修士論文

強化学習における探査率の動的制御

室蘭工業大学 工学研究科 情報電子工学系専攻
コンピュータ知能学コース 博士前期課程 11054032 番

澁谷　和

目 次

第1章　はじめに

1.1　ロボットの社会への普及

近年，ロボットは工場や研究室のみならず，様々な形で社会に普及している．例えば，家庭用掃除ロボットのルンバや 2 足歩行型ロボットである HONDA 社の「ASIMO」[1] や NEC 製のチャイルドケアロボット「Papero」[2] などがある．もちろん, 工場で使われる従来の産業用ロボットも多く存在するが，近年ではより人間に近い動作ができるように，柔軟物を取り扱えるロボットの研究などもすすめられている [3]．他にも，人間に代わって危険な場所での作業の代替や力仕事の支援をするロボットの開発もさかんに行われている．例えば，介護用ロボット [4–6] がある．看護者にとって，寝たきりの患者を抱きかかえたり，搬送する作業は腰痛などの原因となる作業である．その作業を代替えする介護用ロボットに期待が寄せられている．また，放射線量が高い原子炉内の作業を代替する原発作業用ロボット [7–9] や海中ロボット [10,11] や宇宙ロボット [12–14] やレスキューロボット [15,16]，人間が立ち入ることが難しい場所で作業を行うロボット，いわゆる極限作業ロボットの開発も行われている．他にもエンターテイメント用 [17,18] ロボットとして，人間を楽しませるようなロボットも開発されている．このように，ロボットは様々な用途で使用されていて，我々の生活に欠かせないものとなりつつある．

1.2　学習の必要性

ロボットの日常生活環境への普及にあたって，ロボットの用いられる環境への適応が課題の一つとして挙げられる．我々人間が生活する環境は動的環境と呼ばれ，ロボットがそれまでに用いられてきた工場や研究室など変化の少ない環境と異なり，時々刻々と変化し続けている．例えば，家庭の環境ひとつとっても，常に変化している．周辺の家具の位置や，床に散らばっている本などの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，変化が複雑で予想することが難しい．このような動的環境においても，ロボットには自身のタスクを遂行することが期待されるため，ロボットがいかにその環境に適した行動を取るかが問題となってくる．

この問題を解決するためのアプローチの一つとして，ロボットの直面する環境を予測し，行動を完

全に設計するという方法が考えられる．これはロボットの設計者が，ロボットが直面するであろうあらゆる環境を予測し，各環境に適した動作をロボットに設計するという方法である．この方法は，動的環境においては，ロボットの直面する環境が限られているため有効な方法であると考えられる．しかし動的環境においては，この手法の有効性は低くなることが予測される．なぜなら，動的環境は環境が時間とともに変化するものであり，この変化のパターンが無数に存在するためである．そのため，設計者が全てのロボットが直面する環境を予測することは困難である．

よって，動的環境においてロボットが環境に適応するためには，人間のようにロボットが自ら直面する状態を認識し，その環境に応じた行動を獲得することが望ましい．このようなことを可能にするために，機械学習 [19] という方法がある．機械学習とは，人間が以前の経験を活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．

機械学習の行動の違いを Fig. 1.1 に示す．この例では，今までロボットが通った通路に突然，障害物が現れたとする．ロボットの直面する環境を予測し，行動を完全に設計するという方法の場合，突然障害物が現れたときに，対処できない．しかし，機械学習の場合，何度かは障害物と衝突するが，そこに障害物があることを認識し，よけるという行動をロボットは獲得することができる．

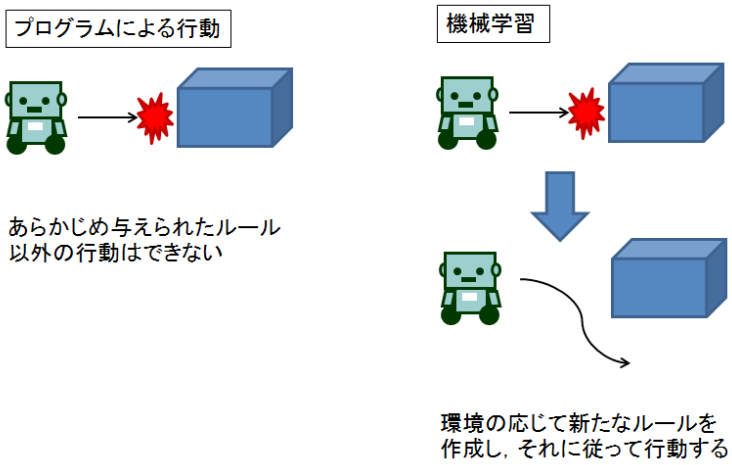


Fig. 1.1　プログラムと学習の違い

1.3　機械学習

機械学習には大きく分けて，教師あり学習，教師なし学習，強化学習の 3 つが存在する強化学習は，教師あり学習か教師なし学習かという議論がよくされる．報酬の検出方法を予め与えなければならないため，教師あり学習であると考えることができるが，出力に対し，直接教師信号を与えるわけでは

ないので教師あり学習でもないといえる．よって，本論文では強化学習を教師あり学習や教師なし学習のどちらでもない学習方法として扱う．

1.3.1　教師あり学習

教師あり学習は，教え示された知識に基づいて学習を進める方法である．教師あり学習では，あるデータについてそれが正しいのか正しくないのかを教える教師がいて，その教師の指示に基づいて学習を進める (Fig. 1.2)．また，学習の方法を教師に教わることもできる．

教師あり学習では，効率的で精密な学習を行うことが可能である．学習について教師から良し悪しをただちに教えてもらえるので，ほかの学習方法と比較して素早くかつ正確に学習を進めることが可能である．しかし，教師あり学習は学習結果を汎化する能力や学習データには現れないような状況に対応する能力において劣ってしまう可能性がある．教師あり学習は，学習のための教師信号が適切でなければ，学習結果も適切でないものになってしまう．これは，教師を誤ったものにしてしまえば，ロボットの動きも誤ったものになってしまう，ということである．

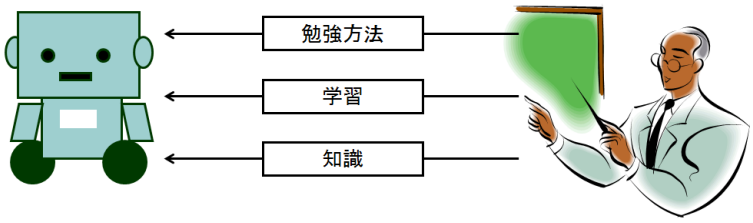


Fig. 1.2　教師あり学習

1.3.2　教師なし学習

教師あり学習に対して，個々の学習事例や学習方法について指示を受けないで学習を進める方法を教師なし学習と呼ぶ．正しい出力は与えられないため，何らかの基準を設けてそれを最適にするような出力の割り当てを求めることになる．

1.4　強化学習

1.4.1　強化学習とは?

強化学習 [20] は，ある状態で取った行動の結果に着目し，このときの評価が良くなるように学習を行うものである (Fig. 1.3)．このときに利用するのが報酬と呼ばれるスカラ値の情報である．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．人間は報酬さえ設定すれば，あとはロ

ボットに任せればロボットは報酬獲得までの行動を自動的に獲得できる．このため，ロボットの行動獲得という目的への応用が期待され，また多くのロボット学習において用いられている手法である [21, 22, 24, 24]．応用例として，ロボット以外にも適用されている．例えば，温度や湿度に合わせて強化学習で屋根の角度を学習させる研究 [26] や株の売買 [27, 28] にも適用されている．

Fig. 1.3　強化学習概要

1.4.2　強化学習における問題点

強化学習には探査 (exploration) と知識利用 (exploitaiton) の間のトレードオフをいかに扱うかという問題がある．多くの報酬を得るために，ロボットは過去に試みた行動の中で報酬を得るために効果的な行動を選択し続けなければならない．しかし，このような行動を発見するためには，過去に試みたことのない行動も選択しなければならない．つまり，ロボットは報酬を得るためにすでに持っている知識を利用し，将来的に行動選択を改善するために探査も行わなくてはならない．知識を利用することと，探査することは同時に行うことはできない．そのため，この探査と利用のバランスが重要となる [29]．

強化学習において，探査と利用のバランスは行動の選択方法によって決定される．行動の選択方法の代表的なものに，ε-greedy 法がある [20]．この ε-greedy 法は $(1-\varepsilon)$ の確率で，ランダムな行動を取る．これが，探索行動である．ε の確率で，所持している知識の中でよい行動を選ぶ．これが利用行動である．ε が大きいほど，ランダムな行動，つまり探査行動を取る確率が大きくなる．大きすぎると，その時点で最適ではない行動をとる回数が増え，学習効率の低下につながる．一方，確率的要素が小さすぎると，その時点の学習効率は良くなるが，それよりも良い行動があっても，なかなかそれを発見できないため，将来的な学習効率を考えると得策ではない．よって，ロボットを適用する環境に応じて適切に設定する必要がある．ε は一般に設計者が経験に基づいて設定している．そのため，あらかじめ類似の問題についてシミュレーション実験を行って，適切なパラメータ値を探っておく必要がある．その設定自身に試行錯誤を要してしまうため，強化学習本来の良さが損なわれてしまうことがある．一般的に，ε の設定指針は現在のところ存在しない．

1.5 従来研究

強化学習における，探査と利用のトレードオフに関しては，宮崎らの一連の研究がある [30,31]．宮崎は環境同定を目的とした環境同定器と報酬獲得を目的とした報酬獲得器とからなる行動決定部を有する強化学習システム (MarcoPolo) [31] を提案し，ユーザが指定する任意のトレードオフ比に基づく強化学習を実現できることを示した．しかし，MarcoPolo では設計あるいは運用の段階でユーザが環境同定と報酬獲得のトレードオフ比を陽に与える必要があり，その設計指針は試行錯誤に頼らざるを得ない．

近年では，探索-利用のバランスに関わるパラメータをいかに制御するかという点に着目し始めた．まず，遺伝的アルゴリズムを用いて強化学習パラメータを決定する研究がいくつかある [32–35]．しかし，遺伝的アルゴリズムを用いたことで，計算時間がかかるのと，GA のパラメータ調整に手間がかかるという点から根本的解決には至っていない．また，エージェント自身がパラメータを調整するのではなく，最適な値の範囲を拡大する研究もある [36]．しかし，これも設計者が，パラメータを設定する必要があり，根本的な解決にはいたっていない．

1.6 研究目的

本研究ではロボットが環境に応じて自律的に探索-利用のバランスを調整することを目的とする．

1.7 アプローチ概要

強化学習ではエージェントの目的はできるだけたくさんの報酬を得ることである．そのためには，エージェント自身が報酬をどれだけ得られるかという予測が必要になる．もしも，予測が可能である，つまり状態遷移が既知であるならば，今所持している知識 (例えば，Q 値) を利用して報酬予測をすればよい．また，もし予測が不可能であれば，報酬の予測が立つまでの知識を獲得していないため，知識を獲得しにいくことが望ましい．では，この予測可能であるか不可能であるかという判断を情報量で行う．例えば，情報量が低いならば，状態の遷移先の候補が少なく，予測しやすいので，今までの知識を利用し，報酬を獲得しに行くことが望ましい．反対に，情報量が高いならば，状態の遷移先の候補は多く，その状態に遷移するかの予測は困難になるので，探査をすることが望ましい．このように情報量の大小に応じて，探査率 ε を動的に制御することを提案する．

1.8 本論文の構成

1 章ではロボットという言葉が誕生した過程から，ロボットの普及や機械学習，特に強化学習について概要を述べた．特に本研究で扱う強化学習のことについて触れ，強化学習の問題点と本研究の目

的，問題解決へのアプローチの概要を述べた．

2 章では本研究で対象とする強化学習について具体的に述べる．さらに強化学習における問題点を実験とともに検証する．

3 章ではトレードオフ問題が起こっていることと，環境に応じて最適な ε が異なることを実験によって示す．

4 章では 2 章・3 章で述べた問題点を解決するためのアプローチや具体的な手法の提案を行う．

5 章では 3 章で提案した手法の有効性を検証するため実験を行う．ここでは特にシミュレーションに限定して実験を行う．

6 章では本論文のまとめを行う．さらに今後の課題についても述べる．

第2章　強化学習

本章では，本論文で対象とする強化学習について述べる．はじめに，第 1 節では，強化学習の概要について述べる．第 2 節では，強化学習におけるエージェントの行動評価手法について述べる．第 3 節では，強化学習におけるエージェントの行動選択法について述べる．最後に第 4 節では，強化学習における問題点の一つである探索-利用のトレードオフ問題について述べる．

2.1　強化学習概要

2.1.1　強化学習とは？

強化学習は，学習者 (ロボットなど) が現在の状態を認識し，環境に適した行動を獲得する学習法である．学習者は環境との相互作用を繰り返しながら，行動の結果によって環境から与えられる報酬を基に自らの行動を改善する．受け取った報酬を元に学習者は自身の行動の良し悪しを判断し，より環境に適応しようとする．Fig. 2.1 に強化学習の概要図を示す．

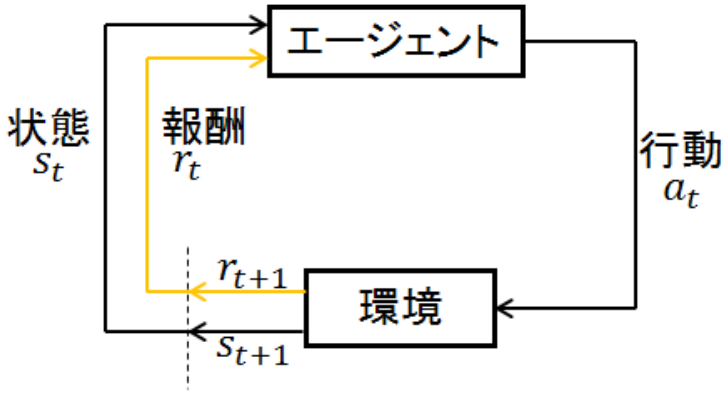


Fig. 2.1　強化学習の概要

教師ありでは状況を予測し，正しい出力にあった教師を設定しなければならない．しかし，場合によっては教師を設定できないような環境に陥ることも考えられる．強化学習であれば，教師あり学習

のように行動にいちいち教師を設定しなくても，学習者に達成させたい目的に合わせて報酬を設定すればよい．そのため，人間が学習者を適用する環境を知る必要がなく，学習者は未知の環境でも学習することが可能になる．

2.1.2 強化学習の構成要素

ここでは，強化学習の構成要素である，エージェント・環境・行動学習手法・報酬・価値について述べる．また，強化学習の要素間の関係を Fig. 2.3 に示す．

- エージェント

  学習者 (ロボット) のことを指す．本論文では，学習者 (ロボット) のことをエージェントとする．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エージェン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

  報酬は強化学習問題において目標を定義する．目標は設計者がエージェントに学習させる状態や行動である．この関数は状態行動対を報酬という数値情報として出力する．報酬は現在の状態に備わった望ましさを表している．ゆえに，報酬関数は即時的な意味合いでエージェントにとって何が良いのかを示している．一般に報酬関数は設計者が設計するものでエージェントが変更することはない．

- 行動価値

  報酬が即自的な意味合いで何が良いのかを示しているのに対して，行動価値は，最終的な状態または行動の価値を決定する．価値とは，エージェントがその状態を起点として将来にわたって入手できる報酬の期待値である．報酬はその環境が即時的で固有の望ましさを決定するのに対して，価値はその後に続きそうな状態郡とそれらの状態郡で得られそうな報酬を考慮に入れた上での長期的な望ましさを示すものである．例えば，ある状態では常に低い報酬しか得られ

ないかもしれないが，高い報酬が得られるような状態が規則的にそれに続くのならば，高い価値を持つ．

本論文では，行動価値を Q 値と表す．ここで，$Q(s_t, a_t)$ は時間 t において，状態 s_t で行動 a_t を選択したときの行動価値を表す．この Q 値を集めた空間を Q 空間とする．Q 空間は状態軸と行動軸，Q 値軸で構成される．状態軸はエージェントが認識できる状態から構成され，行動軸はエージェントが取ることができる行動から構成される．Q 空間の例を Fig. 2.2 に示す．例えば，エージェントが認識可能な状態を s_1, s_2, s_3 とし，取ることができる行動が a_1, a_2, a_3 の 3 通りあるとする．このとき Q 空間は 9 つの Q 値 $Q(s_1, a_1), Q(s_1, a_2), Q(s_1, a_3), Q(s_2, a_1), Q(s_2, a_2), Q(s_2, a_3), Q(s_3, a_1), Q(s_3, a_2), Q(s_3, a_3)$ から構成される．

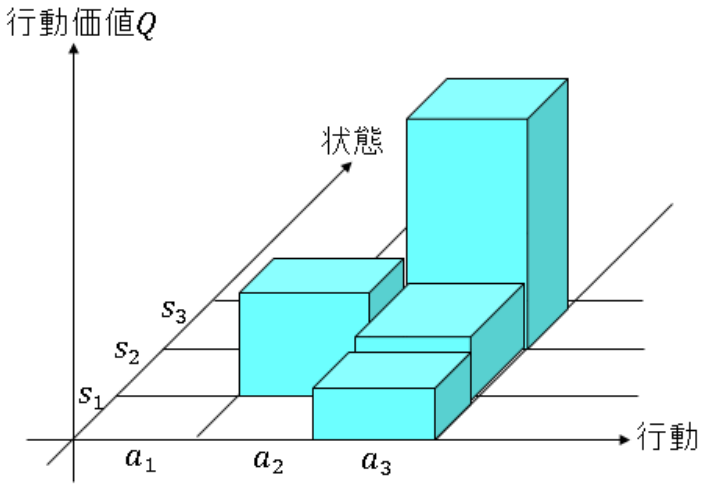


Fig. 2.2　Q 空間の例

- 行動価値推定手法

  価値関数は報酬関数を基に更新されてゆく．行動学習手法は報酬関数をもとに価値関数を更新する手法である．例えば，過去に得られた全ての報酬の平均するといったように，どのように価値関数を更新するかを規定したものが行動学習手法である．詳しくは 2.2 節で述べる．

- 行動選択手法

  行動を決定する手法である．例えば，ある状態で行動を決定する際にその行動をランダムに決定する手法，今までで最も高い報酬を得られた行動に決定する手法，過去の報酬の累積から確率的に決定する手法，というように行動の決定のルールが行動選択手法である詳しくは 2.3 節

で述べる．

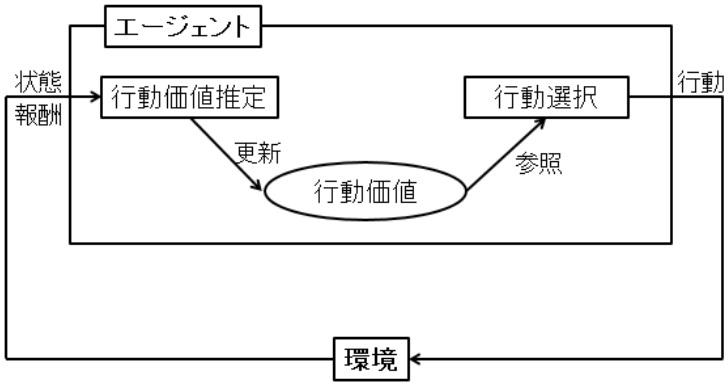


Fig. 2.3　強化学習の要素

2.1.3　強化学習の流れ

強化学習の流れを Fig. 2.4 に示す．環境から知覚した状態 s_t によって，エージェントは自身の選択可能な行動から，行動 a_t を選択し実行する．この際エージェントは状態 s_t における行動価値に基づき，行動選択を行う．学習者は適用している行動選択法によって選択する行動が変化する．行動の結果得られた報酬 r_t に基に，エージェントは状態 s_t における行動 a_t の価値 (行動価値) $Q(s_t, a_t)$ を更新する．この際エージェントは行動価値推定手法を用いて価値の更新を行う．この繰り返しによって学習者は学習を行い，学習者が次回同様の状態に直面した際においての行動選択に活かす．

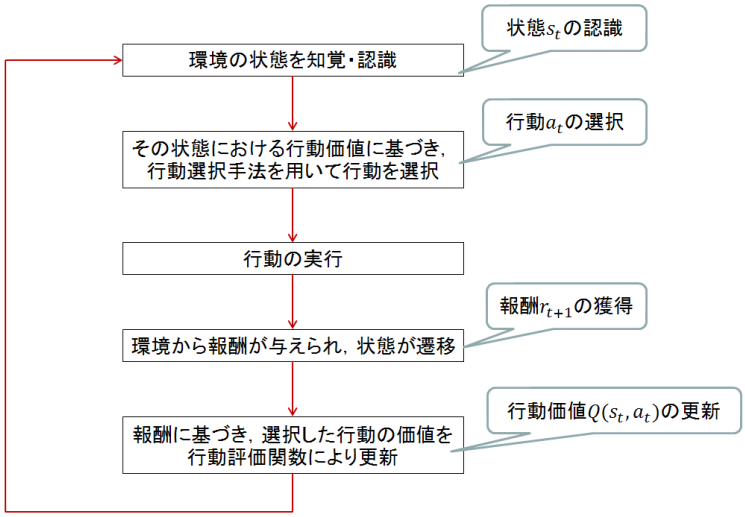


Fig. 2.4　強化学習の流れ

2.2　行動価値推定手法

本節では，強化学習におけるエージェントの行動価値の推定手法について述べる．強化学習において，エージェントは行動の真の価値そのものを知ることはできないため，毎回の行動によって得られる報酬からその行動の真の価値を推定する．そして，その推定価値を使って行動選択手法を通して行動を選択する．この行動の真の価値を推定するための方法が行動評価手法である．本節では，行動価値評価手法として，標本平均手法，加重平均法，Q 学習法の 3 手法について述べる．

2.2.1　標準平均手法

標本平均手法では，ある行動が選ばれたときに実際に得られた報酬を単純に平均化していく．平均化された報酬が行動価値となる．状態 s_t，行動 a_t について標本平均手法を用いたときの時間 t での行動の価値 $Q(s_t, a_t)$ は式 (2.1) で表せる．

$$Q(s_t, a_t) = \frac{r_{s_t1} + r_{s_t2} + \cdots + r_{s_t k_{s_t,a_t}}}{k_{s_t,a_t}} \tag{2.1}$$

ここで，$Q(s_t, a_t)$ は時間 t において，状態 s_t で行動 a_t を選択したときの行動価値を表す．また，k_{s_t,a_t} は状態 s_t における行動 a_t の累計行動選択回数である．$r_{s_t1} + r_{s_t2} + \cdots + r_{s_t k_{s_t,a_t}}$ は状態 s_t において行動 a_t が選択されたときのそれぞれの時間における獲得報酬を表す．

2.2.2 加重平均法

加重平均法は，遠い過去の報酬よりも最近に受け取った報酬を重要視する推定手法である．より直近に受け取った報酬に重みを与えるため，定数値のステップサイズ・パラメータ α を使用する．行動 a_t を取ったときの行動価値 $Q(s_t, a_t)$ を更新するための更新式は式 (2.2) で表せる．ステップサイズパラメータ α は $(0 \leq \alpha \leq 1)$ であり，これが大きいほど，より最近の報酬を重視する．また r_{t+1} は，報酬を表す.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_t - Q(s_t, a_t)] \tag{2.2}$$

2.2.3 Q 学習

Q 学習 [37] は，目標状態のみ報酬を与えるような遅延報酬でも利用できる手法である．Q 学習では，行動の結果，遷移した先の状態の行動価値によって現在の行動価値を更新する (Fig. 2.5)．例えば，迷路問題の場合，ゴール時にもらえる報酬値をスタートからゴールまでのルートに対し伝播させることが可能となる．これによりスタートからゴール間までのそれぞれの状態に対して，その状態の価値が算出されるため学習が可能となる．また，Q 学習の利点は，環境が離散有限 MDP 環境であれば，十分な試行により Q 値が最適地に収束し，最適政策を獲得できることが保証されている．そのため，多くの研究で用いられている [38, 39].

Q 学習における行動評価式は，式 (2.3) で表せる．ここで，現時刻 t の状態と次時刻 $t+1$ 遷移後の状態をそれぞれ s_t，s_{t+1} とする．状態 s_t における行動 a_t の価値を $Q(s_t, a_t)$ とする．r_{t+1} は，遷移先で獲得できる報酬である．$max_a Q(s_{t+1}, a_t)$ は遷移先の状態が持つ最大の Q 値を示す．$\alpha(0 \leq \alpha \leq 1)$ は学習率であり，価値の更新量を調整するものである α の値が大きければ，更新量は大きくなり，小さければ更新量は小さくなる．$\gamma(0 \leq \alpha \leq 1)$ は割引率であり，遠い将来の報酬ほど割引いて考えることを表している．γ の値が大きければ遠い将来の報酬まで考慮することになり，小さければ即時的な報酬を優先することになる．

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{2.3}$$

以下に，Q 学習における学習の流れを記述する．

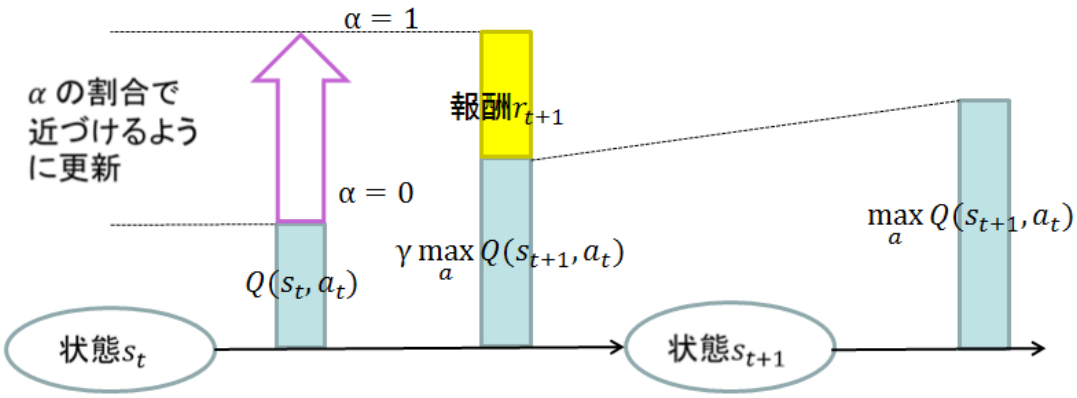


Fig. 2.5　Q 学習における Q 値更新

1. エージェントは環境の状態 s_t を観測する．
2. エージェントは任意の行動選択方法 (探索戦略) に従って行動 a_t を実行する．
3. 環境から報酬 r_t を受け取る．
4. 状態遷移後の状態 s_{t+1} を観測する．
5. 式 (2.3) により Q 値を更新する．
6. 時間ステップ t を $t+1$ に進めて手順 1 へ戻る．

2.3　行動選択手法

本節では強化学習における行動選択法について述べる．行動選択手法では，前節で推定した価値を元に行動を選択する．行動選択の際に重要となるのは，単に現在の推定価値が最大となる行動を選択するのみだけではなく，より価値の高い行動を求める探索を行うことである．ここでは，greedy 法，ε -greedy 法，softmax 法について述べる．

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が高いと評価された行動を選択する．例を Fig. 2.6 に示す．Fig. 2.6 の例で言うと，行動価値が最も高い行動は a_2 であるため，a_2 を選択する．

この greedy 法は常に直面する状態の価値が最も高い行動を選択するが，価値が低いと評価された行動は一切選択しない．すなわち，価値が低いと評価される行動に対しては，その行動の価値が一時的に低いだけで，本当は価値の高い行動であるという可能性を確かめるための試行を一切行わないそのため，その行動価値が一時的に低いだけで，本当は価値が高い行動であるという可能性を考慮しない．

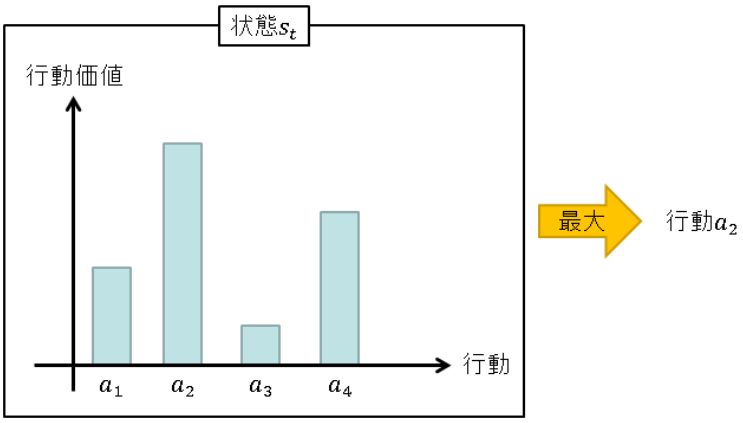


Fig. 2.6　greedy 法の例

2.3.2　ε -greedy 法

ε -greedy 法とは基本的には greedy 法同じく価値が最も高い行動を選択するが，探査率 $\varepsilon(0 \leq \varepsilon \leq 1)$ で行動価値に関わらすランダムに行動を選択する．例を Fig. 2.7 に挙げる．$1-\varepsilon$ の確率で greedy 法と同じように最も価値が高い行動 a_3 を選ぶ．また，ε の確率で a_1〜a_4 の行動のうちいずれか一つをランダムに選択する．

ε -greedy 法の利点は，小さい確率 ε で探索を行うことである．これによって，greedy 法の欠点が解消される．ε -greedy 法の欠点としては，確率 ε における行動選択の際に，行動価値に関わらずランダムに行動を選択することが挙げられる．このため，確率 ε での行動選択において，ほとんど最悪と思われる行動を選択する可能性とほぼ最適に近い行動を選択する可能性が同じくらいに高くなるということがある．

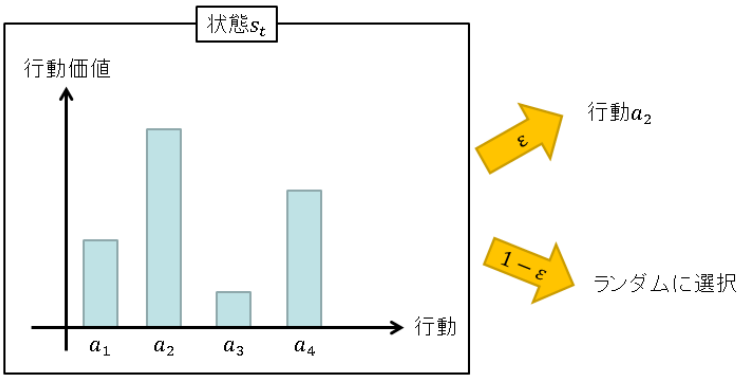


Fig. 2.7　ε-greedy 法の例

2.3.3　softmax 法

softmax 法は，各行動の価値の推定値 (Q 値) と温度定数 τ により行動選択確率が決められる．各行動には，それぞれの Q 値により重み付けされた選択確率が決められる．

softmax 法では，ε -greedy 法と異なり，行動選択確率 $\pi_t(s_t, a_t)$ を行動選択に使用する．softmax 法では行動 a の価値が相対的に見て他の行動価値よりも高い程，行動 a の選択される確率が 1 に近づく．具体的には，時間 t における状態 s で行動 a を選択する確率 $\pi_t(s, a)$ は式 (2.4) で与えられる．

$$\pi(s_t, a_t) = \frac{\mathrm{e}^{Q(s_t,a_t)/\tau}}{\sum_{b=1}^{n} \mathrm{e}^{Q(s_t,b)/\tau}} \tag{2.4}$$

ここで，τ は温度と呼ばれる正定数である．τ が大きい場合は全ての行動がほぼ同程度に選択されるように選択確率が設定される．τ が低い場合には，行動価値の異なる動作において選択確率の差がより大きくなるように設定される．そして，$\tau \leftarrow 0$ の極限において，softmax 法は greedy 法と一致する．また分母は，行動に関する選択確率の和を 1 にするように正規化の役割をしている．

例を Fig. 2.8 に挙げる．この例では，行動価値は大きい順に a_2, a_4, a_1, a_3 となっている．よって，行動選択確率も大きい順に $\pi(s_t, a_2), \pi(s_t, a_4), \pi(s_t, a_1), \pi(s_t, a_3)$ となる．

softmax 法では，価値関数に応じて選択確率を変化させるため，前述した ε -greedy 法の欠点を解消することができる．しかし，softmax 法ではパラメータの設定に，行動価値の数値にある程度見通しが立っていることと，e のべき乗の性質をよくわかっていることが必要である．

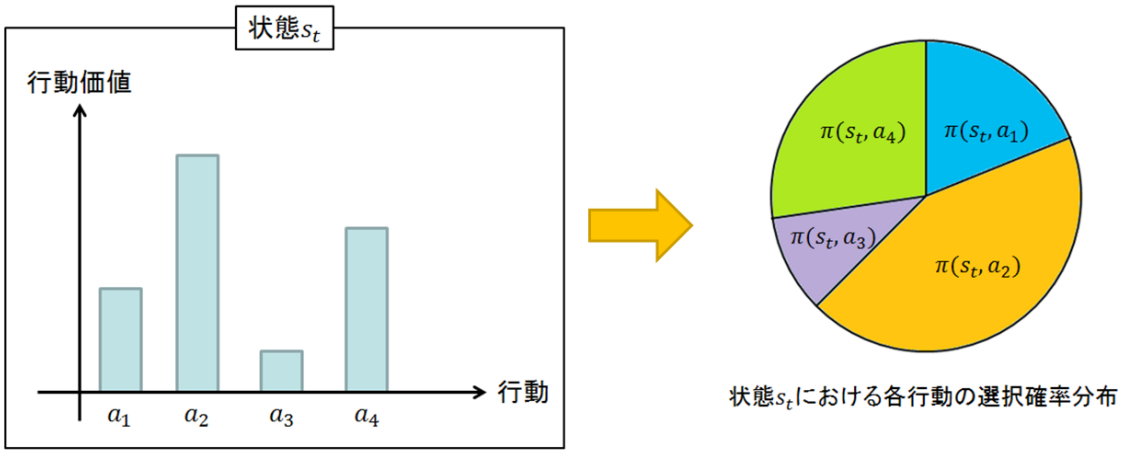


Fig. 2.8　softmax 法の例

2.4　強化学習における問題点

強化学習の問題点の一つに探索-利用のトレードオフ問題がある．強化学習を行うロボットの目標はより多くの報酬を得ることである．多くの報酬を獲得し続けるには，過去に試みた行動の中で，最も多くの報酬を獲得できるような行動を取り続けなければならない．このような行動を取り続けることは，ロボットが所有している知識を利用 (exploitaiton) していることとなる．しかし，現在所有している知識が最適なものとは限らない．そのため，より多い報酬を獲得するためには過去に試みていない行動を行わなければいけいない．つまり，未知の状態を経験することが必要である．このような未知の状態を経験するために行動することを探索 (exploration) という．探索と利用の行動のうち，たとえ同じ状態でもその状態に対して，探索行動を取るべきか利用行動を取るべきかということは一意に定まらない．なぜなら，探索すべきか利用すべきか，というのはロボットの学習環境に応じて変化するためである．よって，ロボットが探索と利用のバランスを自律的に制御することが必要である．

強化学習において，探索と利用のバランスは行動決定法のパラメータによって決定する．本論文では ε-greedy 法を対象にする．2.3 節で述べたとおり，ε-greedy 手法は，現在の行動価値が最も高い行動 (グリーディな行動) を $(1-\varepsilon)$ の確率で選択するか，小さい確率 ε でランダムに行動を選択するという手法である．$(1-\varepsilon)$ の確率で行う，現在の行動価値が最も高い行動の選択が利用に相当し，ε の確率での，ランダムな行動が探索に相当する．ε が小さいほど，現時点で最適とされる行動が行われる回数が多くなるが，真に最適な行動を見つけ出すまでに時間がかかってしまう．それゆえ，ε の値を調整し，探索と知識利用のバランスの取り方を考える必要がある．

強化学習エージェントが学習を行う学習環境は全て同じものではなく，大きさや複雑さが異なる．学習環境の大きさ・複雑さなどによって最適な ε の値は変化する．従来では，ε の値は学習環境に応じて人間が設定していた．しかし，学習環境に応じて，適切な ε の値は変化するため，設定には手間がかかってしまう．

第3章　n本腕バンディットにおけるトレードオフ問題の検証

本章では，N 本腕バンディット問題について述べ，この問題における探索-利用の問題点について述べる．

3.1　N 本腕バンディット問題とは？

本節では，本実験の適用タスクである N 本腕バンディット問題について述べる．N 本腕バンディットとは，N 本のレバーのあるスロットマシンのようなものである．このレバーのことをバンディット問題では腕と呼ぶ．N 本腕バンディットのイメージを Fig. 3.1 に示す．各腕には獲得報酬が設定されており，腕を引く毎に「当たり」か「はずれ」が決定する.「当たり」を引くと，その腕に設定された分の報酬を獲得することができる．ただし，エージェントはどの腕が「当たり」であるかということは事前に知らない．N 本腕バンディット問題におけるエージェントの目的とは最も報酬が得られる腕探し出し，引き続けることである．

N 本腕バンディット問題においては，如何にして高い報酬を得られる腕を引き続けることが重要になる．エージェントはより高い報酬を得られる腕を探し出し，引き続けることが必要である．

N 本腕バンディット問題におけるエージェントの目的は，より多くの報酬が得られる腕を引くことである．多くの報酬を得るためには過去に引いた腕の中で，最も多くの報酬が得られるような腕を引き続けなければならない．しかし，現在所有している知識が最適とは限らない．つまり，現在引いている腕が最高の報酬を得られるとは限らない．そのため，最高の報酬を得るためには，未だ引いていない腕を引いてみることが必要である．なぜなら，そのまだ引いていない腕により多くの報酬が割り当てられているかもしれないからである．

しかし，強化学習では探索-利用の度合いのパラメータを一意に決めることが多い．本実験では，行動選択手法の代表的な手法である ε-greedy 法を用いる．複数の ε において実験を行うことで，報酬獲得量に影響が出ることを示す．

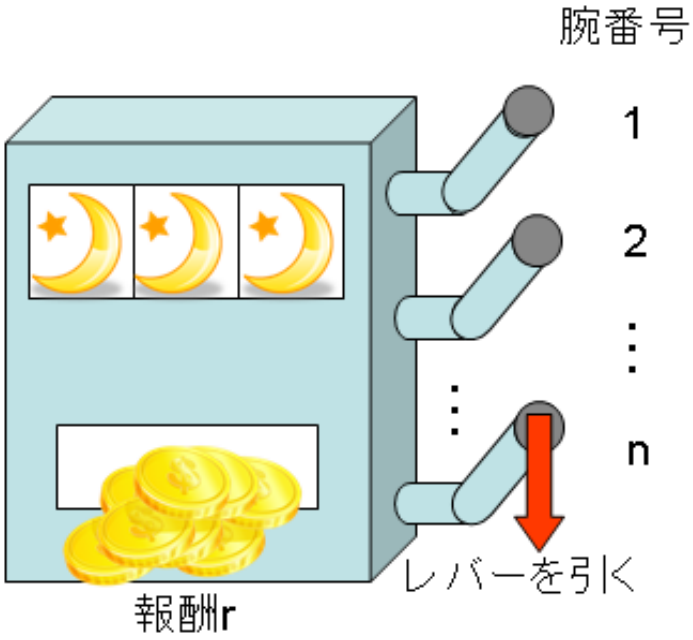


Fig. 3.1　N 本腕バンディット問題のイメージ

3.2　実験目的

強化学習において学習中は探索と利用のバランスを決定するパラメータは学習中に変更せず一意に決定されることが多い．そのため，既知の状態に関して，探索するといった不要な行動を取ってしまうこともある．本実験では，上記のような現象が起き，獲得報酬が低下してしまうという問題点を検証する．

3.3　実験設定

まず，Fig. 3.2 に実験概要を示す．まず，M 台のエージェントを用意する．そして，M 台のエージェントにバンディット問題を適用する．エージェントに適用するバンディット問題の腕や報酬等の設定はすべて共通の設定を用いる．また，各エージェントは同じ行動評価手法，行動選択手法を用いる．ただし，行動選択手法における探索-利用のバランスを決定するパラメータを各エージェントで異なった値に設定する．本実験では ε-greedy 法の ε を探索-利用のバランスを決定するパラメータとする．そして，上記の条件の下で各エージェントにおける獲得報酬を比較する．

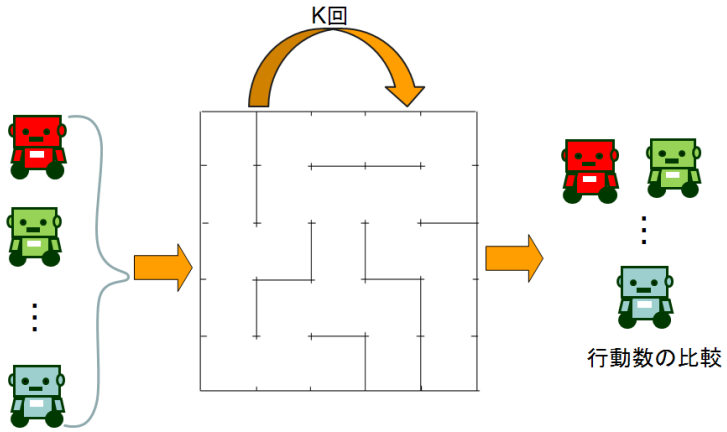


Fig. 3.2　実験概要

3.3.1　エージェントの設定

本実験で用いるエージェントは全て共通の行動評価手法，行動選択手法を用いる．これはバランス決定のパラメータの値 (本実験では ε に相当する) の違いにおいて結果の比較を行うためである．本実験では，エージェントの行動評価手法に標本平均化手法を用いた．標本平均化手法の式を式 (3.1)

に示す.

$$Q_t(a) = \frac{r_1 + r_2 + \cdots + r_{k_a}}{k_a} \tag{3.1}$$

ここで，$Q_t(a)$ は時間 t において行動 a を選択したときの行動価値を表す．k_a は行動 a の累計選択回数を表す．$r_1, r_2, \cdots, r_{k_a}$ は行動 a が選択されたときの各選択における獲得報酬を表す.

また行動選択手法には ε-greedy 法を用いる．ε-greedy 法とは ε の確率でランダムな行動を取り，それ以外では greedy な行動を取るという行動選択手法である．本実験では，ε の値のみを変化させたエージェント間において比較を行う．本実験ではエージェントは 6 台用意し，各エージェントの ε の値はそれぞれ 0.00，0.01，0.05，0.10，0.15，0.20 とする．

Table 3.1　エージェントの設定

エージェント番号	1	2	3	4	5	6
ε	0.00	0.01	0.05	0.10	0.15	0.20

3.3.2　タスクの設定

Table 3.2 に本実験で使用するバンディットの設定を示す．本実験では腕の総数は 10 本とする．本タスクの最適値 (1 回腕を引くことによる最大獲得報酬量) は 1.5 である.

Table 3.2　バンディットの各腕に対する報酬

腕の番号	1	2	3	4	5	6	7	8	9	10
報酬	0.0	0.1	0.0	0.2	0.3	1.2	1.0	0.8	0.5	1.5

3.3.3　パラメータの設定

Table 3.3 に本実験のパラメータを示す．ここではエージェントが腕を 1 回選択する (腕を引く) ことを 1 試行とする.

Table 3.3　各パラメータの値

項目	内容
試行回数	1000 回
行動選択手法	ε -greedy
ε	0.00,0.01,0.05,0.10,0.15,0.20
行動評価手法	標本平均化手法
行動価値 Q の初期値	0.00
ロボットの台数 M	6 台

3.4　実験結果

次に，各エージェントが同一タスクを 2000 回行ったときの各試行における平均獲得報酬の推移を見る．結果を Fig. 3.3 に示す．グラフの横軸が試行数であり，縦軸が各試行数における平均獲得報酬を表している．この結果を見ると ε の値が小さいエージェント ($\varepsilon = 0.01$) は学習収束は早いが，最適値に到達するまで時間がかかるといった欠点が存在する．逆に，探索を重視しすぎると ($\varepsilon = 0.15$ や $\varepsilon = 0.20$) のように，最適値に到達しているにもかかわらず獲得報酬が下がってしまうということが起こる．このように，利用を重視すると最適値に収束する時間が遅くなるが，収束後の学習は安定する．逆に，探索を重視すると，最適値を素早く発見できるが，発見後も探索的な行動 (ランダムな行動) を取り続けているため，学習が安定せず，結果として獲得報酬が低くなってしまうということが挙げられる．

Fig. 3.5 は各エージェントの獲得報酬の比較である．Fig. 3.5(a) が $\varepsilon = 0.00$, Fig. 3.5(b) が $\varepsilon = 0.01$, Fig. 3.5(c) が $\varepsilon = 0.05$，Fig. 3.5(d) が $\varepsilon = 0.10$，Fig. 3.5(e) が $\varepsilon = 0.15$，Fig. 3.5(f) が $\varepsilon = 0.20$ のときの各試行の獲得報酬量をそれぞれ表している．横軸が試行回数を表し，縦軸が獲得報酬を表している．

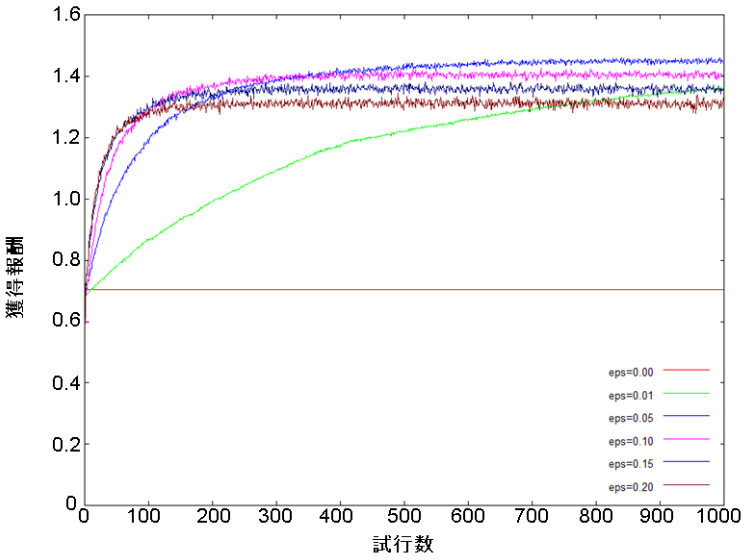


Fig. 3.3　各エージェント間の平均報酬量の比較

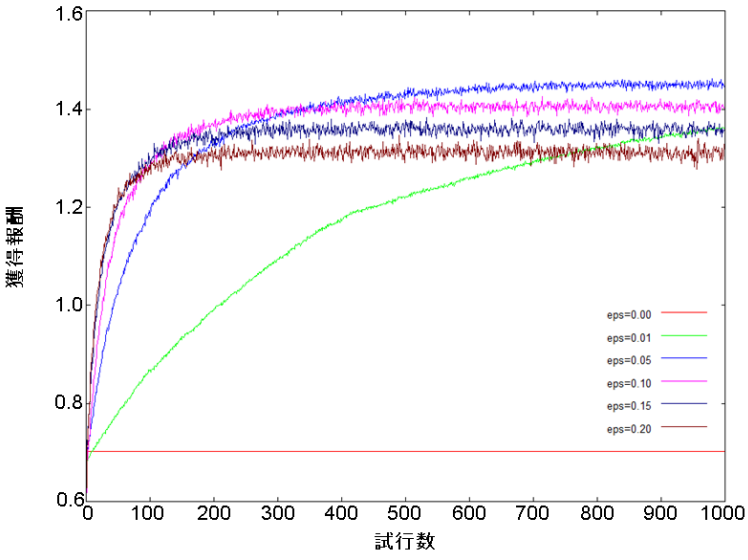
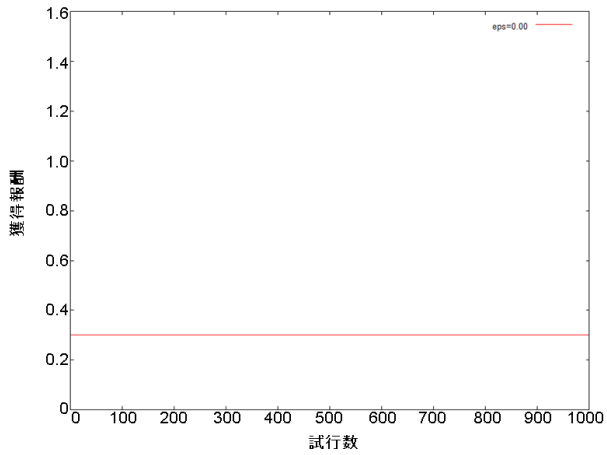
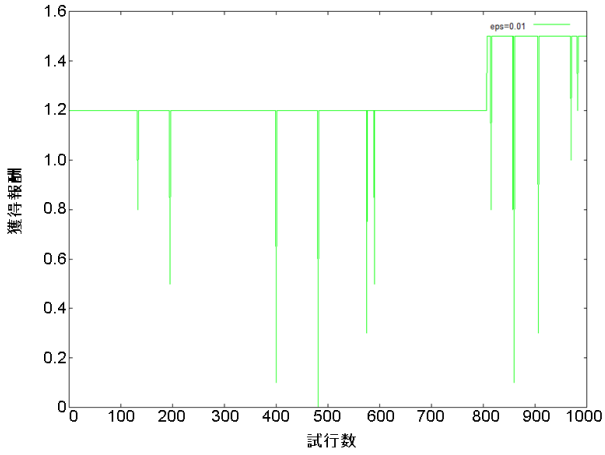


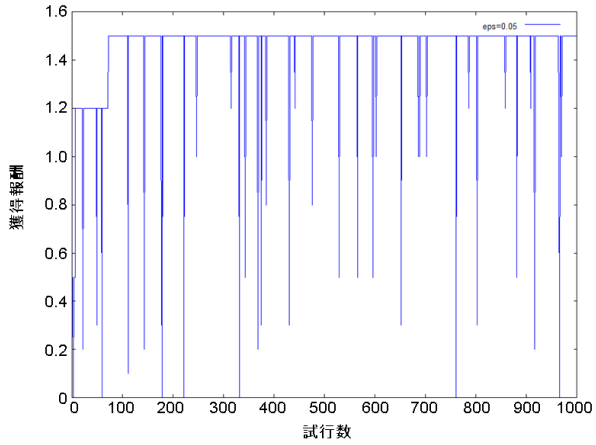
Fig. 3.4　Fig. 3.3 の拡大図



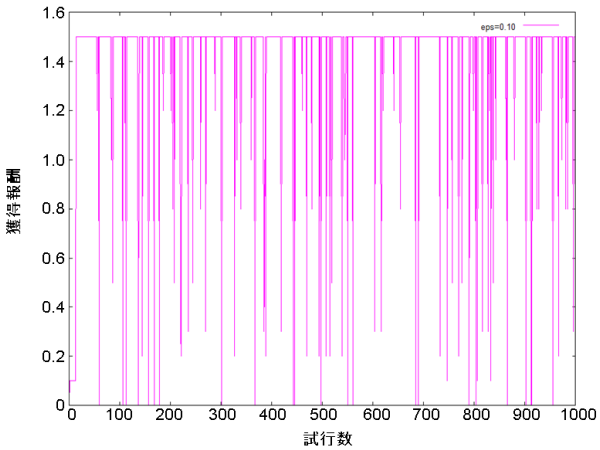
(a)$\varepsilon = 0.00$



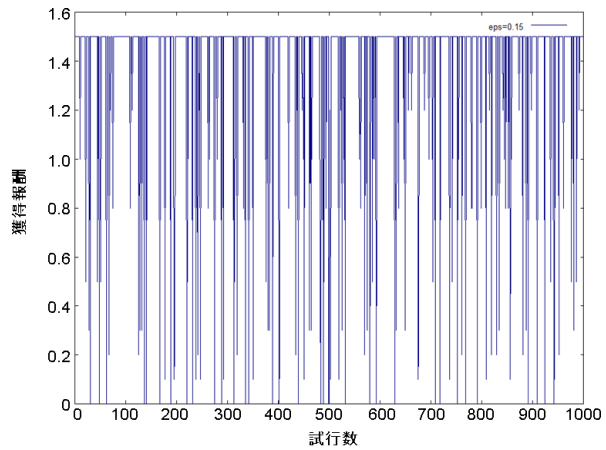
(b)$\varepsilon = 0.01$



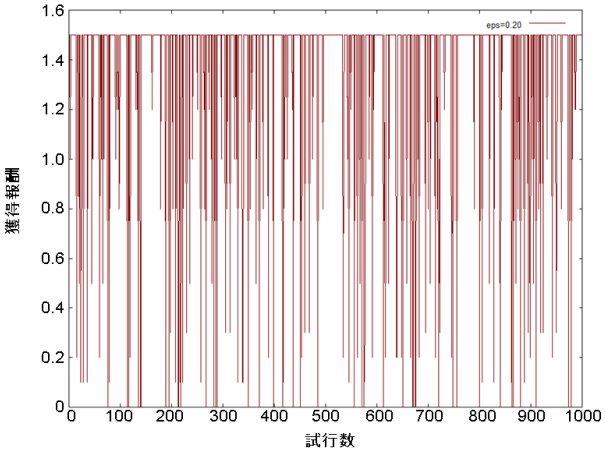
(c)$\varepsilon = 0.05$



(d)$\varepsilon = 0.10$



(c)$\varepsilon = 0.15$



(d)$\varepsilon = 0.20$

Fig. 3.5　獲得報酬の推移

まず，$\varepsilon = 0.00$ について述べる．$\varepsilon = 0.00$ の場合，探索を行わないため，1 試行目で引いた腕に報酬が割り当てていれば，その腕を引き続ける．最初の 1 試行目だけは Q 値が全て同じ値であるため，どれかの腕をランダムに引く．ランダムに引いた腕に報酬が割り当ててあれば，その腕の Q 値が最も高くなるため，その腕を引き続ける．$\varepsilon = 0.00$ は他に高い報酬が得られる腕があることを考慮しないため，常に同じ腕を引き続ける．そのため，獲得報酬量が他の ε に比べて，格段に低下する．

そのほかの ε は，いずれも最適解 (本実験の場合，1.5) を引いている．さらに，ε が大きいほど，最適解を見つけるのが早いのがわかる．例えば，$\varepsilon = 0.01$ では 800 試行付近で最適解を発見しているが，$\varepsilon = 0.05$ では，より早い段階の 100 試行付近で発見している．$\varepsilon = 0.10$，$\varepsilon = 0.15$，$\varepsilon = 0.20$ では 100 試行未満で既に，最適解を発見している．

しかし，最適解を発見した後も，ε が大きいほど，ランダムな行動を一定の割合で取り続けている．$\varepsilon = 0.20$ では早い段階で最適値を発見できているのに関わらず，$\varepsilon = 0.20$ の確率でランダムに腕を選択するため，報酬値が低い腕を選択していることが多い．そのため，最適解を発見後も学習が安定せず，低い報酬を取る回数が多くなっていることがわかる．逆に $\varepsilon = 0.01$ や $\varepsilon = 0.05$ の場合，最適解の発見は遅いが，最適解発見後は，報酬値が低い腕を選択することもあるが，$\varepsilon = 0.20$ よりも多く報酬値が高い腕を選択していることがわかる．

以上の結果から，ε を固定パラメータとすると，問題が生じることがわかった．知識利用を重視するように ε の値を低く設定すると，最適解の発見が遅れ，獲得報酬が低下する恐れがある．しかし，逆に ε の値を高く設定すると，最適解発見の後も探索行動 (ランダムな行動) をし，無駄な行動を取ることが多くなり，これも獲得報酬の低下につながる．そのため，パラメータを固定値にするより，ロボットが学習の進行度や環境に合わせて調整することが，獲得報酬の向上につながる．

第4章　探索-利用バランスの自律的調整の提案

本章では，第 4.1 節で，本研究が提案する学習機構の概要を示す．第 4.2 節では，エージェントが環境の遷移確率を算出し，そこから情報量を算出する．情報量を算出することで，環境に対する学習を行う．最後に第 4.3 節では，情報量と学習の進行度から ε-greedy 法における ε を算出する．

4.1　提案手法の概要

本論文では，我々は探索と利用のジレンマを解消することを目指す．我々は特に行動選択法のパラメータによって，探索－利用の度合いを調整していることから，パラメータを学習者が学習中に直接制御することを提案する．本論文では，特に ε-greedy 法に着目し，パラメータ ε を学習者が情報量を用いて制御することを目指す．ここでの情報量とは環境の遷移確率を基に算出したものであり，環境遷移に関する情報量となる．

ε と情報量が比例関係にあると仮定する．例えば，情報量が高いときは遷移先が多く，遷移先が一意に確定できない．また，遷移先が等確率のときも情報量が高くなる．反対に，情報量が低いときは遷移先が常に固定，遷移先がある程度に絞られる．

提案手法の概要を Fig. 4.1 に示す．提案手法は「経験情報の獲得」・「探査率 ε の算出」の 2 つのモジュールに分かれている．「経験情報の獲得」では，環境のモデルを知るために，エージェントは行動回数と遷移回数を行動毎に記録する．その情報からある状態のときある行動を取ると，どのくらいの確率で次状態に遷移するかという遷移確率を算出する．このモジュールに関しては 4.2 節で詳細を述べる．「探査率 ε の算出」では「経験情報の獲得」で算出した遷移確率を用いて，各状態行動対に対して遷移先がどの程度ランダム性を持っているのかという平均情報量を算出する．そして，その平均情報量から ε を算出する．このモジュールに関しては，4.3 節で詳細を述べる．「経験情報の獲得」はエージェントが行動毎に行うが，「探査率 ε の算出」は試行のはじめに行う．

4.2　経験情報の獲得

「環境についての学習」では，エージェントは行動毎に環境の遷移確率を算出し，情報量を算出するモジュールである．本研究では，エージェントは環境遷移に関する情報量を用いて ε を制御する．

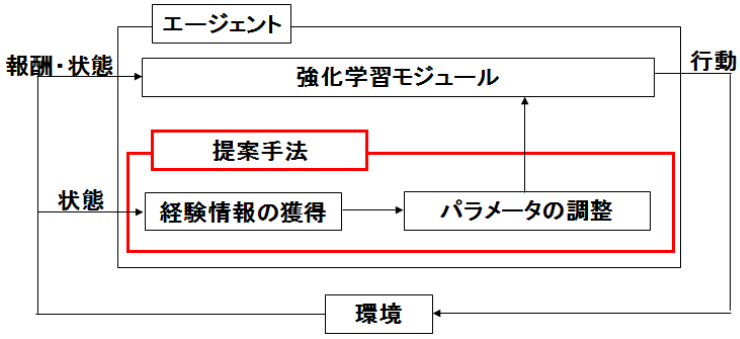


Fig. 4.1　提案手法概要

情報量を計算するためには，環境の遷移確率が必要であるが，環境の遷移確率を学習者ははじめから知ることはできない．よって，エージェントは行動するたびに，経験情報を獲得し，環境の遷移確率を自身の経験から算出する．本論文では，経験情報のことを行動回数と遷移確率とする．

まずは，環境の遷移確率の定義式を式 (4.1) に示す．

$$P_{a_j}(s_i, s_k) = Pr\{s_k | s_i, a_j\} \tag{4.1}$$

ここで，s_i はエージェントが認識している状態を表す．a_j はエージェントが選択した行動を表す．s_k は状態 s_t において行動 a_t を選択した行動を表す．

次に遷移確率の算出方法を説明する．遷移確率は式 (4.2) で計算される．

$$P_{a_j} = \frac{R(s_i, a_j, s_k)}{N(s_i, a_j)} \tag{4.2}$$

$N(s_i, a_j)$ は状態 s_i において，行動 a_j を選択した回数である．これを行動回数と呼ぶ．$R(s_i, a_j, s_k)$ は状態 s_i において，行動 a_j を選択し，次状態 s_k に遷移した回数である．これを遷移回数と呼ぶ．この 2 つをまとめて経験情報と定義する．式 (4.2) 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，環境の遷移確率とは一致しない場合がある．

経験情報の獲得例を Fig. 4.2 に示す．例では状態 s_1 において行動 a_1 を選択した際に，次状態 s_2 に遷移したとする．このときに学習者は $N(s_1, a_1)$ と $R(s_1, a_1, s_2)$ をそれぞれインクリメントする．これを繰り返すことにより経験情報を獲得する．

上に述べた方法で経験情報の獲得を行う．しかし，このままでは，一度遷移しただけで，遷移確率が 100%になってしまう．そこで，学習者がまだ知らない遷移先があるのではないかという余裕を持たせるために, 新たに未発見状態 s_u という状態を追加する．初めて遷移する状態に遷移した際に，$R(s_t, a_t, su)$ に+1 する．つまりは，遷移先の状態数と同じ値になる．Fig. 4.3 に例を示す．ここ

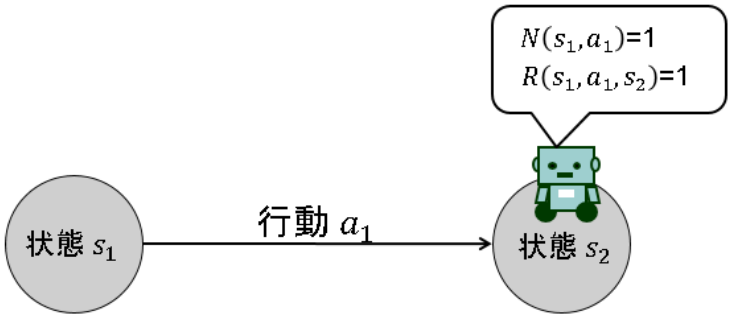


Fig. 4.2　経験情報の獲得例

では，上で述べた例と同じ例を用いる．ここでは，はじめて状態 s_2 に遷移したとする．そのときに $R(s_t, a_t, su)$ を+1 する．さらに，$N(s_t, a_t)$ も+1 する．これは，行動回数も+1 しなければ，遷移確率が 1 とならないためである．

4.3　探査率 ε の算出

ここでは式 (4.2) で求めた確率を用いて，状態 s_i における行動 a_j の平均情報量を求める．ここでは，以前に一度も選択したことのない行動はどこに遷移するか未知であるため，平均情報量は最大値を取る．これは，未知ということは考えられる状態すべてに等確率で遷移するということを想定している．一度でも行動を取った場合，確率が求められるため，環境遷移確率を用いて，平均情報量を算出する．式 (4.3)，式 (4.4) に状態 s_i における行動 a_j の平均情報量の算出式を示す．ここでは，n は状態数を表す．この状態数はエージェントが今まで訪れたことのある状態数であり，環境の状態数とは必ずしも同値にならないことに注意されたい．

$$N(s_i, a_j) = 0 \text{ のとき } H(s_i, a_j) = \log_2 n \tag{4.3}$$

$$N(s_i, a_j) \neq 0 \text{ のとき } H(s_i, a_j) = -\sum_{k=0}^{n} P_{a_j}(s_i, s_k) \log_2 P_{a_j}(s_i, s_k) \tag{4.4}$$

式 (4.3)，式 (4.4) で求めた平均情報量を用いて ε を算出する．状態 s_i における行動 a_j を足し合わせたものを ε とする． ε の式を式 (4.5) に示す．分母は分子が取りうる値の最大値となっている．ここでは n は状態数，m は行動数である．

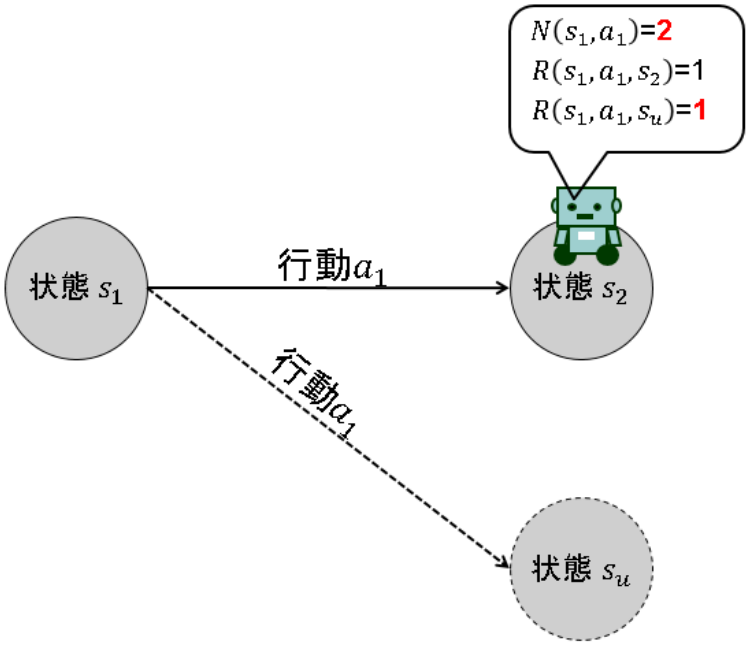


Fig. 4.3　未発見状態の扱い

$$\varepsilon = \frac{1}{nm}\sum_{i=0}^{n}\sum_{j=0}^{m}\frac{H(s_i,a_j)}{\log_2 n} \tag{4.5}$$

$H(s_i,a_j)$ は状態 s_i における行動 a_j の情報量を表している．n は状態数である．この状態数 n はエージェントが今までに認識したことのある状態数であり，環境の状態数とは必ずしも一致しない．m は行動数である．$\log_2 n$ は $H(s_i,a_j)$ が取りうる値の最大値であり，これによって $\frac{H(s_i,a_j)}{\log_2 n}$ を 0-1 の範囲になるように正規化している．

4.4　強化学習におけるパラメータの制御方法

ここでは流れを記述する．

1. エージェントは環境の状態 s_t を観測する．
2. エージェントは ε -greedy 法に従って行動 a_t を選択する．
3. 環境から報酬 r_t を受け取る．
4. 状態遷移後の状態 s_{t+1} を観測する．

5. $N(s_t, a_t)$ と $R(s_t, a_t, s_{t+1})$ をそれぞれ+1 する．
6. s_{t+1} に遷移するのが初めてであれば，$R(s_t, a_t, s_u)$ を+1 する．(同時に $N(s_t, a_t)$ も+1 する)
7. エージェントは行動評価方法に従って，状態 s_t における行動 a_t の価値を更新する．
8. 時間ステップ t を $t+1$ に進めて手順 1 へ戻る．ゴール状態であれば，次の手順へ．
9. 全状態行動対における確率を式 (4.2) によって算出する．
10. 前の手順で算出した確率を基に式 (4.3)，式 (4.4) によって情報量を算出する．
11. 前の手順で算出した情報量を基に，式 (4.5) によって次エピソードにおける ε を算出する．
12. 手順 1 に戻る

第5章　迷路問題における提案手法の有効性の検証

4章では，強化学習の問題点である，探査-利用のトレードオフ問題を解決するためのεの動的制御手法を提案した．本章では，提案手法の有効性を迷路問題において検証する．

5.1　実験目的

本実験の目的は，提案手法の有効性を示すことである．本実験ではタスクとして，迷路問題を用意し，環境の大きさに応じてεの最適値が動的に制御されていることを示す．

5.2　実験概要

Fig. 5.1に実験概要を示す．まず，M種類のパラメータεを持ったM台のエージェントを用意する．そして，M台のエージェントに迷路問題を適用する．エージェントに適用する迷路やスタートやゴールの位置などはすべて共通の設定を用いる．また，各エージェントは同じ行動評価手法，行動選択手法を用いる．これはバランス決定のパラメータの値(本実験ではεに相当する)の違いにおいて結果の比較を行うためである．本実験ではε-greedy法のεを探索-利用のバランスを決定するパラメータとする．実験はN試行数の実験をK回行う．これは1回の実験のみでは，偶然その結果が良かったものなのかが判断しにくいため，K回実験を行い，その平均ステップ数を各エージェント間で比較する．ただし，本実験の場合，迷路には1台のエージェントしかおらず，他のエージェントと衝突することは無いと考える．エージェントは上，下，右，左の4行動を取ることができ，壁に衝突した際には，同じ状態にとどまる．また，本実験ではエージェントが何らかの行動を取ることを1ステップとし，スタートからゴールに辿り着くまでを1試行とする．

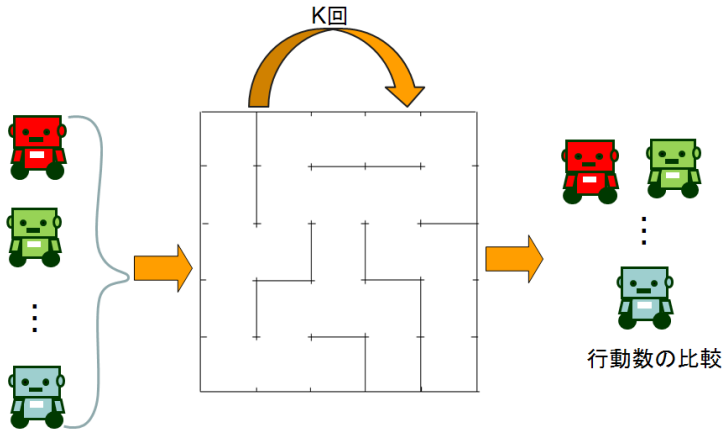


Fig. 5.1　実験概要

本実験では，エージェントの行動評価手法に Q 学習法を適用する．Q 学習の式を式 (5.1) に示す．

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{5.1}$$

ここで，現時刻 t の状態と次時刻 $t+1$ 遷移後の状態をそれぞれ s_t，s_{t+1} とする．ある状態 s における行動 a の価値を $Q(s, a)$ とする．

また行動選択手法には ε-greedy 法を用いる．ε-greedy 法とは ε の確率でランダムな行動を取り，それ以外では greedy な行動を取るという行動選択手法である．本実験では，ε の値のみを変化させたエージェント間において比較を行う．本実験ではエージェントは 12 台用意し，各エージェントの ε の値はそれぞれ 0.00，0.01，0.02，0.03，0.04，0.05，0.06，0.07，0.08，0.09，0.10 とする．そして，もう 1 台は提案手法を適用したエージェントを用意する．

5.3　静的環境下における実験

本節では，静的な環境で実験を行う．迷路問題における静的環境とは学習中に迷路の構造が変わらない実験環境である．

5.3.1　実験設定

ここでは，静的環境における迷路構造の特徴について述べる．本実験で用いる迷路はスタートからゴールまでの経路が複数ある迷路を対象とする．例を Fig. 5.2 に示す．この例ではスタートが左上で，ゴールが右下とする．

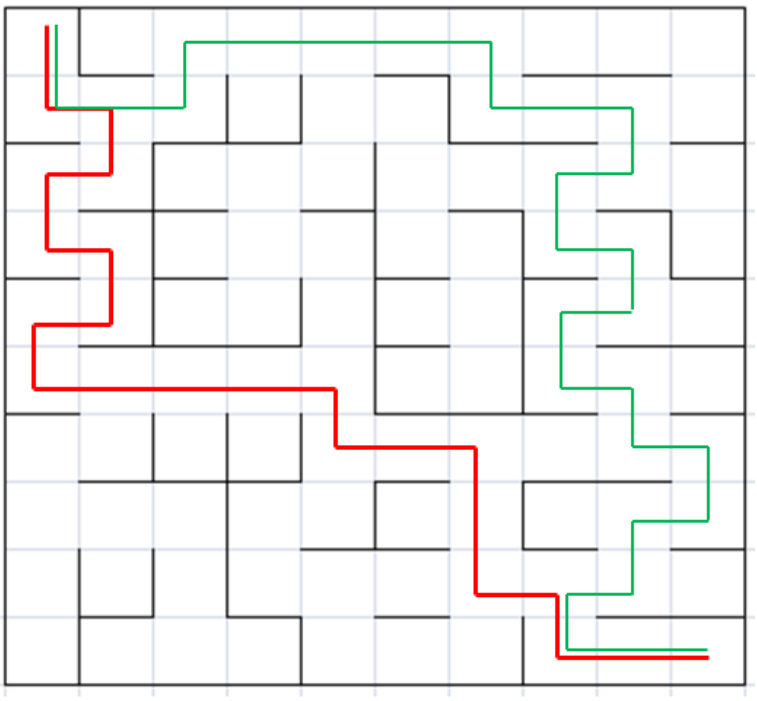


Fig. 5.2　スタートからゴールまでの経路が複数ある迷路の例

このような迷路ではスタートからゴールまでの経路が複数あるため，エージェントは複数の経路から最短経路を発見しなければならない．

本実験では迷路のサイズとして 30 × 30，40 × 40，50 × 50 の迷路でそれぞれ実験を行った．Fig. 5.3，Fig. 5.4，Fig. 5.5 が実際に使用した迷路である．赤色のマスがスタート地点を表し，青色のマスがゴール地点を表している．スタートからゴールまでの最短ステップ数は 30 × 30 が 53 ステップであり，40 × 40 が 69 ステップであり，50 × 50 が 95 ステップである．

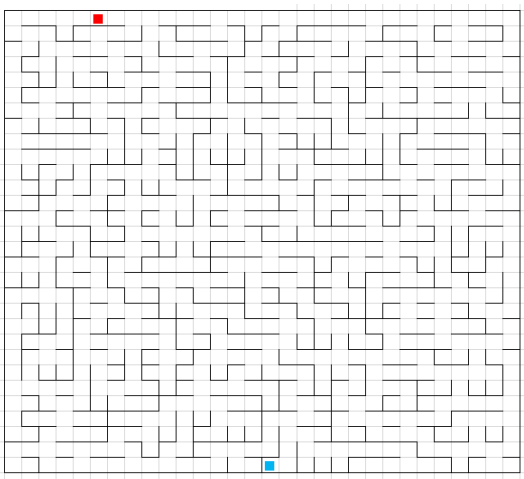


Fig. 5.3　30 × 30 の迷路

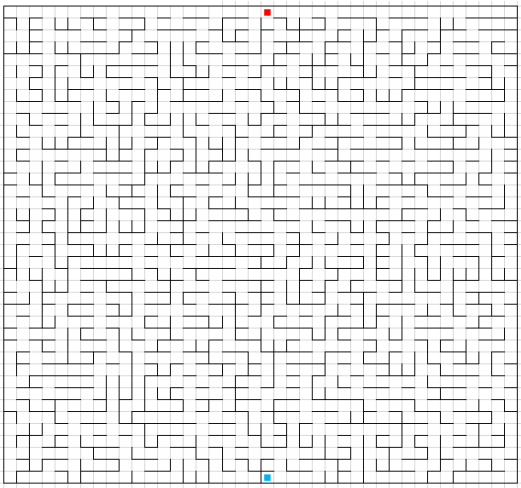


Fig. 5.4　40 × 40 の迷路

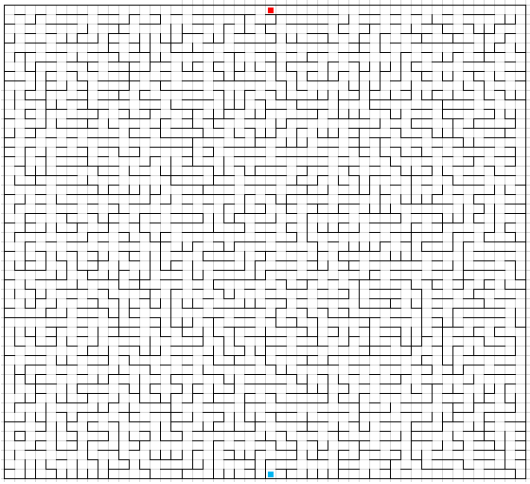


Fig. 5.5　50 × 50 の迷路

Table 5.1 に本実験のパラメータを示す．パラメータは全て共通のものとしている．

Table 5.1　全てのサイズの迷路に共通の設定

項目	内容
試行回数	10000 回
実験回数	100 回
行動選択手法	ε -greedy
行動評価手法	Q 学習
行動価値 Q の初期値	0.00
学習率 α	0.9
割引率 γ	0.99
ロボットの台数 M	12 台

5.3.2　迷路:30 × 30 の結果

Fig. 5.6 に提案手法における ε の推移を示している．この ε は 100 回実験を行い，出力した結果の平均値を表している．X 軸が試行数を示し，Y 軸が ε の平均値を示す．1 試行目では行動を一切していない状態であるため，ε は 1.00，つまりランダム行動を取るようになっている．2 試行目から ε の値が急激に低下するのは，1 試行目で得た遷移情報を基に確率計算を行ない，情報量を算出している

ためである．その後，確率が確定的になっていくため，徐々に低下する．

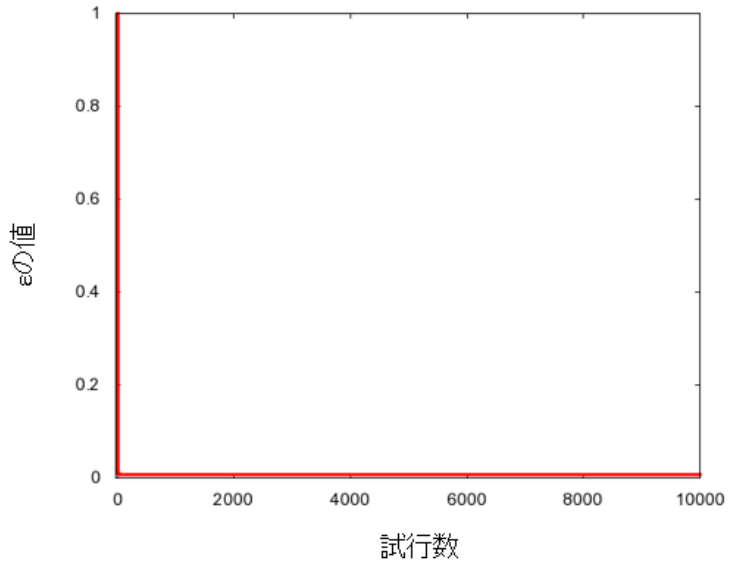
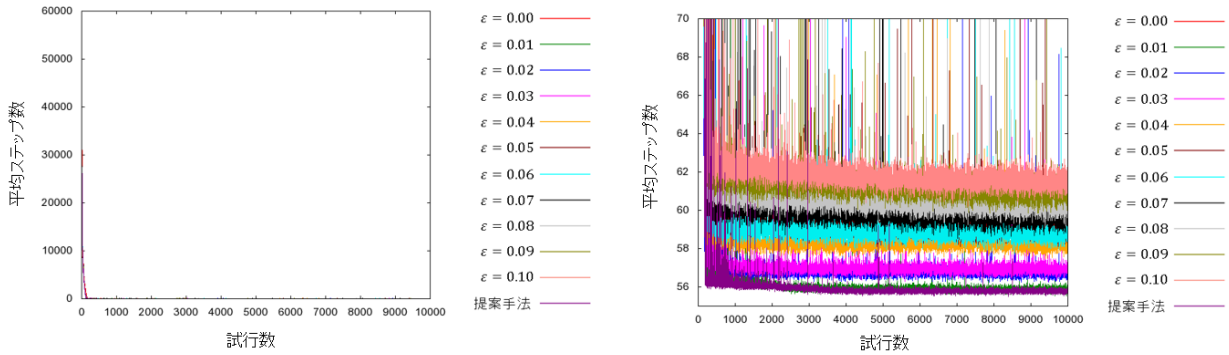


Fig. 5.6　提案手法における ε の推移

Fig. 5.7 に各エージェント間のステップ数の比較を示す．X 軸が試行数を示し，Y 軸に平均ステップ数を示す．Y 軸の範囲は 55-75 である．Fig. 5.7(a) が拡大前の図であり，Fig. 5.7(b) は Y 軸を 55-75 に拡大した図である．

Fig. 5.8 に各エージェント間のステップ数の比較を示す．Fig. 5.8 は Fig. 5.7 を各 ε 毎にグラフ化した図である．X 軸が試行数を示し，Y 軸がステップ数の平均を示す．Y 軸の範囲は 55-75 である．

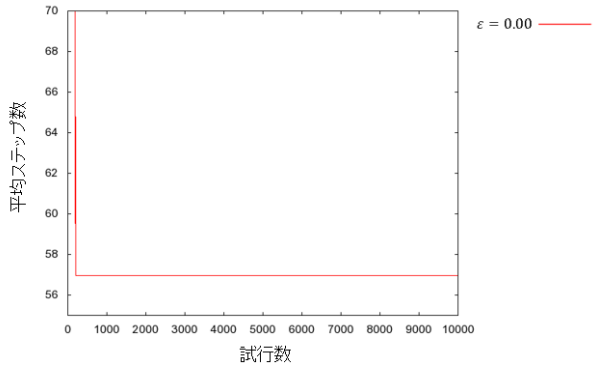
ε= 0.00 は一度 1 つ経路を学習すると，探索をせず，知識の利用のみしかしない．そのため，その経路が最短経路でなくとも同じ経路しか，通らないため．一定値しか出力しない．ε= 0.01 では 100 回に 1 回の割合で，探索行動 (ランダムな行動) を取る．そのため，一度通った経路だけでなく，他の経路も通る可能性がある．よって，試行数が進むにつれて，経路を発見するため，1000 試行目と 9000 試行目を比べると，平均ステップ数が減っているのがわかる．そのため，ε を大きくしていけば，最短経路を発見する確率も高くなると考えられる．しかし，ε が大きくなると，平均ステップ数も $\varepsilon = 0.01$ に比べ増えてしまっている．これは，ε の確率で起こるランダム行動が頻繁に起こるためである．最短経路発見後も，ランダム行動を一定の確率で取り続けるため，このような結果になったと考えられる．提案手法では，経験情報がたまっていない学習初期では，探査をするため，ステップ数は他の $\varepsilon = 0.01$ と変わりない．しかし，経験情報がたまってくると，徐々に ε が低下していく．そのため，最短経路を学習し，ランダム行動による探査がほぼ行われなくなるため，このように提案手



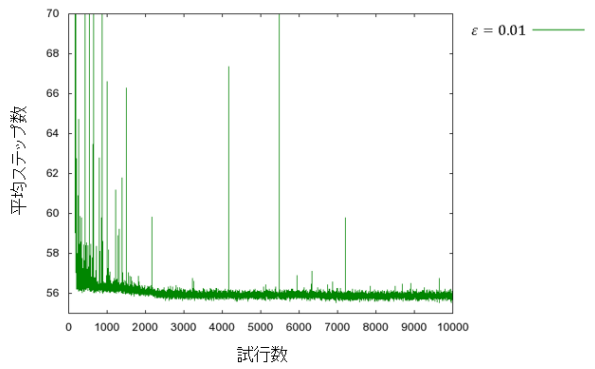
(a) 拡大前　　(b) Y 軸

Fig. 5.7　平均ステップ数の比較

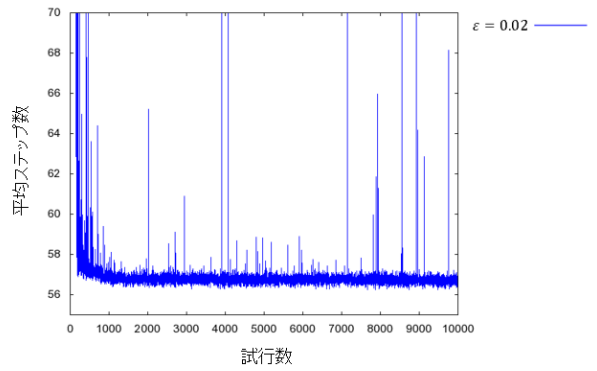
法の平均ステップ数が少ない結果となったと考えられる.



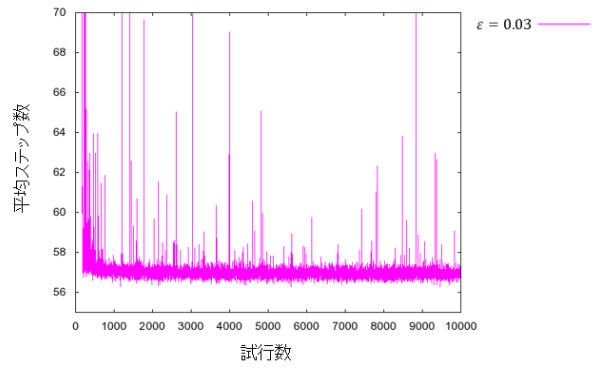
(a)$\varepsilon = 0.00$



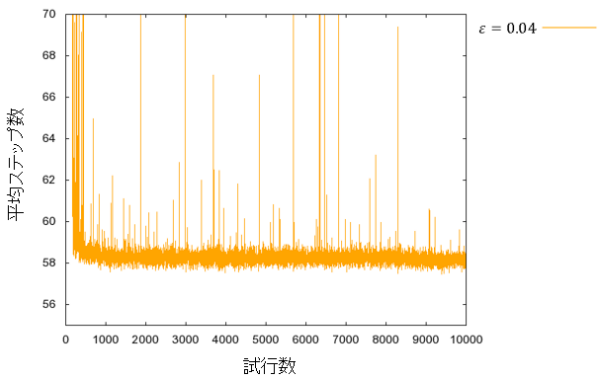
(b)$\varepsilon = 0.01$



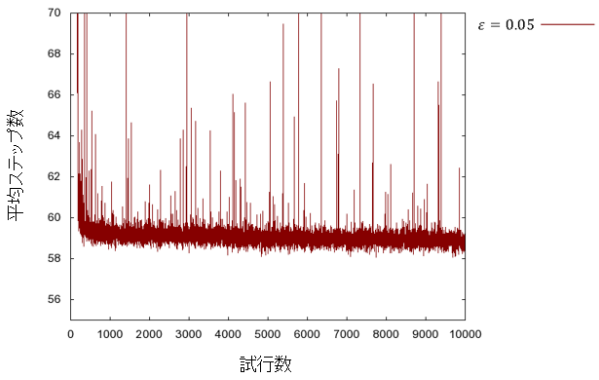
(c)$\varepsilon = 0.02$



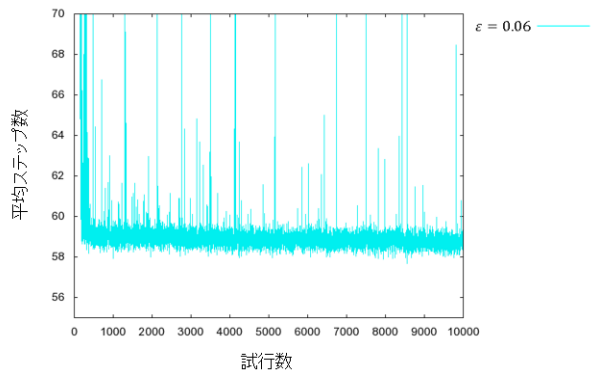
(d)$\varepsilon = 0.03$



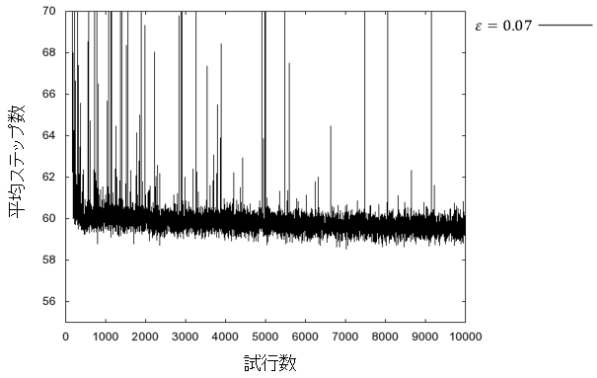
(e)$\varepsilon = 0.04$



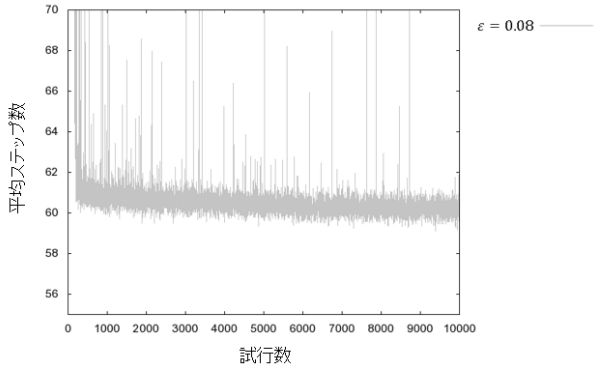
(f)$\varepsilon = 0.05$



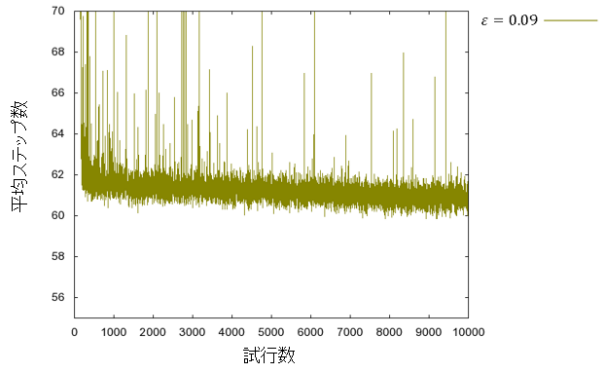
(g)$\varepsilon = 0.06$



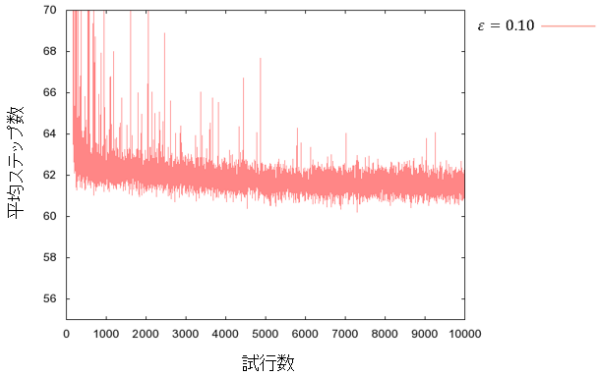
(h)$\varepsilon = 0.07$



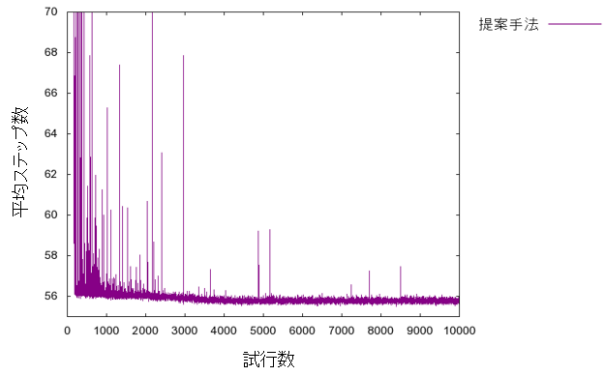
(i)$\varepsilon = 0.08$



(j)$\varepsilon = 0.09$



(k)$\varepsilon = 0.10$



(l) 提案手法

Fig. 5.8　各ε毎の平均ステップ数

5.3.3　迷路:40 × 40 の結果

Fig. 5.9 に提案手法における ε の推移を示している．この ε は 100 回実験を行い，出力した結果の平均値を表している．X 軸が試行数を示し，Y 軸が ε の平均値を示す．

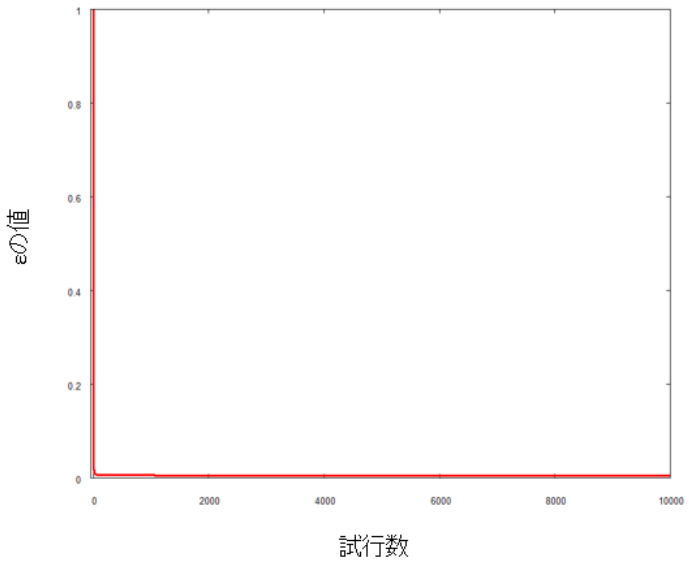
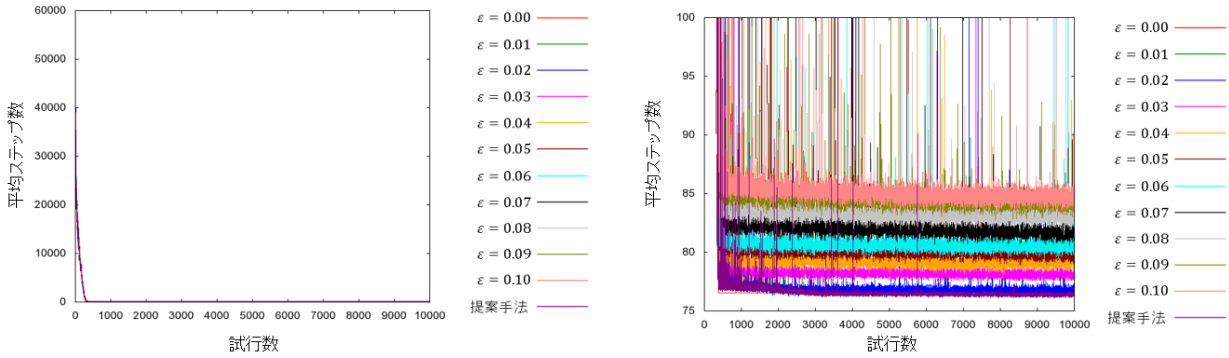


Fig. 5.9　提案手法における ε の推移

Fig. 5.10 に各エージェント間のステップ数の比較を示す．X 軸が試行数を示し，Y 軸に平均ステップ数を示す．Y 軸の範囲は 55-75 である．Fig. 5.10(a) が拡大前の図であり，Fig. 5.10(b) は Y 軸を 60-100 に拡大した図である．

Fig. 5.11 に各エージェント間のステップ数の比較を示す．Fig. 5.11 は Fig. 5.10 を各 ε 毎にグラフ化した図である．X 軸が試行数を示し，Y 軸がステップ数の平均を示す．Y 軸の範囲は 60-100 である．

40 × 40 の場合は，約 500 試行目〜2000 試行目まで，提案手法よりも ε = 0.00 の方が平均ステップ数が少ない．これは，提案手法が 2000 試行目まで，最短経路を学習していないためである．2000 試行目を過ぎると，提案手法の方が平均ステップ数は少なくなっている．



(a) 拡大前　　(b) Y 軸

Fig. 5.10　平均ステップ数の比較

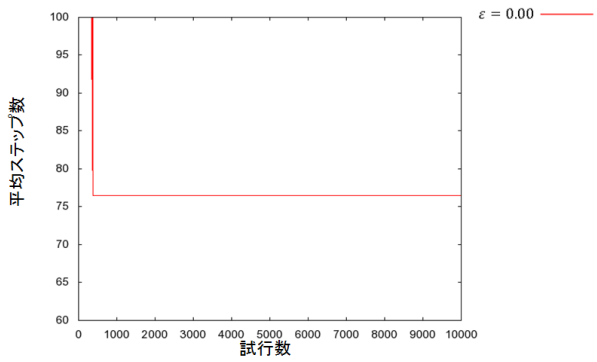
5.3.4　迷路:50 × 50 の結果

Fig. 5.12 に提案手法における ε の推移を示している．この ε は 100 回実験を行い，出力した結果の平均値を表している．X 軸が試行数を示し，Y 軸が ε の平均値を示す．

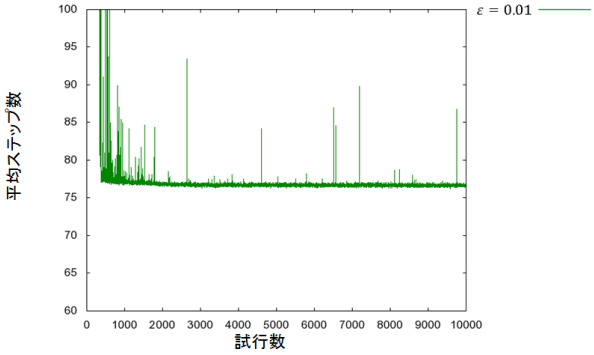
Fig. 5.13 に各エージェント間のステップ数の比較を示す．X 軸が試行数を示し，Y 軸に平均ステップ数を示す．Y 軸の範囲は 55-75 である．Fig. 5.13(a) が拡大前の図であり，Fig. 5.13(b) は Y 軸を 90-120 に拡大した図である．

Fig. 5.14 に各エージェント間のステップ数の比較を示す．Fig. 5.14 は Fig. 5.13 を各 ε 毎にグラフ化した図である．X 軸が試行数を示し，Y 軸がステップ数の平均を示す．Y 軸の範囲は 90-120 である．

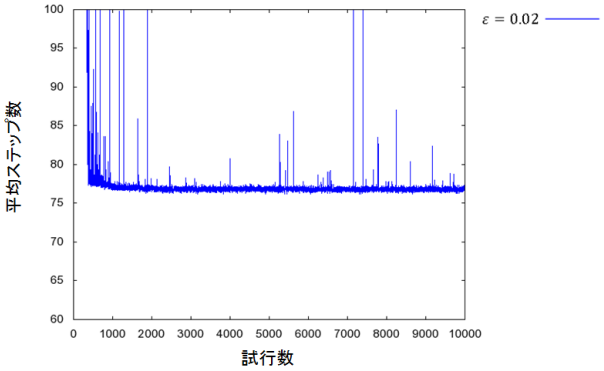
静的環境の実験では，大きさの異なる 3 種類の迷路を用意した．どの迷路においても提案手法を用いた場合，ε を固定したものと比べ，最も少ないステップ数でゴールにたどり着いていたことがわかった．これは，学習初期において，提案手法の ε が高いため，この間に探索が行われる．この間に提案手法は経験をため，環境の遷移確率を算出する．そして，経験を積むと，ε が低下し，利用行動の割合が多くなる．このため，従来の ε-greedy 法よりも，平均ステップ数が低くなったと考えられる．また，30 × 30，40 × 40，50 × 50 の 3 種類の迷路で実験した結果，全ての大きさで，提案手法が最も少ない平均ステップ数であったことから，環境に応じた ε の制御ができていることを確認した．



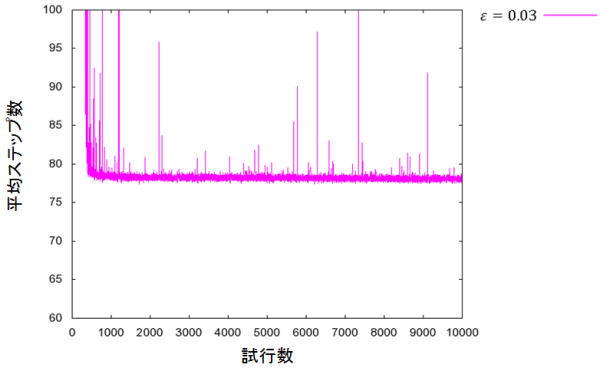
(a)$\varepsilon = 0.00$



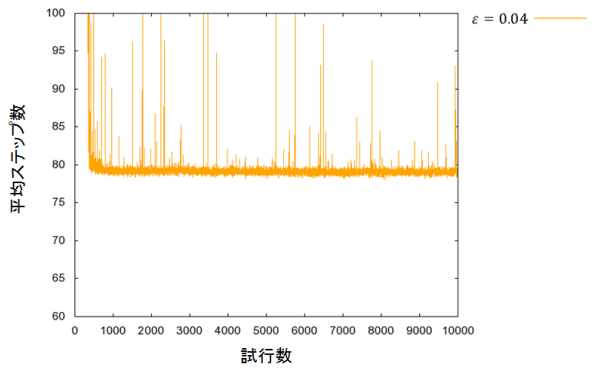
(b)$\varepsilon = 0.01$



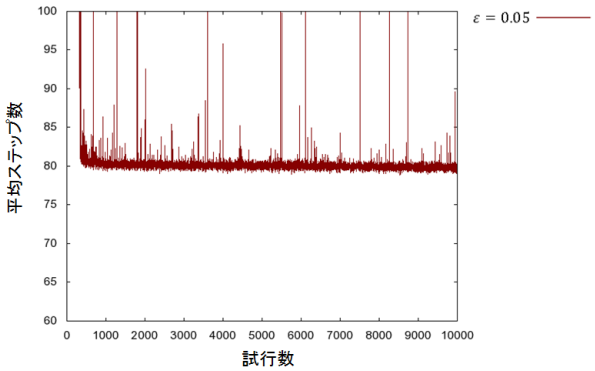
(c)$\varepsilon = 0.02$



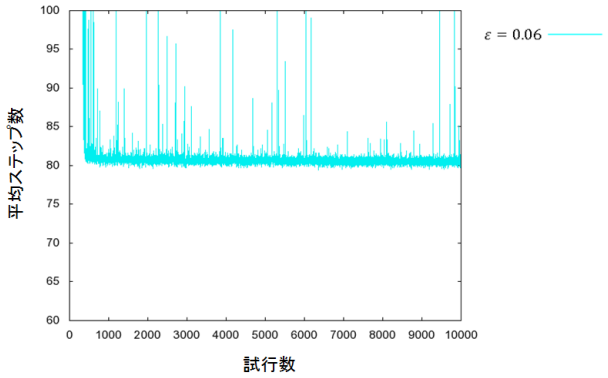
(d)$\varepsilon = 0.03$



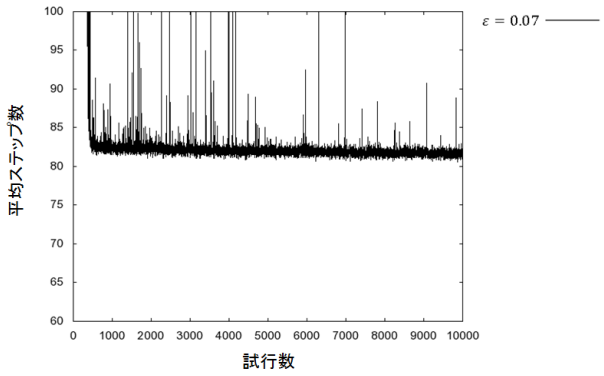
(e)$\varepsilon = 0.04$



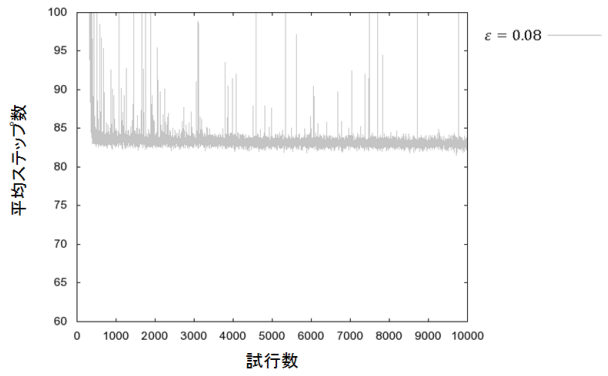
(f)$\varepsilon = 0.05$



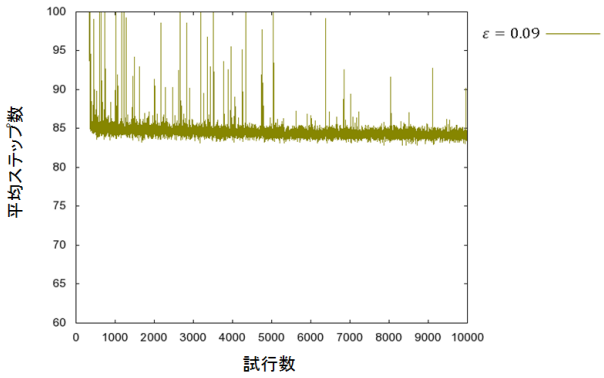
(g)$\varepsilon = 0.06$



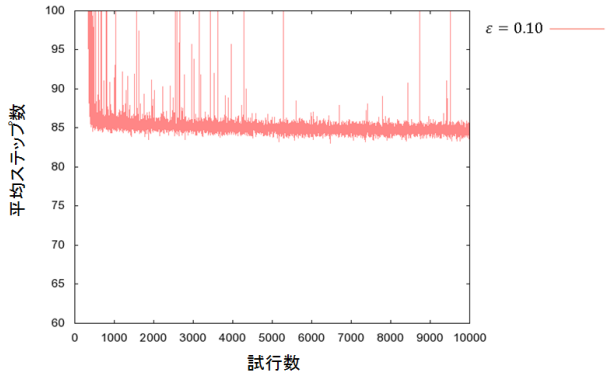
(h)$\varepsilon = 0.07$



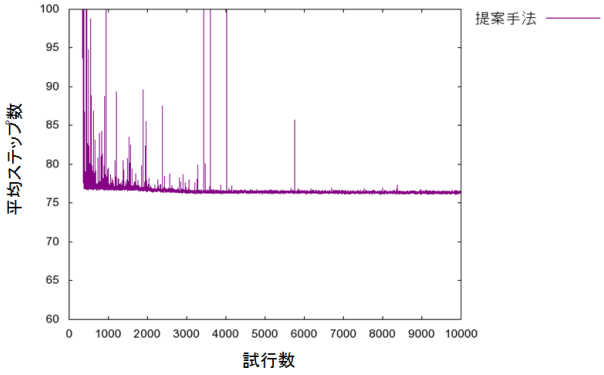
(i)$\varepsilon = 0.08$



(j)$\varepsilon = 0.09$



(k)$\varepsilon = 0.10$



(l) 提案手法

Fig. 5.11　各ε毎の平均ステップ数

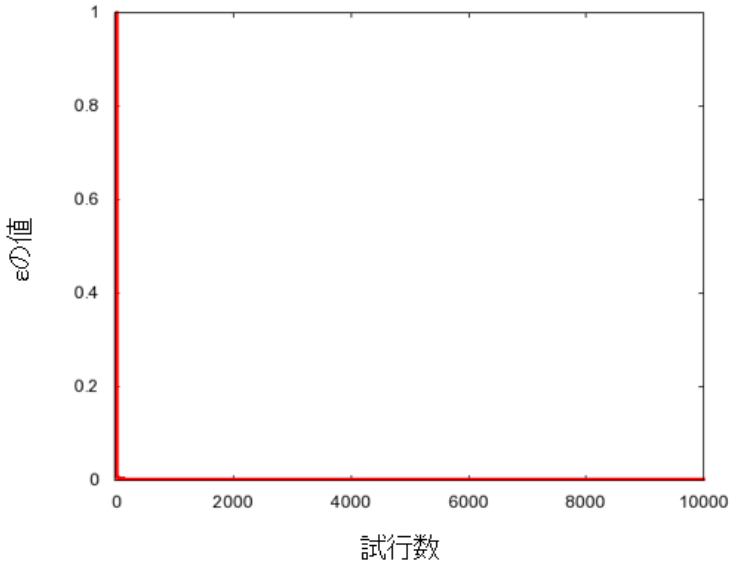
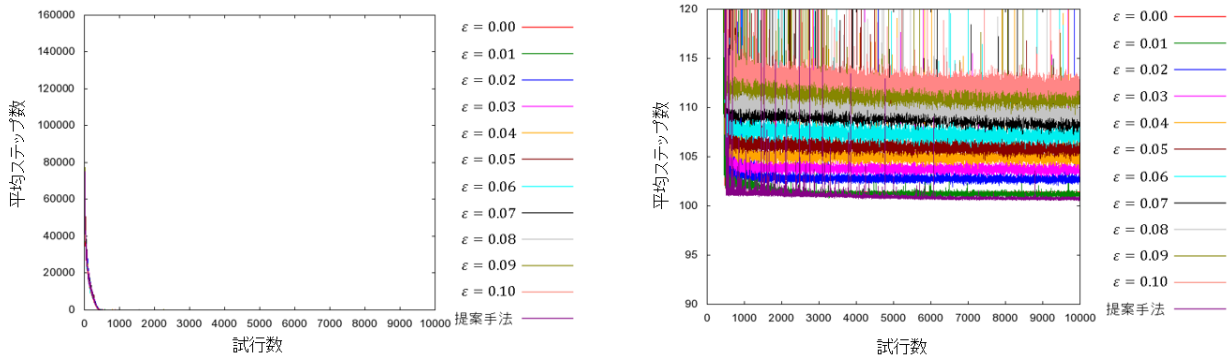
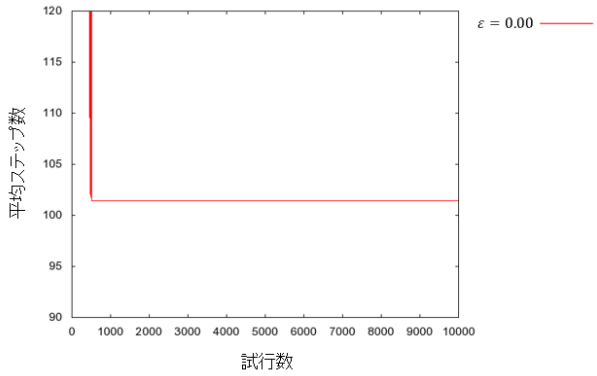


Fig. 5.12　提案手法における ε の推移

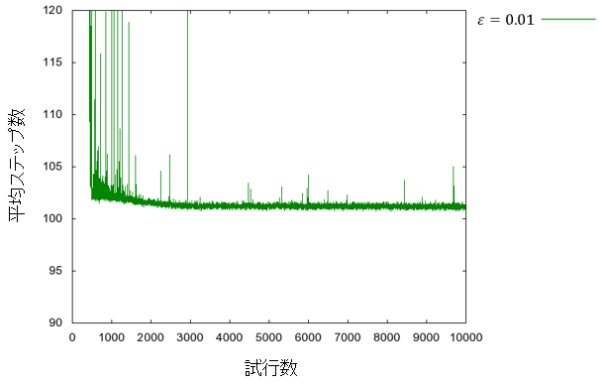


(a) 拡大前　　(b) Y 軸

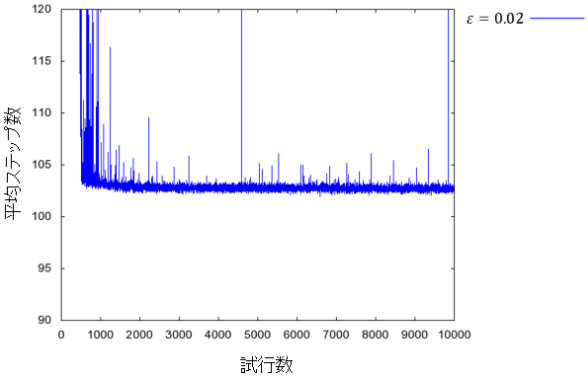
Fig. 5.13　平均ステップ数の比較



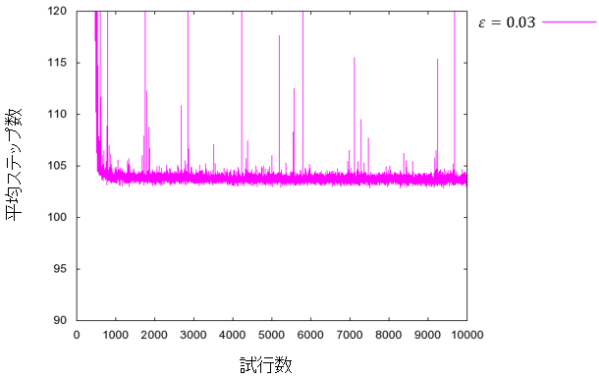
(a)$\varepsilon = 0.00$



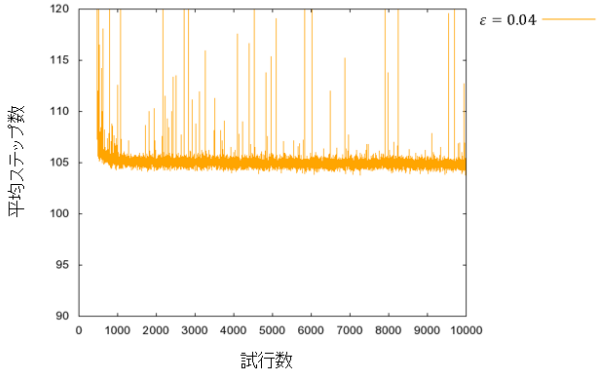
(b)$\varepsilon = 0.01$



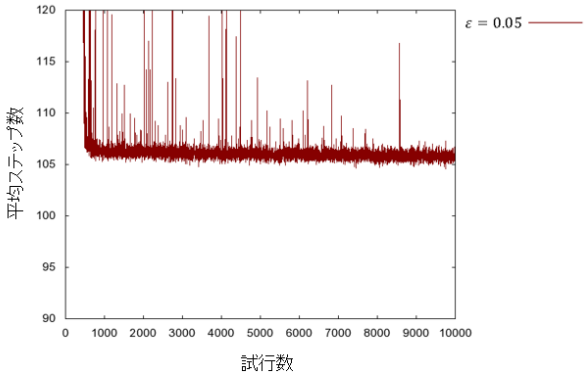
(c)$\varepsilon = 0.02$



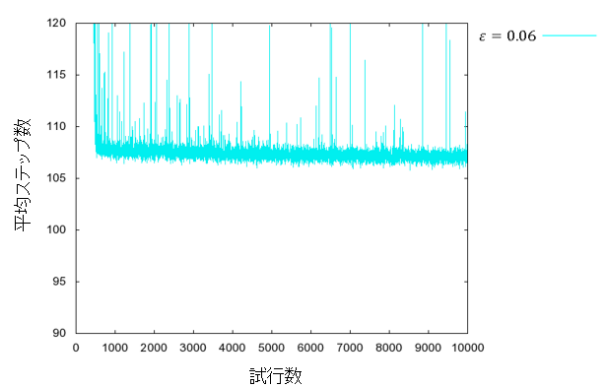
(d)$\varepsilon = 0.03$



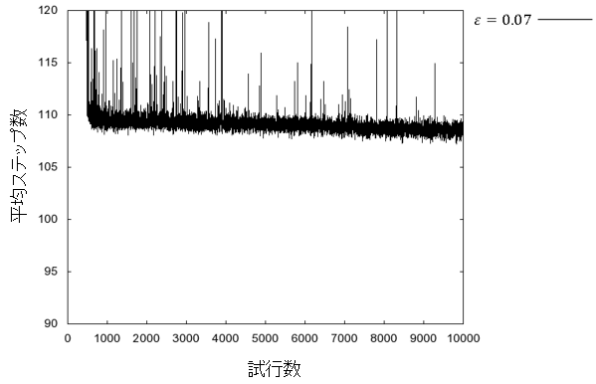
(e)$\varepsilon = 0.04$



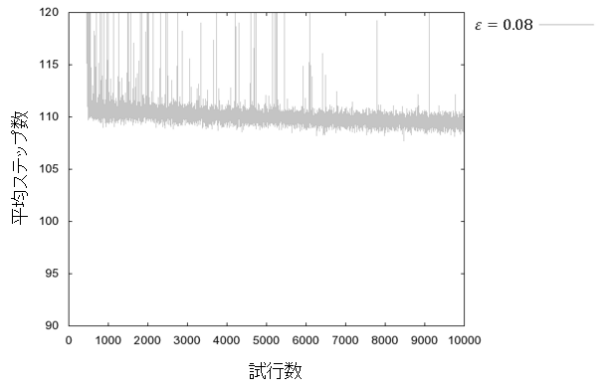
(f)$\varepsilon = 0.05$



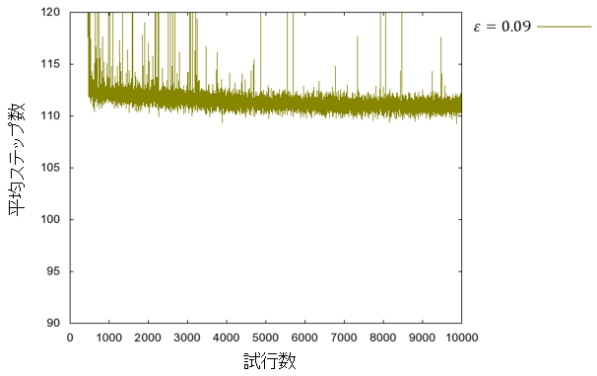
(g)$\varepsilon = 0.06$



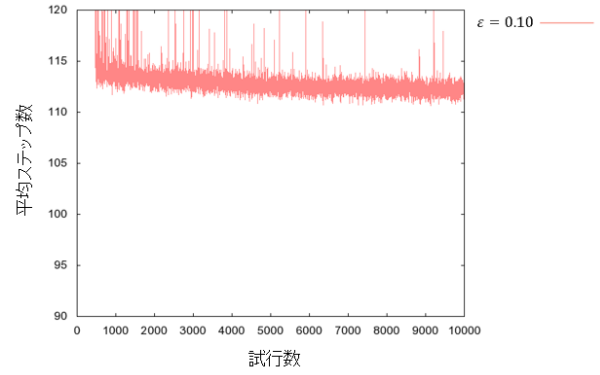
(h)$\varepsilon = 0.07$



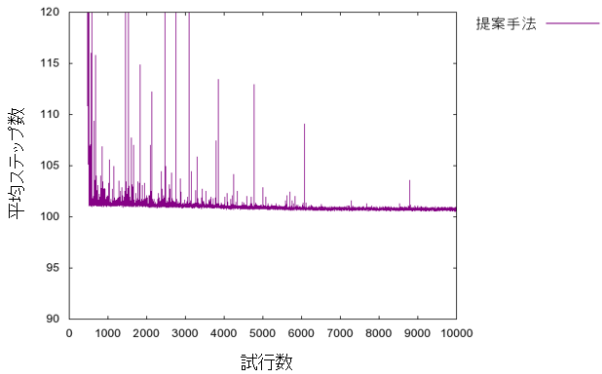
(i)$\varepsilon = 0.08$



(j)$\varepsilon = 0.09$



(k)$\varepsilon = 0.10$



(l) 提案手法

Fig. 5.14　各 ε 毎の平均ステップ数

5.4　動的環境下における実験

本節では，動的な環境で実験を行う．

5.4.1　実験設定

本実験では，一定試行毎に環境変化を起こすことで，動的環境を実現する．迷路問題においては，特定のマスに障害物を設置することで環境変化をさせることとする (Fig. 5.15)．障害物の置かれたマスにエージェントは移動できないとする．

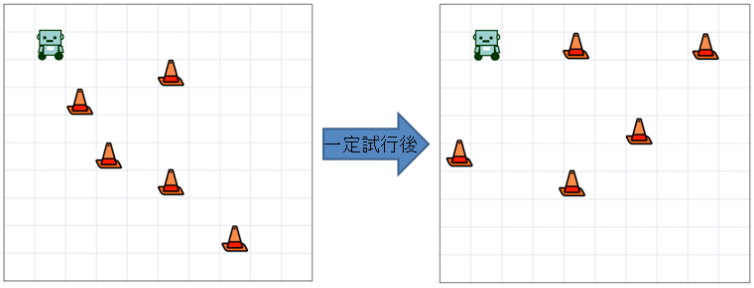


Fig. 5.15　環境変化

このような環境変化が伴う迷路では 1 度発見した経路が環境変化により，通行できない可能性がある．その場合，エージェントは新しい経路を発見しなければならない．

本実験では迷路 (オープンスペース) のサイズとして 10 × 10 の迷路のみで実験を行った．Fig. 5.16 が実際に使用した迷路である．赤色のマスがスタート地点を表し，青色のマスがゴール地点を表している．

Table 5.2 に本実験のパラメータを示す．障害物は一定試行毎にランダムで置かれる．

5.4.2　結果

Fig. 5.17 に提案手法における ε の推移を示している．この ε は 100 回実験を行い，出力した結果の平均値を表している．X 軸が試行数を示し，Y 軸が ε の平均値を示す．

環境変化が起きているのは 200，400，600，800 試行目である．200，400 試行目では ε が上昇しているのがわかる．これは環境変化が起きた状態の遷移先に関する平均情報量が上昇したためである．600，800 試行目で ε が下降したり，変化していないように見えるのは，600，800 試行目で環境が大きく変化し，ε を上昇させる前に，変化した状態で多く行動を行ったため，ε が低下している．

Fig. **??**に各エージェント間の累積ステップ数の比較を示す．X 軸が試行数を示し，Y 軸に累積ステップ数を示す．Y 軸の範囲は 55-75 である．Fig. **??**(a) が拡大前の図であり，Fig. **??**(b) は Y 軸を

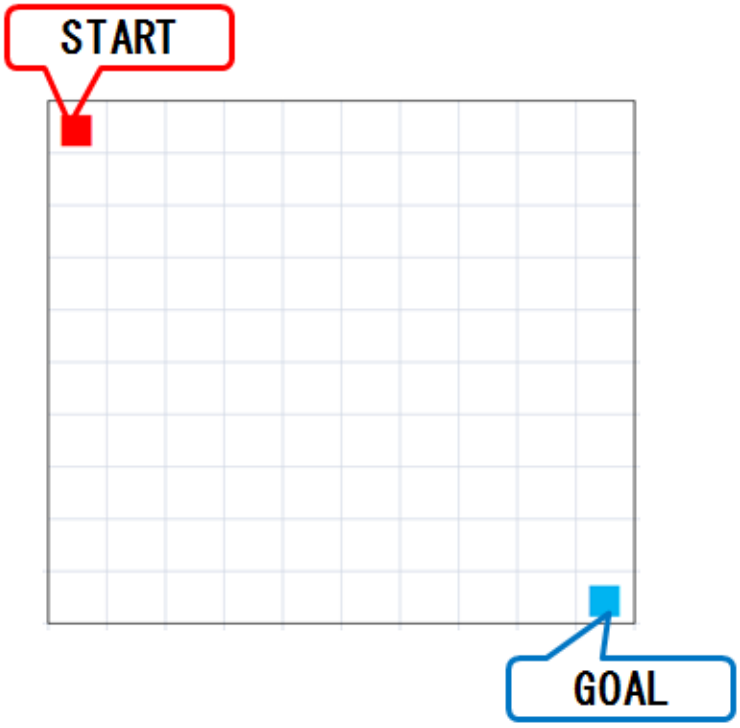


Fig. 5.16　10 × 10 の実験環境

55-75 に拡大した図である．ε が大きいほど，累積ステップ数が少なく，ε が小さいほど，累積ステップ数が多くなる傾向が見れる． ε が低い場合では，環境変化が発生しても，環境が変化する前の知識を利用し，行動を決定することが多いため，障害物を避ける事ができない．結果，障害物にぶつかることが多く，環境変化が起きた 200,400,800 試行付近で，急激にステップ数が増加してしまう．

しかし，提案手法では $\varepsilon = 0.09$ や $\varepsilon = 0.09$ よりもステップ数が多くなっている．これは障害発生時に ε が 0.10 程度しか上昇しないためである．原因としては，環境変化により情報量は上昇したが，環境の一部のみ変化したため，情報量が変化しない状態がある．そのため，全ての状態行動対に関して，総計した場合，環境変化した部分の影響が弱くなり， ε の上昇率が悪くなる．

Table 5.2　パラメータ設定

項目	内容
試行回数	1000 回
実験回数	100 回
環境変化までの試行数	200 試行毎
障害物の数	20
行動価値 Q の初期値	0.00
学習率 α	0.1
割引率 γ	0.9
ロボットの台数 M	12 台

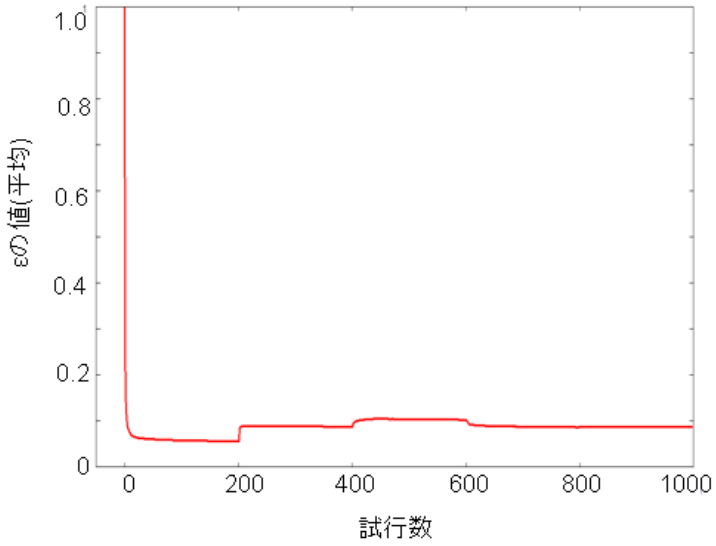


Fig. 5.17　提案手法における ε の推移

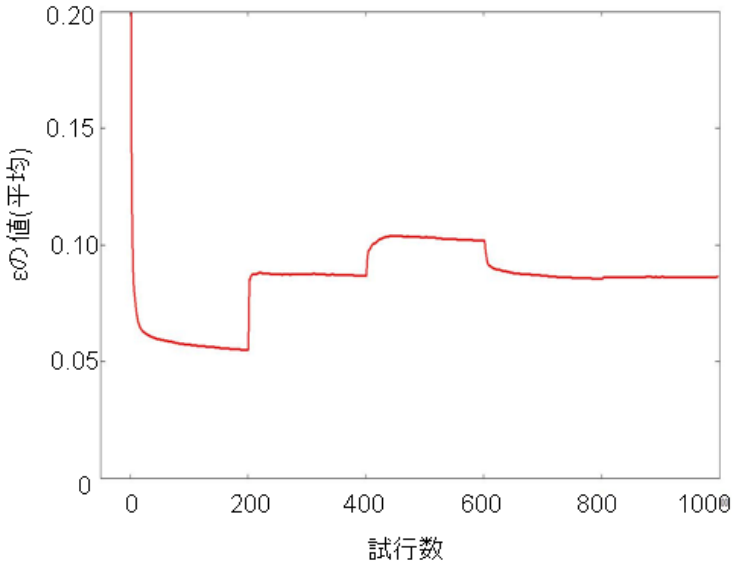
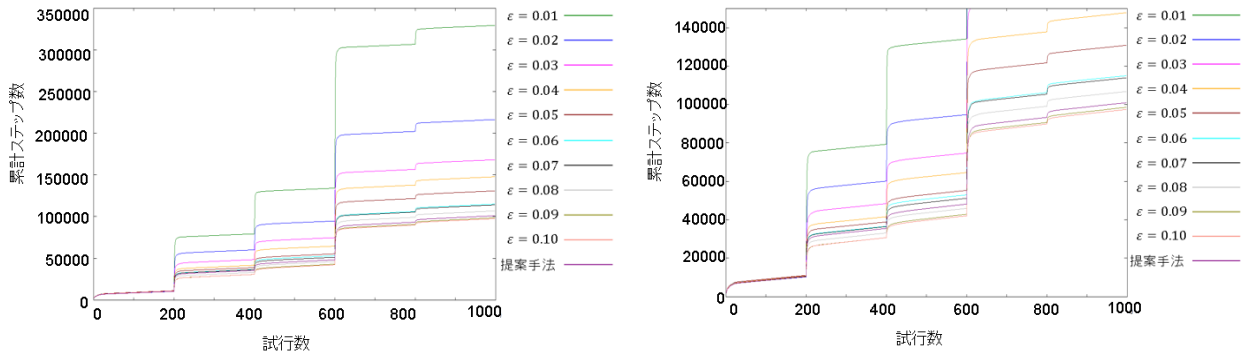


Fig. 5.18　提案手法における ε の推移



(a) 拡大前　　(b) Y 軸

Fig. 5.19　累計ステップ数の比較

第6章　おわりに

6.1　まとめ

本論文では，強化学習における探査と利用のトレードオフ問題に着目した．強化学習においては，探査-利用のトレードオフ問題があり，探査-利用のバランスは行動選択手法のパラメータによって決定されている．本論文では，行動選択手法のうち ε-greedy 法に着目し，探査率 ε を環境に合わせて，動的に制御することを目的とした．

本論文では情報量に着目し，状態遷移に関する情報量に応じて，探査率 ε を制御することを提案した．情報量が大きければ，遷移先が定まっておらず，定めるためにも探索をした方が良い．また，情報量が小さければ，遷移先を定まっているため，これ以上の探索をしても新しい情報が得られないので，利用をした方が良い．この条件に基づき，ε を情報量の大きさに合わせて動的に制御する．

提案手法は 2 つのモジュールから構成される．1 つは「経験情報の獲得」である．経験情報の獲得では，環境のモデルを知るために，エージェントは行動回数と遷移回数を行動毎に記録する．その情報からある状態のときある行動を取ると，どのくらいの確率で次状態に遷移するかという遷移確率を算出する．2 つめのモジュールは「探査率 ε の算出」である．経験情報の獲得で算出した遷移確率を用いて，各状態行動対に対して遷移先がどの程度ランダム性を持っているのかという平均情報量を算出する．そして，その平均情報量を 0 から 1 に正規化し，合計した値を ε とした．ε は，試行毎に算出している．このようにして，学習中に ε を動的制御する方法を提案した．

検証実験では迷路問題をエージェントに適用し，環境は静的環境と動的環境の 2 種類を用意した．ε を学習中一定に固定した従来の強化学習と提案手法を比較した．検証の結果，ε が学習中に動的に変化すること，環境に応じて，変化することを確認した．よって，探査率 ε を環境に合わせて，動的に制御することができたと考える．

この提案手法を用いて，探査率 ε が学習中，動的に制御されているかをシミュレーション実験を通して検証した．経路計画問題において提案手法と探査率 ε を学習中一定に固定した従来の強化学習法を比較した．

実験は静的環境と動的環境の 2 種類を用意した．静的環境下では，ε が低いほど，環境に適した行動が取れた．逆に動的環境下であると，ε の値が高いほど，障害物を避けて，ゴールに早くたどり着

くことができる．このような場合でも，提案手法では探査率 ε を変化させ，環境に追従できることが確認できた．しかし，動的環境の場合，環境が一部しか変わらない場合，全体を総計してみているので，εが全体で見ると，上昇しないこともある．

6.2 今後の課題

今後の課題として，まず学習の進度を ε の計算式に組み込むことを考える．提案手法では ε の制御方法として，情報量のみを考慮にいれた．しかし，エージェントの学習はQ値を更新することで行われる．Q値は何度も更新を行うことで，真の値 Q^* に収束する．そのため，学習の進度も ε に反映させることで，より優れた手法になると考える．

上記の問題点とは別に，実ロボットへの適応も今後の課題として挙げられる．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，本論文で提案した手法を実ロボットにも適応したいと考えている．しかし，シミュレーション実験と異なり，実ロボットに用いる場合に存在する問題点がある [40]．提案手法を実ロボットへ用いる場合に考えられる問題点として以下の2点が挙げられる．

- 不完全知覚下への対応

  本論文の実験では，エージェントのセンサ能力について特に制限することはなかった．エージェントは，実験環境すべての状態を別々の状態として認識し，学習を行っていた．しかし，実ロボットにおいて搭載できるセンサには上限がある．本来は異なった状態であるが，センサの不足によって，同じ状態と誤って認識してしまうことがある．このことを不完全知覚という．

  例えば，Fig. 6.1 のような迷路問題を考える．ロボットは前後左右も壁の有無のみ認識できるとする．また，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，Fig. 6.1 の s_5 と s_9 は右と下のみ壁がある状態で，ロボットからみると同じ構造に見える．s_5 から上へ移動すると s_2 へ遷移し，s_9 から上に移動すると s_6 へ遷移する．このとき，同じ状態から行動すると，遷移先の状態が異なるため，提案手法では情報量が高くなってしまう．しかし，どんなに探索しても，情報量が低くなることはなく，無駄な探索をしてしまう．この問題を解決するためには，不完全知覚がおきている状態を別々の状態に切り分けるなどの対応が必要である．

- ロボットのメモリ制限

  ロボットのメモリ制限については，実ロボットを用いる場合には，使用できるメモリの量には限りがある．しかし，本論文では，行動回数と遷移回数を保持できる量に制限を設けなかった．そのため，行動回数と遷移回数を保持できる量に制限を設け，参照回数が低い行動回数と遷移回数の忘却を行う必要がある．

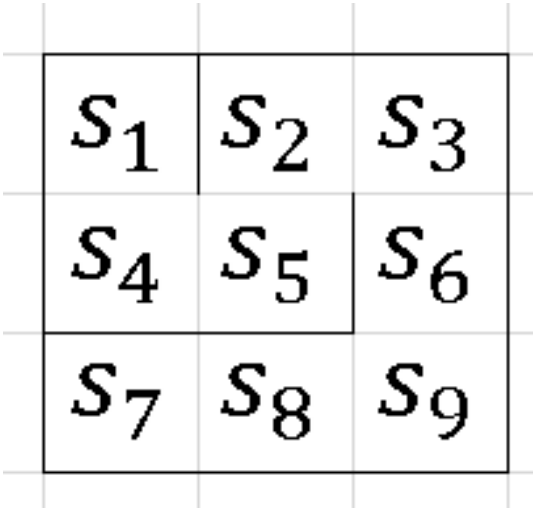


Fig. 6.1　不完全知覚の例

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助教に深く感謝の意を表します．また，ご助言，ご指導を頂いた畑中雅彦教授，佐賀聡人教授，本田泰准教授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杉本大志君，高泉昇太郎君，三浦丈典君，二階堂芳君，片山和宣君，小橋遼君，千葉秀平君に感謝致します．

参考文献

[1] Masato Hirose, Kenich Ogawa, ”Honda humanoid robots development”, Philosophical Transactions of the Royal Society A, Vol.365, No.1850, pp.11-19, 2007

[2] 西沢敏弘，服部浩明，”サービスロボットの安全化事例　－チャイルドケアロボット「Papero」－”，日本ロボット学会誌，Vol.25, No.8, pp.1159-1161, 2007

[3] 鷲見和彦，”柔軟物も取り扱える生産用ロボットシステムの開発”，日本ロボット学会誌，Vol.27, No.10, pp.1082-1085, 2009

[4] 佐藤侑，郭士傑，稲田誠生，向井利春，”介護支援ロボット RIBA-II の動作設計と評価実験”，日本機械学会論文集 C 編，Vol.78, No.789, pp.1899-1912, 2012

[5] 遠山茂樹，エコプルワント，米竹淳一郎，”超音波モータを用いたパワーアシストスーツの開発：モータ駆動, スーツ機構開発 (パワーアシスト 2)”，福祉工学シンポジウム講演論文集，pp.119-122, 2004

[6] 本間敬子，山田陽滋，松本治，李秀雄，小野栄一，”介護支援ロボットの実証試験における倫理審査と被験者保護について：排泄介護総合支援ロボット「トイレアシスト」の事例報告”，日本ロボット学会誌，Vol.28，No.2，pp.181-190, 2010

[7] 鈴木正憲, ”原子力発電プラント水中検査用 ROV の開発”, 日本ロボット学会誌, Vol.22, No.6, pp.697-701, 2004

[8] 伊藤智之, 木村元比古, ”小型水中点検ロボットの開発”, 日本ロボット学会誌, Vol.22, No.6, pp.702-705, 2004

[9] K.Ohono, S.Kawatsuma, T.Okada, E.Takeuchi, K.Higashi, S.Tadokoro, ”Robotic control vehicle for measuring radiation in Fukushima Daiichi Nuclear Power Plant”, IEEE International Symposium on Safety, Security and Rescue Robotics 2011, pp.38-48, 2011

[10] 浦環, ”海中に求められるロボット”, 日本ロボット学会誌, Vol.22, No.6, pp.692-696, 2004

[11] 山本郁夫，”魚ロボットの開発”，日本ロボット学会誌，Vol.22, No.6, pp.706-708, 2004

[12] 中須賀真一，”人工知能は宇宙開発・宇宙利用に貢献できるか？”, 日本ロボット学会誌，Vol.21, No.1, pp.2-13, 2006

[13] 久保田考, ”惑星別探査ローバ”，日本ロボット学会誌, Vol.21, No.5, pp.468-471, 2003

[14] Richard Volpe, Richard Doyle, ”Recent Robotics Developments at NASA/JPL”, 日本ロボット学会誌, Vol.27, No.5, pp.490-493, 2009

[15] 大野和則，永田圭司，秋山英久，”レスキューロボットの地図構築”，日本ロボット学会，Vol.28, No.2，pp.169-172，2010

[16] 広瀬茂雄, ”ヘビ型ロボットの移動機構”, 日本ロボット学会誌, Vol.28, No.2, pp.151-155, 2010.

[17] 西村明浩，”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

[18] 加納政芳，清水太郎，”なにもできないロボット Babyloid の開発”，日本ロボット学会誌，Vol.29, No.3, pp.298-305, 2011

[19] 小高知宏，”はじめての機械学習”，株式会社 オーム社，2011

[20] Richard S. Sutton and Andrew G. Brato，”Reinforcement Learning”，The MIT Press，1998

[21] 丸山淳一, 松原崇充, Joshua G. Hale, 森本 淳, ”強化学習を用いたヒューマノイドロボットによる転倒回避ステップ動作の学習”，日本ロボット学会誌，Vol.27, No.5, pp527-537,2009

[22] Minoru Asada, Shoichi Noda, Sukoya Tawaratsumida and Koh Hooda, ”Purposive Behavior Accquisition for a Real Robot By Vision-Based Reinforcement Learning”, Machine Learning, vol23, pp.279-303,1996

[23] Jun Morimoto and Kenji Doya, ”Accuisition of stand-up behavior by a real robot using hierarchical reinforcement learning”, Robotics and Autonomous System, vol.36, pp37-51, 2001

[24] Hajime Kimura, Toru YamaShita, Shigenobu Kobayashi, ”Reinforcement Learning of Walking Behavior for a Four-Legged Robot”, 40th IEEE Conf. on Decision and Control, pp411-416, 2001

[25] Maja J. Mataric, ”Reinforcement Learning in the Multi-Robot Domain”, Autonomous Robots 4, pp.73-83, 1997

[26] 田中文英，”AI 化建築へ:マルチタスク強化学習による住居屋根制御”，建築雑誌, Vol.117 No.1488 p.85, May 2002

[27] 松井藤五郎，後藤卓，”強化学習を用いた金融市場取引戦略の獲得と分析 (¡特集¿ファイナンスにおける人工知能応用)”, 人工知能学会誌, Vol.24, No.3, pp.400-407, 2009

[28] 松井藤五郎, 後藤卓, 和泉潔, 大和田勇人，”強化学習を用いた債券取引戦略の獲得”，2008 年度人工知能学会全国大会 (第 22 回), 2C3-01, 2008

[29] S.Thrun, ”The Role of Exploration in Learning Control”, Handbook for Intelligent Control: Neural, Fuzzy and Adaptive Approaches, Van Nostrand Reinhold, 1992

[30] 宮崎和光，山村雅幸，小林重信，”k-確実探査法：強化学習における環境同定のための行動選択戦略”, 人工知能学会誌, Vol.10, No.3, pp.124-133, 1995

[31] 山村雅幸，宮崎和光，小林重信，”MarcoPolo：報酬獲得と環境同定のトレードオフを考慮した強化学習システム”, 人工知能学会誌, Vol.12, No.1, pp.78-89, 1997

[32] Kamei, Keiji and Ishikawa, Masumi，”A Genetic Approach to Optimizing the Values of Parameters in Reinforcement Learning for Navigation of a Mobile Robot”，Neural Information Processing，Springer Berlin Heidelberg，2004

[33] Eriksson A.，Capi G. and Doya K, ”Evolution of meta-parameters in reinforcement learning algorithm”, Proceeings of the IEEE/RSJ Internatinal Conference on intelligent Robots and Systems, 2003

[34] 亀井圭史，石川眞澄，”パラメータの相互依存性を考慮した強化学習の最適パラメータ推定”，電子情報通信学会技術研究報告. NC, ニューロコンピューティング，2007

[35] 亀井圭史，石川眞澄，”遺伝的アルゴリズムによる移動ロボットの強化学習パラメータ最適化”，電子情報通信学会技術研究報告. NC, ニューロコンピューティング，2005

[36] 小野 兼嗣，岩田 一貴，林 朗，末松 伸朗，ソフトマックス行動選択のパラメータ調整の手間を省くための新たな関数の導入，電子情報通信学会技術研究報告. NC, ニューロコンピューティング，pp.107-112, 2010

[37] Watkins, C. J. C. H. and Dayan, P. ,"Technical Note: Q-Learning", Machine Learning 8, pp. 279–292 (1992).

[38] MAHADEVAN S., "Automatic Programming of Behavior-based Robots using Reinforcement Learning", Proc. AAAI-91, pp.768-773,1991

[39] Clouse, Jeffery A. and Utgoff, Paul E., "A Teaching Method for Reinforcement Learning", Proceedings of the 9th International Workshop on Machine Learning, pp.92-110, 1992

[40] 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)", 人工知能学会誌, Vol.12, No.6, pp.831-836

研究業績

1. Nodoka Shibuya, Yoshiki Miyazaki, Kentarou Kurashige, "Suggestion of Probabilistic Reward-Independent Knowledge for Dynamic Environment in Reinforcement Learning", 2011 Int. Symp. on Micro-NanoMechatronics and Human Science CD-ROM, pp.140-145, November 6-9, 2011, Nagoya, Japan