目次

第 1 章　はじめに 1
1.1　背景 1
1.2　従来研究 1
1.3　研究目的 2
1.4　本論文の構成 2

第 2 章　強化学習 3
2.1　強化学習概要 3
2.1.1　強化学習とは 3
2.1.2　強化学習の利点 3
2.1.3　強化学習の構成要素 4
2.1.4　強化学習の流れ 5
2.1.5　マルコフ決定過程 5
2.2　強化学習（学習部) 6
2.3　強化学習（行動選択部） 8
2.3.1　greedy 法 8
2.3.2　ε-greedy 法 8
2.3.3　追跡手法(pursuit 法) 9
2.4　強化学習の問題点 10

第 3 章　確率表現を用いた報酬非依存型知識の提案に関する先行研究 13
3.1　報酬非依存型知識概要 13
3.2　報酬非依存型知識の定義 14
3.2.1　定義：報酬非依存型知識 14
3.2.2　定義：知識テーブル 15
3.3　強化学習における報酬非依存型知識の利用方法 16
3.3.1　強化学習における報酬非依存型知識の利用：アプローチ 16
3.3.2　強化学習における報酬非依存型知識の利用：流れ 17
3.4　報酬非依存型知識の獲得 18
3.5　報酬非依存型知識の利用 19
3.5.1　報酬非依存型知識の利用の流れ 19
3.5.2　報酬非依存型知識の利用における価値関数の更新 21
3.6　報酬非依存型知識を用いた動的環境への対応 22

3.6.1　動的環境とは 22
3.6.2　報酬非依存型知識による環境変化の認識と対応 23
3.7　先行研究の問題点 24

第 4 章　報酬非依存型知識の確率化の提案 27
4.1　確率的報酬非依存型知識の概要 27
4.1.1　確率的報酬非依存型知識とは 27
4.1.2　確率的報酬非依存型知識利用の流れ 27
4.2　報酬非依存型知識の定義 29
4.2.1　定義：確率的報酬非依存型知識 29
4.2.2　定義：知識テーブル 30
4.3　確率的報酬非依存型知識の獲得 31
4.4　確率的報酬非依存型知識の利用 33
4.4.1　確率的報酬非依存型知識の利用の流れ 33
4.4.2　確率的報酬非依存型知識の利用における価値関数の更新 35

第 5 章　実験 37
5.1　実験目的 37
5.2　実験概要 37
5.1.1　概要 37
5.1.2　実験環境 37
5.1.3　エージェントの設定 38
5.3　実験 39
5.3.1　実験環境 39
5.3.2　実験パラメータ設定 41
5.3.3　結果 42
5.3.4　考察 54

第 6 章　結論 55
6.1　まとめ 55
6.2　今後の課題 55

参考文献 57
謝辞 58

第 1 章　はじめに

1.1　背景

現在，ロボットは様々な形で社会に普及している．例えば，掃除ロボットや受付嬢ロボット, エンターテイメントロボットなどあらゆる分野でロボットは活躍している[1]-[3].

数十年前までは，ロボットは工場など変化が少ない環境において稼働していた．さらにロボットの動作も同じ動作を繰り返し行うといった単純なものであり，特定の用途でのみ用いられていた．このような変化が単純の環境であれば，設計者がロボットの直面する環境を想定して，動作を設計することが可能であった．しかし，現在ロボットの普及に伴い，ロボットが直面する環境が複雑なものとなった．例えば，家庭内で動作するロボットを考える．周辺の家具の位置や，床に散らばっているおもちゃなど物体はその時々で場所を変える．また，子供やペットといった，常に動き回っているものもある．このような環境は動的かつ複雑な環境であり，設計者がロボットの直面する環境を予測し，動作を設計するのは不可能である．このような複雑な環境下に対応するための方法の一つとしてロボット自身が学習を行う機械学習がある[4].

機械学習とは人間が以前の経験を生かし環境に適応していくように，ロボットにも状況に合わせた行動を取れるように知能を持たせる方法である．機械学習には，ニューラルネットワークや遺伝的アルゴリズム，強化学習[5]などがある．中でも強化学習は実ロボットに適用が多い手法として注目されている．強化学習は実ロボットの行動獲得[6]以外にも建築の分野にも応用されている[7].

強化学習とは，試行錯誤により徐々に環境に適応していく機械学習の一種である．強化学習は報酬と呼ばれるスカラ値を用いて学習する．ロボットは行動を取ることでその行動に見合った報酬が得られる．

人間社会へのロボットの普及によって，人間はロボットに多様な仕事を求めるようになった．ある特定の環境下で 1 つの目的を達成するのではなく，変化のある環境下で複数の目的を達成するロボットを求めている．強化学習ではある目的に対してのその目的の達成方法しか学習しない．そのため，目的が変更されると前の目的に対しての学習結果により，効率的に学習できないといった問題点がある．また，強化学習では完全に学習するまでに時間がかかってしまうという問題点もある．

1.2　従来研究

1.1 節で述べたような強化学習の問題点に対応するために，従来研究では階層型強化学習[8]やファジィ推論を用いた強化学習[9]，目的に依存しない知識を強化学習に適用させる研究などがある．本研究では「強化学習における報酬非依存型知識の利用」[10]を先行研究と

して扱う．

1.3 研究目的

報酬非依存型知識は環境遷移に関する情報を定義したものである．具体的には，エージェントが認識する状態と行動，行動の結果の遷移先によって構成されている．この状態行動対と遷移先は 1 対 1 対応である．報酬非依存型知識は環境が変化した際に以前獲得した知識を捨て，実際の行動結果を新しい報酬非依存型知識として上書きする．しかし，環境の変化する速さが早い場合，報酬非依存型知識を利用するときに，すでに遷移先が変わってしまい，間違った環境予測をしてしまう恐れがある．それにより，動的環境では学習効率が落ちる恐れがある．

そこで，本論文では報酬非依存型知識を確率化し，動的環境に対応することを目的とする．ここで動的環境とは環境変化が起こる環境のことを指す．

1.4 本論文の構成

以下に本論文の構成を述べる．

第 2 章では，強化学習の基本的な概念や手法について述べる．さらに強化学習の問題点についても述べる．

第 3 章では，本研究の先行研究について述べる．また，動的環境についての定義を行い，先行研究の問題点を述べる．

第 4 章では，本研究の提案手法である，確率的報酬非依存型知識の定義・利用方法を述べる．

第 5 章では，4 章で述べた提案手法の有効性を確認するために実験を行う．

第 6 章では，本論文の結論及び今後の課題を記述する．

第 2 章　強化学習

2.1　強化学習概要

2.1.1　強化学習とは

強化学習は試行錯誤を通して，環境に適した行動パターンを獲得する機械学習の一種である．ロボットは周囲の環境を認識し，より良い行動を取るように学習を進めていく．

強化学習では報酬と呼ばれるスカラ値を用いて学習を行う．ロボットは行動を選択するとその結果として報酬を環境から受け取る(図 2.1)．受け取った報酬を基にロボットは選択した行動の良し悪しを判断する．この報酬はロボットに達成してほしいタスクに合わせて人間が設定する必要がある．報酬のみを与えるだけでロボットは最適な行動を獲得することができる．そのため，何が最善の行動なのかということを指定する必要はない．

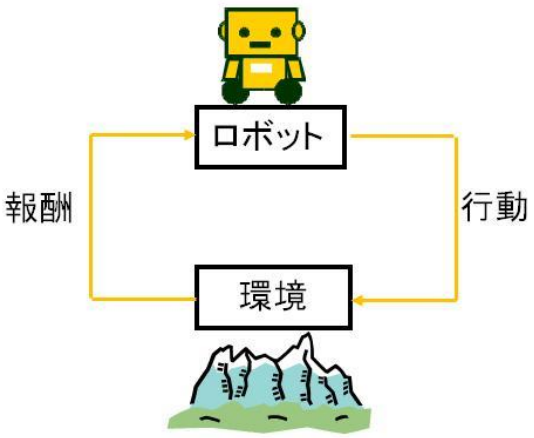


図 2.1 ロボットと報酬の関係

2.1.2　強化学習の利点

強化学習の特徴・利点を以下にあげる．

- 未知の環境を扱うことが可能

  強化学習ではロボットに報酬を与えることによって，未知の環境下でも学習できる．そのため，実ロボットの行動獲得に用いられることが多い[].

- 学習の際に教師を必要としない

  教師あり学習には，どのように行動すればよいかという学習過程を教師と与えてやることが必要である．しかし，強化学習の場合は達成すべき目的を報酬して人間が設定することで，目的達成までの行動はロボットが自動的に獲得する．

- 試行錯誤による探索

強化学習は試行錯誤によって多くの報酬を得るための最適な行動を学習する．この試行錯誤によりロボットは様々な状態を経験する．多くの状態を経験することにより最適な行動を獲得できる．

2.1.3 強化学習の構成要素

強化学習の構成要素を以下に挙げる．

- ロボット(エージェント)

学習者のことを指す．本論文では以後，エージェントと表す．エージェントはセンサを有し，そのセンサによって状態を認識することが可能である．また，エージェントは認識した状態に対して，何らかの行動を取ることができる．ただし，認識できる状態や取ることのできる行動はロボットが有するセンサやアクチュエータに依存する．

- 環境

ロボットを取り巻く環境．ロボット以外の全てから構成される．環境はいくつかの要素により構成され，その要素を認識することで状態を知覚する．

環境は静的環境と動的環境の 2 種類に分けられる．静的環境とは変化する要素がない環境のことである．動的環境とは変化する要素がない環境のことである．

- 報酬関数

報酬関数は強化学習における目的を表している．報酬関数は目的に合わせて人間が設定する．この報酬関数によりロボットにとって何が良い出来事で何が悪い出来事であるかを知ることができる．ロボットにとって唯一の目的は受け取る報酬を最大化することである．

- 価値関数

報酬関数が即時的な意味合いで何が良いのかを示しているのに対して，価値関数は最終的に何が良いのかを示す．行動価値とは，ロボットが行動を基点として，その行動以降に獲得できる報酬の期待値である．

- 学習部

受け取った報酬を基に行動の価値を評価・推定する部分である．詳しくは 2.2 節で説明する．

- 行動選択部

学習部で評価した価値を基に次に取る行動を選択する部分である．詳しくは 2.3 節で説明する．

2.1.4 強化学習の流れ

強化学習の流れを図 2.2 に示す．

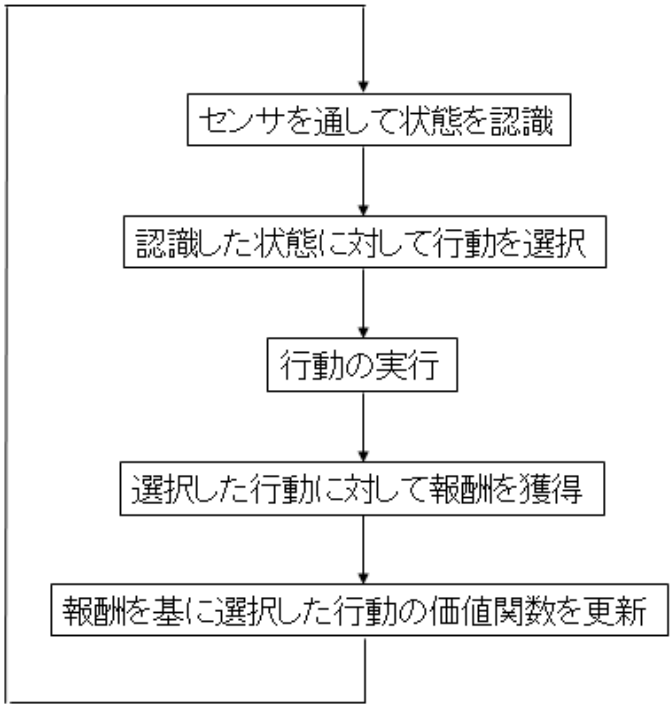


図 2.2 強化学習の流れ

エージェントはセンサを有しており，このセンサによって周囲の状態を認識する．そして，これまでの学習結果から認識した状態に対して行動を選択する．このとき，行動選択手法により選択する行動は決定される．そして，選択した行動に対して環境から報酬を受け取る．受け取った報酬を元に選択した行動に対して価値関数を更新する．学習法によって報酬からどのように価値関数を更新するかが決まる．そして，再び学習結果を用いて行動を選択する．このサイクルを繰り返すことでロボットは目的に対して最適な行動を学習する．

2.1.5 マルコフ決定過程

強化学習で扱う環境は，有限状態数のマルコフ決定過程(MDP)としてモデル化された環境である．この環境は以下のような特徴を持つ．

- 環境は状態を持ち，状態は完全に認識することが可能である．
- 状態s_{t+1}への遷移が，そのときの状態s_tと行動aにのみ依存し，それ以前の状態や行動には関係ない．
- 任意の状態s_tからスタートし，無限時間経過した後の状態分布確率は 最初の状態とは無関係である．

この環境モデルに関しては，全ての行動を十分な回数選択しさえすれば．最適解が求められることが可能であると証明されている．

マルコフ決定過程の環境では，ロボットによる環境の状態が完全であることが仮定されている．しかし，現実ではノイズやセンサの能力が不十分なため，状態認識に不確実性・不完全性がある場合が多い．部分観測マルコフ決定過程(POMDP)[11]はロボットの状態観測に不確実性を付加した数理モデルである(図 2.3)．

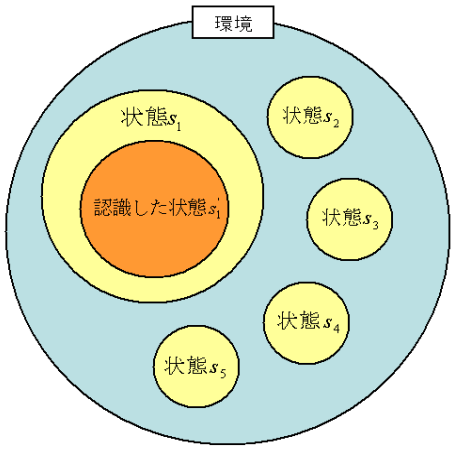


図 2.3 認識した状態と実際の状態の対応

2.2 強化学習 (学習部)

学習部とは獲得した報酬を用いて行動の価値を推定する部分である．学習部では獲得した報酬から自身の取った行動の価値を評価・推定する．獲得した報酬を用いた価値の推定方法には様々な方法が存在する．中でも有名な手法にQ学習と呼ばれるものがある．本実験ではQ学習を用いているため，以下にQ学習の詳細を述べる．

Q学習はQ値という行動価値の推定値を持つ．このQ値は各状態においてどの行動を取れば，どの程度報酬が得られるかの期待値を表している．Q値はエージェントが認識した状態とその時に取ることのできる行動の一つを対にしたものに与えられる値である．例えば，エージェントの状態をsとし，この状態で可能な行動がa_1・a_2・a_3・a_4の 4 通りあるとする．このときエージェントは 4 つのQ値，$Q(s,a_1)$・$Q(s,a_2)$・$Q(s,a_3)$・$Q(s,a_4)$

を元に取る行動を決定する(図 2.4).

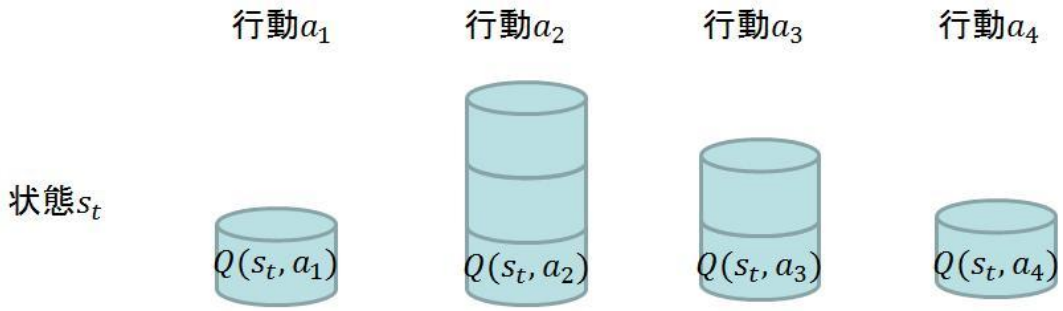


図 2.4 Q値の例

Q学習では報酬を獲得できたかどうかに関わらず，1 行動毎にQ値を更新し，学習を進める．Q値の更新は式（2.1）を用いて行う．式（2.1）は得られた報酬だけではなく，行動した先の状態が持つQ値を用いて行動の良し悪しを決めている(図 2.5).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (2.1)$$

s_tはロボットが時刻tにおいて認識した状態であり，a_tは状態s_tで取った行動を表す．$Q(s_t, a_t)$は状態s_tのとき行動を取ったときの評価値であり，更新対象のQ値を表す．s_{t+1}は状態s_tのとき行動a_tを取ったときの遷移先である．r_{t+1}は遷移先の状態s_{t+1}において得られる報酬である．$\max_a Q(s_{t+1}, a)$は遷移先の状態が持つ最大のQ値を表す．ここでaは状態s_{t+1}において最大のQ値を持つ行動を表している．αは学習率と呼ばれる定数$(0 \leq \alpha \leq 1)$であり，Q値の更新の度合いを表している．γは割引率と呼ばれる定数$(0 \leq \gamma \leq 1)$であり，将来得られると期待される報酬が現在においてどれだけの価値があるか表す．$\gamma = 0$の場合，遷移先の報酬のみでQ値を決定する．つまり，即時報酬のみを採用する．

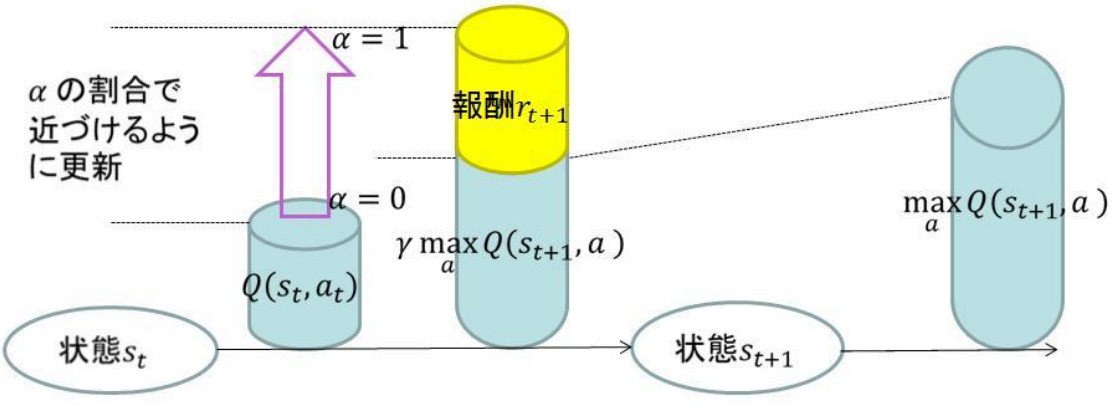


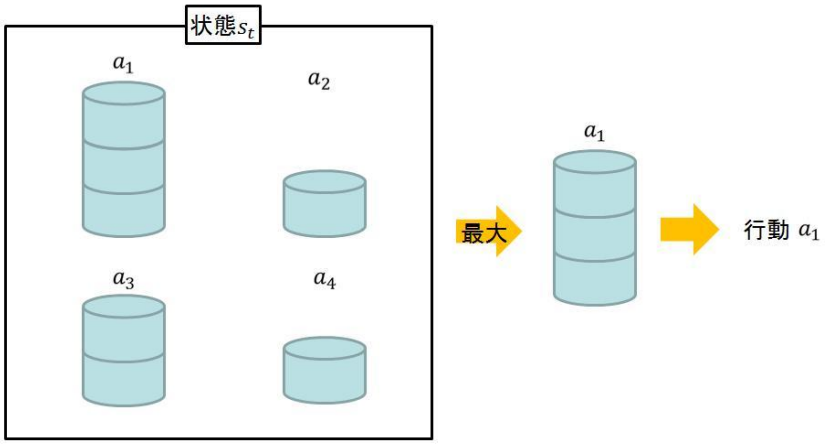
図 2.5　Q 学習における Q 値更新

2.3　強化学習（行動選択部）

この節では強化学習における行動選択部について述べる．行動選択部とは学習部で学習した Q 値(評価値)を元に適切な行動を選択する部分である．行動選択法にはここでは greedy 法，ε-greedy 法，追跡手法の 3 つの行動選択法について説明する．

2.3.1　greedy 法

greedy とは「貪欲な」という意味である．その名の通り，greedy 法では直面する状態において最も価値(評価値)が高いと評価された行動を選択する(図 2.6)．この方法は常に即時の報酬を最大にするために，現在の知識を利用する．反面，価値が低いと判断された行動に対しては，その行動がより良い行動に繋がるかもしれないという可能性を考慮しない．そのため，状態数や行動数が多い場合には局所解に陥ることがあるため，学習にはあまり向かない．



状態s：ロボットが認識した状態
行動a_1～a_4：状態sで取ることができる行動

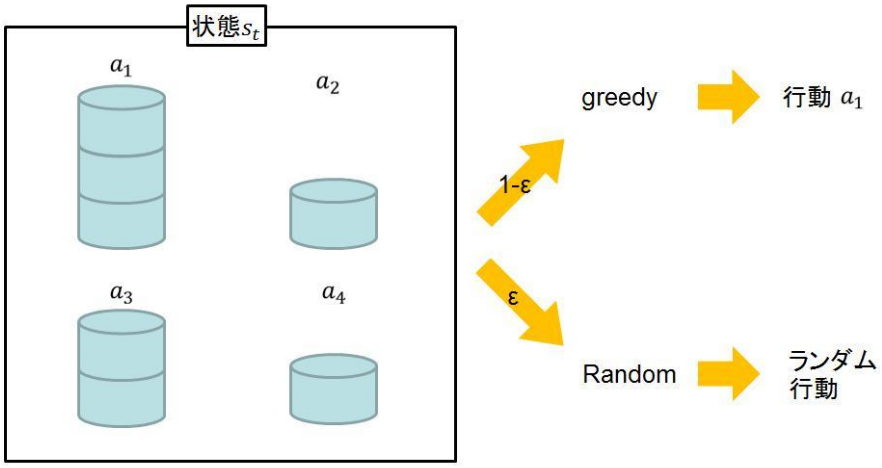

：行動の評価値

図 2.6　greedy 法

2.3.2　ε-greedy 法

ε-greedy 法とは基本的には greedy 法と同様に振舞うが，確率 ε $(0 \le \varepsilon \le 1)$ で行動の評価値の大きさに関わらずランダムに行動を選択する．逆に，$(1-\varepsilon)$ の確率で最も価値が高い

と評価された行動を選択する(図 2.7). ε の値を大きく設定すると，ランダムに行動を選択する確率が高くなり，より良い行動を探すことが多くなる．逆に，ε の値を小さくすると，学習した行動の評価値を利用することが多くなる．この ε の値は人間の手で設定される．そのため，行動の探査を中心とするか，学習した得た結果を利用するかの設計者の意図が反映させやすい．



状態s：ロボットが認識した状態
行動a_1～a_4：状態sで取ることができる行動


：行動の評価値

図 2.7　ε -greedy 法

2.3.3　追跡手法(pursuit 法)

追跡手法では状態 s において行動 a を選択する確率を表す $\pi_s(a)$ を使用する．この $\pi_s(a)$ はロボットやエージェントが行動する毎に更新される．最も価値が高い行動の選択確率が高くなるように式(2.2)により確率分布を更新する．その他の行動に関しては式(2.3)によって選択確率が低くなるように確率分布を更新する．追跡手法における選択確率の更新例を図 2.8 に示す．

$$\pi_s(a_{\max}) = \pi_s(a_{\max}) + \beta[1.0 - \pi_s(a_{\max})] \qquad (2.2)$$

$$\pi_s(a_{other}) = \pi_s(a_{other}) + \beta[0.0 - \pi_s(a_{other})] \qquad (2.3)$$

ここで $a_{\max}$ とは状態 s において Q値 が最も高い行動を表す．また，a_{other} は $a_{\max}$ 以外の行動を表す．β は確率の変動幅を決める定数で，0 以上 1 以下の値をとる．この β が大きいほど選択確率が大幅に変動し Q 値が大きい行動を選択しやすくなる．また，β の値が小さ

いほど選択確率があまり大きく変動しないため，様々な状態を経験しやすくなる．

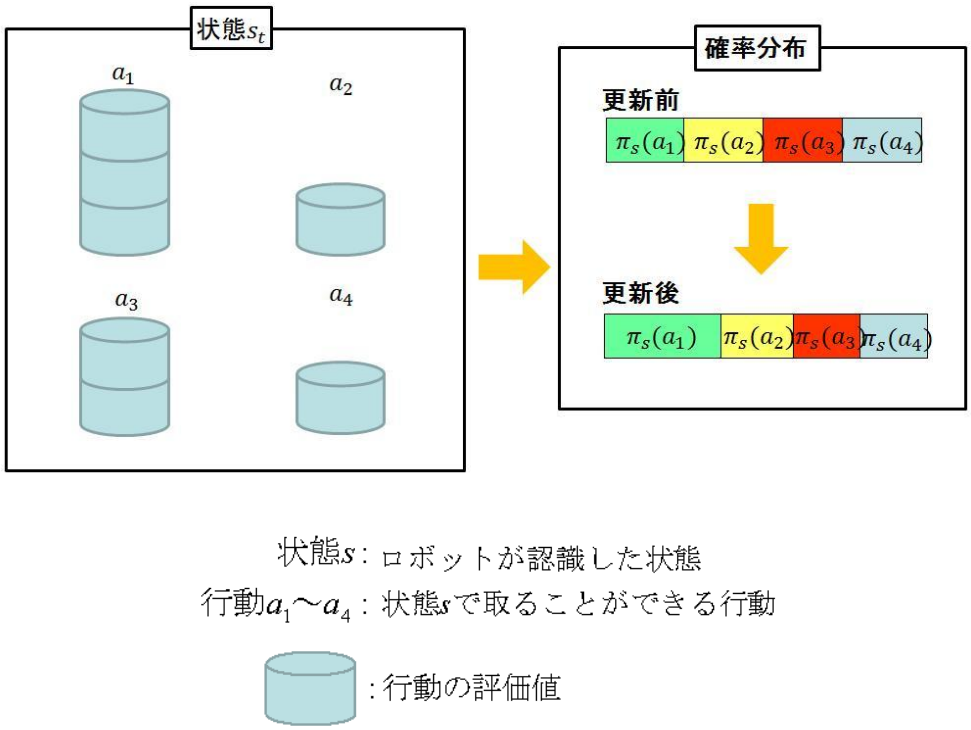


図 2.8　追跡手法における確率分布の更新例

2.4　強化学習の問題点

強化学習は報酬による学習である．報酬によって学習することにより様々な特徴・利点がある．しかし，報酬による学習という特徴によって引き起こされる問題点も存在する．いかにその問題点を述べる．

- 目的が変化すると対応が遅れることがある．

  強化学習では報酬を設定することで，ある 1 つ目的に対して試行錯誤を通して学習を行う．しかし，試行錯誤をしている間に経験したとしても，目的の達成方法以外について学習を行わない．そのため目的が変わった場合，それまでの学習で経験したとしても 1 からの学習になってしまう．また，変わる前の目的に対する学習の結果から影響を受けて新たな目的に対する学習が進みにくいことがある．

  例を図 2.9 に示す．ある場所から駅までの道をロボットが学習する場合，ロボットは駅に着くと報酬を獲得することができる．駅に着くまでの間に，ロボットはコンビニや郵便局など様々な場所を通過する．しかし，強化学習ではコンビニや郵便局などの通過した場所に対しての学習を行うことは無い．最終的に学習するのは駅に向かう道

のみである．駅への道を学習した後に，今度は郵便局への道を学習させる．よって，報酬を郵便局に着いたときのみ貰えるよう設定する．この時，ロボットは「駅に行けば報酬を貰える」と認識しているため，まず駅に向かう．しかし，駅に行っても報酬は貰えないため，郵便局への道を再度学習しなければならない．郵便局への道を学習した後に，駅への道を学習させる場合も再度学習させなければならない．このように目的が変化するような場合では過去に経験した状態であっても，再び学習しなければならない．

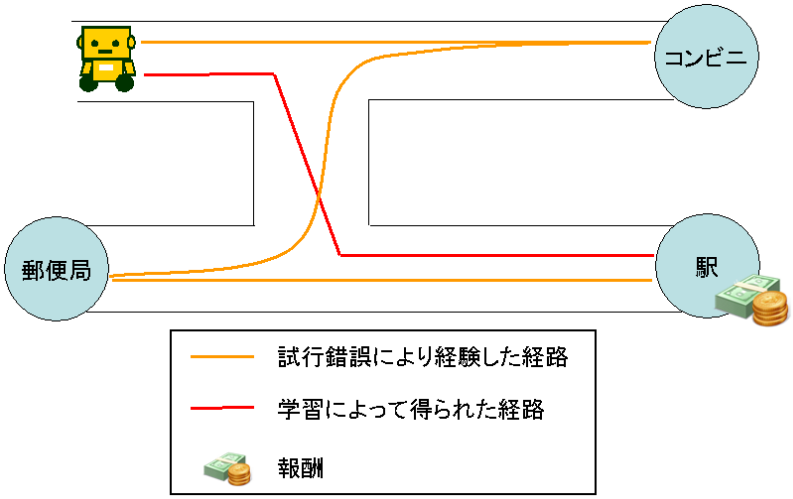


図 2.9 学習によって得られた経路

- 学習に時間がかかる

強化学習では試行錯誤を行うことで目的状態までの行動を獲得する．この時，ロボットが認識する状態が増加するほど，経験する状態が増加する．経験する状態が多いほど，様々な経験をするために試行錯誤する回数は増えてしまう．また，各状態において取ることができる行動も多いほど試行錯誤する回数は増える．このように状態が多くなったり，行動の選択肢が増えるほど学習に時間がかかる傾向にある．

ロボットが直面する状態・行動が少ない場合の例を図 2.6 に示す．また，ロボットが直面する状態・行動が少ない場合の例を図 2.7 に示す．図 2.6 の場合は，ロボットが認識できる状態は 4 種類ある．各行動において取ることのできる行動は 1 つのみである．対して，図 2.10 の場合は，ロボットが認識できる状態は 6 種類ある．各行動において取ることのできる行動は 2 つである．図 2.13 の場合は状態数と行動数が多く，試行錯誤をしなければいけない回数も増える．

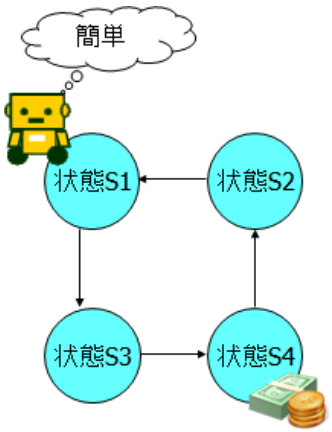


図 2.10 直面する状態・行動が少ない場合

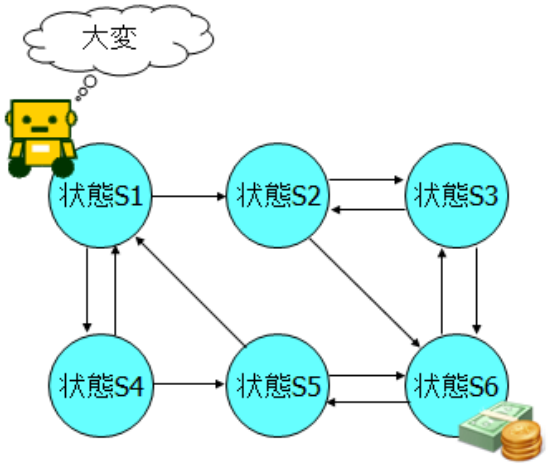


図 2.11 直面する状態・行動が少ない場合

第 3 章　確率表現を用いた報酬非依存型知識の提案に関する先行研究

2 章では強化学習は報酬によって学習を行うということを述べた．また，問題点として同一環境内でも目的が変わるような問題には対応が追い付かないことがあるということも述べた．これらの問題点を解決するために，先行研究である宮崎により「報酬非依存型知識」が提案された．3 章では，この報酬非依存型知識の定義や獲得・利用方法などを述べる．

3.1　報酬非依存型知識概要

報酬非依存型知識とは，環境遷移に関する情報のことである．具体的には，エージェントが認識する状態と，エージェントが取る行動の組み合わせからどの状態に移り変わるかという情報である．例を図 3.1 に示す．図 3.1 ではエージェントが認識する状態は「電気が消えている」という状態である．エージェントが取る行動は「電気を消す」という行動である．「電気が消えている」という状態で，「電気を消す」という行動を取ったときに，「電気がついている」という状態に移り変わる．このようにエージェントが認識した状態，エージェントが取る行動，行動によって移り変わった状態の 3 点を報酬非依存型知識として扱う．この報酬非依存型知識を獲得することで，環境遷移に関する知識を蓄える．

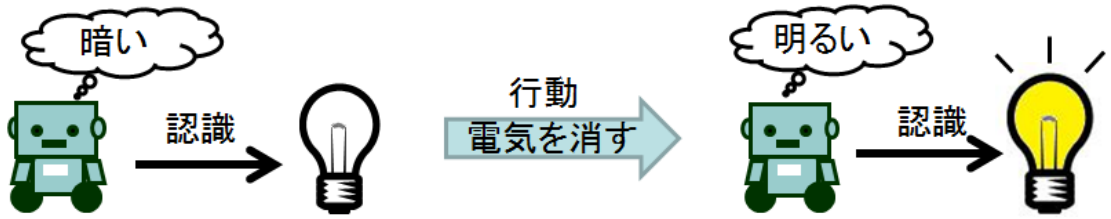


図 3.1 報酬非依存型知識として扱う情報

図 3.1 にシステムの概要図を示す. 強化学習では環境からエージェントに報酬が与えられ，自身の状態をセンサによって認識する．この時にエージェントはセンサを通して認識した状態と，エージェントが取った行動，行動結果による遷移先を報酬非依存型知識として蓄える．この報酬非依存型知識を蓄えることにより強化学習とは別に環境に対して学習を行う．蓄えた報酬非依存型知識はエージェントがどのように行動すれば目的の状態に辿り着くかを予測するために利用する．

強化学習に報酬非依存型知識を導入することにより，目的が変更されても対応が可能になる．これは報酬非依存型知識が目的に依存しない情報であるためである．強化学習はある目的の達成方法しか学習しない．そのため，目的が変更されると環境が同じでも 1 から

の学習になってしまう．しかし，報酬非依存型知識を用いることにより変更された目的までの達成方法を予測することができる．その結果，目的が変わっても予測した行動を取ることで，素早く学習することができる．

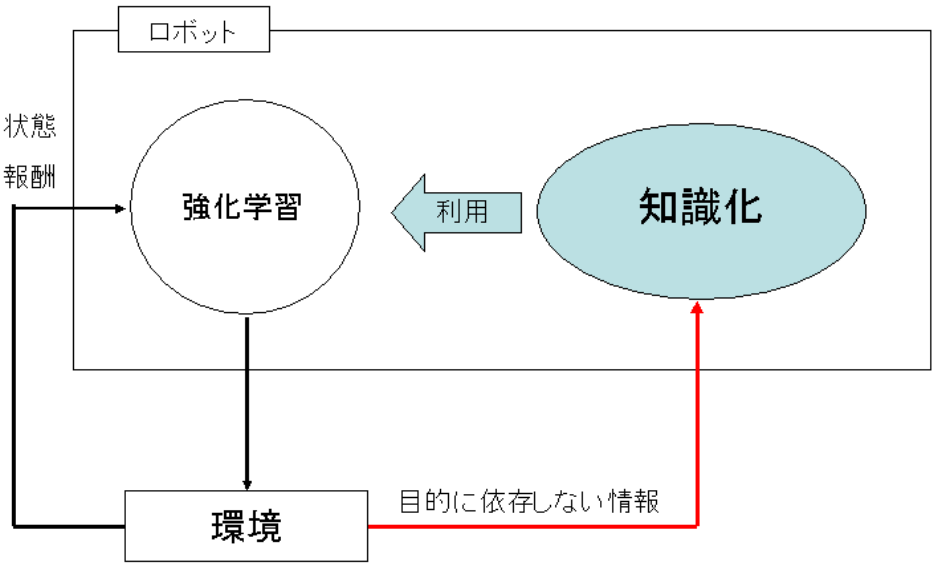


図 3.2 先行研究システム概念図

3.2 報酬非依存型知識の定義

3.2.1 定義：報酬非依存型知識

報酬非依存型知識の定義を式(3.1)に示す．

$$k_i := (s_t, a_t) \rightarrow (s_{t+1}) \qquad (3.1)$$

ここで，k_i は報酬非依存型知識を表す．s_t はエージェントが認識している状態を表す．a_t はエージェントが取った行動を表す．s_{t+1} は状態 s_t において行動 a_t を取った時の遷移先の状態を表す．この報酬非依存型知識はエージェントが認識する状態と，そのときに選択可能な行動の 1 つとの状態行動対による次の状態を確定情報として定義している．

図 3.3 に例を示す．エージェントは明るさを認識できる光センサを有しているとする．このとき，電気がついている状態を「状態 s_0」，電気が消えている状態を「状態 s_1」とする．また，電気を消す行動を「行動 a」，電気をつける行動を「行動 b」とする．この場合，報酬非依存型知識は以下の 2 つのようになる．

- $(\text{状態}s_0, \text{行動}a) \rightarrow (\text{状態}s_1)$
- $(\text{状態}s_1, \text{行動}b) \rightarrow (\text{状態}s_0)$

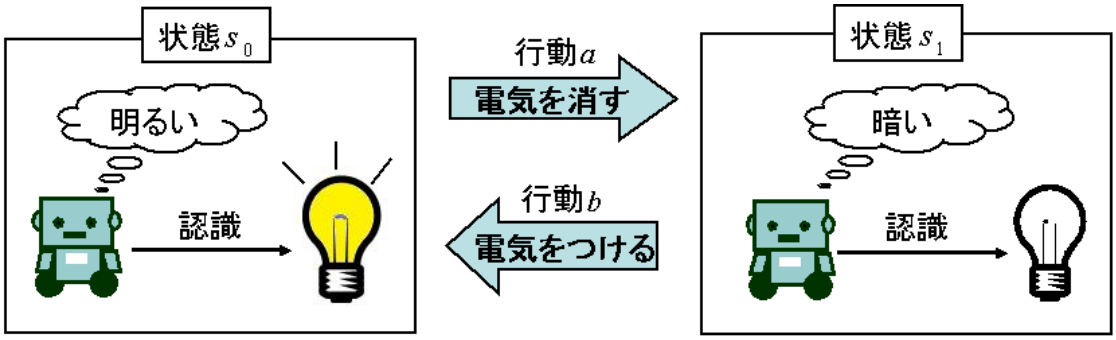


図 3.3 報酬非依存型知識として扱う情報の例

3.2.2 定義：知識テーブル

報酬非依存型知識はある状態行動対に対して 1 つ存在する．しかし，報酬非依存型知識単体では環境の一部分の情報しかない．目的までの行動を予測するためには，報酬非依存型知識が複数個必要になる．そのため，いくつもある報酬非依存型知識をエージェントが保持するためのテーブルが必要になる．報酬非依存型知識を保持するための知識テーブルを K と表し，式(3.2)により定義する．

$$K := \{k_i \mid i = 1,2,\cdots,n\} \qquad (3.2)$$

ここで，n は報酬非依存型知識の個数を表す．

図 3.4 に知識テーブルの概要図を示す．

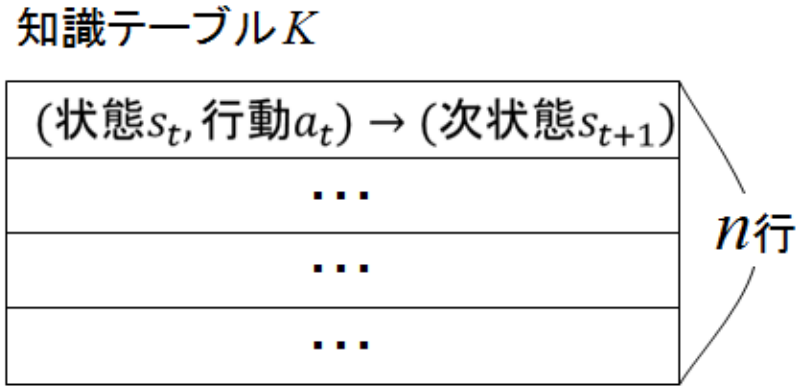


図 3.4 知識テーブルの概念図

定義した知識テーブルの特徴を以下に挙げる．

- 知識テーブルは各行に報酬非依存型知識を 1 つ保持する

  知識テーブルは各行に報酬非依存型知識を 1 つ保持する．保持する報酬非依存型知識が n 個であれば，知識テーブルの行数は n 個となる．

- 知識テーブルの行数は拡張可能である．

  定義した知識テーブルの行数はエージェント自身がいつでも拡張することができるとする．エージェントが n 個の報酬非依存型知識を持っていて，n 行の知識テーブルを持っているとする．知識テーブルにない報酬非依存型知識を追加するときに行数は $n+1$ 行となる．

- 重複する報酬非依存型知識をもたない

  知識テーブル内は重複する報酬非依存型知識は存在しない．ここで重複とは「状態」・「行動」・「次状態」の 3 つの内，全てが同じものを指す．3 つの内，1 つあるいは 2 つが同じものは別の報酬非依存型知識である．

3.3　強化学習における報酬非依存型知識の利用方法

報酬非依存型知識を利用する目的は，強化学習において目的変更に対応させることである．そのため，本節では強化学習における報酬非依存型知識の利用方法について述べる．

3.3.1　強化学習における報酬非依存型知識の利用：アプローチ

強化学習では学習の初期段階において，過去に経験した状態を何度も経験することが多い．また目的が変更された場合，前の目的で経験した状態を新たな目的のために再び経験しなければならない．このような無駄な経験をなくすためには，目的に対して取るべき行動を予測すれば良い．そこで，図 3.5 のように報酬非依存型知識を用いて目的に対して状態遷移の予測を行う．

強化学習では価値関数を基にして行動を選択する．そのため，報酬非依存型知識を強化学習に用いるときは，予測した行動を取りやすいように価値関数を更新し，強化学習の効率化を図る．

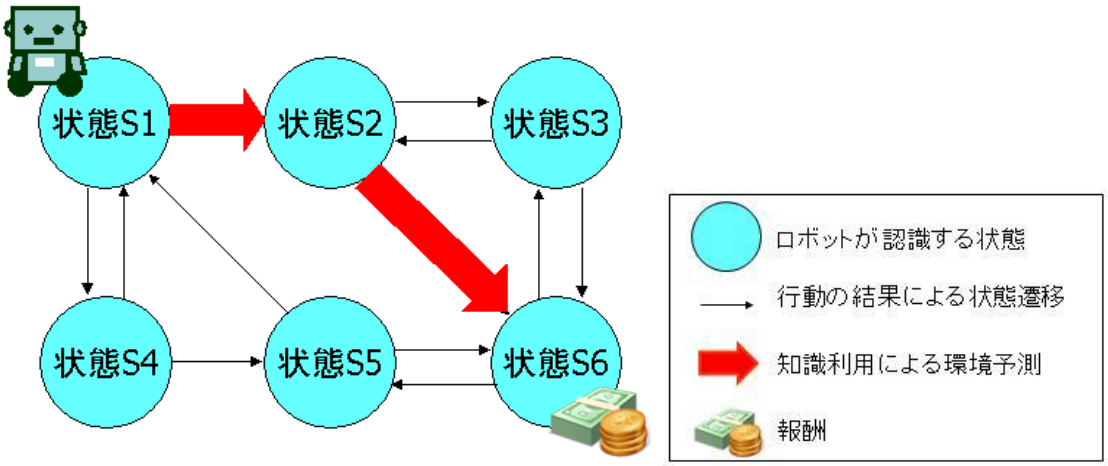


図 3.5 報酬非依存型知識の利用例

3.3.2 強化学習における報酬非依存型知識の利用：流れ

報酬非依存型知識を利用するためには大きく 2 つの部分に分かれる(図 3.6).

1 つ目は「報酬非依存型知識の獲得」である．報酬非依存型知識は状態行動対とそれによる遷移先の情報である．よって，報酬非依存型知識を獲得するためには実際に行動し，遷移先の状態を認識する必要がある．そのため，強化学習で認識した状態や行動から報酬非依存型知識を獲得する．詳細な獲得方法は 3.4 節で述べる．

2 つ目は「報酬非依存型知識の利用」である．ここでは，獲得した報酬非依存型知識を元に環境遷移を予測し，価値関数を更新する部分である．詳細な利用方法は 3.5 節で述べる．

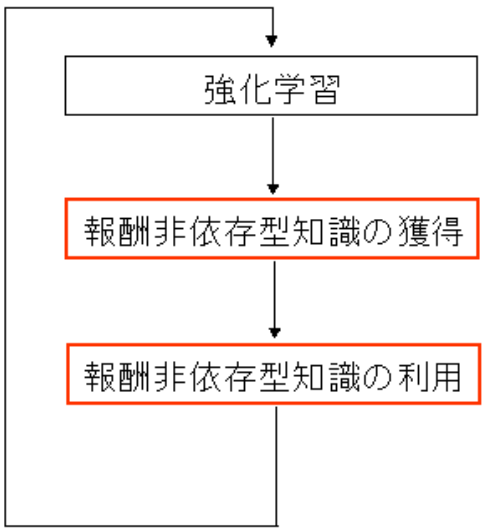


図 3.6 報酬非依存型知識の利用の流れ

3.4　報酬非依存型知識の獲得

この節では 3.2 節で定義した報酬非依存型知識の獲得方法について述べる．3.2 節でも述べたとおり，報酬非依存型知識は状態行動対とそれによる遷移先の状態を表した知識である．この情報を記憶するためには，エージェントが実際に行動する必要がある．そのため，エージェントは実際に行動し，状態を認識することで報酬非依存型知識の獲得を行う．

エージェントは行動毎に自身が持つ知識テーブルに行動による遷移情報を報酬非依存型知識として追加する．報酬非依存型知識を獲得する際に，エージェントは同じ報酬非依存型知識がないか知識テーブル内を検索する．同じ報酬非依存型知識がなかった場合には，知識テーブルの行数を 1 行増やし，報酬非依存型知識を知識テーブルに追加する．ここで，同じ報酬非依存型知識とは以下の 2 つの例のように「状態」・「行動」・「遷移先の状態」の 3 つの要素が全て同じ知識を指す．

- (状態 s_1，行動 a_1)→(状態 s_2)
- (状態 s_1，行動 a_1)→(状態 s_2)

同じではない報酬非依存型知識の例を以下に挙げる．

- (状態 s_2，行動 a_1)→(状態 s_2)
- (状態 s_2，行動 a_2)→(状態 s_3)
- (状態 s_1，行動 a_1)→(状態 s_3)

これらはすべて異なる報酬非依存型知識として認識する．

ただし，動的環境下においては状態と行動が同じであっても遷移先が異なる場合がある．その場合は，報酬非依存型知識を実際の行動結果で上書きをする．詳しくは 3.6 節で述べる．

図 3.7 に報酬非依存型知識の獲得例を示す．まずエージェントは現在の状態である「状態 s_1」を認識する．そして，「行動 a_1」を取ることで状態が「状態 s_1」から「状態 s_3」に遷移する．「状態 s_3」を認識した時点で，知識テーブルに $(s_1, a_1) \rightarrow (s_3)$ という報酬非依存型知識が追加される．

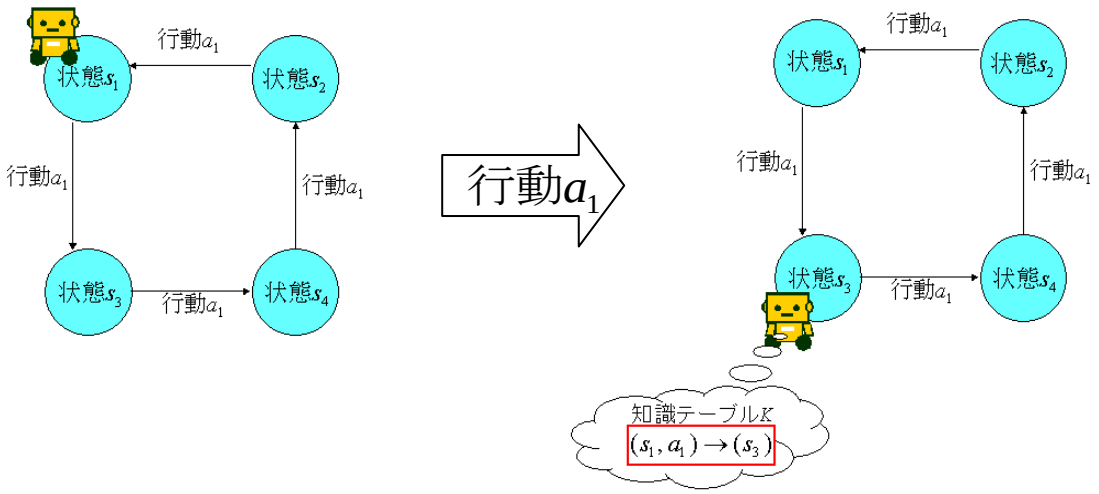


図 3.7 報酬非依存型知識の獲得例

3.5 報酬非依存型知識の利用

この節では 3.2 節で定義した報酬非依存型知識を用いてのルートの発見, 価値関数の更新方法などについて述べる.

3.5.1 報酬非依存型知識の利用の流れ

強化学習における報酬非依存型知識の利用は大きく以下の 3 ステップに分かれる.

Step1：目的状態が変化したことを知覚する

Step2：知識テーブル内で報酬を得た状態までの遷移ルートを探す

Step3：Step2 で見つけたルートに対いて価値関数の対応する部分を更新する.

Step1 でエージェントは目的状態(報酬を獲得した状態)の変化を知覚する. 目的状態の変化の認識方法は, 報酬を獲得した状態を比較することで行う. 1 試行前の報酬を獲得した状態を記憶しておき, 記憶した状態と現在の報酬を獲得した状態を比較する. 2 つを比較した結果, 状態が異なっていたら, 目的が変更したと認識する.

例を図 3.9 に示す. 図 3.9 の例でいうと, まずエージェントは N 試行目において報酬を獲得した状態を s_4 と記憶する. 次の N+1 試行目において報酬を獲得した状態を s_6 と認識する. このときに N 試行目において報酬を獲得した状態と N+1 試行目において報酬を獲得した状態を比較する. つまり, ここでは s_4 と s_6 を比較する. $s_4 \neq s_6$ であるので, エージェントは目的が変更されたと認識する.

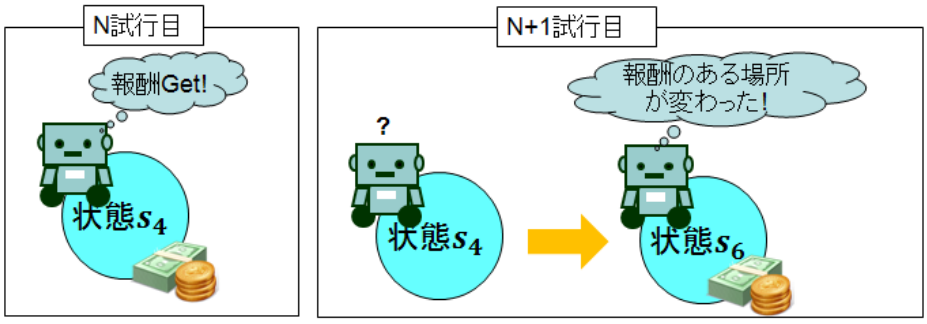


図 3.9 目的変化の認識例

Step1 で目的変化を認識した後, Step2 においてエージェントは知識テーブル内で初期状態から目的状態までの遷移ルートを探索する. はじめに目的状態を遷移先に持つ報酬非依存

型知識を探す．次に，その報酬非依存型知識が持つ状態に遷移するような報酬非依存型知識を探索する．最終的に初期状態に辿り着くか，ルート探索中に同じ状態に辿り着いたらルート探索は終了となる．

例を図 3.10 に示す．ここでは目的状態を s_6，初期状態を s_1 とする．目的状態を遷移先の状態に持つ報酬非依存型知識は $(s_2,a_1)\rightarrow(s_6)$ ・ $(s_5,a_2)\rightarrow(s_6)$ ・ $(s_3,a_1)\rightarrow(s_6)$ の 3 つが該当する． s_2 ・ s_5 ・ s_3 はいずれも初期状態ではないので，次に s_2 ・ s_5 ・ s_3 の遷移先を持つ報酬非依存型知識を知識テーブル内で探す．まず， s_2 の場合は $(s_1,a_1)\rightarrow(s_2)$ が該当する． s_1 は初期状態であるため， s_2 の場合のルート探索はここで終了となる． s_5 の場合も同様に s_5 を遷移先に持つ報酬非依存型知識を知識テーブル内で探す．この場合は，$(s_4,a)\rightarrow(s_5)$ が該当する． s_4 は初期状態ではないため，さらに s_4 を遷移先の状態に持つ報酬非依存型知識を探す．ここでは $(s_1,b)\rightarrow(s_4)$ が該当する． s_1 は初期状態であるため，s_5 の場合のルート探索はここで終了となる． s_3 の場合は，知識テーブル内を探しても， s_3 に遷移するような報酬非依存型知識は存在しない．そのため， s_3 の場合はここでルート探索を終了する．

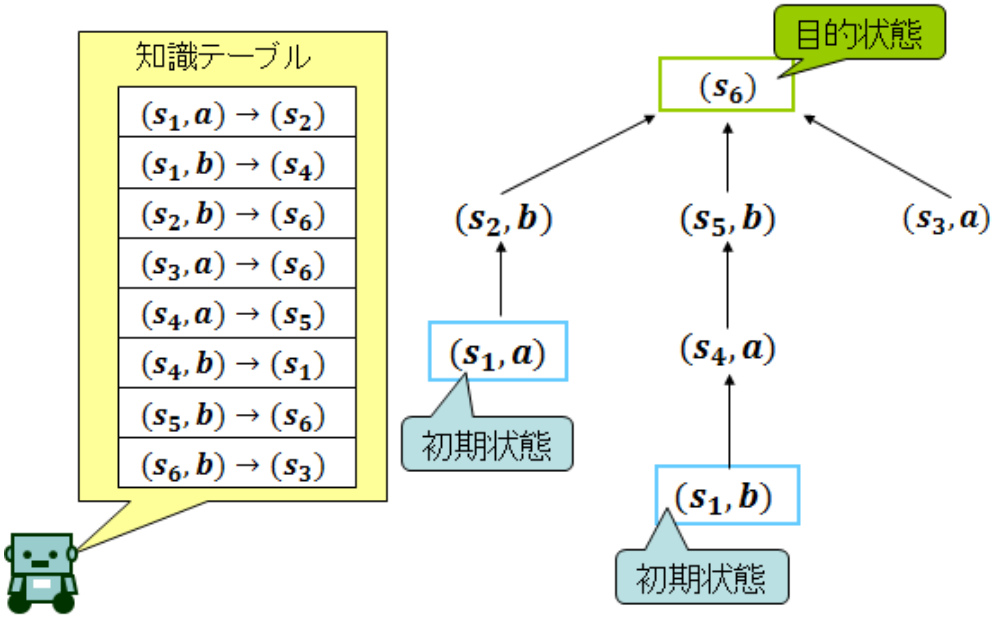


図 3.10 遷移ルートの探索例

Step3 では Step2 で見つけたルートに対して各状態行動対に対応する価値関数を更新する．この価値関数の更新は目的が変更される毎に行われる．目的が変更されなくとも，新たなルートを発見した場合も価値関数の更新が行われる．価値関数の価値関数の更新式については 3.5.2 節で述べる．

3.5.2　報酬非依存型知識の利用における価値関数の更新

3.5.1 で述べたように報酬非依存型知識の利用は 3 ステップに分かれ，Step3 において価値関数が更新されることとなる．報酬非依存型知識を用いた価値関数の更新式を式(3.1)，式(3.2)に示す．ここでは，学習部のシステムに Q 学習を用いているため，価値関数は Q 値と表す．

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + (f \times \gamma)^{d-1} \times r \times SG(s_t) \qquad (3.1)$$

$$SG(s_t) = \begin{cases} 1.0 & (s_t\text{が初期状態から目的状態のルート上}) \\ 0.5 & (\text{それ以外}) \end{cases} \qquad (3.2)$$

式(3.1)において $Q(s_t,a_t)$ は更新対象の Q 値を表す．d は状態 s_t から目的状態までの最短距離を表す．この最短距離 d は知識テーブル内の最短距離である．そのため環境の最短距離と一致するとは限らない．最短距離 d は遷移ルートによって目的状態との距離が変わる場合がある．例を図 3.11 に示す．同じ状態 s_1 でも，行動 a と行動 b を取った場合で遷移ルートが異なる．この場合，各遷移先によって最短距離 d は変わる．$s_1 \rightarrow s_2 \rightarrow s_6$ のルートの場合には目的状態 s_6 から s_1 までの距離 d は 2 となる．また，$s_1 \rightarrow s_4 \rightarrow s_5 \rightarrow s_6$ のルートの場合には目的状態 s_6 から s_1 までの距離 d は 3 となる．

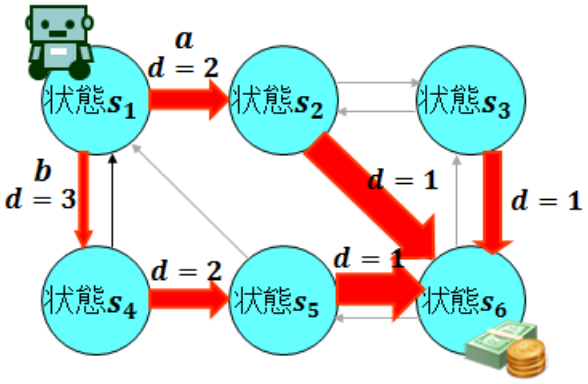


図 3.11　d の設定例

f は報酬非依存型知識の利用度を表すパラメータである．f は $0 \le f \le 1$ であり，f が大きくなるほど，強化学習における Q 値を大きく更新する．$f = 1$ に近いほど，目的に対する価値関数が学習完了時の状態に近くなるように更新する．また，f が小さいほど更新する Q 値が小さくなる．$f = 0$ の場合は報酬非依存型知識による価値関数の更新は行わず，強化学習のみの学習となる．

r は獲得した報酬を表す．

式(3.2)で示される $SG(s_t)$ は更新対象となる状態行動対の内，状態 s_t がエージェントの初期状態から目的状態までのルートの中にあるかどうかで値が変化する．このようにすることで，初期状態から目的状態までのルートは重要であるとし，それ以外のルートに対しても探索する必要性を考慮している．

式(3.1)，式(3.2)によって目的状態に近い Q 値ほど更新される値が大きくなるようになっている．また，初期状態から目的状態のルート上にある Q 値ほど大きく更新される．図 3.12 に更新の例を示す．例 3.12 に Q 値の更新例を示す．目的状態 s_6 に近い Q 値ほど大きな値で更新される．また，初期状態 s_1 から目的状態 s_6 のルート上にある Q 値は大きく更新される．(s_3, a) のように初期状態から外れたルート上にある Q 値は小さく更新される．

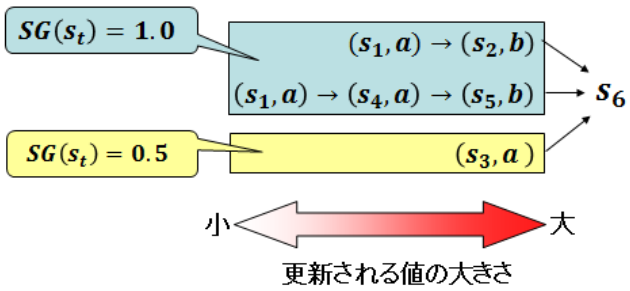


図 3.12 更新される値の例

3.6 報酬非依存型知識を用いた動的環境への対応

本節では，動的環境における報酬非依存型知識を用いた対応方法について述べる．また，本論文で扱う動的環境についても述べる．

3.6.1 動的環境とは

ここでは本論文における動的環境について記述する．本論文で取り上げる動的環境とは環境構造が変化することを指す．環境構造の変化とは各状態行動対による遷移先が変わることである．動的環境における環境変化の例を図 3.13 に示す．

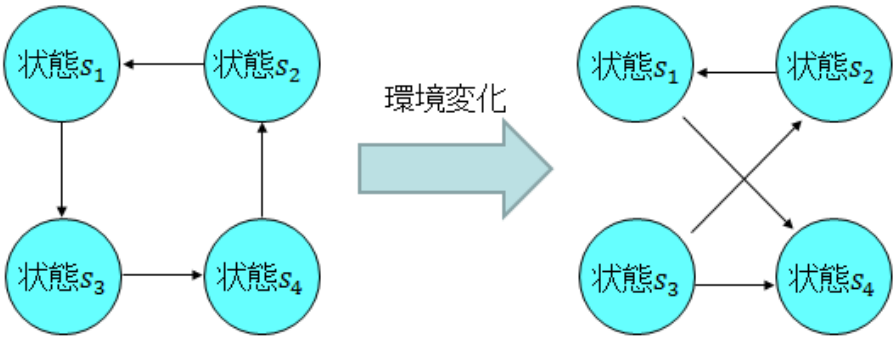


図 3.13　動的環境の例

3.6.2　報酬非依存型知識による環境変化の認識と対応

A)　認識

報酬非依存型知識を有する場合, エージェントが所持している報酬非依存型知識と実際の行動の結果を照らし合わせることで, 環境の変化を認識することができる. 比較した際に報酬非依存型と実際の行動結果が異なれば環境が変化したと認識する.

図 3.14 を例に詳しく説明をする. 図 3.14 で環境が変化する前に状態 s_1 において行動 a を取った場合，エージェントは状態 s_3 に遷移する．この時エージェントは報酬非依存型知識として「$(s_1, a) \rightarrow (s_3)$」を獲得する．その後，環境変化後に状態 s_1 において行動 a を取った場合，エージェントは状態 s_4 に遷移する．この時，同じ状態行動対 (s_1, a) の報酬非依存型知識を知識テーブル内から探し, 見つかった場合は比較を行う．図 3.14 の例では知識テーブル内に「$(s_1, a) \rightarrow (s_3)$」があるので，$(s_1, a) \rightarrow (s_3)$ と実際の行動の結果を比較する．ここでは，同じ状態行動対 (s_1, a) に対して遷移先が異なっているため，環境が変化したと認識する.

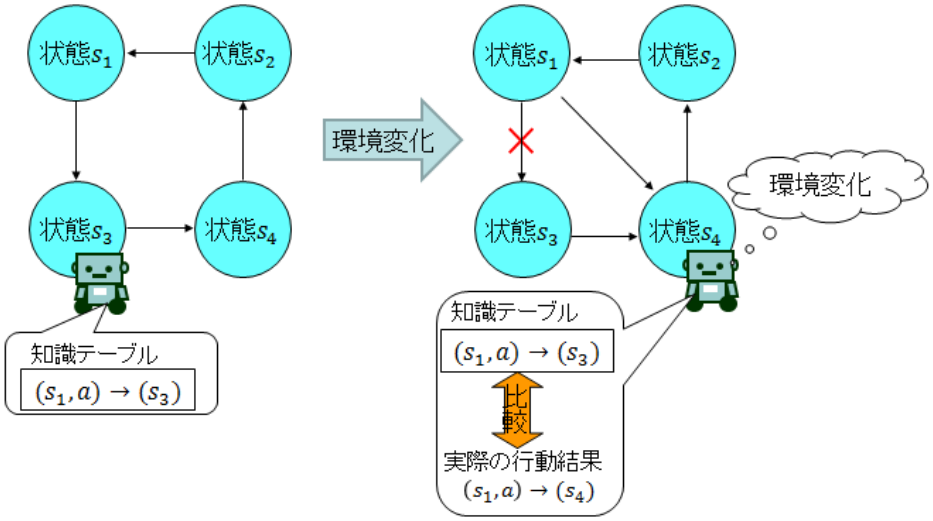


図 3.14 環境変化の認識例

B)　対応

報酬非依存型知識を用いた環境変化の認識については前述の通りである．ここでは，環境の変化を認識した後の対応方法について述べる．

環境変化を認識した後，変化があった部分の報酬非依存型知識を実際の行動結果で上書きをする．また，環境が変化したということは，目的状態(報酬を獲得する状態)までの遷移ルートも変化することになる．そのため，対応する状態行動対の価値関数を初期化することにより，環境変化後の学習を行いやすくする．

図 3.15 に環境変化を認識した際の対応例を示す．報酬非依存型知識$(s_1, a) \rightarrow (s_3)$と実際の行動結果$(s_1, a) \rightarrow (s_4)$を比較し，環境変化を認識する．その後，知識テーブル内にある 報酬非依存型知識$(s_1, a) \rightarrow (s_3)$を実際の行動結果$(s_1, a) \rightarrow (s_4)$で上書きをする．また，環境が変化した$(s_1, a)$に対応する価値関数を初期化する．

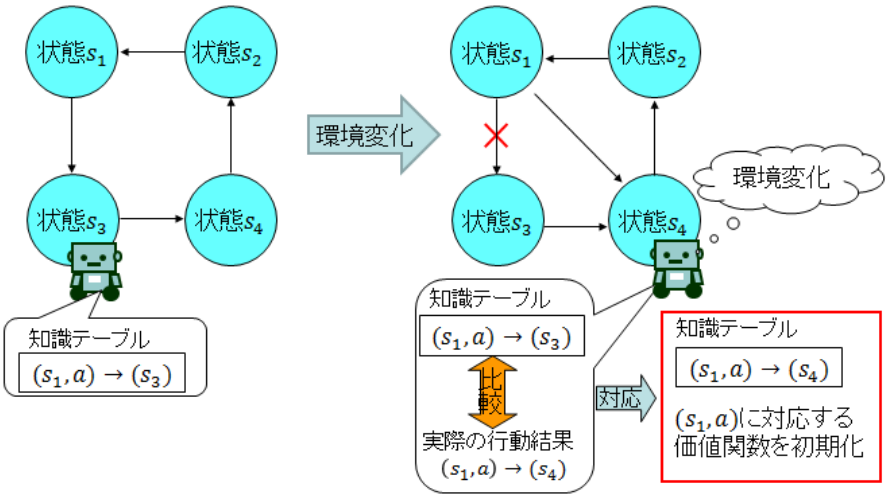


図 3.15 環境変化を認識した際の対応例

3.7　先行研究の問題点

強化学習の問題点は，報酬非依存型知識を用いたシステムを用いたことで解決された．しかし，報酬非依存型知識を用いることによって新たな問題点が生まれた．以下にその問題点を示す．

報酬非依存型知識は状態行動対とそれによる遷移先を表した情報である．報酬非依存型知識は状態行動対と遷移先が 1 対 1 対応である．そのため，3.6 節で述べたように環境変化により遷移先が変化した場合には，実際の行動結果で報酬非依存型知識を上書きする対応を取る．しかし，環境変化が早い場合にはこの方法では対応できない可能性がある．

例を図 3.16～図 3.19 に示す．図 3.16 では，初期状態は S1 とし，目的状態は状態 S5 と

する．また，エージェントは何度か行動し，すでに 5 つの報酬非依存型知識を獲得しているものとする．図 3.16 の時点ではエージェントは状態 S1 から行動 a1 を取り，状態 S2 に遷移している．状態 S2 で，エージェントが行動 a2 を取ると状態 S1 に遷移する．この時，エージェントが持つ(S2,a1)→(S6)という報酬非依存型知識が実際の行動結果である(S2,a2)→(S1)で上書きされる(図 3.17)．さらにこの後，目的状態が S5 から S6 に変化したとする．この時，エージェントは知識テーブル内から S1→S4→S5→S6 という遷移ルートを発見する(図 3.18)．ルートを発見した後に環境変化が起こり，状態 S2 において行動 a1 を取ったときの遷移先が S6 に変わったとする(図 3.19)．この時，エージェントが再び状態 S2 において行動 a2 を取らない限り，S1→S2→S6 というルートは発見できない．そのため，1 行動余分に行動することになる．このように，環境変化が起こると報酬非依存型知識では学習効率が低下する恐れがある．

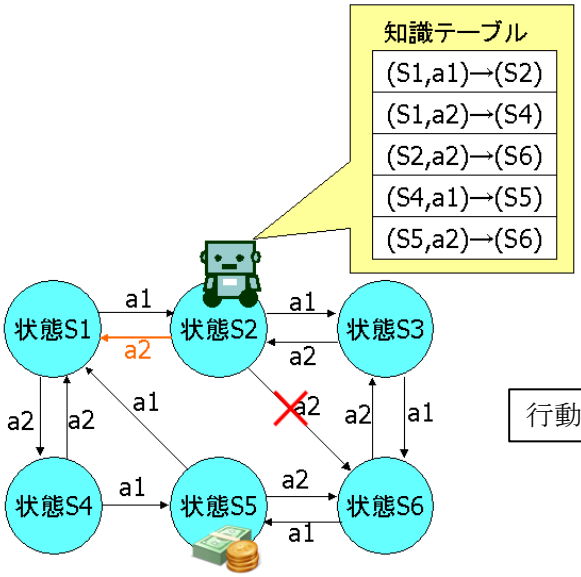


図 3.16 状態 S2

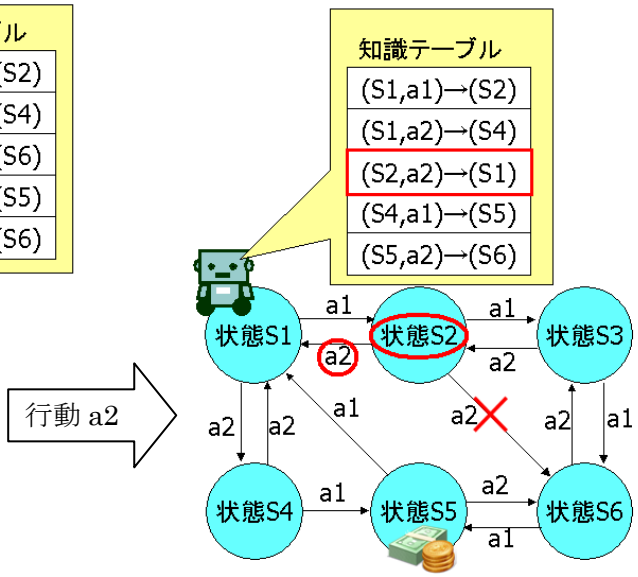


図 3.17 状態 S1 に遷移

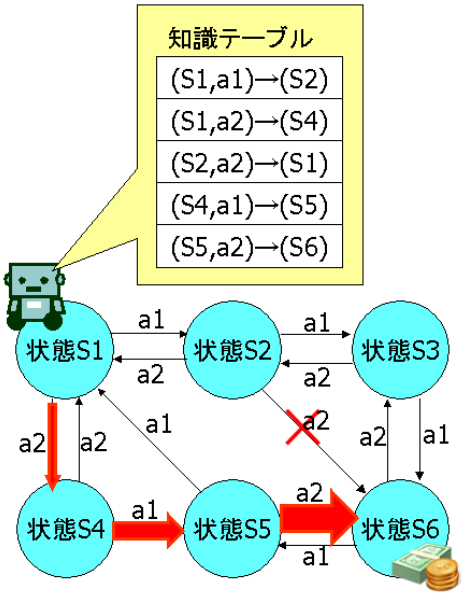


図 3.18 目的状態が S5 から S6 に変化

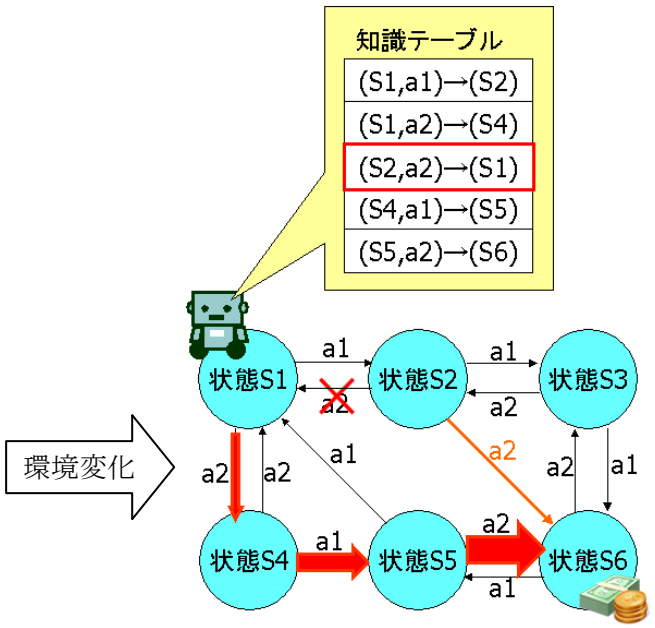


図 3.19 環境変化後

第 4 章　報酬非依存型知識の確率化の提案

3 章では宮崎によって提案された報酬非依存型知識の定義や利用方法を述べた．さらに，問題点として報酬非依存型知識を動的環境に適応した場合，学習効率が低下する恐れがあるということを述べた．これは報酬非依存型知識が状態行動対と 1 対 1 対応していることに原因がある．本論文ではこれらの問題点を解決するために報酬非依存型知識の確率化を提案する．そこで，本章では「確率的報酬非依存型知識」を定義し，獲得・利用方法を述べる．

4.1　確率的報酬非依存型知識の概要

4.1.1　確率的報酬非依存型知識とは

報酬非依存型知識は状態行動対とそれによる遷移先の状態である．この状態行動対と遷移先は 1 対 1 対応である．つまり，ある状態行動対がわかれば遷移先は一意に決まる確定的な知識である．しかし，動的環境においては同じ状態で同じ行動を取っても遷移先が異なる場合がある．そこで，報酬非依存型知識に複数の遷移先を持たせ，それぞれに遷移確率を持たせる．これを「確率的報酬非依存型知識」と定義する．図 4.1 に例を示す．エージェントは状態 s1 において行動 a を 10 回選択したとする．この時，状態 s2 に遷移する回数が 10 回中 8 回あったとする．また状態 s3 に遷移する回数が 10 回中 2 回あったとする．この場合，状態 s2 に遷移する確率は 0.8 となり，状態 s3 に遷移する確率は 0.2 となる．このように確率的報酬非依存型知識はエージェント自身の経験から遷移確率を算出する．

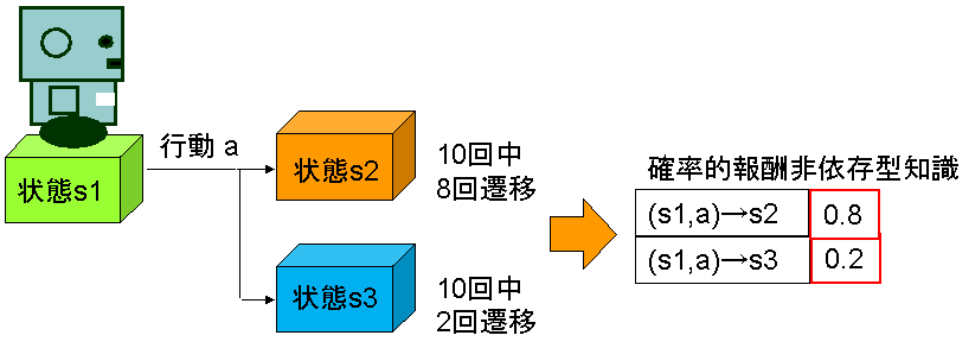


図 4.1 確率的報酬非依存型知識の例

4.1.2　確率的報酬非依存型知識利用の流れ

確率的報酬非依存型知識の利用は先行研究のシステムに沿って行う．図 4.2 に確率的報酬非依存型知識の利用の流れを示す．

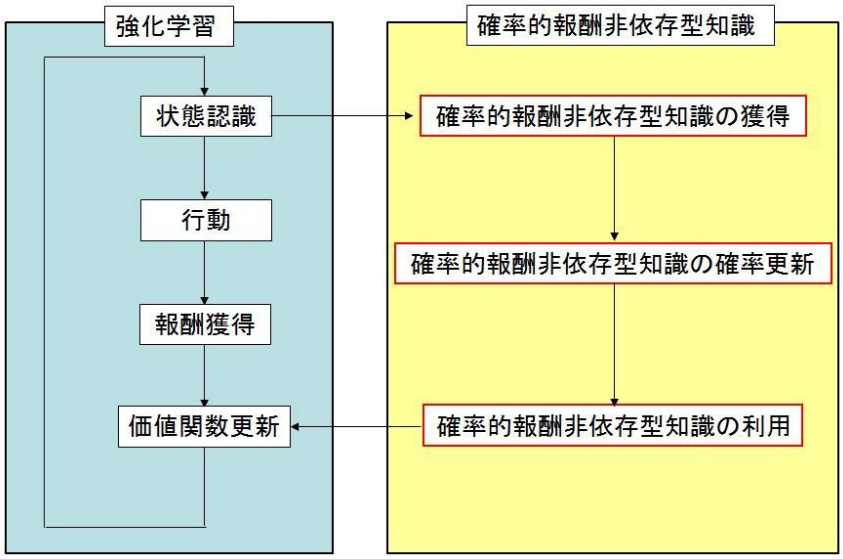


図 4.2 確率的報酬非依存型知識の利用の流れ

強化学習では「状態認識」→「行動」→「報酬獲得」→「価値関数更新」という流れにより学習を進めていく．対して確率的報酬非依存型知識は，「確率的報酬非依存型知識の獲得」→「確率的報酬非依存型知識の確率更新」→「確率的報酬非依存型知識の利用」という流れになる．確率的報酬非依存型知識はエージェントが認識した状態と行動とそれによる遷移先を確率的報酬非依存型知識として蓄える．そして，獲得した確率的報酬非依存型知識の遷移確率を更新する．この確率的報酬非依存型知識を蓄えることにより目的に対する学習である強化学習とは別に環境に対して学習を行う．蓄えた確率的報酬非依存型知識はエージェントが初期状態から目的状態(報酬を獲得できる状態)までの遷移ルートを求めるために利用する．そして，その遷移ルートに沿って行動するように，強化学習の価値関数を更新する．確率的報酬非依存型知識の利用例を図 4.3 に示す．価値関数は目的状態との距離が近いほど大きく更新する．また，遷移確率が大きいほど大きく更新する．詳しくは 4.4 節で述べる．

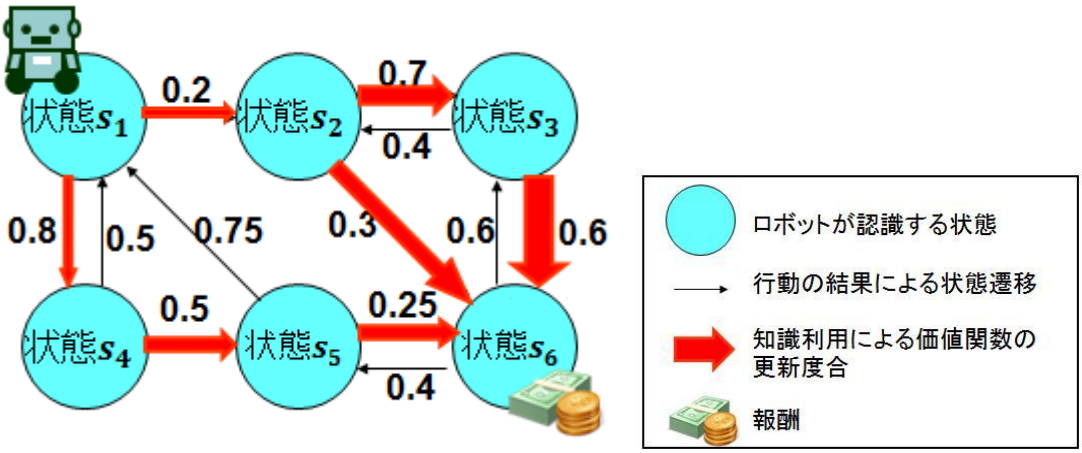


図 4.3 確率的報酬非依存型知識の利用例

確率的報酬非依存型知識の定義を 4.2 節で行う．4.3 節では確率的報酬非依存型知識の獲得方法を述べる．4.4 節では確率的報酬非依存型知識の利用方法について述べる．

4.2 報酬非依存型知識の定義

本節では確率的報酬非依存型知識を定義する．確率的報酬非依存型知識は各状態行動対による各遷移先にそれぞれ遷移確率を持つ．本節では遷移確率や確率的報酬非依存型知識を定義する．また，確率的報酬非依存型知識を集めた集合を知識テーブルと定義する．

4.2.1 定義：確率的報酬非依存型知識

状態 s_t において行動 a_t を取ったときの次状態 s' への遷移確率を式(4.1)に示す．

$$P_{ss'}^{a} = \Pr\{s_{t+1} = s' | s_t = s, a_t = a\} \qquad (4.1)$$

s_t はエージェントが認識している状態を表す．a_t はエージェントが取る行動を表す．s_{t+1} は状態 s_t において行動 a_t を取った時の遷移先の状態を表す． s は状態を表す． a は行動を表す． s' は次状態を表す．

ここではエージェントが状態 s において行動 a を選択したときに次状態 s' に遷移した回数を R とする．状態 s において行動 a を選択した回数を N とする．

$$P_{ss'}^{a} = \frac{R}{N} \qquad (4.2)$$

エージェントは各状態行動対に対して N を持ち，各状態行動対による遷移先に対して R を持つ．この遷移確率はエージェントの経験から導き出す．そのため，環境が変化する確率とは一致しない場合がある．

状態s_tにおいて行動a_t取ったときの次状態s'の遷移確率を集めた集合を式(4.3)に示す. この式(4.2)を確率的報酬非依存型知識をと定義する.

$$k_{sa} = \{P_{ss'}^{a} \mid \forall s' \in S\} \qquad (4.3)$$

ここではエージェントの取りうる状態の集合を$S = \{s_1, s_2, s_3, \cdots, s_n\}$とする. 例として図4.3に状態$s_1$と$a_1$からなる確率的報酬非依存型知識$k_{s_1a_1}$を示す. 図4.4ではエージェントがとする. この例では状態行動対(s_1, a_1)に対して, それぞれの遷移先に対して遷移確率を記憶している.

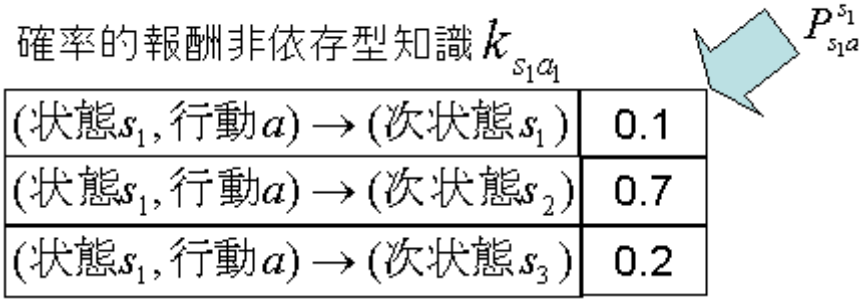


図 4.4 確率的報酬非依存型知識の例

4.2.2 定義：知識テーブル

確率的報酬非依存型知識は各状態行動対に対して, 複数の遷移先と遷移確率を持つ. しかし, 確率的報酬非依存型知識単体では, ある状態行動対に対してのみの情報である. そこで, 全ての状態行動対を集合として扱う. その集合を知識テーブルと定義する. この知識テーブルはエージェント1体につき1つの知識テーブルを持つ.

知識テーブルの定義を式(4.4)に示す.

$$K = \{k_{sa} \mid \forall s \in S, \forall a \in A\} \qquad (4.4)$$

ここでは, エージェントの取りうる状態の集合を$S = \{s_1, s_2, s_3, \cdots, s_n\}$とし, エージェントの取りうる行動の集合を$A = \{a_1, a_2, a_3, \cdots, a_n\}$とする.

図 4.5 に知識テーブルの例を示す.

知識テーブルK

(状態s_1, 行動a) → (次状態s_1)	0.1
(状態s_1, 行動a) → (次状態s_2)	0.7
(状態s_1, 行動a) → (次状態s_3)	0.2
(状態s_2, 行動a) → (次状態s_1)	0.5
(状態s_2, 行動a) → (次状態s_2)	0.25
(状態s_3, 行動a) → (次状態s_3)	0.25
・・・	

k_{s_1a}

k_{s_2a}

図 4.5 知識テーブルの例

4.3 確率的報酬非依存型知識の獲得

この節では 4.2 節で定義した確率的報酬非依存型知識の獲得方法について述べる．4.2 節でも述べたとおり，確率的報酬非依存型知識は各状態行動対に対してある状態に遷移する確率を表している．この遷移確率は式(4.2)で表す．

$$P_{ss'}^{a} = \frac{R}{N} \qquad (4.2)$$

遷移確率を算出するためには，状態 s において行動 a を選択した回数 N を記憶する必要がある．また，同時に状態 s において行動 a を選択したときに次状態 s' に遷移した回数 R も記憶する必要がある．つまり，エージェントは実際にある状態において行動し，N を記憶する．そして，遷移先の状態を認識し，その状態に遷移した回数を選択する．そして，N と R を用い，遷移確率を算出する．

確率的報酬非依存型知識の獲得例を図 4.4 に示す．まず，エージェントは現在の状態である「状態 s_1」を認識する．そして，「行動 a」を取ることで「状態 s_1」から「状態 s_2」に遷移する．「状態 s_2」を認識した時点で，確率的報酬非依存型知識を獲得する．次に獲得した確率的報酬非依存型知識の遷移確率について計算する．「状態 s」において「行動 a」を選択した回数は 1 回，「状態 s」において「行動 a」を選択したときに「状態 s_2」に遷移した回数は 1 回である．よって，(状態s_1, 行動a_1) → (次状態s_2) の遷移確率は 1.0 となる．次に「状態 s_1」から「行動 a」を取ることで「状態 s_1」から「状態 s_3」に遷移したとする．「状態 s_3」を認識した時点で，確率的報酬非依存型知識を獲得する．次に獲得した確率的報酬非依存型知識の遷移確率について計算する．「状態 s」において「行動 a」を選択した回数は 2 回，「状態 s」において「行動 a」を選択したときに「状態 s_2」に遷移した回数は 1 回である．よって，(状態s_1, 行動a_1) → (次状態s_3) の遷移確率は 0.5 となる．また，こ

の時同じ状態行動対を持つ確率的報酬非依存型知識があれば，その知識の遷移確率も更新する．図 4.4 の場合であれば，$(\text{状態}s_1, \text{行動}a_1) \rightarrow (\text{次状態}s_2)$ が該当し，遷移確率は 0.5 と更新される．

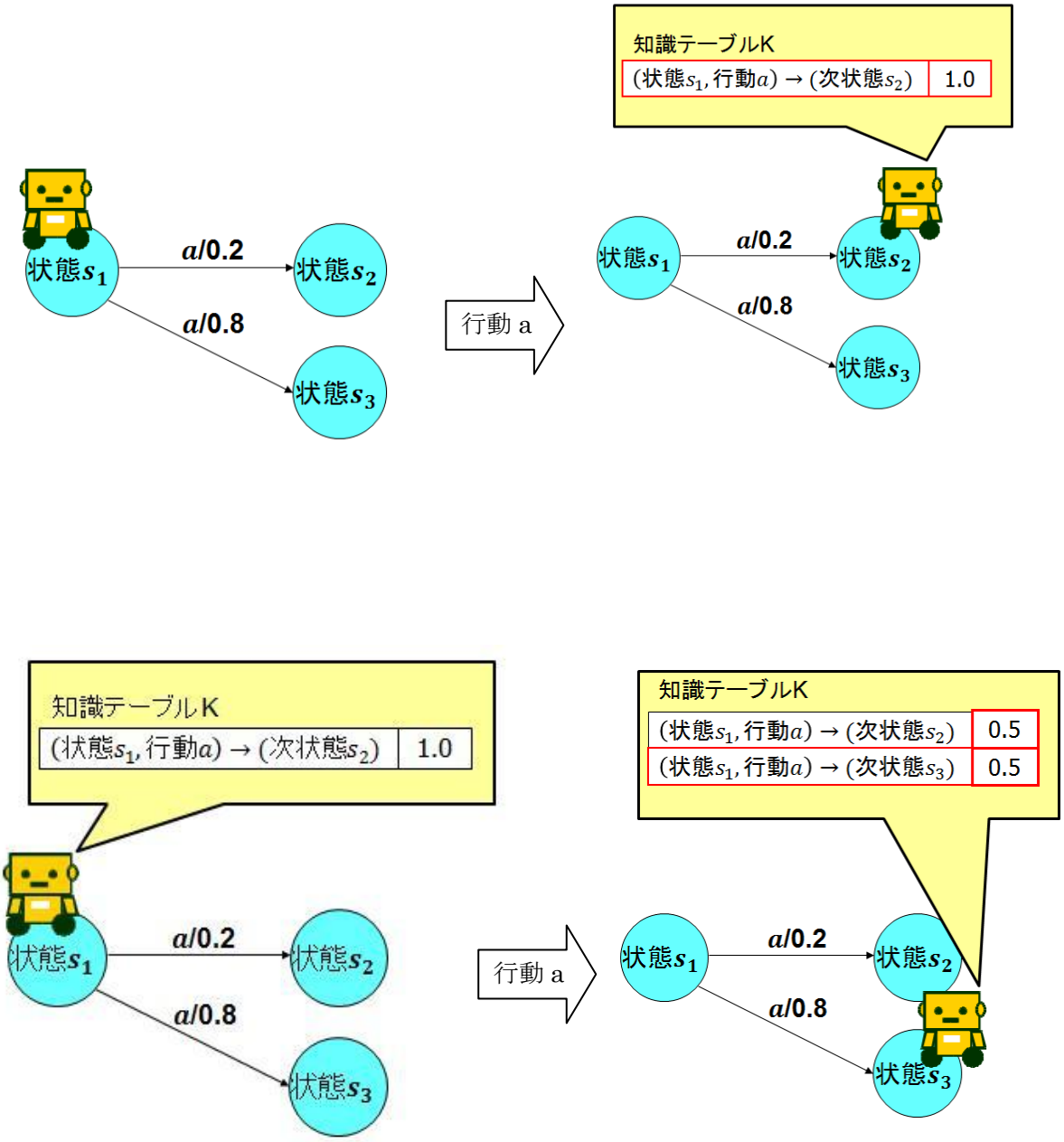


図 4.4 確率的報酬非依存型知識の獲得例

また，確率的報酬非依存型知識の獲得のアルゴリズムを以下に示す．

(1) 状態s_tを認識する
(2) 行動a_tを取る
(3) 状態遷移後の状態s_{t+1}を認識する
(4) (状態s_t,行動a_t)→(次状態s_{t+1})の知識を獲得する
(5) (状態s_t,行動a_t)の状態行動対を持つ知識の遷移確率を更新する
(6) (1)に戻る

4.4 確率的報酬非依存型知識の利用

この節では 4.2 節で定義した確率的報酬非依存型知識を用いたルートの発見・価値関数の更新方法などについて述べる.

4.4.1 確率的報酬非依存型知識の利用の流れ

強化学習における確率的報酬非依存型知識の利用は大きく以下の 4 ステップに分かれる.

Step1：目的状態が変化したことを認識する

Step2：すべての状態の価値関数を初期化する

Step3：知識テーブル内で報酬を得た状態までの遷移ルートを探す

Step4：Step2 で見つけたルートに対して価値関数の対応する部分を更新する

Step1 でエージェントは目的状態(報酬を獲得した状態)の変化を認識する．目的状態の変化の認識方法は，報酬を獲得した状態を比較することで行う．1 試行前の報酬を獲得した状態を記憶しておき，記憶した状態と現在の報酬を獲得した状態を比較する．2 つを比較した結果，状態が異なっていたら，目的が変更したと認識する.

例を図 4.9 に示す．図 4.9 の例でいうと，まずエージェントは N 試行目において報酬を獲得した状態をs_4と記憶する.次の N+1 試行目において報酬を獲得した状態をs_6と認識する．このときに N 試行目において報酬を獲得した状態と N+1 試行目において報酬を獲得した状態を比較する．つまり，ここではs_4とs_6を比較する．$s_4 \neq s_6$であるので，エージェントは目的が変更されたと認識する.

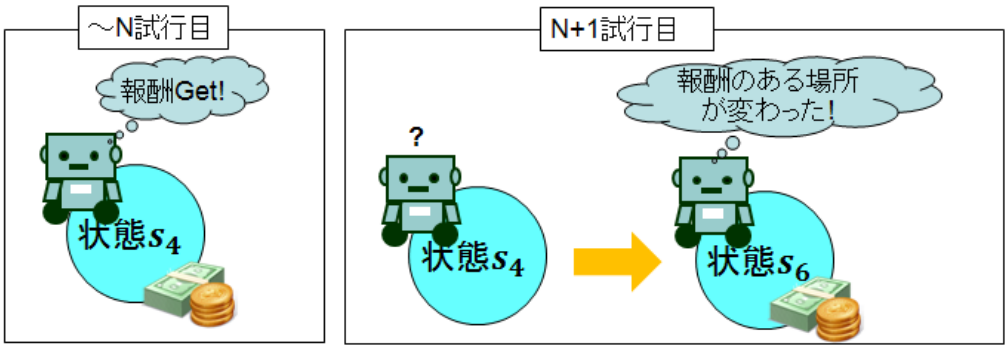


図 4.9 目的変化の認識例

Step1 で目的変化を認識した後，Step2 では価値関数を初期化する．これは目的が変更した際，前の学習結果をリセットし，再学習しやすいようにするためである．

Step2 で価値関数を初期化した後，Step3 においてエージェントは知識テーブル内で初期状態から目的状態までの遷移ルートを探索する．ルート探索はこのときに各知識が持っている遷移確率が 0 より大きいものを対象に全ルート探索を行う．はじめに目的状態を遷移先に持つ確率的報酬非依存型知識を探す．次に，その報酬非依存型知識が持つ状態に遷移するような確率的報酬非依存型知識を探索する．最終的に初期状態に辿り着くか，ルート探索中に同じ状態に辿り着いたらルート探索は終了となる．

例を図 4.10 に示す．ここでは目的状態を s_6，初期状態を s_1 とする．目的状態を遷移先の状態に持つ確率的報酬非依存型知識は $(s_2,a)\rightarrow(s_6)$ ・ $(s_5,a)\rightarrow(s_6)$ ・ $(s_3,a)\rightarrow(s_6)$ の 3 つが該当する． s_2 ・ s_5 ・ s_3 はいずれも初期状態ではないので，次に s_2 ・ s_5 ・ s_3 の遷移先を持つ確率的報酬非依存型知識を知識テーブル内で探す．まず， s_2 の場合は $(s_1,a)\rightarrow(s_2)$ が該当する． s_1 は初期状態であるため， s_2 の場合のルート探索はここで終了となる． s_5 の場合も同様に s_5 ・ s_3 を遷移先に持つ確率的報酬非依存型知識を知識テーブル内で探す．

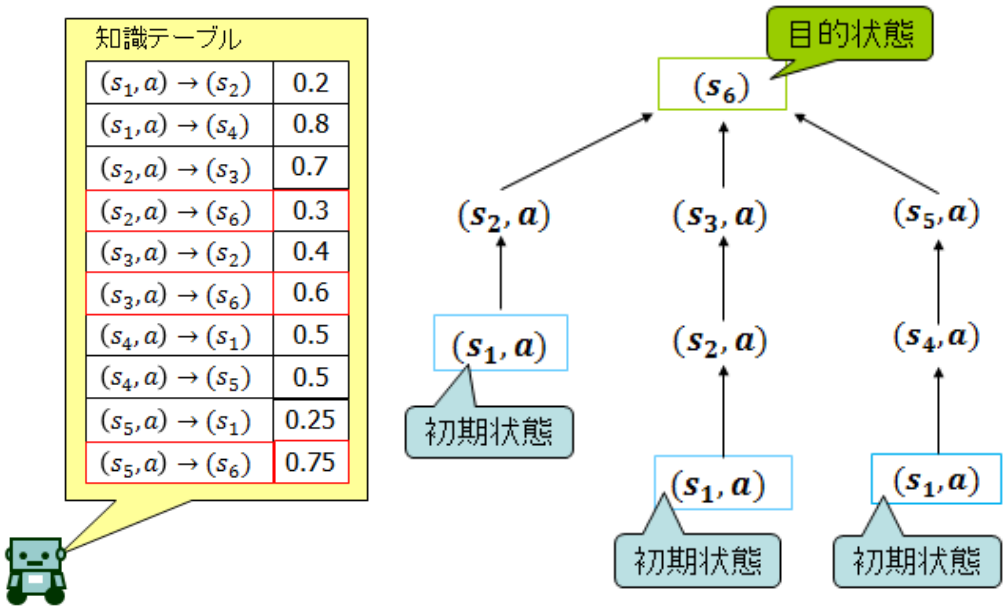


図 4.10 遷移ルートの探索例

Step4 では Step3 で見つけたルートに対して各状態行動対に対応する価値関数を更新する．この価値関数の更新は目的が変更される毎に行われる．価値関数の価値関数の更新式については 4.4.2 節で述べる．

4.4.2 確率的報酬非依存型知識の利用における価値関数の更新

4.4.1 で述べたように確率的報酬非依存型知識の利用は 3 ステップに分かれ，Step3 において価値関数が更新されることとなる．確率的報酬非依存型知識を用いた価値関数の更新式を式(4.1)，式(4.2)に示す．

$$Q(s_t,a) \leftarrow Q(s_t,a) + (f \times \gamma)^{d-1} \times P^{a}_{s_t s'} \times r \times SG(s_d) \quad (4.1)$$

$$SG(s_t) = \begin{cases} 1.0 & (s_t\text{が初期状態から目的状態のルート上}) \\ 0.5 & (\text{それ以外}) \end{cases} \quad (4.2)$$

式(3.1)において $Q(s_t,a_t)$ は更新対象の Q 値を表す．d は状態 s_t から目的状態までの最短距離を表す．この最短距離 d は知識テーブル内の最短距離である．そのため環境の最短距離と一致するとは限らない．最短距離 d は遷移ルートによって目的状態との距離が変わる場合がある．例を図 4.11 に示す．同じ状態 s_1 でも，行動 a と行動 b を取った場合で遷移ルートが異なる．この場合，各遷移先によって最短距離 d は変わる．$s_1 \rightarrow s_2 \rightarrow s_6$ のルート

の場合には目的状態 s_6 から s_1 までの距離 d は 2 となる．また，$s_1 \rightarrow s_4 \rightarrow s_5 \rightarrow s_6$ のルートの場合には目的状態 s_6 から s_1 までの距離 d は 3 となる．

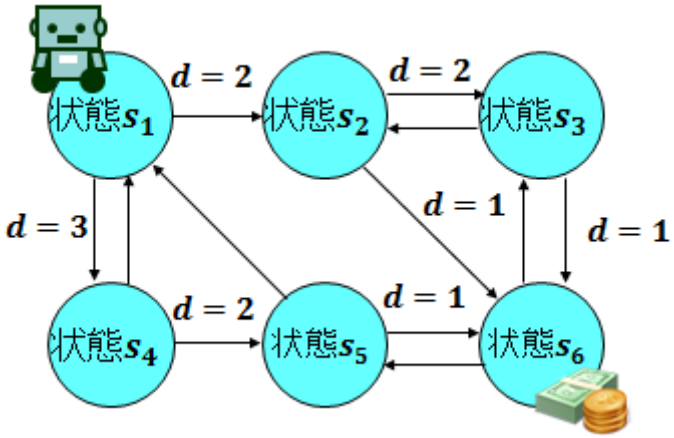


図 4.11　d の設定例

f は確率的報酬非依存型知識の利用度を表すパラメータである．f は $0 \leq f \leq 1$ である．r は獲得した報酬を表す．$P^{a}_{s_t s'}$ は状態 s_t において行動 a を取ったときに状態 s' に遷移する確率を表す．

式(3.2)で示される $SG(s_t)$ は更新対象となる状態行動対の内，状態 s_t がエージェントの初期状態から目的状態までのルートの中にあるかどうかで値が変化する．このようにすることで，初期状態から目的状態までのルートは重要であるとし，それ以外のルートに対しても探索する必要性を考慮している．

第 5 章　実験

5.1　実験目的

確率的報酬非依存型知識を用いたシステムが動的環境下に適応できることを確認する．また，動的環境下への適応だけでなく，目的変更にも対応できることを確認する．

5.2　実験概要

5.1.1　概要

提案手法の有用性を示すためにシミュレーション実験を行った．本実験は迷路問題を用いる．迷路問題ではロボットはスタートからゴールまでの行動数ができるだけ少なくなるように学習を行う．この迷路問題に強化学習を適用したロボットと先行研究である報酬非依存型知識を有するロボット，そして提案手法の確率的報酬非依存型知識を有するロボットを適用する．本実験ではロボットがスタート地点からゴール地点にたどり着くまでを 1 試行とする．一定試行における行動数を比較することによって提案手法の有用性を検証する．実験概要を図 5.1 に示す．

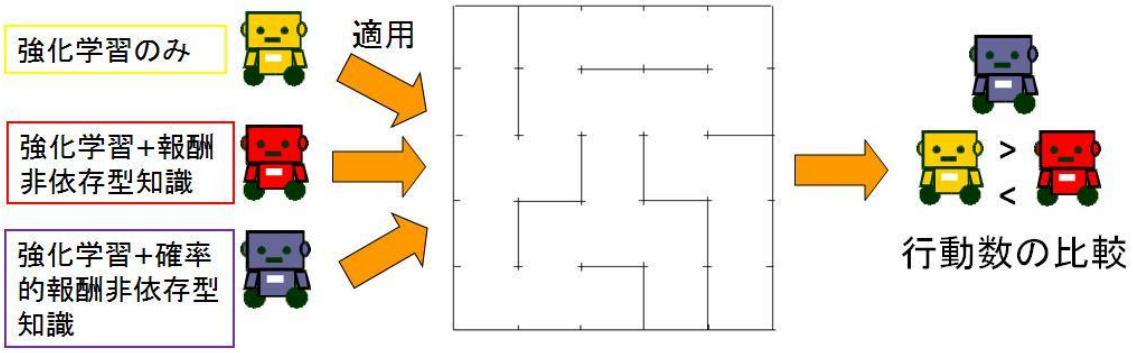


図 5.1　実験概要

5.1.2　実験環境

本実験では実験環境として迷路問題を用いる．ただし，この迷路問題ではゴール変化を伴う．スタート位置は固定で，ゴールの場所は 1 か所である．ゴールの場所は一定試行毎に変化する．この設定は目的変化に追従できることを示すための設定である．また，エージェントと同時にランダムに動く障害物を迷路に投入する．エージェントが 1 行動を取るたび，障害物も上下左右のいずれかの行動をランダムに取る．この障害物が存在するマスにエージェントは移動することができなくなる．この設定は動的環境を生み出すためである．図 5.2 にゴール変化の例を示す．また，図 5.3 に動的環境の例を示す．

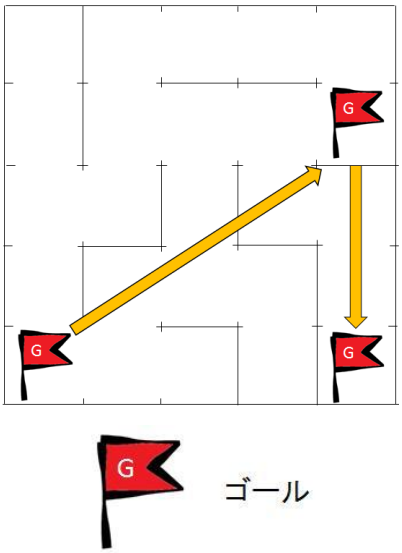


図 5.2 ゴール変化の例

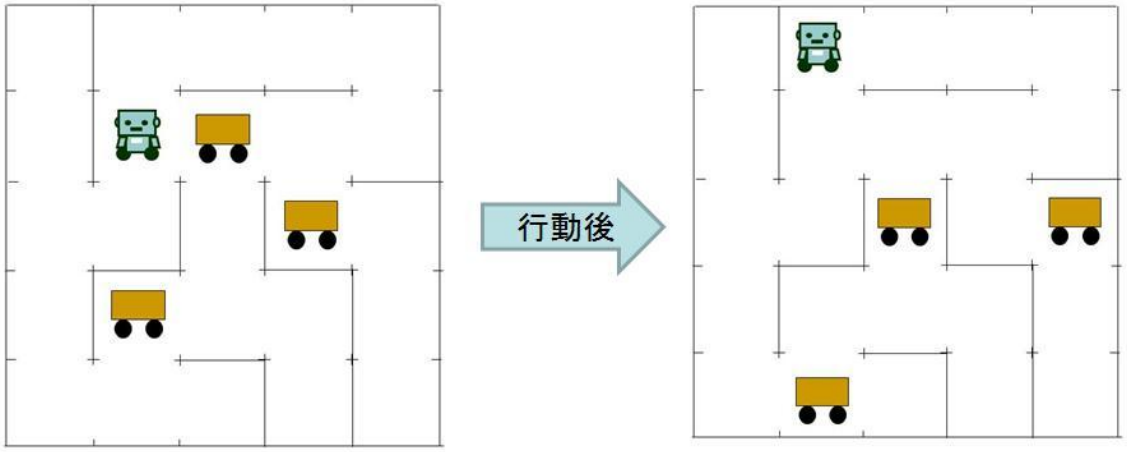


図 5.3 動的環境の例

5.1.3 エージェントの設定

本実験で用いるエージェントの設定を記述する．エージェントは迷路のマスを状態として認識することができる．エージェントは各状態において「上に移動」・「下に移動」・「左に移動」・「右に移動」の 4 つの行動を取ることができる．本実験ではこの 4 つのうちいずれかの行動を取ることを 1 行動とする．壁にぶつかる方向への行動の場合は，行動する前の状態と行動後の状態は同じ状態になるが，この行動も 1 行動とする．

また，強化学習の学習部 Q 学習を用いる．行動選択部には ε -greedy 法と追跡手法を用いる．結果の比較のため，強化学習のみのロボットと報酬非依存型知識を有するエージェント，そして確率的報酬非依存型知識を有するエージェントを用意する．以下に比較するエージェントについてまとめる．

- Agent A　　強化学習のみ
- Agent B　　強化学習＋報酬非依存型知識(先行研究)
- Agent C　　強化学習＋確率的報酬非依存型知識(提案手法)

本実験では，動的環境下であるため報酬非依存型知識は 3.6 節で述べた対応方法をとる．ただし，環境変化を認識した際に価値関数(Q 値)を初期化しない．これは報酬非依存型知識を有するエージェントが連続で障害物に衝突した際に，初期化を繰り返し行い，学習が進まないためである．

5.3　実験

5.3.1　実験環境

今回の実験では 33×33 の迷路 2 つと 49×49 の迷路 1 つでそれぞれ実験を行った. 図 5.4～図 5.6 が実際に使用した迷路である．ゴールは Goal1～Goal10 まであり，Goal1→Goal2→・・・→Goal10 の順番で変化する．

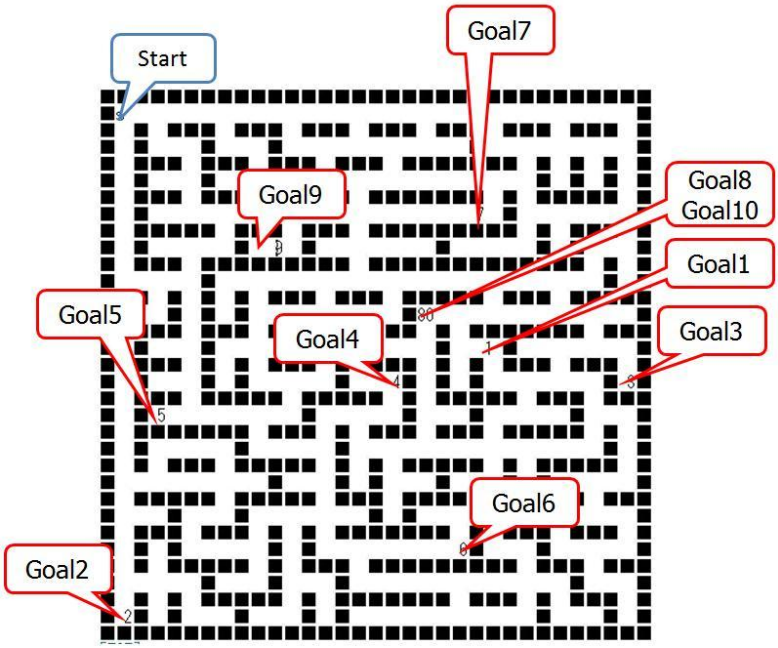


図 5.4 実験で用いた迷路(33×33(a))

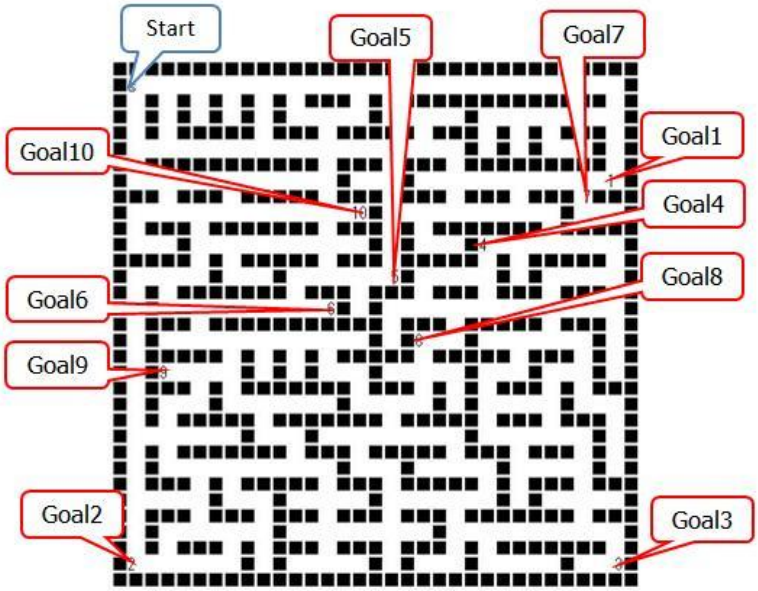


図 5.5 実験で用いた迷路(33×33(b))

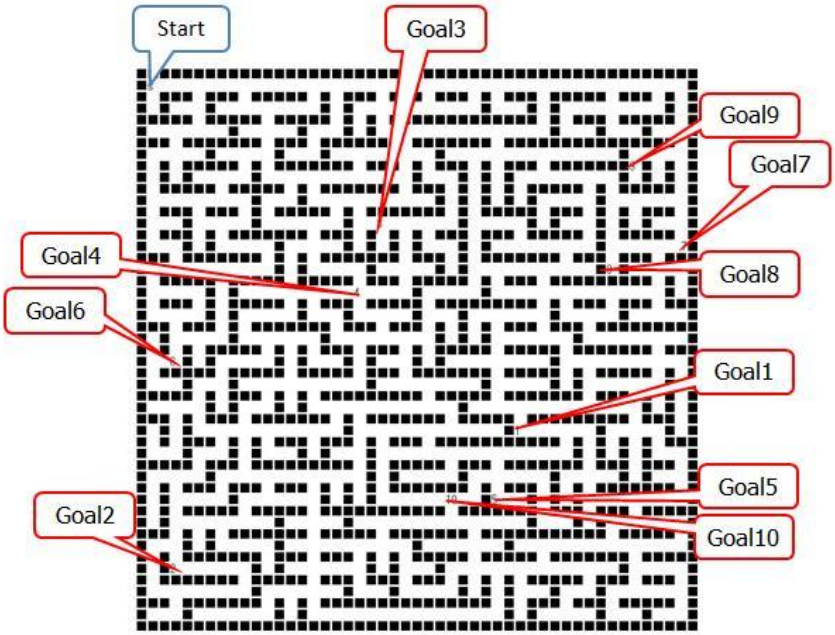


図 5.5 実験で用いた迷路(50×50)

5.3.2　実験パラメータ設定

実験で用いたパラメータ設定を表 5.1～表 5.4 に示す．本実験では 3 つの迷路を用いて実験を行う． そのため，3 つの迷において共通の設定を表 5.1 に示す．各迷路固有の設定を表 5.2～5.4 に示す.

表 5.1　全ての迷路において共通な設定

報酬(ゴールのみ)	100
実験終了までの試行数	2500
ゴール変更試行数	250
Q 値の初期値	0.001
ε (ε -greeedy)	0.05
β	0.7
知識利用率 f	1.0

表 5.2　迷路 33×33(a)での実験パラメータ設定

迷路サイズ	33×33
全状態数	540
学習率 α	0.9
割引率 γ	0.7
障害物の数	40

※1 行動あたりの障害物との遭遇確率 0.07

表 5.3　迷路 33×33(b)での実験パラメータ設定

迷路サイズ	33×33
全状態数	541
学習率 α	0.9
割引率 γ	0.7
障害物の数	40

※1 行動あたりの障害物との遭遇確率 0.07

表 5.3　迷路 50×50 での実験パラメータ設定

迷路サイズ	50×50
全状態数	1219
学習率 α	0.9
割引率 γ	0.9
障害物の数	70

※1 行動あたりの障害物との遭遇確率 0.05

5.3.3 結果

(1) 33×33(a)の迷路の場合

行動選択部の手法にε-greedy 法を用いた実験結果を図 5.6～図 5.8 に示す．図 5.6 では各ゴールにおける行動数の総和の比較を行ったグラフである．各ゴールにおいて行動総数が低いということはエージェントの学習が行われていることを表している．Goal1～Goal3 ではエージェントの遷移回数が少ないため，agent C(提案手法)は agent B(先行研究)と同等かそれ以上の行動数となっている．しかし，Goal4～Goal10 では agent C(提案手法)は agent B(先行研究)よりも行動数は少なくなっている．また agent A(強化学習のみ)と比べても Goal1～Goal10 の場合，行動数は少なくなっている．図 5.7 は agent A(強化学習のみ)と agent C(提案手法)の行動総数を正規化し，比較したものである．この正規化は agent A(強化学習のみ)の行動総数を 1 として，正規化している．また，図 5.8 は agent B(先行研究)と agent C(提案手法)の行動総数を正規化し，比較したものである．この正規化は agent B(先行研究)の行動総数を 1 として，正規化している．図 5.7，図 5.8 の total は Goal1～Goal10 までの全ての行動数を足し合わせて正規化したものである．図 5.7 は全体的に強化学習よりも提案手法の方が行動総数は少ない．Goal7 では学習が後半なのにもかかわらず，強化学習と提案手法に差は見られない．これは，Goal7 にたどり着くまでに長い 1 本道があり，障害物に道をふさがれ中々ゴールできなかったためと考えられる．図 5.8 は学習後半になるにつれて先行研究との行動数の差が多いのが見られる．

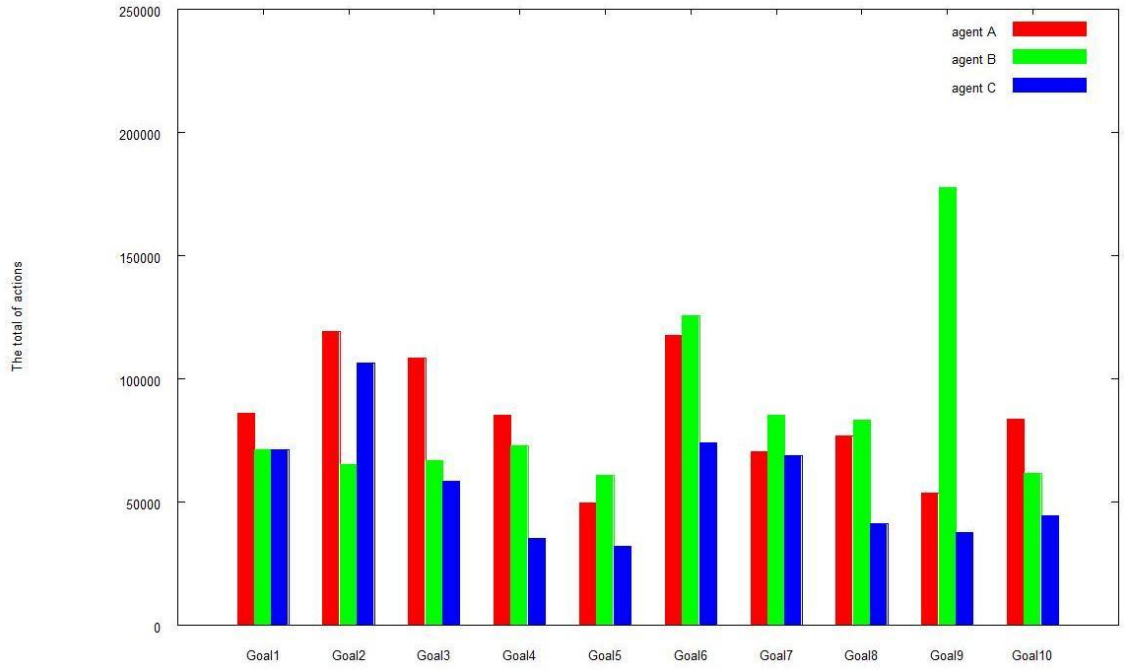


図 5.6 ε-greedy を用いた場合の各ゴールにおける行動数の総和

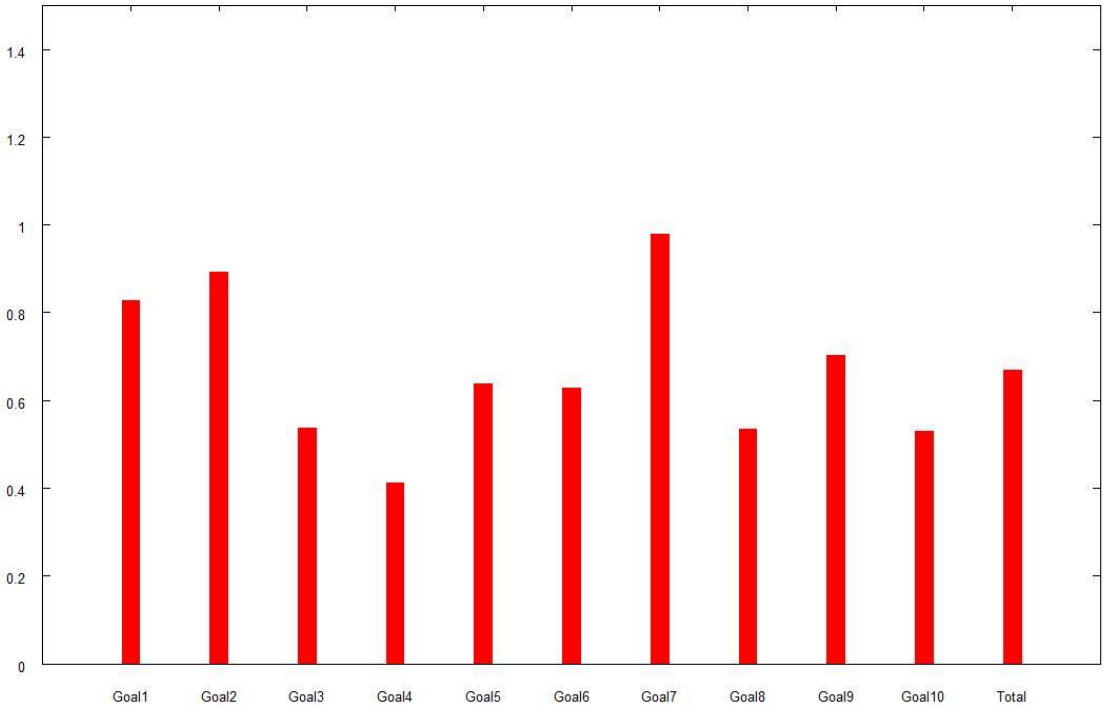


図 5.7　ε -greedy の場合の強化学習に対する提案知識の行動数総和の比較

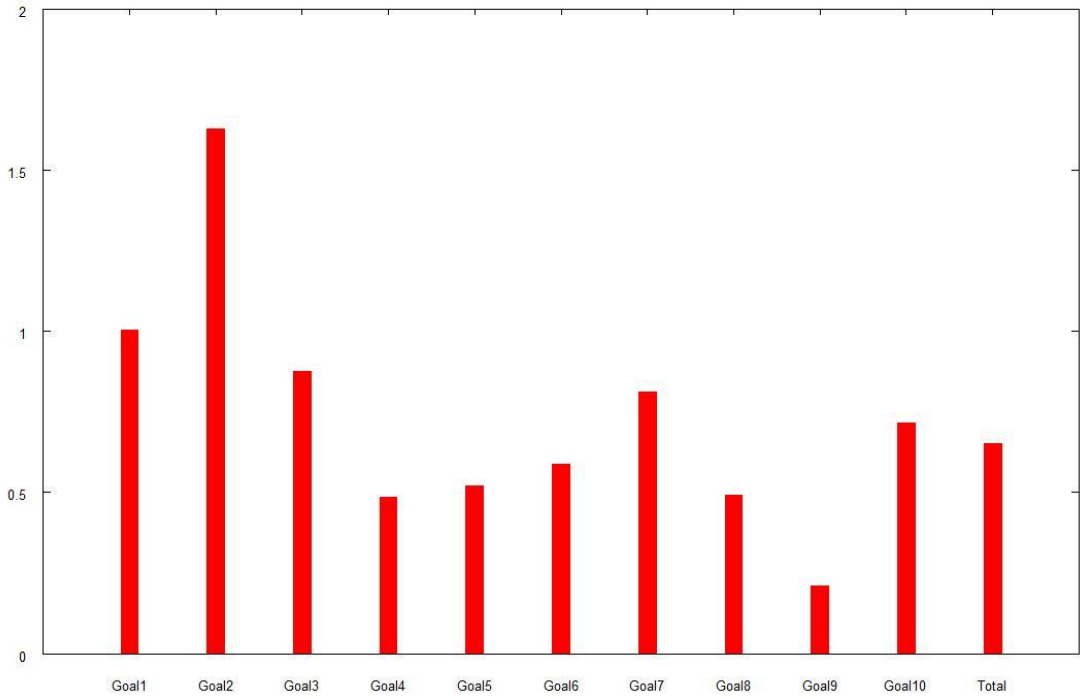


図 5.8　ε -greedy の場合の先行研究に対する提案知識の行動数総和の比較

次に追跡手法を用いた実験結果を図 5.9～図 5.11 に示す．図 5.11 では各ゴールにおける行動数の総和の比較を行ったグラフである．図 5.10 は agent A(強化学習のみ)と agent C(提案手法)の行動総数を正規化し，比較したものである．図 5.11 は agent B(先行研究)と agent C(提案手法)の行動総数を正規化し，比較したものである．図 5.10，図 5.11 の total は Goal1～Goal10 までの全ての行動数を足し合わせて正規化したものである．図 5.10 は ε -greedy 法と同じく Goal7 を除くと，全体的に強化学習よりも提案手法の方が行動総数は少ない．図 5.11 は ε -greedy 法の場合と比べると大きな差はないように見える．これは報酬非依存型知識が正しい遷移ルートを導きだしたためである．そのため，提案知識と同じような遷移ルートを通ったためにあまり差は見られない．

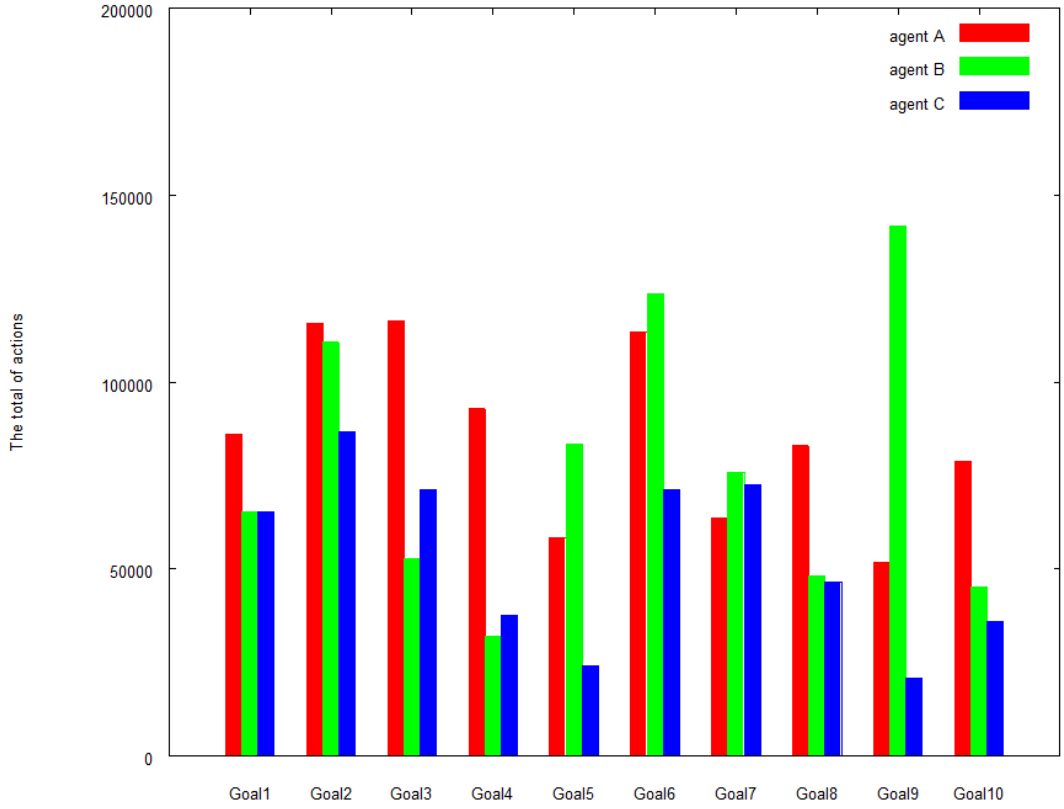


図 5.9 追跡手法を用いた場合の各ゴールにおける行動数の総和

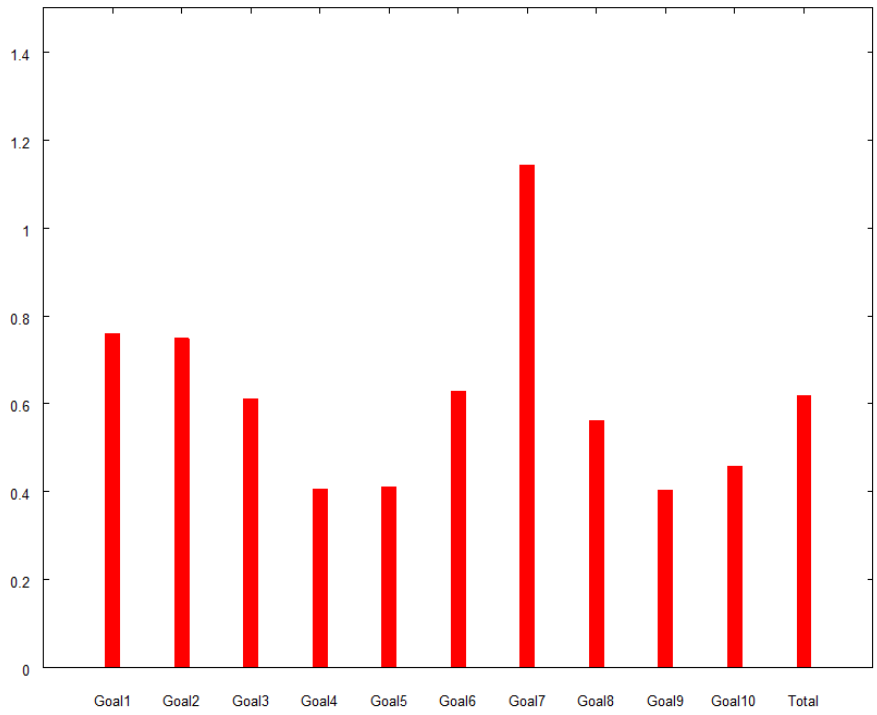


図 5.10 追跡手法の場合の強化学習に対する提案知識の行動数総和の比較

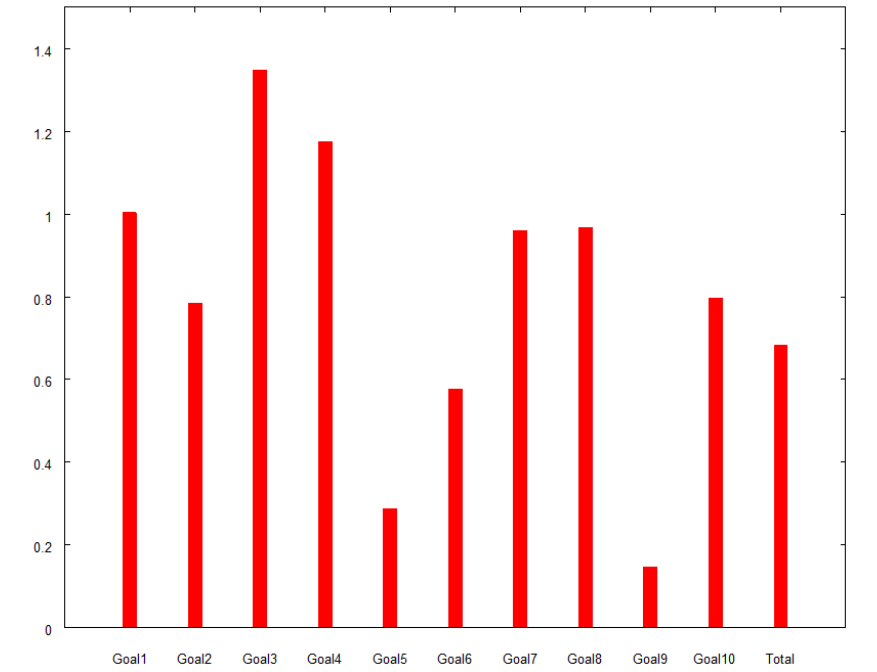


図 5.11 追跡手法の場合の先行研究に対する提案知識の行動数総和の比較

(2) 33×33(b)の迷路の場合

行動選択部の手法にε-greedy を用いた実験結果を図 5.12～図 5.14 に示す．図 5.12 では各ゴールにおける行動数の総和の比較を行ったグラフである．Goal2 ではエージェントの遷移回数が少ないため，正しく遷移確率を導き出せていない．そのため，agent C(提案手法)は agent B(先行研究)より多い行動数となっている．しかし，Goal4～Goal10 では agent C(提案手法)は agent B(先行研究)よりも行動数は少なくなっている．また agent A(強化学習のみ)と比べても Goal1～Goal10 の場合，行動数は少なくなっている．図 5.13 は agent A(強化学習のみ)と agent C(提案手法)の行動総数を正規化し，比較したものである．図 5.14 は agent B(先行研究)と agent C(提案手法)の行動総数を正規化し，比較したものである．図 5.14，図 5.15 の total は Goal1～Goal10 までの全ての行動数を足し合わせて正規化したものである．図 5.12 は全体的に強化学習よりも提案手法の方が行動総数は少なく，目的変化にも対応していることがわかる．図 5.13 は学習後半になるにつれて先行研究との行動数の差が多いのが見られる．特に Goal10 の行動総数に差が出ている．これは Goal5 や Goal6 のゴールの時に Goal10 の周辺を通ったため，遷移回数が多くなり，正しく遷移確率を導き出せたためである．

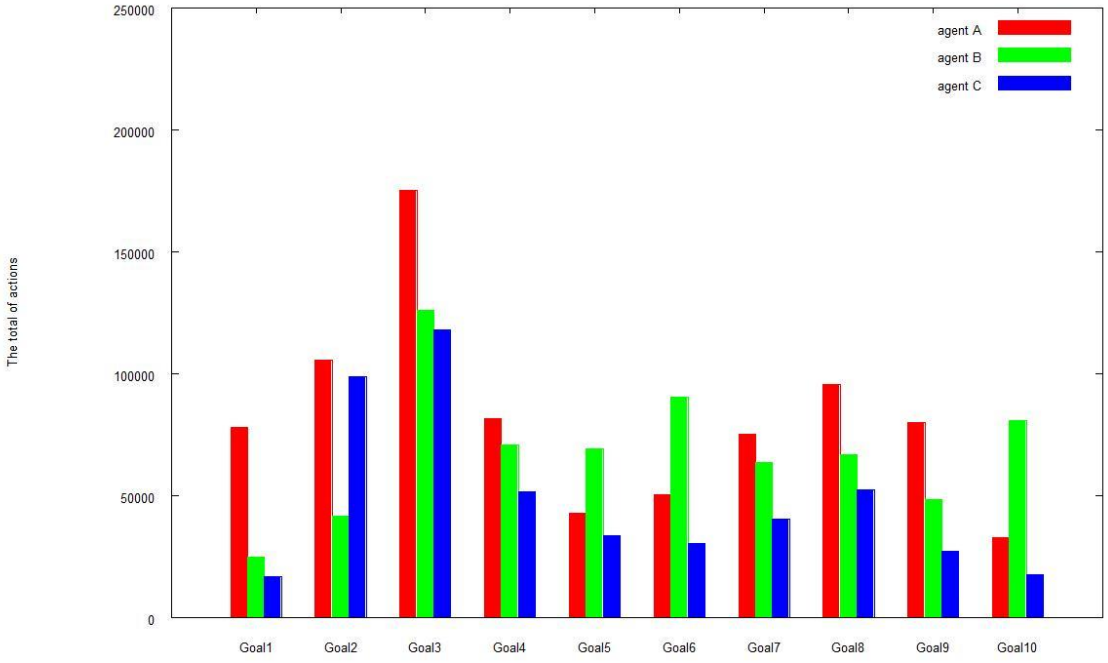


図 5.12　ε-greedy を用いた場合の各ゴールにおける行動数の総和

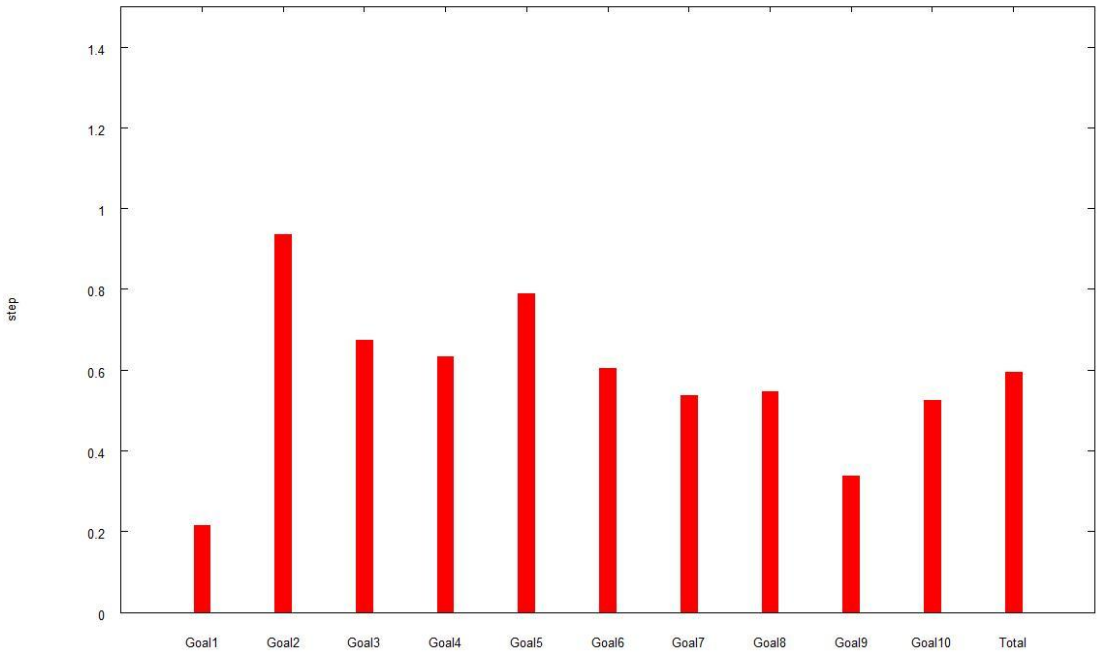


図 5.13　ε -greedy の場合の強化学習に対する提案知識の行動数総和の比較

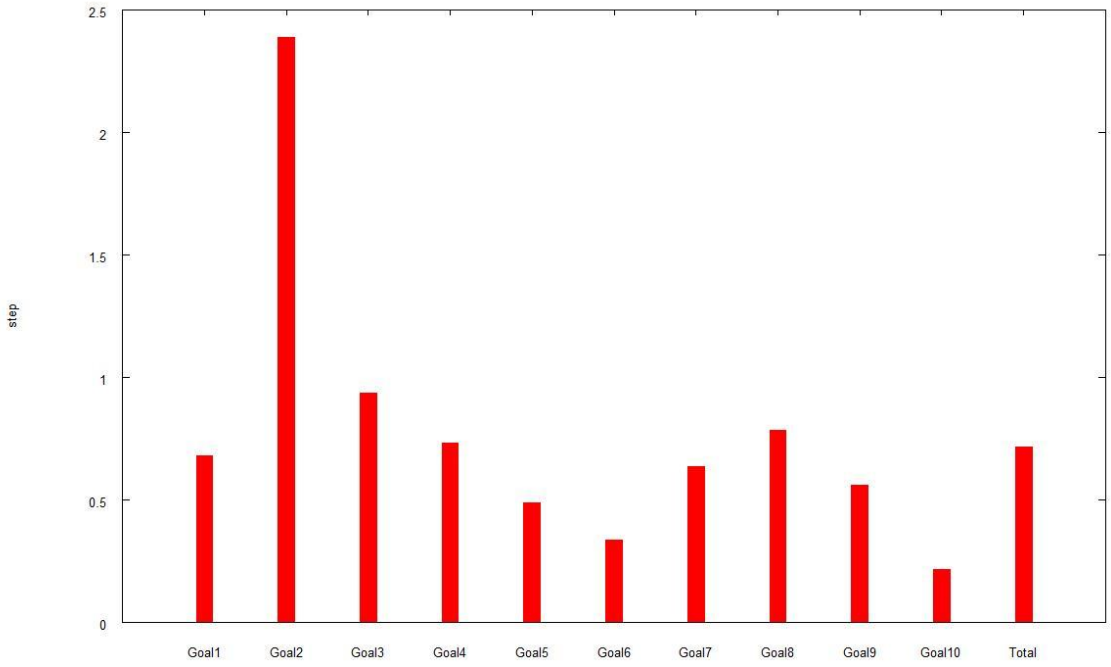


図 5.14 ε -greedy の場合の先行研究に対する提案知識の行動数総和の比較

行動選択部の手法に追跡手法を用いた実験結果を図 5.15～図 5.17 に示す．図 5.15 では各ゴールにおける行動数の総和の比較を行ったグラフである．Goal1～Goal4 ではエージェントの遷移回数が少ないため，遷移確率が環境の遷移確率と大きく異なってしまったと考えられる．そのため，agent C(提案手法)は agent B(先行研究)と同等かそれ以上の行動数となっている．しかし，Goal5～Goal10 では agent C(提案手法)は agent B(先行研究)よりも行動数は少なくなっている．また agent A(強化学習のみ)と比べても Goal1～Goal10 の場合，行動数は少なくなっている．図 5.16 は agent A(強化学習のみ)と agent C(提案手法)の行動総数を正規化し，比較したものである．図 5.17 は agent B(先行研究)と agent C(提案手法)の行動総数を正規化し，比較したものである．図 5.16，図 5.17 の total は Goal1～Goal10 までの全ての行動数を足し合わせて正規化したものである．

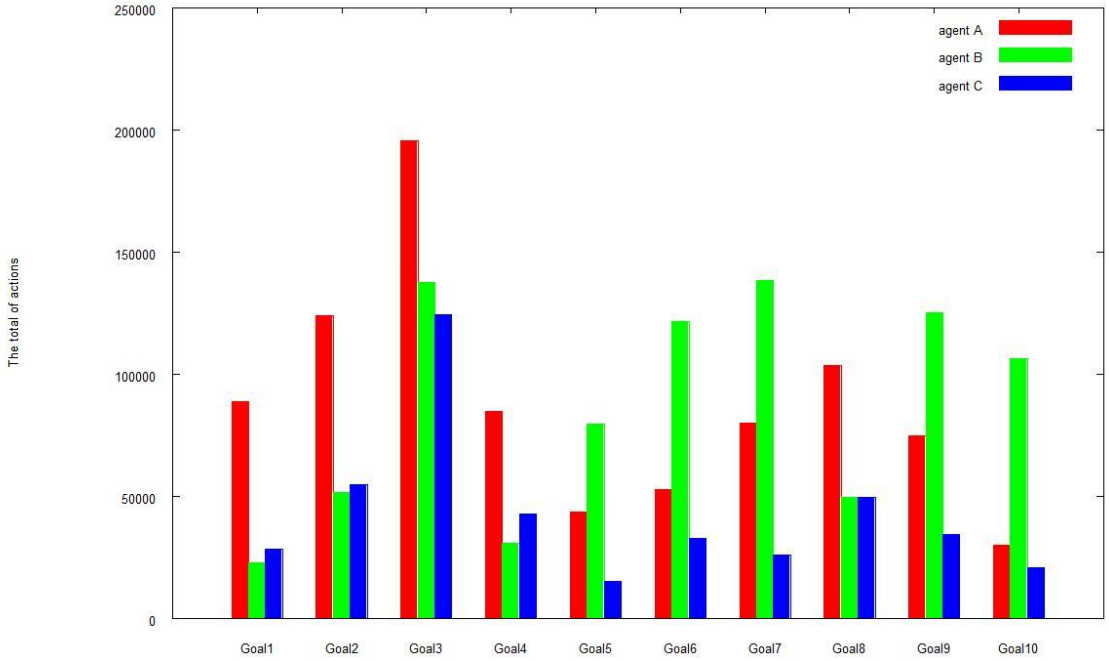


図 5.15 追跡手法を用いた場合の各ゴールにおける行動数の総和

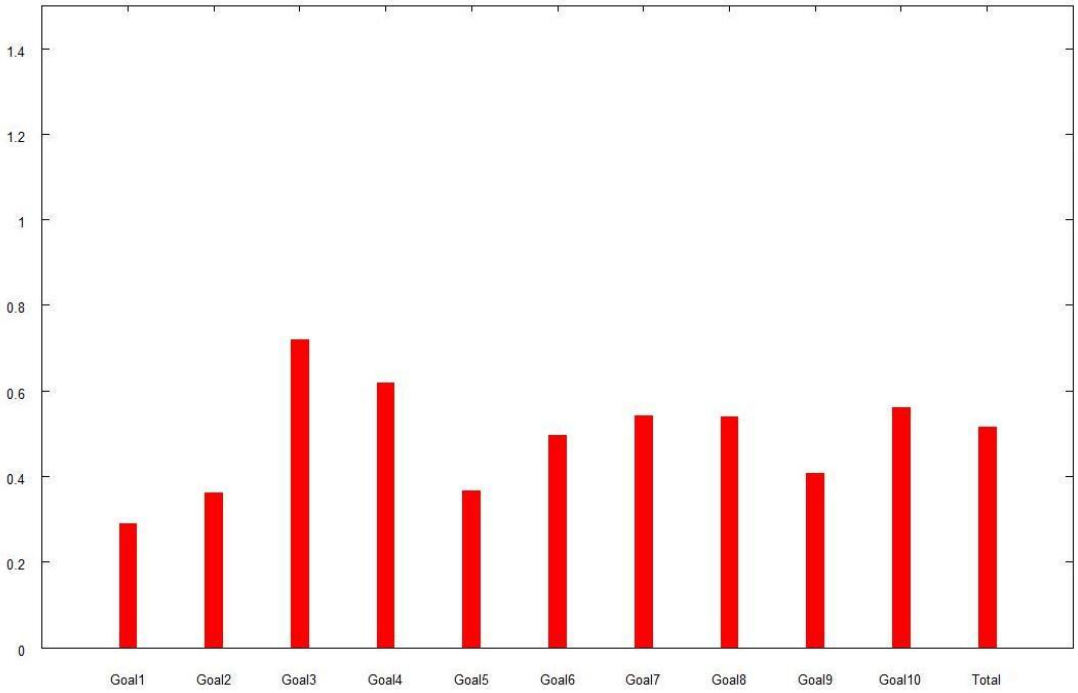


図 5.16 追跡手法の場合の強化学習に対する提案知識の行動数総和の比較

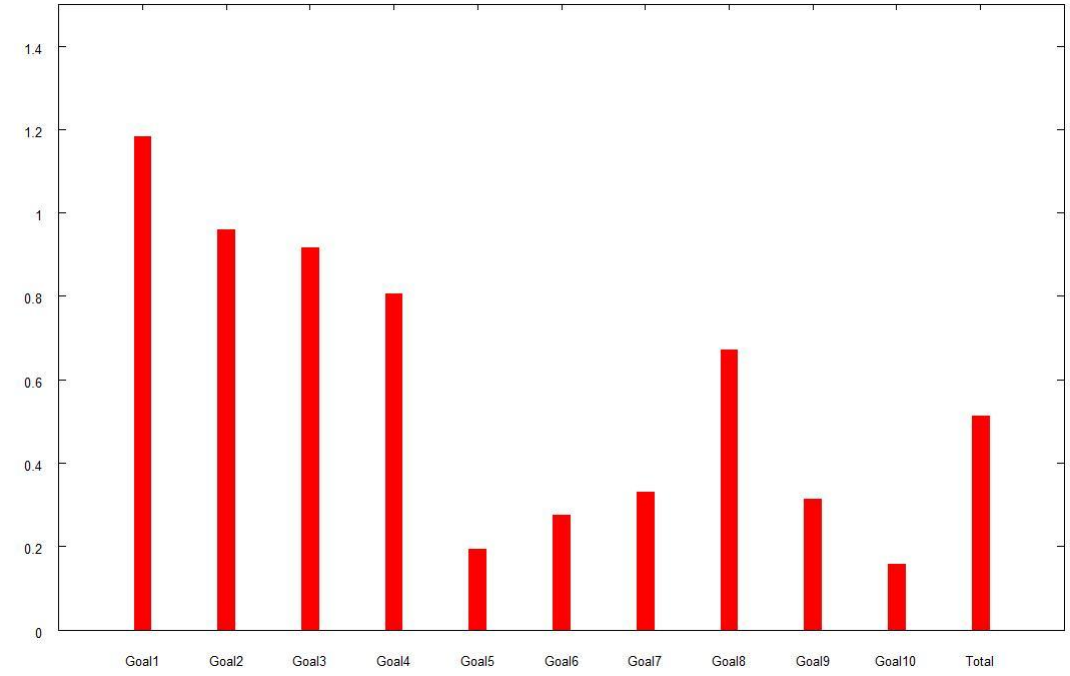


図 5.17　追跡手法の場合の先行研究に対する提案知識の行動数総和の比較

(3) 50×50 の迷路の場合

行動選択部の手法に ε -greedy を用いた実験結果を図 5.18～図 5.20 に示す．図 5.18 では各ゴールにおける行動数の総和の比較を行ったグラフである．図 5.19 は agent A(強化学習のみ)と agent C(提案手法)の行動総数を正規化し，比較したものである．図 5.20 は agent B(先行研究)と agent C(提案手法)の行動総数を正規化し，比較したものである．図 5.19, 図 5.20 の total は Goal1～Goal10 までの全ての行動数を足し合わせて正規化したものである．

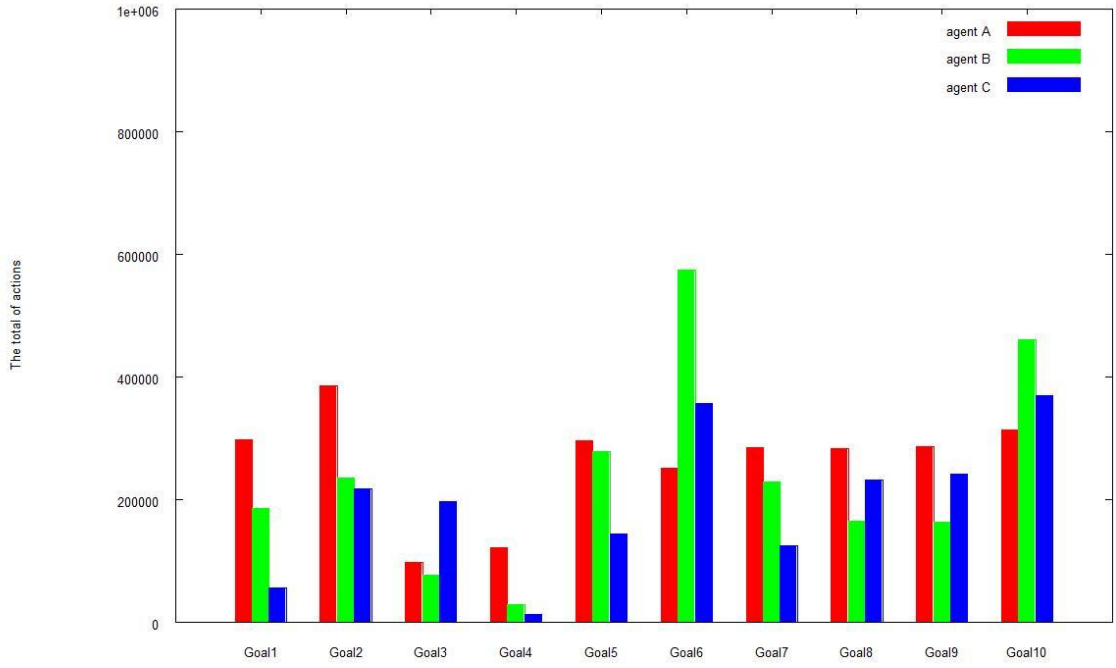


図 5.18 ε -greedy 法を用いた場合の各ゴールにおける行動数の総和

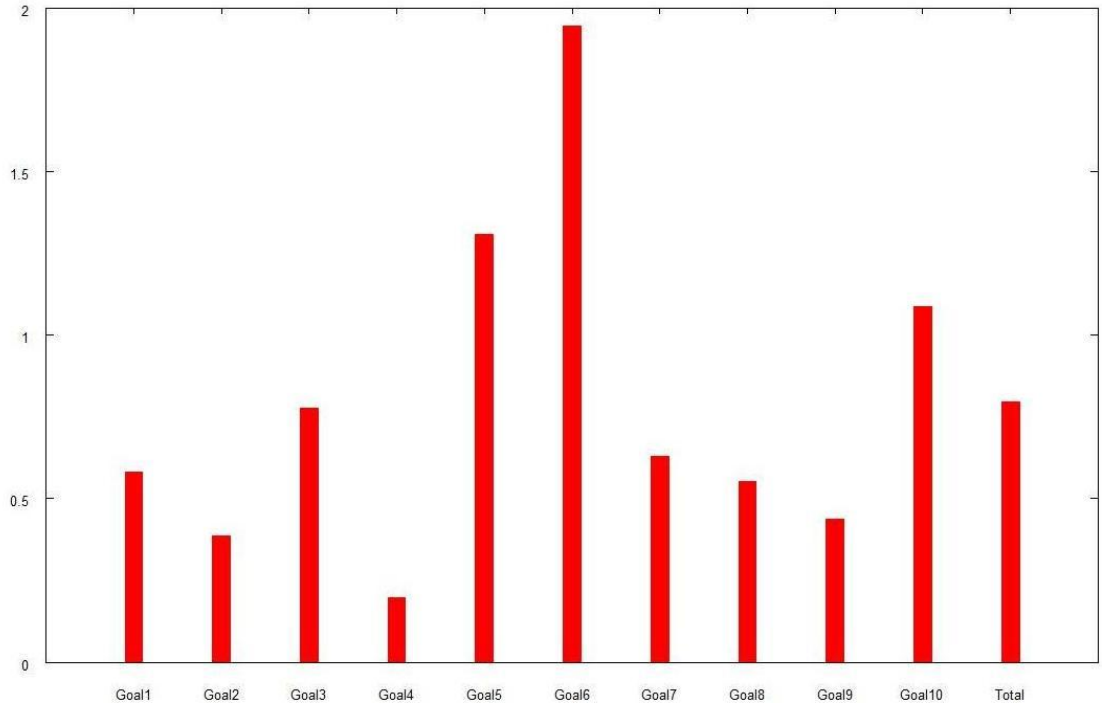


図 5.19　ε -greedy 法の場合の先行研究に対する提案知識の行動数総和の比較

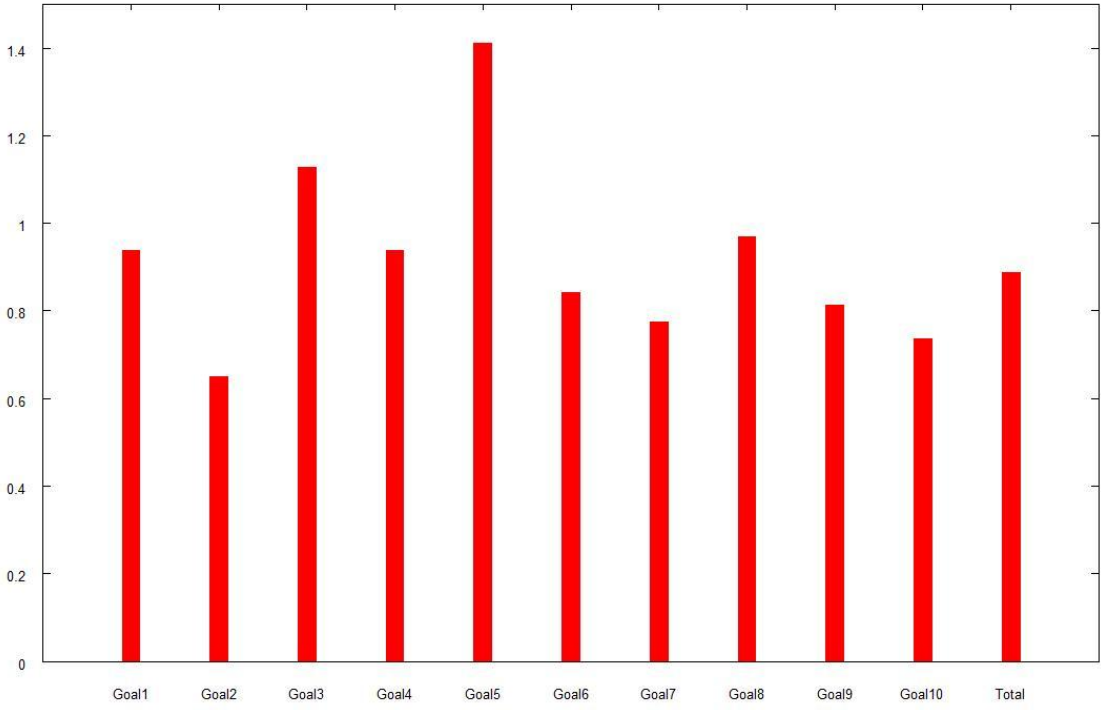


図 5.20　ε -greedy 法の場合の先行研究に対する提案知識の行動数総和の比較

行動選択部の手法に追跡手法を用いた実験結果を図 5.21～図 5.23 に示す．図 5.21 では各ゴールにおける行動数の総和の比較を行ったグラフである．各ゴールにおいて行動総数が低いということはエージェントの学習が行われていることを表している．図 5.22 は agent A(強化学習のみ)と agent C(提案手法)の行動総数を正規化し，比較したものである．図 5.23 は agent B(先行研究)と agent C(提案手法)の行動総数を正規化し，比較したものである．この正規化は agent B(先行研究)の行動総数を 1 として，正規化している．図 5.22，図 5.23 の total は Goal1～Goal10 までの全ての行動数を足し合わせて正規化したものである．

33×33 の迷路と比べて迷路が大きいため，スタート地点周辺ではほとんど Q 値が初期値と同じ値となる．そのため，強化学習と比べて差はないようにも見える．しかし，比較的スタート地点から近い Goal3, Goal4 に対しては強化学習よりも行動数は少なくなっている．

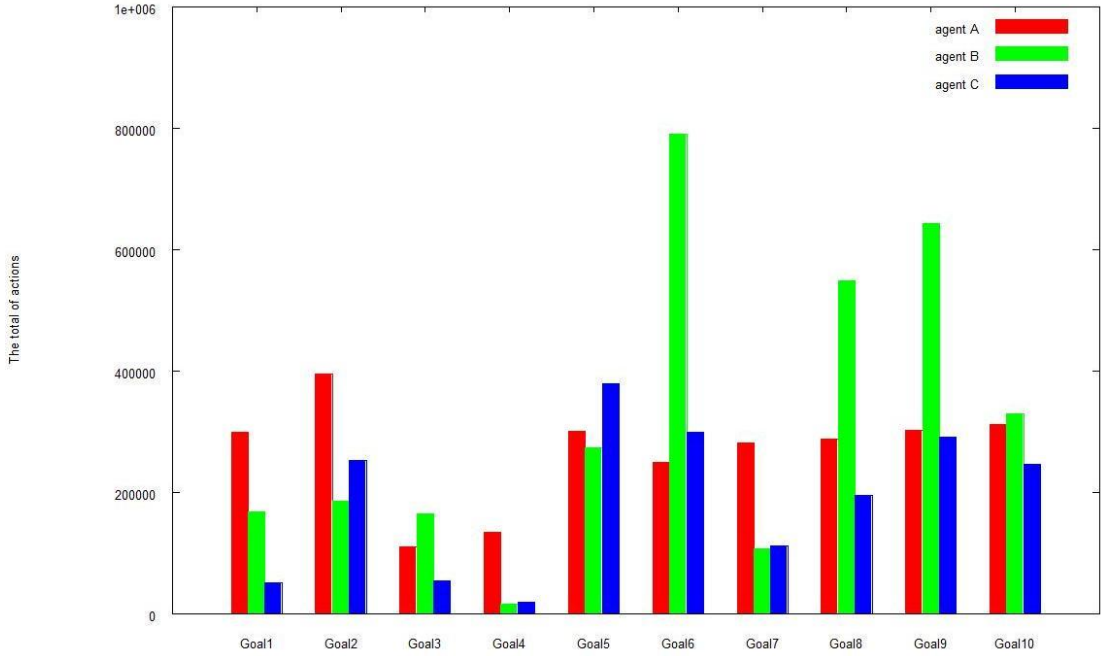


図 5.21 追跡手法を用いた場合の各ゴールにおける行動数の総和

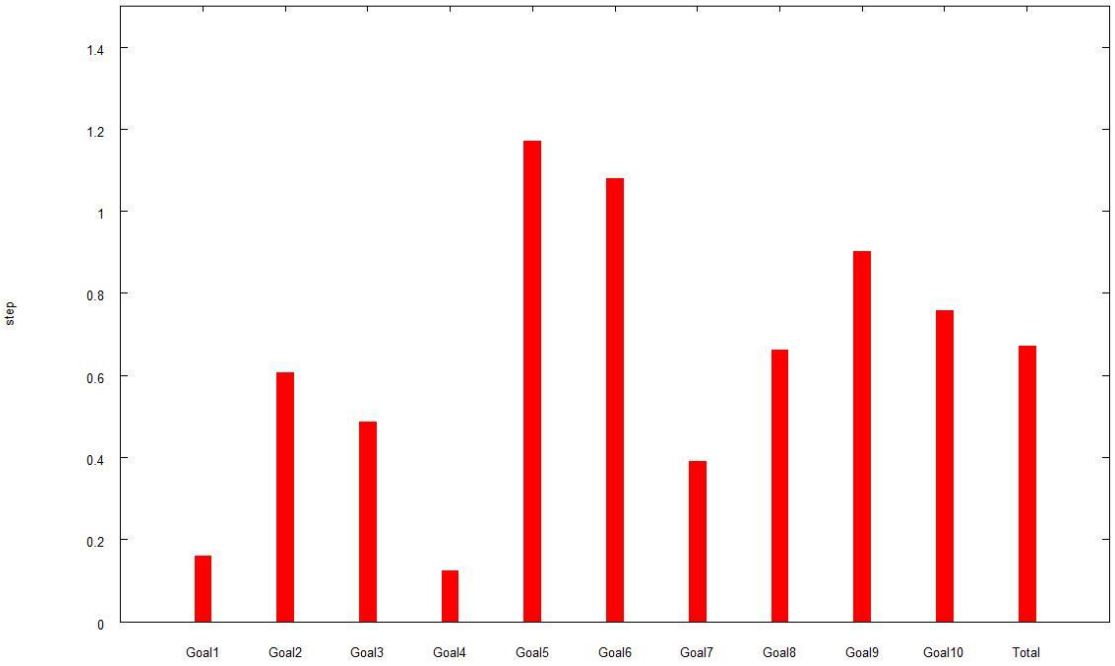


図 5.22 追跡手法の場合の強化学習に対する提案知識の行動数総和の比較

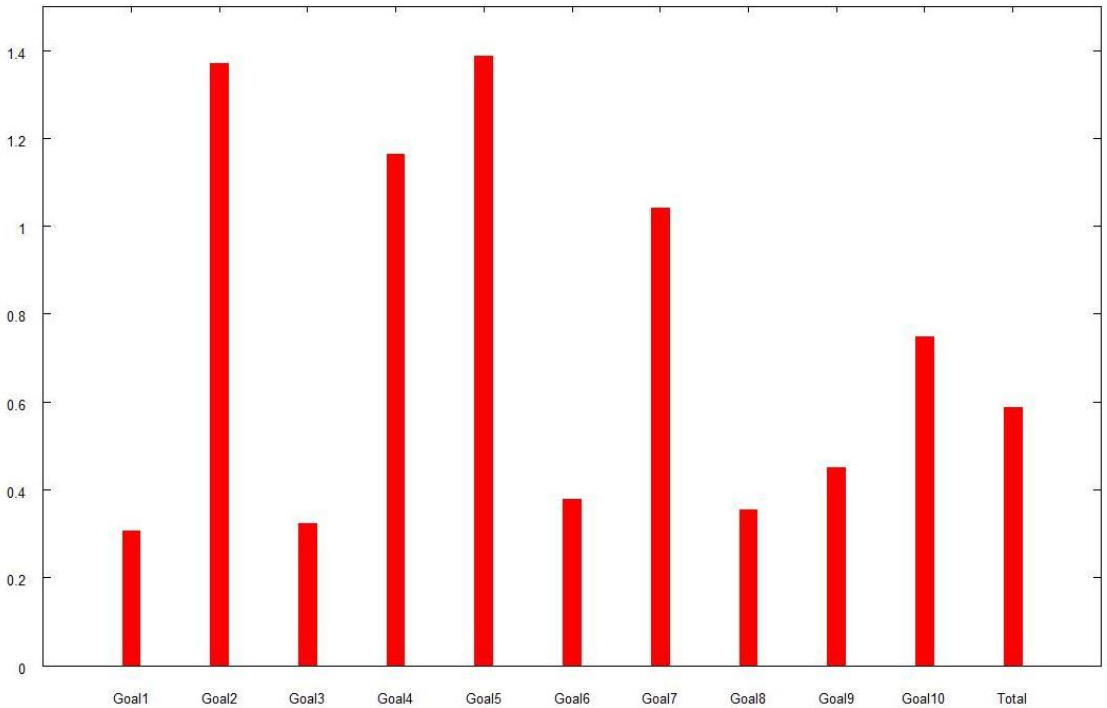


図 5.23 追跡手法の場合の先行研究に対する提案知識の行動数総和の比較

5.3.4 考察

今回の実験では 3 種類の迷路を用意した．どの迷路においても確率的報酬非依存型知識を用いた場合，行動総数が少なくなることが結果としていることができた．学習初期段階には確率的報酬非依存型知識であると，遷移回数が少なく，確率に偏りが生じる場合がある．この場合，強化学習や報酬非依存型知識を有するエージェントよりも行動数が多くなってしまうことがあった．しかし，学習進んでいくにつれて，確率的非報酬依存型知識を有するエージェントの遷移回数が増え，確率に偏りがなくなっていった．これにより，行動数が強化学習や報酬非依存型知識を有するエージェントよりも行動総数が低くなった．これは学習の収束が早いことを意味している．

実験結果によって，確率的報酬非依存型知識を有するエージェントは，動的環境において対応ができているといえる．また，目的変化に対しても対応できているといえる．

第 6 章　結論

6.1　まとめ

本論文では報酬非依存型知識を動的環境下に適応させることを目的とした．先行研究の報酬非依存型知識は各状態行動対とそれによる遷移先の状態により構成されていた．この報酬非依存型知識は各状態行動対と遷移先が 1 対 1 対応であった．そのため，動的環境下では学習効率が低下するという問題点があった．そこで，本論文では各状態行動対に対して遷移先が複数になるような知識を提案した．さらに各遷移先にどのくらいの割合で遷移するかという確率を持たせた．この遷移確率を含めた知識を「確率的報酬非依存型知識」と定義した．そして，エージェントの経験から遷移確率を算出し，その遷移確率に合わせて強化学習で用いる価値関数にバイアスをかけ，動的環境の適応を目指した．

実験は強化学習のみのエージェント，先行研究である報酬非依存型知識を有するエージェント，そして提案手法の確率的報酬非依存型知識を有するロボットの 3 台で比較を行った．3 種類の迷路に適用し，いずれの場合も動的環境に対応し，かつ目的変更にも対応できた．これらの結果を得ることで提案した確率的報酬非依存型知識の有効性を示すことができた．

6.2　今後の課題

強化学習は実ロボットに用いられることが多い手法である．ゆえに本論文で提案したシステムを実ロボットへ適用したいと考えている．しかし，実機で用いる場合に存在する問題点もある．提案システムを実機で用いる場合に考えられる問題点として以下のものがある．

- 不完全知覚下における検証
  本論文の実験では，ロボットのセンサ能力について特に記述することはなかった．しかし，実機のロボットを使用する場合，ロボットの認識能力には制限が出てきてしまう．
  例えば, 迷路問題を例として考える. ロボットは前後左右の壁の有無のみ認識できる(絶対位置がわからない)，ロボットは東西南北の方角がわかっているとする．その場合，図 6.1 の S5 と S9 はロボットには同じ構造に見える．そのため，S5 と S9 の遷移先が異なり，確率的報酬非依存型知識において対応できる可能性がある．

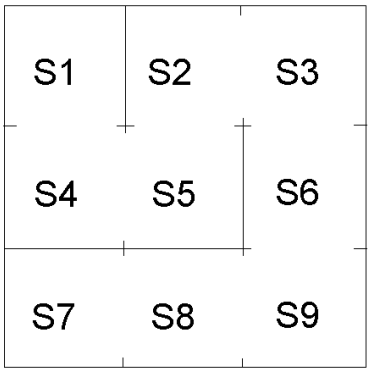


図 6.1 不完全知覚下の例

- ロボットのメモリの制限

  本論文では確率的報酬非依存型知識を保持できる量については特に制限を持たせなかった．しかし，実ロボットを用いる場合には使用できるメモリには限りがある．そのため，知識テーブルの大きさを制限し，使用率の低い確率報酬非依型知識の忘却を行う必要がある．

- 実ロボットの行動に要する時間とこのシステムで計算を行う際の時間との差異

  本研究において，初期状態から目的状態のルートを求める際に，知識テーブル内の確率的報酬非依存型知識を検索する．この検索する時間が多くかかることになる．これはシミュレーション上の実験では問題がないが，実環境では意思決定に無限時間はかけられない．なぜなら，意思決定を行う時間が多ければ，実環境下ではすでに環境が移り変わっている可能性が高いからである．この問題を解決するためには，確率的報酬非依存型知識を持てる限界を決定したり，知識テーブル内の検索方法を変える必要がある．

参考文献

[1] 田島茂樹，青山元，関鉄太郎，石川和良，横田和良，横田和孝，尾崎功一，山本純雄”ロボットによる高層ビルの清掃システム開発”，日本ロボット学会誌，Vol.22，No.5，pp.595-602，2004

[2] 小林宏,”表情豊かな顔ロボットの開発と受付システムの実現”，日本ロボット学会誌，Vol.24，No.6，pp.708-711，2006

[3] 西村昭浩,”ミュージックロボット miuro(ミューロ)”，日本ロボット学会誌，Vol.26，No.8，pp.887-888，2008

[4] 森川幸人，“マッチ箱の AI”，新紀元社，2000

[5] Richard S. Sutton and Andrew G. Barto., “Reinforcement Learning”, The MIT Press, 1998

[6] 木村元，山下透，小林重信，“強化学習による 4 足歩行ロボットの歩行動作獲得”，電気学会 電子情報システム部門誌，Vol.122-C,No.3，pp.330-337，2007

[7] 田中文英，山村雅幸,”AI 化建築へ : マルチタスク強化学習による住居屋根制御”，第 3 回 MYCOM オンライン資料，pp.103-110，2002

[8] 高橋泰岳，浅田稔，“階層型学習機構における状態行動空間の構成”，日本ロボット学会誌，Vol.21，No.2，pp164～171，2003

[9] 星野孝総，亀井且有，“ファジィ環境評価ルールを用いた強化学習の提案とチェスへの応用”，日本ファジィ学会誌，Vol13，No.6，pp626～632，2001

[10] 宮崎愛央, “強化学習における報酬非依存型知識の提案”, 室蘭工業大学卒業研究, 2009

謝辞

本論文を結ぶにあたり，日頃より懇切なるご指導を賜りました倉重健太郎先生に深く感謝の意を表します．また，ご助言，ご指導をいただいた畑中雅彦先生，蓮井洋志先生，佐賀聡人先生，本田泰先生に感謝の意を表します．そして，論文の査読や助言をしていただいた認知ロボティクス研究室の木島康隆さん，中南義典さん，宮崎愛央さん，梅津祐介君，北山直樹君に感謝致します．