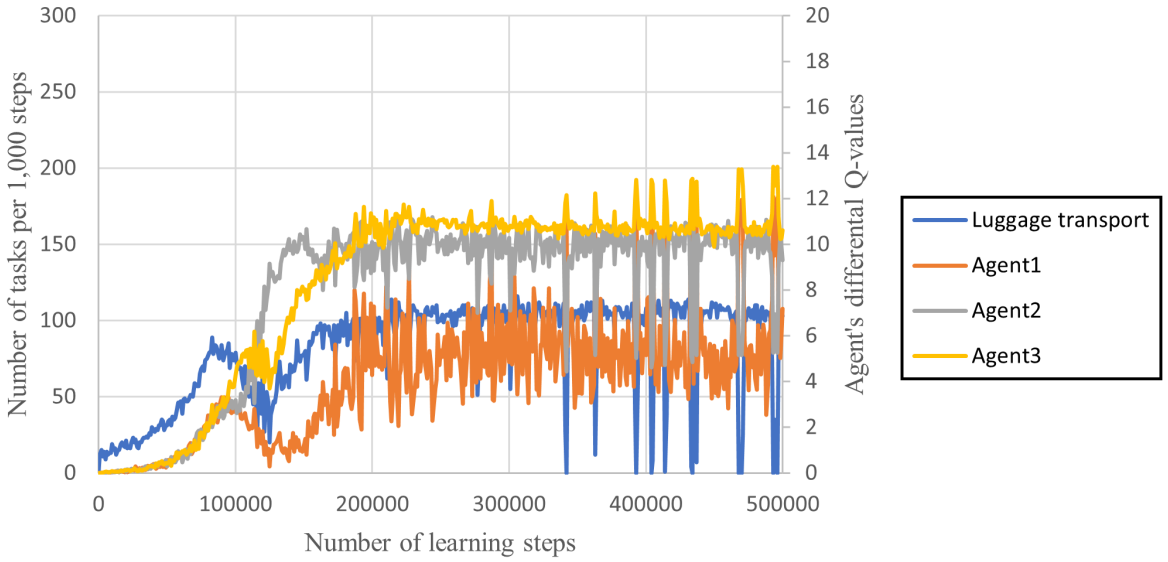

図 3.17: エージェント数が 3 体での Softmax 方策を用いた実験環境 2 の結果

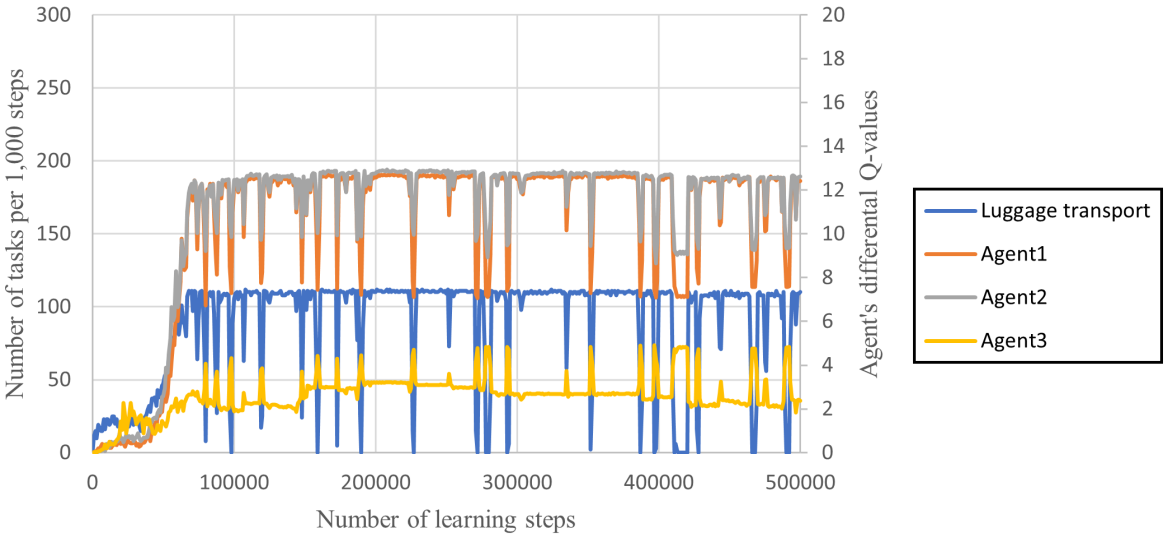

図 3.18: エージェント数が 3 体での Softmax 方策を用いた実験環境 4 の安定しない結果

次に，4 体エージェントの実験において実験環境 4 が収束しない結果について考察する．4 体エージェントの実験において，実験環境 4 と比較して実験環境 5 では，安定して荷物搬送を行えている．これは，視界範囲に他のエージェントを観測することなく荷物搬送を行うことができるためであると考えられる．また，環境の構造としてエージェントが十分に周回できる広さである場合，停止や往来を行動選択する確率が減少することも安定した学習の要因であると考えられる．実験環境 2 においては，エージェントの振る舞いが安定

せず，荷物搬出量と Q 値の差分平均が低いため，正しく学習が行えていない．このことから，図 3.3 に示すような環境では，エージェントの Q 値にのみ従った行動選択を十分に行うことが困難であり，望ましいと思われる行動先に他のエージェントが存在しやすいことが考えられる．その他実験環境及び強化学習の特性として，部分観測のエージェントは視界内に他のエージェントが映ると停止または迂回することにより，視界内にエージェントが映らない行動を選択しやすい．これは，他のエージェントが視界に映る場合の学習が環境同定されるが，より高い報酬を受け取ることが可能な視界状態はエージェントを視界内に映さないことによる進路妨害の予防であることが考えられる．他のエージェントが視界内に映る場合，エージェントを避けるために停止または迂回することでエピソード長が長くなり，式 (3.3) によって分配される Q 値が低くなる．よって，各エージェントがタスク周辺環境で視界内にエージェントを映さないような行動選択を行うことが難しい実験環境 4 は，実験環境 5 と比較して安定して荷物搬送を行えない．

一方で，2 体エージェントでの実験では狭い環境条件である実験環境 2 の結果が収束している．これは，エージェントの振る舞いが追従と似た行動選択のためである．実験環境 2 のような狭い環境では，視界内にエージェントを映すことで報酬にたどりつける場合も考えられる．エージェントが 2 体での実験環境 2 の結果では，1 体のエージェントがもう 1 体のエージェントに追従する形で荷物搬送を行う．このとき，図 3.19 に示すように実験環境 2 の荷物搬入口では同時に荷物を受け取ることができないため，1 体のエージェントはもう 1 体が荷物を受け取るまで環境左側で待機し，もう 1 体のエージェントが受け取り完了後に追従して荷物を受け取り搬送する行動を行う．このように，他のエージェントの影響を受けづらい環境やエージェント数の場合，追従型の学習を行う個体も現れる．特に，実験環境 2 のように待機後に行動を行わなければならない場合，環境内に追従しなければならない対象のみ存在する条件であれば，視界内にエージェントが映る状態での安定した行動選択が可能になると考えられる．

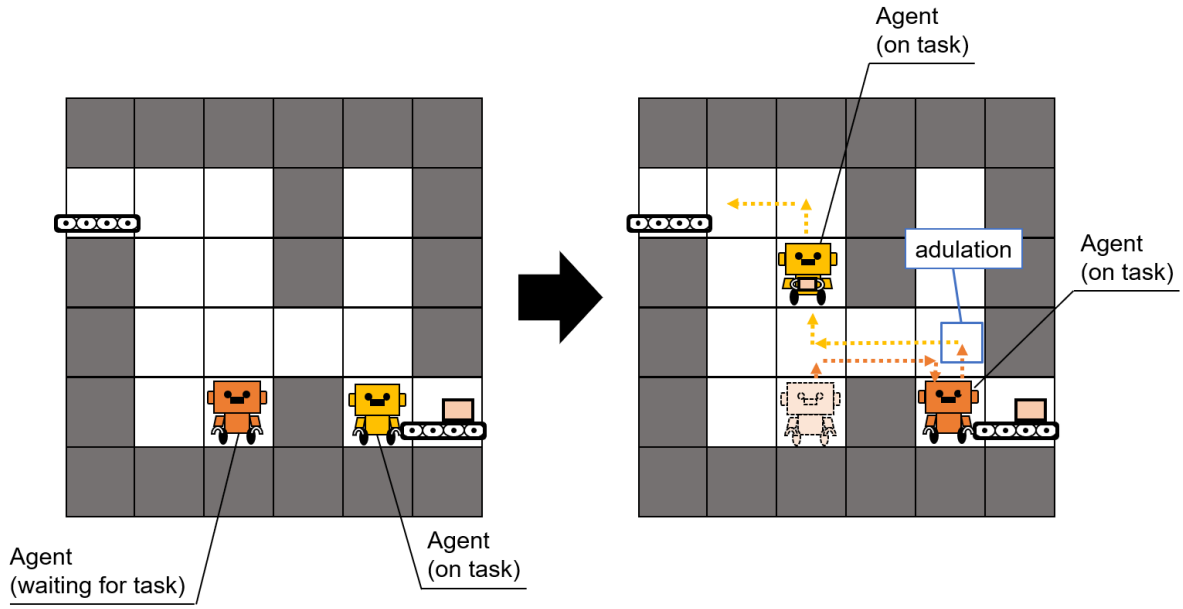

図 3.19: 実験環境 2 において待機後に他のエージェントを追従する行動例

最後に，組み合わせ方策について考察する．ϵ-greedy 方策は，環境がモデルベースの場

合や内部状態が確率モデルである場合に最適解に収束することが示されている．しかし，本実験条件では環境が未知であり，候補行動集合に停止行動が存在するため，組み合わせ方策はSoftmax 方策よりも荷物搬出量が低い結果となった．また，これに準ずる形でモデルフリーな手法が有効であるため，Softmax 方策を用いた結果では荷物搬出量が最大値となりやすい．このため，ϵ-greedy 方策を用いて学習を行う場合は，実験環境に対する先験的な知識を基とする ϵ の値調整や，実験手法である Q-PSPLearning がモデルフリーな手法であり，実験条件であるエージェントの停止行動と貪欲行動の相性が悪い点などから，有効性はSoftmax 方策より低いと考えられる．また，ϵ-greedy 方策を用いた学習では，学習ルールの不規則さに伴う Q 値の差分平均減少により，方策切り替え後のエージェントの行動選択の不安定化につながると推察される．以上のことから，組み合わせ方策は未知な環境の探索に対して安定化の問題がある．しかし，適切なエージェント数や環境に用いれば，エージェント数が 3 体時の実験環境 2 のように学習を安定化させることが可能である．

第4章　相対ベクトルを導入したルールの評価実験

4.1.　マルチエージェントシステムを用いた荷物搬送問題の問題点

マルチエージェントシステムを用いた荷物搬送問題では，第 2 章に述べるようにシングルエージェントと比較して搬送効率が上昇するが，環境によっては互いに進行阻害することから学習が正しく行えない問題がある．特に，第 3 章で行った実験のように狭い環境やエージェント数によっては，人間が環境全体を観測した場合に単純な解と考えらえれる状況であっても正しく学習が行えないことがある．このため，本章では候補行動集合を生成するルールベースに着目し，狭い環境に対応できるルールを提案することで，学習を安定化させることを目標とする．

4.2.　相対ベクトルを導入したルールの提案

第 3 章では，エージェントの視界環境のみをもとにした環境ルールにより学習を行った．しかし，部分観測環境におけるエージェントの学習は，他のエージェントとの相互作用により，視界環境のみで学習を行うことは困難である．特に，第 3 章で示した実験環境 4 のようなタスク環境の狭い環境では，視界内にエージェントが映ることによる環境同定の条件増加及び進路妨害により，荷物搬送効率の低下が発生する．このため，本論文ではエージェントの学習において，視界環境を用いた環境同定のみでなく，現在地点とその n ステップ前との位置を用いた相対的なベクトルをルールベースに導入する手法を提案する．本研究では，このルールを以下 “相対ベクトルルール” と記す．

ここでの相対ベクトルとは，現在地点と n ステップ前の (x,y) 座標の位置ベクトルを絶対値でとった値である．図 4.1 に示すように，相対ベクトルを導入するにあたり，環境ルールのルール条件部に用いる環境同定に加え，現在から n ステップ前の行動との相対的なベクトルをもとに重み付き候補行動集合を生成する．このため，用いる相対ベクトルはエージェントの内部で生成され，各エージェントはこの値を共有しない．また，n ステップ前の行動を用いることができない場合は Null とする．清本らの研究 [22] では，初期位置と現在位置の位置ベクトルで学習を行うが，現在地とその n ステップ前の相対ベクトルと用いることにより，課題であった初期位置の依存性が緩和されると考えられる．

相対ベクトルルールの *if-then* ルール条件式において，エージェントが荷物を持っていない場合の条件を式 (4.1)，エージェントが荷物を持っている場合の条件を式 (4.2) に示す．

$$if \quad \text{環境 and 荷物なし and 相対ベクトル} \quad \text{then} \quad \text{候補行動集合} \tag{4.1}$$

$$if \quad \text{環境 and 荷物あり and 相対ベクトル} \quad \text{then} \quad \text{候補行動集合} \tag{4.2}$$

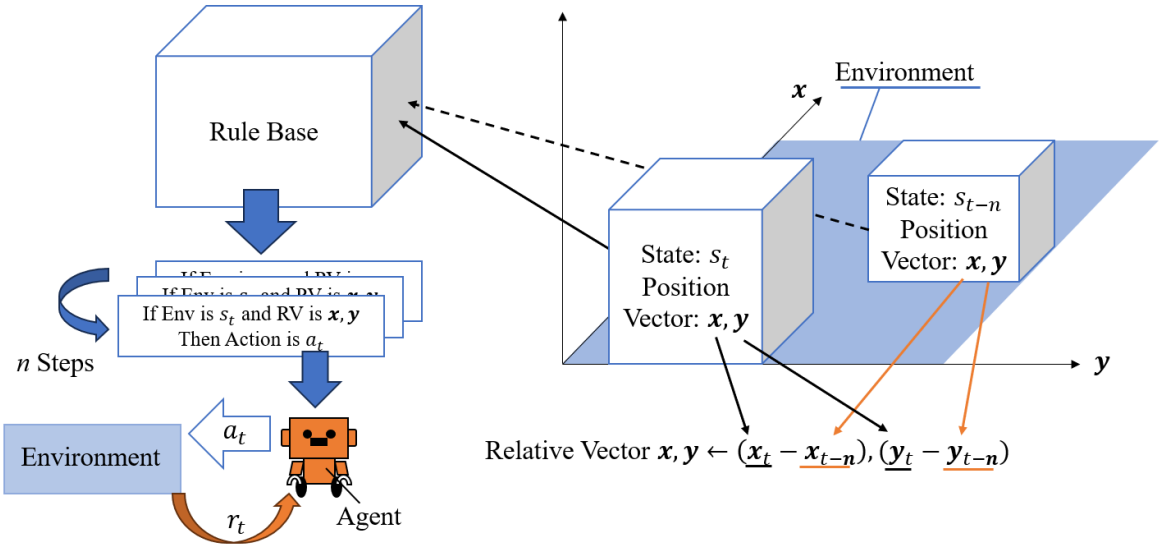

図 4.1: 相対ベクトルを導入したルール

相対ベクトルルールのアルゴリズムとしては，相対ベクトルを含む環境同定を行い，その後行動選択を行う．以下に，相対ベクトルルールのアルゴリズムを示す．

Step1 現在環境の位置と n ステップ前の位置から相対ベクトルを生成する．

Step2 過去ルールを参照し，該当するルールがなければ新規ルールとする．

Step3 行動選択手法を用いた行動選択を行い，報酬を得た場合は Step4 にへ進み，得られない場合は Step1 へ戻る．

Step4 更新式を用いた学習を行う．

Step5 試行が終了しなければ，Step1 へ戻る．

相対ベクトルルールでは，Step2 におけるルール参照の際に，従来の環境同定に加えて相対ベクトルの同定も行う．ルール条件に相対ベクトルを加えることにより，従来の環境からの状態だけでは学習を行うことが困難であった POMDP 環境に対して，ルール数が増加する．このことから，マルコフ性の保証および学習の安定化が期待される．

相対ベクトルルールと環境ルールを比較すると，環境ルールでは視界から得た状態を観測し，この状態に対して有効と思われる行動を出力する．この行動から得た報酬をもとに状態価値関数を更新する．相対ベクトルルールでは，視界から得られる状態のみでなく，相対ベクトルも環境同定の条件として加える．ここで，相対ベクトルを用いるのは，部分観測環境下では自身の環境に対する位置が不明確であり，部分観測に影響されない前ステップとの相対的なベクトルが有効と考えられるためである．相対ベクトルを条件に付加することにより，視界環境に依存する環境ルールより正確な学習が行えると考えられる．また，相対ベクトルを用いる際に必要な情報がエージェント自身の数ステップ間であるた

め，初期位置の変化や学習中の環境変化に対応しやすい．一方，環境条件が増加することで，学習にかかる時間と学習が収束するまでの時間が増加するため，学習に用いる視界範囲などの情報を減らす必要がある．

4.3. 荷物搬送問題の評価実験

4.3.1. 実験手法

本実験では，第 3 章と同様に荷物搬送問題を行う. このとき，Softmax 方策と組み合わせ方策を用いたエージェント数が 4 体時に学習が困難であった実験環境 4 に対して，4.2 で述べた相対ベクトルルールを用いることにより学習の安定化を図る．エージェントの Q 値の更新式は，第 3 章と同様に Q-PSPLearning を用いて行う．また，相対ベクトルルールに用いる行動選択手法は Softmax 方策であり，学習途中で手法を変化させることはない．

4.3.2. 実験内容

本論文の実験条件として，動作確認及び結果比較のための図 3.1 に示す実験環境 1，第 3 章で正しく学習を行えなかった図 3.5 に示す実験環境 4 の 2 種類の環境で行う．エージェントの数は 4 体のみで行う．また，エージェントの条件も第 3 章と同様に視界を周辺 1 マスとしており，上下左右と停止の 5 種類の行動を選択できる．このため，相対ベクトルルールでは，最短で視界外へ行動ができる $n = 2$ ステップ前との相対ベクトルを用いる．2 ステップ前と現在地との相対ベクトルを環境同定に組み込むことにより，視界のみの環境ルールでは学習が困難であった実験環境 4 に対しての有効性を検証する．すべてのエージェントが行動を選択し，環境が遷移することを 1 ステップとする．学習に用いたパラメータの設定は表 4.1 に示すもので行った．また，環境ルールと相対ベクトルルールはエージェントが荷物を持っている場合，持っていない場合と異なるルールを用いて学習を行った．

本実験は，第 3 章で用いた実験環境 1 と実験環境 4 について，相対ベクトルルールで 500,000 ステップの試行を行う．また，Q 値の更新に用いる式 (3.3) における温度定数 T は $T = 0.52$，報酬 $r_t = rd^i$ における公比 d は式 (2.15) から行動の種類が 5 種類であるため，$d = \frac{1}{5} = 0.20$ で行った．

表 4.1: 相対ベクトルルールを用いる実験の設定パラメータ

Number of agents	4
Relative vector between steps	2
Initial Q value	0.00
Learning rate α	0.06
Discount rate γ	0.95
Planning discount rate d	0.20
Temperature value T	0.52
Reward r	1.0
Total steps	500,000
Limit continuous steps	2000

4.4. 実験結果と考察

4.4.1. 実験結果

実験環境 1 において，環境ルールと相対ベクトルルールの結果をそれぞれ図 4.2 と図 4.3 に示す．図 4.3 の左縦軸が 1000 ステップごとの荷物搬出量であり，右縦軸が各エージェントの候補行動集合内の最大 Q 値から 2 番目に高い Q 値の差分を 1000 ステップごとの平均値で表した差分平均である．横軸は試行ステップ数である．実験環境 1 における結果は，第 3 章の環境ルールを用いた実験と同様に安定した荷物搬出を行えている．また，各エージェントの Q 値の差分平均は 14 程度と高いため，安定的な行動選択が行えており，相対ベクトルルールは実験環境 1 において正しく学習が行えている．しかし，荷物排出量は 300 程度であり理論値である 334 回より低いため，環境条件の増加による最適方策の収束が環境ルールと比較して難化していると考えられる．

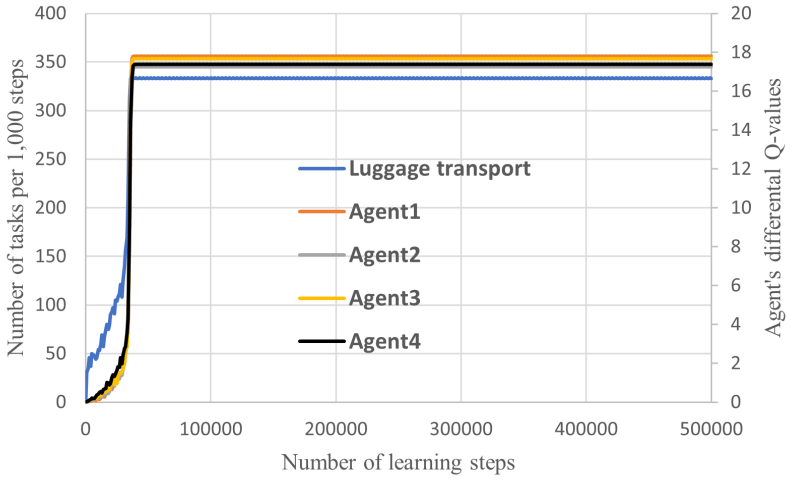

図 4.2: 実験環境 1 において環境ルールを用いた実験結果

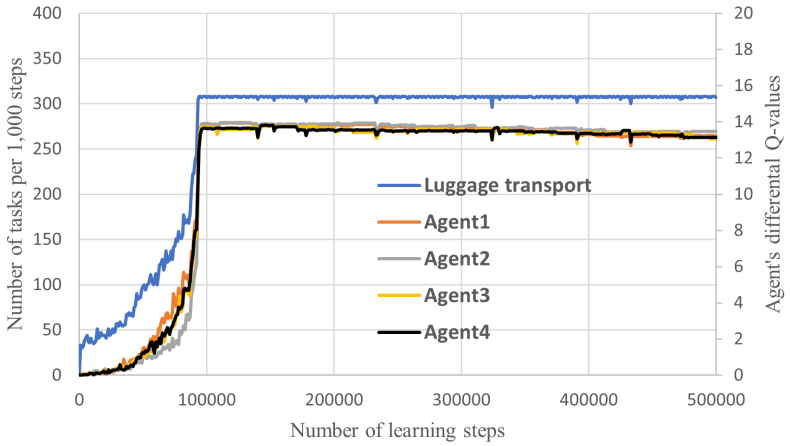

図 4.3: 実験環境 1 において相対ベクトルルールを用いた実験結果

実験環境 4 において，環境ルールと相対ベクトルルールによる結果をそれぞれ図 4.4 と図 4.5 に示す．環境ルールの場合，第 3 章で述べたようにエージェントは図 4.4 のように荷物の搬送が安定して行えず，学習が収束しない．

ここで，図 4.5 に示す Agent1 は環境ルールにおける Agent3 と同様に他のエージェントと比較して低い Q 値の差分平均であり，エージェントの荷物搬出量も Agent1 のみほとんど搬送が行えなかった．一方，Agent2 から Agent4 までの各エージェントは，1000 ステップごとに 100 から 150 間で荷物搬出を安定して行えている．また，このときのエージェントの振る舞いを図 4.6 に示す．図 4.6 では，エージェント 1 が停止行動および往来行動を選択し続け，他のエージェントの行動を阻害しないように振る舞う．このため，Agent1 は他のエージェントを優先する利他的行動を選択しているように推察できる．このため，相対ベクトルルールにより実験環境 4 の学習が改善されている．

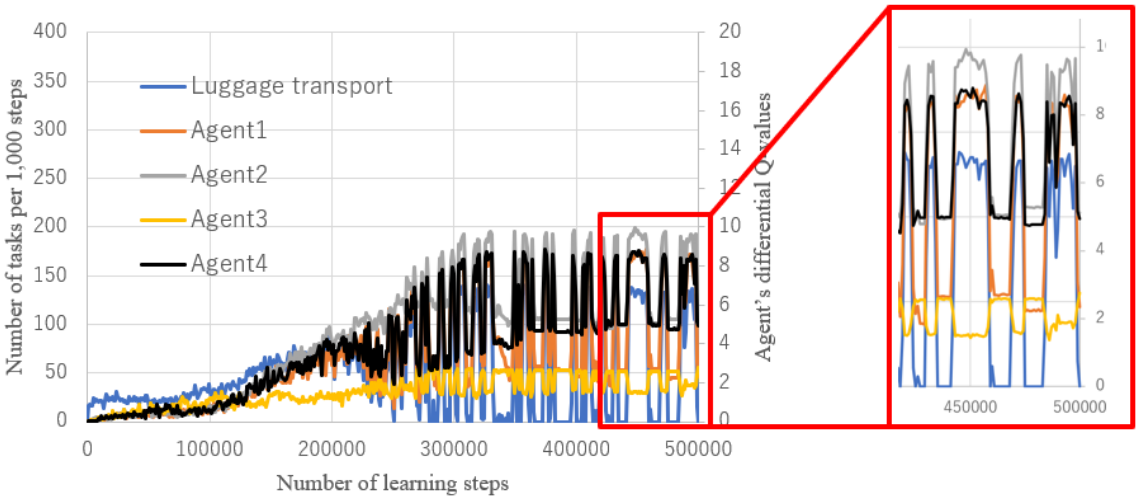

図 4.4: 実験環境 4 において環境ルールを用いた実験結果

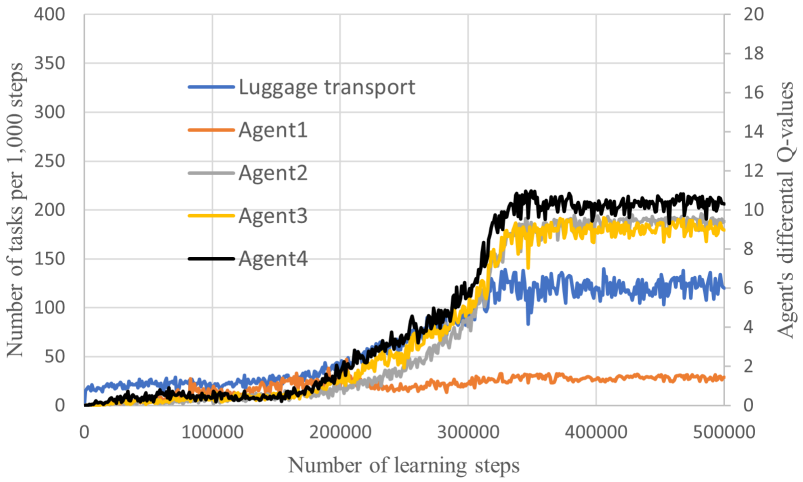

図 4.5: 実験環境 4 において相対ベクトルルールを用いた実験結果

表 4.2: 環境ルールと相対ベクトルルールを用いた各環境別荷物搬出量の平均値と標準偏差

Used method(Environment)	Theoretical value per 1,000 steps	Average tasks per 1,000 steps	standard deviation
Environment Rule(Env.1)	333.3	333.3	0.48
Relative Vector Rule(Env.1)	333.3	307.7	0.50
Environment Rule(Env.4)	250.0	103.4	51.49
Relative Vector Rule(Env.4)	250.0	120.9	7.25

表 4.3: エージェント 1 の Q 値の平均値と標準偏差

Used method (Environment)	Agent1 average Q-values	Agent1 standard deviation
Environment Rulel(Env.1)	17.80	0.003
Relative Vector Rule(Env.1)	14.99	0.29
Environment Rule(Env.4)	7.05	2.33
Relative Vector Rule(Env.4)	1.39	0.10

表 4.4: エージェント 2 の Q 値の平均値と標準偏差

Used method (Environment)	Agent2 average Q-values	Agent2 standard deviation
Environment Rule(Env.1)	17.26	0.004
Relative Vector Rule(Env.1)	13.98	0.01
Environment Rule(Env.4)	8.41	1.81
Relative Vector Rule(Env.4)	9.42	0.18

表 4.5: エージェント 3 の Q 値の平均値と標準偏差

Used method (Environment)	Agent3 average Q-values	Agent3 standard deviation
Environment Rule(Env.1)	17.68	0.003
Relative Vector Rule(Env.1)	14.82	0.34
Environment Rule(Env.4)	2.01	0.33
Relative Vector Rule(Env.4)	9.01	0.20

表 4.6: エージェント 4 の Q 値の平均値と標準偏差

Used method (Environment)	Agent4 average Q-values	Agent4 standard deviation
Environment Rule(Env.1)	17.37	0.007
Relative Vector Rule(Env.1)	13.41	0.11
Environment Rule(Env.4)	7.58	1.39
Relative Vector Rule(Env.4)	10.36	0.23

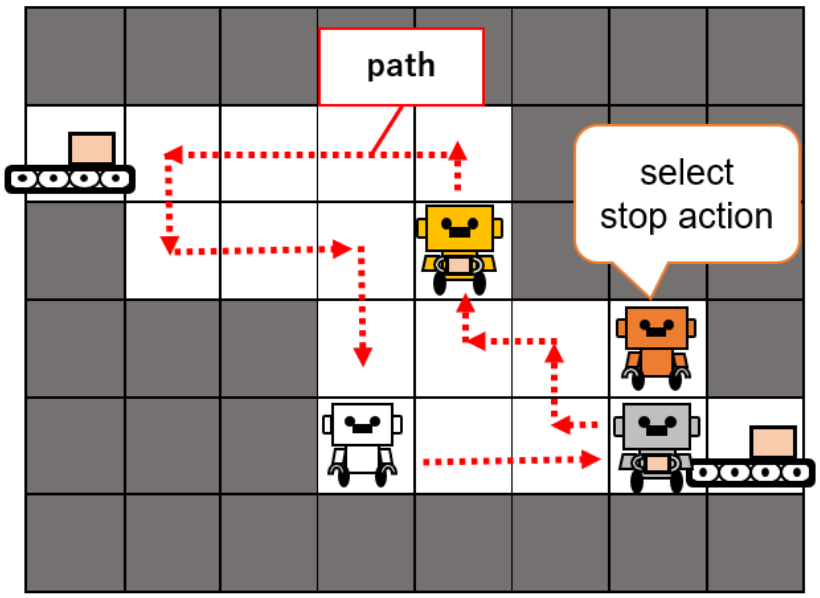

図 4.6: 実験環境 4 において相対ベクトルルールを用いた場合のエージェントの振る舞い

4.4.2. 考察

実験環境 1 の結果比較について考察する．表 4.2 に示すように，相対ベクトルルールでは環境ルールの結果と比較して荷物搬出量が少ない．各エージェントの Q 値の平均値は相対ベクトルルールが低くなる．また，Q 値の標準偏差に関してもわずかに大きい．これは，学習過程にエージェントの往来が発生したためであると考えられる．相対ベクトルルールで用いる相対ベクトルは，2 試行前のベクトルを参照するため，2 試行前と同じ位置にいる場合には参照することができない．このとき，エージェントのルール系列に往来が発生した場合，相対ベクトルによる識別ができずに無効ルールも含めたルール系列すべてに報酬が与えられる．この結果，往来する行動選択を学習過程で多く行い，Q 値が低くなったと考えられる．

相対ベクトルルールを導入した場合に，実験環境 4 では環境ルールと異なり安定して荷物搬送を行えた結果について考察する．第 3 章では環境ルールを用いた実験環境 4 の結果では，図 4.4 に示すように荷物排出量と差分平均が安定しない．一方で，相対ベクトルルールを導入した結果では，図 4.5 に示すように，Agent1 の差分平均が低いが，環境ルールと比較して標準偏差が低くなっている．環境ルールでは，Agent3 が同様に差分平均が低く，この影響により行動が安定しない．このため，相対ベクトルルールを導入すれば，低い差分平均でもその場にとどまり続けるようになる．こうなる理由として，相対ベクトルルール導入によるベースルールの複雑化が考えられる．環境ルールでは，視界環境のみで行動選択の選択を行うため，視界内にエージェントが映らなくなる状態，すなわち他のエージェントが荷物を搬送しているタイミングでその場から移動し，結果として他のエージェントを妨害してしまう．相対ベクトルルールを用いる場合は，エージェント自身が来た方向を認識できるため，特定の方向から進行した場合に阻害され，停止行動を選択した場合でも他のエージェントの移動による環境変化に対応できると考えられる．また，2 ステップ間の相対ベクトルを用いることによる自身の位置推定も要因であると考えられる．エージェントが他のエージェントに阻害され，停止行動を選択する場合は，その後の行動が停止もしくは往来を選択しやすい．これは，視界内に他のエージェントが映り続けていることが多いためである．この停止および往来の行動は，2 ステップ間の相対ベクトルとしてみた場合に差分が 0 であるため，他のエージェントの行動による視界環境が変化した場合でも自身が同じ位置にいることを認識できる．この相対ベクトルの差分からも，その場にとどまり続けることができ，他のエージェントを妨害しない行動選択を行えていると考えられる．このため，この行動選択は他のエージェントに対する利他的行動のように推察される．

第5章　停止行動と利他的行動に対する考察

5.1.　停止行動導入による振る舞い

停止行動を導入した場合，第3章および第4章の実験によりエージェントの振る舞いには2種類の行動が確認された．本節では，これら2種類のエージェントの振る舞いについて考察する．

1種目は，視界に他のエージェントが映る場合に，視界にエージェントが映らなくなるまで停止行動を選択し続けるものである．これは，他のエージェントが進行不可のオブジェクトと認識されているため，衝突回避や迂回のため停止行動を選択する．また，学習中のエージェントは視界に他のエージェントが映らないように行動選択を行いやすいため，視界にエージェントが映らない行動選択として停止行動を選択しやすい．

2種類目は，第4章の実験環境4の結果で現れた，他のエージェントに行動を譲るための停止行動である．この停止行動は，1種目の停止行動と異なり，視界内にエージェントが映らない場合にも停止行動を選択し続ける．このため，この停止行動は他のエージェントに対する利他的行動になりうる．

また，停止行動を導入することにより ϵ-greedy 方策が有効に機能しなくなることがある．これは，ϵ-greedy 方策が貪欲行動により多くの確率で Q 値の最も大きい行動を選択するが，停止行動が最大 Q 値であった場合は視界環境が遷移するか確率 ϵ のランダム行動を行うまでその場にとどまり続けるからである．このため，学習初期に報酬獲得までのルール上で停止行動が行われていた場合，それ以降の学習に遅れが生じる．また，確率 ϵ のランダム行動により他のエージェントによる妨げを受けやすく，停止行動をとらなければならない状態が発生しやすい．よって，停止行動を導入する場合には Softmax 方策を行動選択手法に選ぶことが必要といえる．

5.2.　利他的行動に対する考察

第4章における相対ベクトルルールを導入した実験環境4の結果では，Agent1が停止し続けることによる荷物搬出の安定化が確認できた．この Agent1 の停止行動は，利他的行動であるかを内部 Q 値から考察する．相対ベクトルルールを用いた実験環境4の結果において，荷物搬送を行えるエージェントと荷物搬送を行えないエージェントのルール内の各ステップごとの最大 Q 値を比較したグラフを図5.1に示す．図5.1で示しているのは，試行ステップ数が480,000回以降のルールである．荷物搬送を行えた場合，ステップ長が18であり，最大 Q 値が表3.1に示す $Q = 20.0$ に近くなる．このため，学習が正しく行えており，おおよそ割引率 $\gamma = 0.95$ がステップごとに割り引かれている．一方，荷物搬送

を行えないエージェント，ここでは停止や往来を選択し続けるエージェントではステップ長が 20 であり，ステップ間が 12 から 14 で一定値をとっている．この一定値をとっている位置は，図 4.6 においてエージェントが停止行動をとる位置とその周辺である．また，一定値をとっているステップの前後はより低い Q 値となっている．多くの場合で，エージェントの学習では Q 値のより高い方へ行動選択を行うため，この周辺で停止か往来を選択し続ける．すなわち，あるエージェント 1 体が他のエージェントに進路を譲り続けるのは，利他的行動ではなく停止行動や往来行動の最大 Q 値がその他行動と比較して高いからであると推察される．この行動を選択し続けるのは，第 4 章で述べたように，相対ベクトルルールを導入することによるルールベースの複雑化と相対ベクトルの差分が 0 となることによるものと考えられる．

以上から，従来研究における利他的行動獲得例との比較を行う．Zamora らの研究 [43] では，あるエージェント 1 体が互恵的利他行動を学習すれば，チーム全員が利他的行動をとると述べている．また，Maeedi らの研究 [59] では，エージェント間に情報共有を導入し，郡内エージェントの是正策を講じることで，学習の遅れているエージェントを他のエージェントが引き付ける相互利他性を可能としている．しかし，本実験におけるエージェントは 1 体のみが通信や情報共有を行わず犠牲となり，他のエージェントが利他的行動を選択していない．このため，本研究におけるエージェントは互恵的利他行動の学習を行っていないと推察される．

上田らの研究 [60] では，強化学習においてエージェントが利他的行動を獲得するのは，要因として適度な憐憫悲 (ここでは他のエージェントの不幸を悲しむ感情) が寄与するとされている．特に，他のエージェントのからの憐憫悲を負の報酬として学習を行う場合に利他的行動を獲得できるとしている．本研究における実験条件にこのような感情要素を導入しておらず，罰などの負の報酬を導入していないため，他のエージェントの行動評価による行動選択を行っているといえない．

以上の考察より，相対ベクトルルールを導入した場合のエージェントの停止行動は，利他的行動ではないが，それに準ずるような行動選択であることが確認できた．この結果から，情報共有および感情の寄与を行わない利他的行動獲得の要因として，相対ベクトルルールなどの，ルールベースに対する条件付けを行うことが有効であると考えられる．

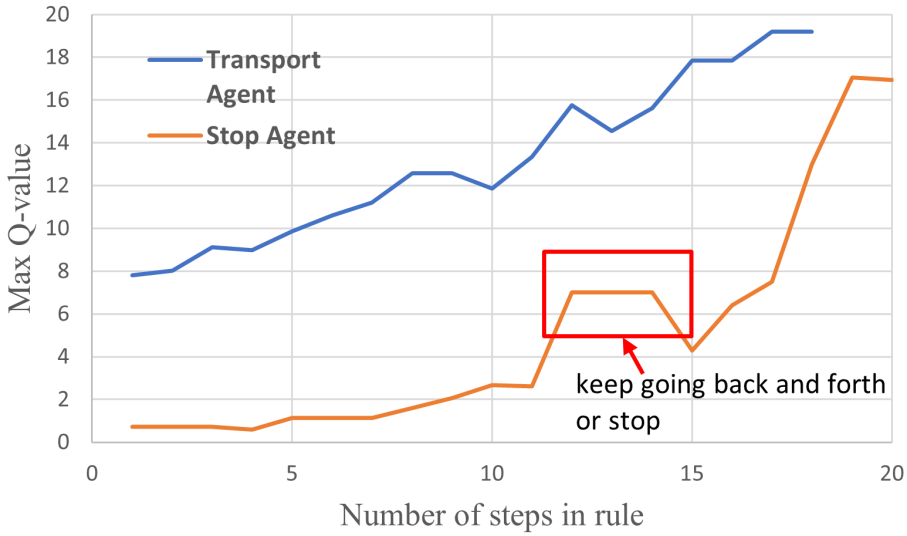

図 5.1: 荷物搬送を行えるエージェントと行えないエージェントのルール内ステップと最大 Q 値

第6章　結論

6.1.　研究成果のまとめ

本研究では，マルチエージェントシステムを用いた荷物搬送問題において，強化学習を用いることによる荷物搬送の自律化を目的とし，これを行うための環境別の比較と各手法，ルールの検証を行った．各環境での実験結果では，Softmax 方策が実験の収束性に秀でているが，実験環境 2 や実験環境 4 などのタスク環境が狭い環境では，エージェント数が多い場合に学習が困難となることが検証できた．また，エージェントの振る舞いに関しては，停止行動を導入することによる待機のような行動が確認できたほか，相対ベクトルルール導入下では利他的行動に似た振る舞いを行うことを検証できた．組み合わせ方策を用いた学習では，エージェント数 2 体の環境では Softmax 方策に劣る荷物搬出量であったが，エージェント数が 3 体での実験環境 2 の結果では，Softmax 方策より多い荷物搬出を行うことができた．

ルールベース別の学習では，環境ルールが学習困難であった実験環境 4 に対し，相対ベクトルルールを用いることで学習の安定化を図ることができた．この学習の安定化は，ルールベースの複雑化と相対ベクトルの差分が 0 となることによるものと考えられるため，利他的行動ではないと考えられるが，実質的に他のエージェントに対しての利他的行動となっている．このため，相対ベクトルルールを導入することにより，タスク環境の狭い実験環境 4 のような環境でも，マルチエージェントシステムを用いた強化学習による学習の安定化を図ることができた．一方で，実験環境 1 のような学習しやすい環境に対して，環境ルールよりも低い荷物搬出量となった．これは，ルールベースの複雑化および往来に対する無効ルールを抑制していないためであると考えられる．

6.2.　今後の研究課題と展望

今後の研究課題として，報酬と罰の与え方によるエージェントの振る舞い検証が挙げられる．本研究では，エージェントにいかなる状況であっても罰を与えずに学習を行った．エージェントに罰を与えない場合，学習に適さない状態への遷移を多く行うため第 3 章におけるエージェントが 3 体以上の場合における実験環境 2 のように，ほとんどの結果で学習が安定しない．このため，エージェントが環境及び他のエージェントに対して望ましいと思われない行動に対して罰を与え，無効ルールの抑制を行う必要がある．同様に，相対ベクトルを導入した手法は，実験環境 1 のような比較的簡易な環境において従来手法と比較して荷物搬送効率が低下したため，提案手法に罰および無効ルール抑制を行う必要がある．また，提案手法の用いる参照ベクトル先の検証が必要である．本研究では，2 ステップ前と現在ステップとの相対ベクトルを用いたが，その他ステップと現在のステップとの相対ベクトルを用いた実験を行っていない．このため，より多くの実験環境およびパラ

メータ設計で学習を進めていく必要がある．また，本研究では，実験環境 4 のような荷物搬入口と搬出口周辺が狭い環境に対し，提案手法によりある程度学習が収束することが分かった．しかし，実験環境 4 における荷物排出量の理論値は 1000 ステップあたり 250 回であり，提案手法では 120 回程度の排出量となった．このため，より正確な環境の同定による荷物排出量の向上が課題となる．

今後の展望として，階層追記型のルールベースを用いた学習による視界範囲の拡大が考えられる [38]．階層追記型の学習であれば，Q 値テーブルを従来手法より多く参照させることが可能となるため，より広い視界範囲による学習を行えると考えられる．これにより，従来手法と比較してより多くの環境同定が可能となり，従来手法では困難であった実験環境 3 のような環境に対して有効な学習が行えると推察される．また，本研究では Q-PSPLearning を用いて行ったが，相対ベクトルルールでの Profit Sharing Plan で無効ルールを抑制しながら学習を行うことにより，環境ルールと比較して実験環境 1 のように荷物搬出量が低下した環境に対して最適行動を獲得できると考えられる．

謝辞

本研究を行うにあたり，ご多忙の中数々の御指導をしてくださいました，高知工科大学システム工学群電子・光システム工学教室 星野孝総准教授に感謝の意を表し，心からお礼申し上げます．また，修士論文を執筆するにあたりご意見を頂きました，高知工科大学システム工学群電子・光システム工学教室小林弘和准教授，山本真行教授に深く感謝いたします．さらに，本研究に関して日頃から様々な協力や相談を受けてくださいました，高知工科大学システム工学群 Soft Intelligent System on Chip 研究室の皆様にもこの場を借りてお礼申し上げます．最後になりましたが，大学・大学院生活 6 年間を支えてくれた家族に深く感謝申し上げます．

参考文献

[1] Yu, Zhuang Fu Yan-zheng ZhaoQing-xiao , Can Yuan: "Research of the localization of restaurant service robot", *International Journal of Advanced Robotic Systems*, Vol. 7, No. 3, p. 18, 2010.

[2] 田口冬樹ほか: " 流通イノベーション研究: アマゾンの成長過程と競争優位の源泉", 専修経営学論集, Vol. 108, pp. 41–76, 2019.

[3] Takeshi Shimmura, Takashi Okuma Hiroyuki Ito-Kei Okada Tomomi Nonaka, Ryosuke Ichikari: "Service robot introduction to a restaurant enhances both labor productivity and service quality", *Procedia CIRP*, Vol. 88, pp. 589–594, 2020.

[4] Wiederhold Gio, Feigenbaum Ed, McCarthy John: "Arthur Samuel: pioneer in machine learning", *Communications of the ACM*, Vol. 33, No. 11, pp. 137–139, 1990.

[5] Sutskever Ilya Oriol Vinyals, Quoc V. Le: "Sequence to sequence learning with neural networks", *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[6] 鈴木大慈: " 機械学習の概要", 応用数理, Vol. 28, No. 1, pp. 32–37, 2018.

[7] Akkus Zeynettin, Hoogi Assaf Rubin Daniel L Erickson Bradley J, Galimzianova Alfiia: "Deep learning for brain MRI segmentation: state of the art and future directions", *Journal of digital imaging*, Vol. 30, pp. 449–459, 2017.

[8] Sohn Kihyuk, Carlini Nicholas Zhang Zizhao Zhang Han Raffel Colin A Cubuk Ekin Dogus Kurakin Alexey Li Chun-Liang, Berthelot David: "Fixmatch: Simplifying semi-supervised learning with consistency and confidence", *Advances in neural information processing systems*, Vol. 33, pp. 596–608, 2020.

[9] Kaelbling Leslie Pack, Moore Andrew W, Littman Michael L: "Reinforcement learning: A survey", *Journal of artificial intelligence research*, Vol. 4, pp. 237–285, 1996.

[10] Sutton Richard S, Barto Andrew G: "Reinforcement learning: An introduction". MIT press, 2018.

[11] 大槻知史: " 最強囲碁 AI アルファ碁 解体新書 深層学習, モンテカルロ木探索, 強化学習から見たその仕組み". 翔泳社, 2017.

[12] 野田五十樹: " マルチエージェント社会シミュレーションが浮き彫りにする緊急時避難の課題", 学術の動向, Vol. 23, No. 3, pp. 3_42–3_47, 2018.

[13] 鈴木雄太, 糸井川栄一: ” 地震火災における延焼予測のばらつきに対して安全な避難経路の最適化-不完全情報下におけるリアルタイム避難誘導のための提案”, 地域安全学会論文集, Vol. 33, pp. 175–185, 2018.

[14] 浅田稔: ” 強化学習の実ロボットへの応用とその課題 (<特集>強化学習)”, 人工知能, Vol. 12, No. 6, pp. 831–836, 1997.

[15] Akihiko Yamaguchi, Tsukasa Ogasawara, Jun Takamatsu: ”Constructing action set from basis functions for reinforcement learning of robot control”, *2009 IEEE International Conference on Robotics and Automation*, pp. 2525–2532. IEEE, 2009.

[16] Argall Brenna D, Veloso Manuela Browning Brett, Chernova Sonia: ”A survey of robot learning from demonstration”, *Robotics and autonomous systems*, Vol. 57, No. 5, pp. 469–483, 2009.

[17] Kalinovcic Luka, Bogdan Stjepan Bobanac Vedran, Petrovic Tamara: ”Modified Banker’s algorithm for scheduling in multi-AGV systems”, *2011 IEEE International Conference on Automation Science and Engineering*, pp. 351–356. IEEE, 2011.

[18] Xu Qinghong, Koenig Sven Ma Hang, Li Jiaoyang: ”Multi-goal multi-agent pickup and delivery”, *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9964–9971. IEEE, 2022.

[19] 望月優加理, 澤田賢治, 新誠一: ” 分散協調制御システムに基づく迷路探索問題における通信手段と探索効率の関係”, システム制御情報学会論文誌, Vol. 32, No. 3, pp. 101–112, 2019.

[20] Si Jennie, Wang Yu-Tsung: ”Online learning control by association and reinforcement”, *IEEE Transactions on Neural networks*, Vol. 12, No. 2, pp. 264–276, 2001.

[21] Watkins Christopher JCH, Dayan Peter: ”Q-learning”, *Machine learning*, Vol. 8, No. 3-4, pp. 279–292, 1992.

[22] 清本盛明, 亀井且有: ” 部分観測マルコフ決定過程における位置ベクトルを用いた強化学習手法の提案”, システム制御情報学会論文誌, Vol. 14, No. 2, pp. 86–91, 2001.

[23] Asadi Kavosh, Littman Michael L: ”An alternative softmax operator for reinforcement learning”, *International Conference on Machine Learning*, pp. 243–252. PMLR, 2017.

[24] 河合宏和, 辰巳昭治, 上野敦志: ” ルーレット選択を用いた Profit Sharing 強化学習における合理性についての一考察”, 人工知能学会全国大会論文集 第 19 回全国大会 (2005), pp. 56–56. 一般社団法人 人工知能学会, 2005.

[25] 野田五十樹: ” マルチエージェント学習下における温度パラメータの調節手法”, 人工知能学会全国大会論文集 第 25 回全国大会 (2011), pp. 1G11–1G11. 一般社団法人 人工知能学会, 2011.

[26] Tokic Michel, Palm Günther: ”Value-difference based exploration: adaptive control between epsilon-greedy and softmax”, *Annual conference on artificial intelligence*, pp. 335–346. Springer, 2011.

[27] 宮崎和光, 山村雅幸, 小林重信: ” 強化学習における報酬割当ての理論的考察”, 人工知能, Vol. 9, No. 4, pp. 580–587, 1994.

[28] 大内東: ” マルチエージェントシステムの基礎と応用”. コロナ社, 2002.

[29] 遠藤理平: ” 倒立振子の作り方 ゼロから学ぶ強化学習”. カットシステム, 2019.

[30] J, Grefenstette John: ”Credit assignment in rule discovery systems based on genetic algorithms”, *Machine Learning*, Vol. 3, No. 2, pp. 225–245, 1988.

[31] 荒井幸代, 宮崎和光, 小林重信: ” マルチエージェント強化学習の方法論: Q-learning と profit sharing による接近”, 人工知能, Vol. 13, No. 4, pp. 609–618, 1998.

[32] 植村渉, 辰巳昭治: ”Profit Sharing 法における強化関数に関する一考察”, 人工知能学会論文誌, Vol. 19, No. 4, pp. 197–203, 2004.

[33] Grondman Ivo, Lopes Gabriel AD Babuska Robert, Busoniu Lucian: ”A survey of actor-critic reinforcement learning: Standard and natural policy gradients”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 42, No. 6, pp. 1291–1307, 2012.

[34] 荒井幸代: ” マルチエージェント強化学習: 実用化に向けての課題・理論・諸技術との融合 (<特集>「マルチエージェント技術における新しい可能性」)”, 人工知能, Vol. 16, No. 4, pp. 476–481, 2001.

[35] 山村雅幸, 宮崎和光, 小林重信: ” エージェントの学習 (<特集>「エージェントの基礎と応用」)”, 人工知能, Vol. 10, No. 5, pp. 683–689, 1995.

[36] Yang Yaodong, Li Minne Zhou Ming Zhang Weinan Wang Jun, Luo Rui: ”Mean field multi-agent reinforcement learning”, *International conference on machine learning*, pp. 5571–5580, 2018.

[37] Wanasinghe Thumeera R, Gosine Raymond G, Mann George KI: ”Distributed leader-assistive localization method for a heterogeneous multirobotic system”, *IEEE Transactions on Automation Science and Engineering*, Vol. 12, No. 3, pp. 795–809, 2015.

[38] Yukinobu Hoshino, Katsuari Kamei, Akira Sakakura: ”A proposal of the learning system using the recordable multi-layer type rule base and its application for the fire panic problem”, *Proceedings of the 2006 International Conference on Game Research and Development*, pp. 137–140, 2006.

[39] 横尾, 北村, 泰彦, 大阪市立大学ほか: ” 淘汰を用いたマルチエージェント実時間探索の高速化: 協調探索への競争の導入”, コンピュータソフトウェア, Vol. 14, No. 4, pp. 379–387, 1997.

[40] Francis, Bach: ”Breaking the curse of dimensionality with convex neural networks”, *The Journal of Machine Learning Research*, Vol. 18, No. 1, pp. 629–681, 2017.

[41] 望月優加理, 澤田賢治: ” ヘテロジニアスなエージェント群のグラフ探索における役割交換と探索効率の関係”, システム制御情報学会論文誌= Transactions of the Institute of Systems, Control and Information Engineers, Vol. 34, No. 10, pp. 269–278, 2021.

[42] 白川英隆, 木村元, 小林重信: ” 強化学習による協調的行動の創発に関する実験的考察”, 知能システムシンポジウム資料, Vol. 25, pp. 119–124, 1998.

[43] Zamora Javier, Murciano Antonio, Millán José R: ”Learning and stabilization of altruistic behaviors in multi-agent systems by reciprocity”, *Biological cybernetics*, Vol. 78, No. 3, pp. 197–205, 1998.

[44] Sneyd James, Bonabeau EricVE Deneubourg Jean-Louis Franks Nigel R, Theraula Guy: ”Self-organization in biological systems”. Princeton university press, 2001.

[45] 松野文俊: ” 群行動の理解と群ロボット研究”, 日本ロボット学会誌, Vol. 35, No. 6, pp. 428–431, 2017.

[46] 柴田克成, 上田雅英, 伊藤宏司: ” 強化学習による個性・社会性の発現・分化モデル”, 計測自動制御学会論文集, Vol. 39, No. 5, pp. 494–502, 2003.

[47] 宮崎和光: ” 離散マルコフ決定過程における強化学習”, 日本ファジィ学会誌, Vol. 9, No. 4, pp. 447–450, 1997.

[48] 牧野貴樹, 澁谷長史, 白川真一浅田稔, 麻生英樹, 荒井幸代, 飯間等, 伊藤真, 大倉和博, 黒江康明, 杉本徳和, 坪井祐太, 銅谷賢治, 前田新一, 松井藤五郎, 南泰浩, 宮崎和光, 目黒豊美, 森村哲郎, 森本淳, 保田俊行, 吉本潤一郎: ” これからの強化学習”. 森北出版, 2016.

[49] Shani Guy, Kaplow Robert, Pineau Joelle: ”A survey of point-based POMDP solvers”, *Autonomous Agents and Multi-Agent Systems*, Vol. 27, pp. 1–51, 2013.

[50] TJ, Spaan Matthijs: ”Partially observable Markov decision processes”. *Reinforcement learning: State-of-the-art*, pp. 387–414. Springer, 2012.

[51] 宮崎和光, 荒井幸代, 小林重信: ”POMDPs 環境下での決定的政策の学習”, 人工知能, Vol. 14, No. 1, pp. 148–156, 1999.

[52] Wurman Peter R, Mountz Mick, D’Andrea Raffaello: ”Coordinating hundreds of cooperative, autonomous vehicles in warehouses”, *AI magazine*, Vol. 29, No. 1, pp. 9–9, 2008.

[53] Zhang Guoxian, Qian Ming, Ferrari Silvia: ”An information roadmap method for robotic sensor path planning”, *Journal of Intelligent and Robotic Systems*, Vol. 56, pp. 69–98, 2009.

[54] 山内智貴, 宮下裕貴, 菅原俊治ほか: ” マルチエージェント搬送問題のためのグラフ理論を活用したデッドロック回避手法の提案”, 研究報告知能システム (ICS), Vol. 2022, No. 4, pp. 1–7, 2022.

[55] Jiang Jianxun, Xin Jianbin: ”Path planning of a mobile robot in a free-space environment using Q-learning”, *Progress in artificial intelligence*, Vol. 8, pp. 133–142, 2019.

[56] 堀内匡, 藤野昭典, 片井修, 椹木哲夫: ” 経験強化を考慮した Q-Learning の提案とその応用”, 計測自動制御学会論文集, Vol. 35, No. 5, pp. 645–653, 1999.

[57] 宮崎和光, 荒井幸代, 小林重信: ”Profit Sharing を用いたマルチエージェントと強化学習における報酬配分の理論的考察”, 人工知能, Vol. 14, No. 6, pp. 1156–1164, 1999.

[58] De Ryck Matthias, Debrouwere Frederik, Versteyhe Mark: ”Automated guided vehicle systems, state-of-the-art control algorithms and techniques”, *Journal of Manufacturing Systems*, Vol. 54, pp. 152–173, 2020.

[59] Maeedi Ali, İrfanoğlu Bülent, Khan Muhammad Umer: ”Reciprocal altruism-based path planning optimization for multi-agents”, *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–9. IEEE, 2022.

[60] 上田祐彰, 谷澤俊彰, 高橋健一, 宮原哲浩: ” マルチエージェントシステムにおける利他的な行動規則の獲得”, 電子情報通信学会論文誌 D, Vol. 88, No. 9, pp. 1278–1286, 2005.