論文 / 著書情報
Article / Book Information

題目(和文)	離散マルコフ決定過程における強化学習
Title(English)	
著者(和文)	宮崎和光
Author(English)	
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第3300号, 授与年月日:1996年3月26日, 学位の種別:課程博士, 審査員:
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第3300号, Conferred date:1996/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

離散マルコフ決定過程における強化学習

研究者 宮崎和光

指導教官 小林重信 教授

目 次

1 序論

1.1 研究の背景

近年多方面でロボットは活躍しており, ロボットに要求されることは技術の進歩に伴い複雑で高度なものへと移行しつつある. 今日そのような状況の中で, ロボットが作業を行う際に, 如何に作業を行えば能率が良いか, 如何に作業環境に適応すべきかなど, ロボット自らが判断し, 行動することのできる自律ロボットに対する要求が高まりつつある. 本研究では, このような背景をもとに, 宇宙空間や深海などの人の手が及んでいない未知なる環境に適応できる自律ロボットの構築を研究の最終目標としている.

未知なる環境では, 予め, 起こり得る事象のすべてをプログラミングすることは到底不可能である. そのためロボットには学習機能が必要となる. しかしそこでの学習には, 一般に, 手取り足取り正解を教えてくれるような教師を想定することはできない. そのためロボットは試行錯誤を繰り返し, 結果的に良かったか悪かったかという情報のみから学習できなければならない. ここではそのような情報を総称して報酬と呼ぶ.

報酬による学習は, 未知なる環境のみならず, 環境が予め既知であっても有効である. なぜなら環境が既知であっても, ロボットの動作を逐一プログラミングすることが困難な場合は多々考えられるからである. 殊に, 目標だけを報酬という形で設定しておけば, 人間が考え出した以上の解を得る可能性を秘めているので, この報酬による学習はたいへん興味深い課題と言える. 現在, この課題に真正面から立ち向かっている学習手法が強化学習 (reinforcement learning) である.

強化学習の由来はパブロフの犬まで遡れるが,[Samuel 59] の checker player のように人工知能の草創期における機械学習の主要なパラダイムのひとつであった. その後, 当時の計算機パワーの限界もあって, 研究者の関心はより効率的な知識処理に移っていった. 数年前から自律ロボットなどからの潜在的な需要と計算機パワーの飛躍的進歩を背景として, 再び脚光を浴びつつある. 特に,[Barto 83] の Adaptive Heuristic Critic に始まり, [Sutton 88] の Temporal Difference およびその発展形と捉えることができる [Watkins 92] の Q-learning に至る解析面での著しい発展のため, 現在, 爆発的な広まりをみせつつある.

本論文では, はじめに述べたような自律ロボットの構築には, 報酬による学習が不可欠との立場から強化学習に注目し, 研究を行う.

1.2 研究の目的

強化学習の目的は, できるだけ多くの報酬を獲得することにある. より多くの報酬を得るためには, 環境を広く同定しなければならない. しかし環境の同定を重視して行動すると, 学習途中での報酬獲得が軽視されがちになる. このように強化学習では, 報酬獲得と環境同定といった相反する目的が要求される.

強化学習の接近法は経験強化型と環境同定型の 2 つに大きく類別される [山村 95]. 経験強化型は報酬による経験の強化を強調する接近法であり, そこでは, 学習途中での報酬獲得が特に重視される. 環境同定型は経験の蓄積を強調する接近法であり, 学習途中はともかく, 学習終了後に最適な報酬獲得戦略が得られれば良いとする立場をとる.

経験強化型では報酬により行動が直ちに強化されるので, 一般に, 素早い学習が可能である. しかしその反面, 誤った行動が強化されやすいという欠点を持つ. そこで本論文では, 学習が素早いという経験強化型の利点は残しつつ, 学習結果の合理性を保証することを第一番目の研究目標に設定する. そのような手法が実現されれば, 強化学習手法としての経験強化型の有効性が明らかにされるものと考える.

ところで, 経験強化型では, 環境全体を同定することは, 一般には, 困難である. そのため学習終了時に最適な報酬獲得戦略が得られるとは限らない. 一般に, 最適性をより効率よく保証するためには, 環境を効率よく同定する必要がある. すなわち環境同定に徹底した手法が重要となる. 本論文では, 以上のような立場から, 環境同定を重視した手法の提案を第二番目の研究目標に設定する. 環境が同定され既知となれば, 既存の様々な手法が利用可能となるので, このような手法は十分意味のある結果を生むと考える.

環境同定型は学習途中での報酬獲得を完全に無視した手法である. しかし強化学習の工学的な応用を考えた場合などでは, 特に, 学習の初期段階からそこそこの報酬を獲得しつつ, 同時に環境の同定を行い, 報酬獲得の水準を段階的に向上させるような挙動が求められる. そこで本論文では, 経験強化型と環境同定型の相補的性質に着目して, 両者を巧みに統合した強化学習システムの提案を第三番目の研究目標に設定する. これにより従来困難であった強化学習の工学的応用への道が開かれるものと期待する.

1.3 論文の構成

本論文は”離散マルコフ決定過程における強化学習”と題し, 7 章より構成されている. 第 1 章は序論である. 第 2 章では, 本研究が対象としている問題領域の説明, および, 既存の研究のサーベイを行う. 第 3 章では, 経験強化型の代表である profit sharing を取り上げ, 学習

結果の合理性を保証するための定理について述べる. 第 4, 第 5 章では, 環境同定を重視した接近である k-確実探査法およびその発展形である ℓ-確実探査法を提案する. 第 6 章は, 第 3 章から第 5 章までの結果を利用し, 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である MarcoPolo を提案する. 最後の第 7 章は結論であり, 本研究の成果を総括し, 今後の課題をとりまとめる.

2　問題設定

2.1　取り扱う問題のクラス

未知なる環境に置かれたロボットのような主体 (agent) を考える. 図 1 にその模式図を示す. エージェントは環境からの感覚入力に対して, 行動を選択し, 実行に移す. 一連の行動に対して, 環境から報酬が与えられる. 報酬とはエージェントの存在意義を抽象化した量である. 生物なら「餌をとる」「敵から逃れる」, ロボットなら「目標を達成する」ときに与えられる.

一般に, エージェントは環境の状態変数のすべてを検知できるとは限らないので, **非決定性の処理**が要求される. また報酬は行動に対して即座に与えられるとは限らないので, **遅れの処理**が要求される. エキスパートシステムなどの教師付き学習では, 完全なる教師の存在を仮定する. その教師は, 確実性や遅れのない評価をエージェントに与えることができる. これに対し強化学習では, 非決定性と報酬遅れを持つ弱い情報源しか利用しないところに特徴がある.

このような要求に加え, 本論文では, 取り扱う問題のクラスを**離散マルコフ決定過程** (discrete Markov decision processes : MDPs) に限定する. そこでは, 入出力変数の値域には離散値, 環境の性質にはマルコフ性を仮定する. 時間は認識-行動サイクルを 1 単位として離散化される. 感覚入力は離散的な属性-値ベクトルとして与えられ, 行動は離散的なバリエーションの中から選ばれる. 感覚入力に対して実行可能な行動は**ルール**として記述される. 各感覚入力に対し選択すべきルールを与える関数を**政策**と呼び, 単位行動当たりの期待獲得報酬を最大化する政策を**最適政策**と呼ぶ.

MDPs は研究開発がもっとも活発な領域であり, 理論的および実験的な研究成果が多く蓄積されており, 本研究の有効性を主張する上でも, 格好の問題クラスである. 以下では, まず始めに強化学習システム全般にわたるサーベイを行った後, 本研究の接近法を明らかにする.

2.2　既存の研究に関するサーベイ

MDPs を対象とする学習システム

本研究では, 強化学習システムを状態認識器, 行動選択器, 学習器の 3 つの構成要素からなるシステムとして把握する. 図 2 にその枠組みを示す. 状態認識器には環境からの感覚入力が入力される. MDPs を対象とする強化学習システムでは, ルールの前提部との照合がここ

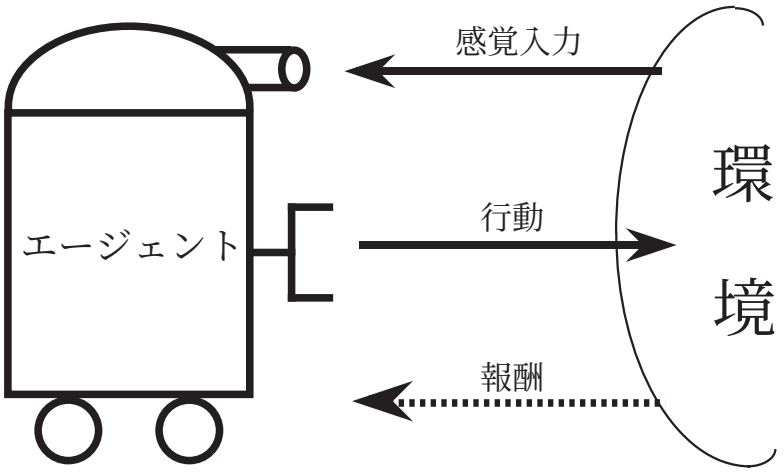


図 1: ロボットと環境との関係

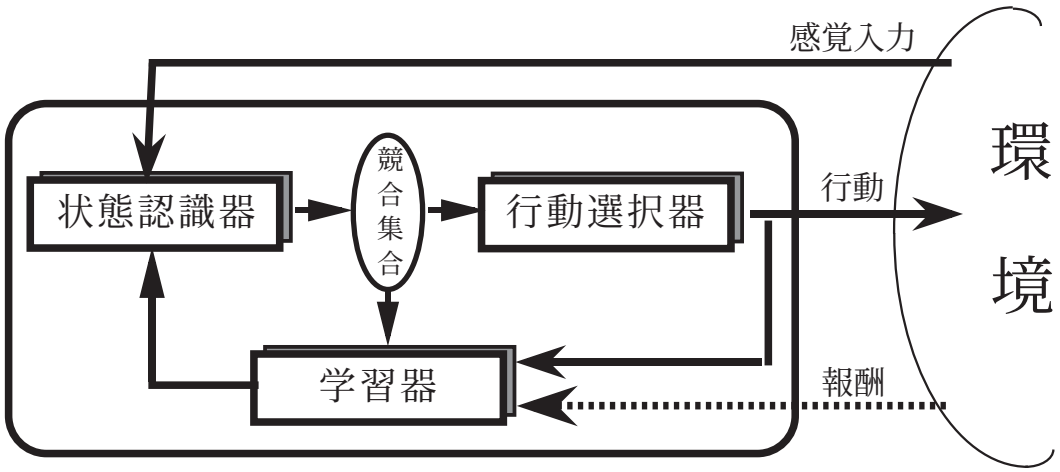


図 2: 強化学習システムの一般的な枠組み

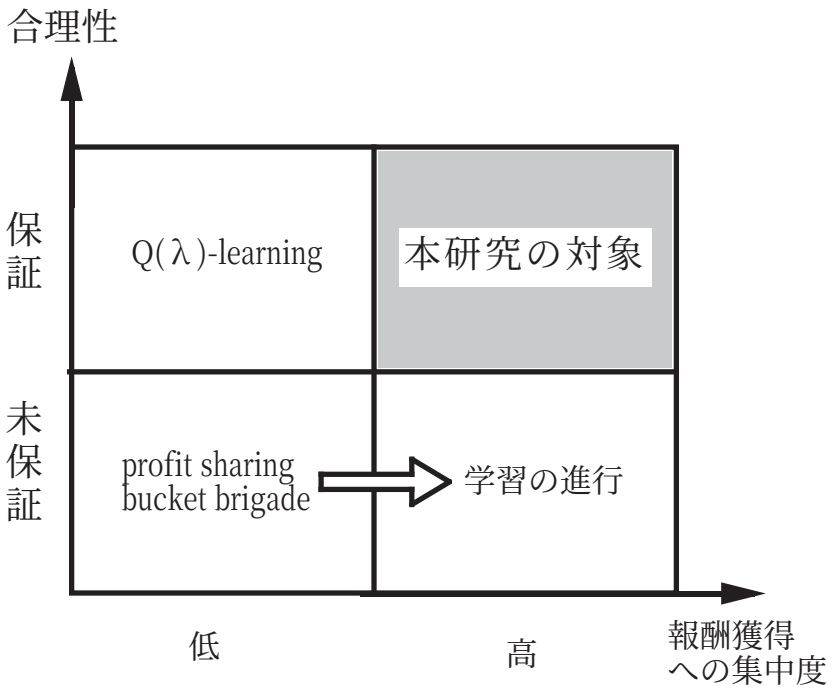


図 3: MDPs を対象とする経験強化型学習システム

で行われる. 感覚入力に照合するルールは競合集合を形成する. その中から行動選択器により行動が選ばれ, 環境へ出力される. 学習器は, エージェントの行動および環境からの報酬に基づいて, ルールに対する強化学習を実行する. MDPs を対象とする強化学習システムでは, 状態認識器は所与と考えても差し支えないので, 学習器と行動選択器だけが設計の対象とされる.

強化学習の目的は, できるだけ多くの報酬を獲得することにある. 一般に, 最適政策を得るためには, 環境を広く同定しなければならない. しかし環境の同定を重視して行動すると, 学習途中での報酬獲得が軽視されがちになる. このように強化学習では, **報酬獲得**と**環境同定**といった相反する目的が要求される. 本論文では [山村 95] の分類にならい, 学習途中での報酬獲得を重視する接近を**経験強化型**, 最適政策の獲得を保証する接近を**環境同定型**と呼ぶ.

MDPs を対象とする既存の研究をこれら経験強化, 環境同定という観点で分類するとそれぞれ図 3 および図 4 のようになる. 図 3 の横軸は報酬獲得への集中度を表し, 図 4 の横軸は環境同定への集中度を表す. 縦軸は, 経験強化型の場合には, 学習結果に何らかの合理性が保証されているかどうかで分類される. 一方, 環境同定型では, 合理性はもとより最適性が保証される. しかし環境側に何らかの仮定が導入される場合が多いため, 縦軸は仮定の

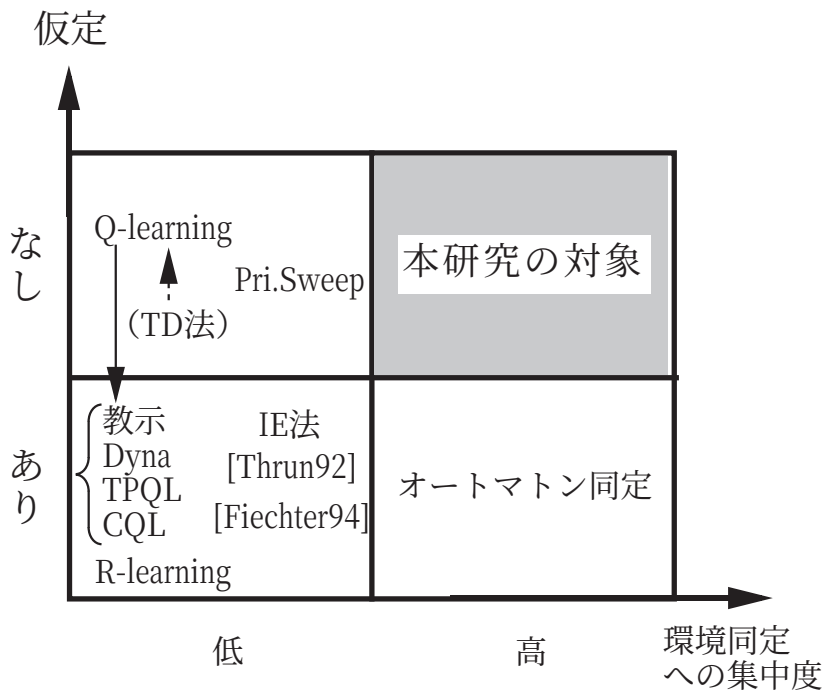


図 4: MDPs を対象とする環境同定型学習システム

有無により分類した.

古典的な強化学習はすべて経験強化型である. その中でも, 主として,Classifier System [Holland 86, Holland 87] の枠組みの中で研究されてきた bucket brigade[Holland 86, Goldberg 89] と profit sharing[Holland 87, Grefenstette 88, Liepins 89] が有名である.

bucket brigade は行動ごとに賭を行なう. あるステージでの勝者には, 報酬および次のステージで競合する行動が支払う賭金の合計が与えられる. 報酬の有無に関わらずルールが強化されるため, 明らかに効率の悪い行動が学習されがちである. これを避けるためには様々な経験的工夫が必要であることが指摘されている [Riolo 87, Smith 91].

profit sharing は, 報酬が与えられたときに, それまでに使われたルール系列を一括的に強化する手法である. そこでは, 報酬をルールにどのように分配するかが非常に重要な問題となる. この分配方法を強化関数という. [Grefenstette 88] は, 強化値を一定としている. [Holland 87] や [Liepins 89] は, 報酬から離れる程単調に強化値を減少させている. profit sharing は bucket brigade に比べ単純であり, 工夫すべきところは強化関数に集約されているという特徴がある. しかし, 従来の強化関数が一般にどれだけ有効であるかは明らかにされていない.

経験強化型は多分に思いつきのアイデアという側面を持ち, その挙動もパラメータに敏

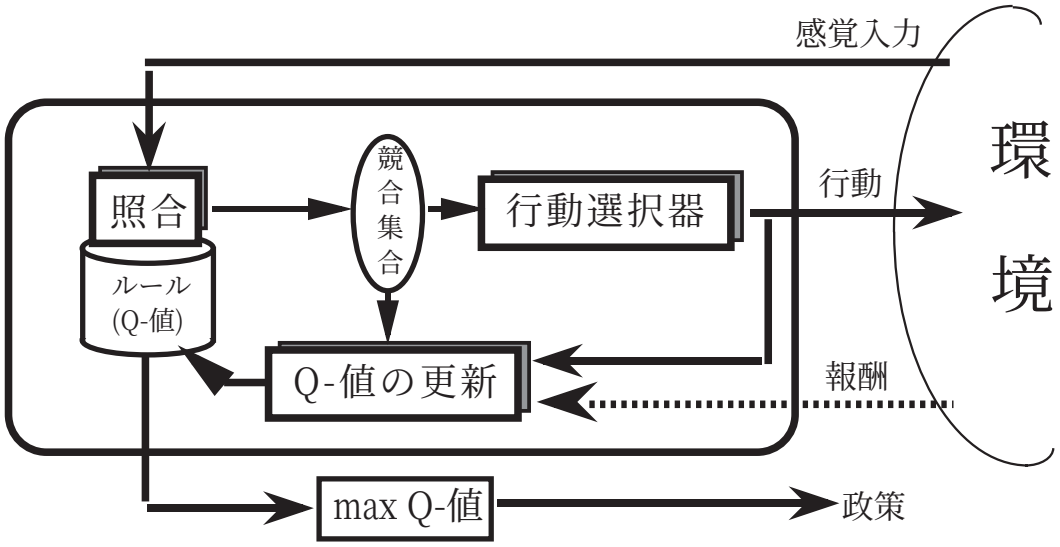


図 5: Q-learning の構成図

感であり安定していない. そもそも経験強化型の目的は, 環境同定に要する行動を節約して, その分, 継続的に報酬を得ることにある. そのため経験強化型が実用に耐える手法になるためには, 一定の合理性を保証する必要がある. すなわち図 3 の右上に位置し得る手法が重要であると考える.

環境同定型の先駆的研究に [Sutton 88] の temporal difference 法 (TD 法) がある. TD 法はマルコフ過程として定式化された環境の各状態の評価を同定する. [Watkins 92] の Q-learning は TD 法の発展形として提案され, 状態の評価だけでなく状態と行動の対の評価を割引期待報酬という量をもとに同定する. すなわち TD 法とは異なり,MDPs を対象とすることができる.

Q-learning の構成を図 5 に基づき説明する. Q-learning の状態認識器はルールベースであり, 各ルールは Q-値と呼ばれる重みを持っている. 行動選択器には,Q-値に基づくルーレット選択や, 90%の確率で最大の Q-値を持つルールを選択するなど, 様々なものが用いられる. いま状態 x で行動 a をとり状態 y に遷移し報酬 r を得た場合を考える. このとき学習器では次式に従い Q-値が更新される.

$$Q(x,a) = (1-\alpha)Q(x,a) + \alpha(r + \gamma \max_b Q_(y,b)) \tag{1}$$

ここで γ は割引率である. あるスケジュールに従って学習率 α を減少させ, 多数の行動の後に Q-値が収束すると, 各状態における最大の Q-値を持つルールの選択が最適政策となることが証明されている.

Q-learningの利点は,環境がMDPsであれば,最適政策の獲得が保証されることである．そのため,現在までに,非常に多くの研究で利用されている．例えば[Lin 90]はQ-learningを用いてエージェントが敵を避けつつ餌を捜し求めるシミュレーションを行っている．[Mahadevan 91a, Mahadevan 91b]は,[Brooks 86]のSubsumption ArchitectureとQ-learningとを組み合わせてOBELIXと呼ばれる移動ロボットを作成している．[Bersini 94]はQ-learningを利用し,恒常性を維持するように振る舞うエージェントに関する研究を行っている．[Munos 94]はQ-learningとbucket brigadeとを組み合わせ, 苗木を避け雑草を刈るエージェントに関するシミュレーションを行っている．[Gambardella 95]はAnt-system[Colorni 91]と呼ばれる蟻の集団を模した分散アルゴリズムにQ-learningを導入し, Traveling Salesman問題を解いている．

一方,Q-learningの欠点は,解析が保証しているのはあくまで最終結果であることと, 解析が先の3つの構成要素のうちの行動選択器を含んでいないことである．その結果,場合によっては無駄な行動を多く含みQ-値の収束までに膨大な行動回数を要することがある．また,学習の途中段階でのQ-値には近似解としての意味はなく, あくまで収束を待たねばそこそこの解すら得られない場合がある．さらにQ-値は環境の構造や学習率などのパラメータに非常に敏感であるため, 実問題へ応用し一定の成果を得るためには,一般に,細かなチューニングが必要となる．

Q-learningの発展形はこれらの欠点の克服を目指している．[Sutton 90a, Sutton 9ob]のDynaは, 報酬を得た経験をエージェント内でリハーサルすることでQ-値の収束を加速している．[McCallum 92]は,Kohonenの Feature Mappingの考えを応用し, 多数の状態を近傍構造に基づいて汎化させた**TPQ-learning** (TPQL)と呼ばれるシステムを提案し,Q-値の収束を加速している．しかしこれら2手法は強化学習の特徴のひとつである非決定性を正しく取り扱えないという欠点を持つ．

Q-値の収束を早めるために**教示**が導入される場合もある．[Lin 91a, Lin 91b]は部屋の中を動き回るロボットのシミュレーションにおいて, 教師が毎回正解を教えることにより, Q-値の収束を加速している．[Clouse 92]は[Lin 91a, Lin 91b]とは異なり,時々,教師が正解を教える枠組みを提案している．教示はQ-値の収束を早めるひとつの有望な手段ではあるが,教示が行えない状況も多々考えられるので,一般的な方法とは言えない．

[Singh 92]はタスクをサブタスクに分解し,各サブタスクを別々のモジュールに学習させる**CQ-learning** (CQL)と呼ばれる手法を提案している．各モジュールでは,与えられた範囲内のQ-値を独立に更新するので, 高速化が期待できる．しかし報酬は1箇所でしか与えられないことおよびタスクが予めサブタスクに分解されていることを仮定している点が問

題点として指摘される.

[Peng 94] は Q(λ)-learning と呼ばれる一度に複数個の Q-値を更新する方法を提案している. これはマルコフ過程における同種の手法である [Sutton 88, Dayan 92] の TD(λ) を発展させた方法である. しかしそこでは MDPs 下での最適性は保証されていない. そのため環境同定型とは言えず, むしろ経験強化型に分類されるべき手法であるが, 学習途中での報酬獲得に徹底しきれていないため, 経験強化型としても中途半端な手法である.

Q-learning から派生した手法に **R-learning**[Schwartz 93] がある. Q-learning では割引期待報酬を最大化する政策を求めるのに対し, R-learning では割引は導入せず, 平均報酬を最大化する政策の獲得を目指す. [Schwartz 93] は, ある特定の環境では, 割引期待報酬よりも平均報酬の方が優れる場合があると主張しているが, 非決定性を含む環境下での性能は実験的に調べれているに過ぎない [Mahadevan 94].

ところで MDPs の環境では, 行動のサンプルを収集することによって環境を統計的に同定することが可能である. 環境が正確に同定されているならば, MDPs における最適政策を求める手法としてよく知られた **Policy Iteration Algorithm** (PIA)[Bertsekas 76, ワグナー 78] を利用することができる [Singh 92]. したがって, 環境を効率よく同定するためのアルゴリズムが重要となる. すなわち環境同定型としては図 4 の右上に位置する手法, すなわち環境をより少ない行動回数で同定できる手法が重要であると言える.

環境を効率よく同定するためには, まだ十分に試されていない行動を優先して試すべきである. [Kaelbling 91] の IE 法では, 以前効果的であった行動と選択回数の少ない行動を同程度に評価することにより, 効率のよい環境同定を期待している. IE 法は移動ロボットへの適用 [Yanco 92] や k-DNF と呼ばれる問題クラス [Kaelbling 94] では一定の成果を得ているが, ルールの評価値に報酬の期待値を反映させていることから不必要な行動を行う可能性があることおよび報酬に遅れのない環境を前提としていることが問題点として指摘される.

[Moore 93] の **Prioritized Sweeping**(Pri.Sweep) では, 現状態で選択回数の少ないルールをより優先的に選択することで効率のよい環境同定を期待している. しかしそこでは連立 1 次方程式の解法としてよく知られる Gauss-Siedel の反復法を利用した, 政策を計算する際の計算量の節約が中心的な課題とされ, 最適政策を得るための行動回数の節約は重視されていない.

[Fiechter 94] は Valiant の PAC-learing の考えに基づき (ε , δ)-最適政策と呼ばれるものを定義し, それをより少ない行動回数で獲得するための手法を提案している. しかしいかなる状態からも即座に始点に戻れる”reset”と呼ばれる行動を仮定しており, エージェントによる学習を想定している強化学習の立場からは不自然な設定となっている.

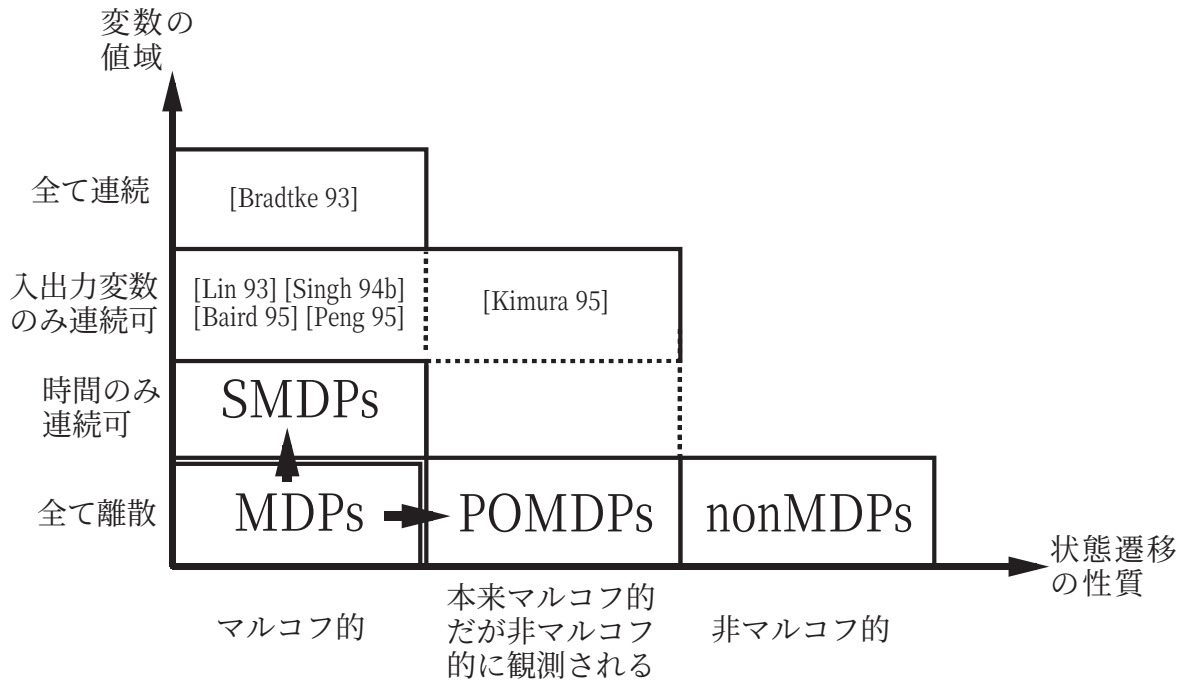


図 6: MDPs 以外に適応可能な強化学習システム

[Thrun 92] は環境モデルを構築し, その構築されたモデルの中で, より不確かな部分を集中的に探索するアルゴリズムを提案している. しかしこの手法は始点からゴールに向かう navigation task に特化されたものであり, さらにいかなる状態からもゴール状態が観測できるという不自然な仮定が導入されている.

ところで環境の同定はオートマトンの同定という形でも研究されている. しかしそこでは強化学習の特徴のひとつである非決定性が正しく取り扱われていない. [Dean 92],[Shen 93],[Rouvellou 95] らは環境は, 本来, 決定的であるとしている. [Basye 95] は非決定性を扱うために, つねに可逆な行動が存在するという仮定を導入している. これはエージェントによる学習を想定している強化学習の立場からは不自然な設定である.

MDPs 以外に適応可能な強化学習システム

MDPs 以外に適応可能な強化学習システムに関する研究も近年活発になりつつある. それらを図 6 にまとめる. 図中の横軸は状態遷移の性質, 縦軸は変数の値域を表す. 変数には, 感覚入力, 行動出力, 時間が考えられる.

MDPs を越えたクラスの中で, 近年, 最も注目を集めているクラスが, **Pertialy Observable Markov Decision Processes (POMDPs)**[Lovejoy 91] である. これは本来環境は

マルコフ的だが, エージェントの感覚能力が不十分なために非マルコフ的とみなされるクラスである. POMDPs に対するアプローチは, モデルを構築することで非マルコフ性を排除する立場と, モデルを構築せずに, あくまで現在の感覚入力だけから行動を決定する立場のふたつに大きく分かれる. 前者は一般に**モデル構築型**と呼ばれ, 後者は**モデルフリー型**もしくは**メモリーレス型**と呼ばれる.

モデル構築型は [Chrisman 92] により提案された手法に端を発する. そこでは同じルールを使ったときの状態遷移確率が学習の初期と後期とで異なっているかどうかをカイ 2 乗検定で検定し, もし相違があると判断された場合, 状態を 2 分割する. また,[McCallum 93] も [Chrisman 92] と同種の手法を提案しているが, そこではカイ 2 乗検定の代わりに信頼区間の考えが導入されている. これら 2 手法はバッチ処理的に状態分割を行うので, 一般に, 学習には膨大な行動回数を要する. [McCallum 95] は逐次的に木構造を更新していくことで効率的な状態分割方法を提案した. しかしモデル構築型では, 一般に, モデルを維持するために多くのメモリを要することおよび完全なるモデルを構築するためには膨大な計算量を要することが問題点として指摘される.

モデルフリー型はモデル構築型が持つ上記の欠点を排除するために [Singh 94a] により提案された. そこでは従来の決定的政策に代わる確率的政策という考えを提示し, MDPs における有効性を論じている. [Jaakkola 94] は確率的政策を得るためのアルゴリズムを提案しているが, 現状では, 局所解への収束が保証されてるに過ぎない. 確率的政策は Q-learning などの環境同定型からの素直な発展であり, 解析が容易であるという特徴を持つ. しかし一般にエージェントは, 確率的政策を出力することはできず, エージェントによる学習を想定している強化学習の立場からは不自然な設定と言える.

また初期の [Whitehead 90] や [Lin 92, Lin 93] などもモデルは構築しないのでモデルフリー型に分類される. [Whitehead 90] はブロック積み上げ問題に Q-learning を適用した際に生じる非マルコフ性を排除する方法について述べているが, 十分な解析は行われていない. [Lin 92, Lin 93] は教示により POMDPs に対応する手法を述べているが, つねに教示が行えるとは限らず, 一般的な手法とは言えない.

[Cassandra 94, Littman 95] らも POMDPs を対象とする学習システムを提案している. そこでは環境モデルが完全に与えられているという仮定の下で, いかにして非マルコフ性を排除するかが議論されている. これは環境が未知であるという前提の下で学習を進める強化学習の立場からは不自然な設定である.

POMDPs 同様,MDPs から発展したクラスに **Semi Markov Decision Processes (SMDPs)** がある. これは時間が連続なマルコフ決定過程である. [Bradtke 94] は SMDPs に適応可能

な強化学習システムを提案している. 基本的なアイディアは, 一定のサンプリングタイムで連続時間を離散化し, Q-learning などの従来手法を利用することにある. SMDPs では, 感覚入力および行動出力は従来通り離散的なので, このような従来手法を直接利用した手法が可能である.

入出力変数の連続化に関しても, 現在, 徐々に研究されつつある. [Lin 93] は移動ロボットのシミュレーションにおいて連続値を取り扱っているが, 実験的に性能が調べられているに過ぎない. [Singh 94b] は連続値を扱うための関数近似システムについて議論している. しかしそこでは政策を得るためのアルゴリズムについては述べられていない. [Baird 95] は関数近似システムを用いて, 連続入出力へ対応している. しかしそこでは非決定性を含む環境下での最適性は完全には保証されていない. [Peng 95] は Memory-Based 的な考えを応用して, 連続入出力へ対応している. しかしこの手法は状態間に距離が定義できるクラスにしか適応できない. [Kimura 95] は山登り法を応用した手法の中で連続入出力の取扱いを可能としている. そこでは,MDPs における最適性は保証されないものの, POMDPs などへの発展が期待されている.

[Bradtke 93] は感覚入力, 行動出力, 時間すべてが連続な場合を取り扱っている. しかしそこでは比較的取扱いが容易な Linear Quadratic Reguration (LQR) と呼ばれる問題のみが対象とされている.

POMDPs 以外の非マルコフ的なクラスをここではまとめて **nonMDPs** と呼ぶ. その中の一例として [Tenenberg 92] の研究がある. そこでは報酬の与えられ方が非マルコフ的になっている. 具体的には, 複数種類の目的が要求され, それらを適切な順序で達成した場合のみ報酬が与えられる. [Tenenberg 92] は各目的ごとに Q-learning を走らせ, それらを後に統合する手法を提案している. しかし一般に, この手法がどの程度有効かは明らかにされていない.

非マルコフ性は, 複数種類のエージェントが同時に学習する場合にも容易に発生する. [Tan 93] はマルチエージェント系における Q-learning の性能を実験的に調べている. [Littman 94] はマルコフゲームという形でふたつのエージェントが各々の目的を同時に追求する枠組みを提示している. そこではゲーム理論の考えを応用し,minimax 点を求めるような Q-learning が提案されている. 現状では, つねに相手の手がみえるといった不自然な仮定が導入されているが, 今後の発展が期待される問題クラスのひとつである.

2.3 本研究の接近法

本論文では MDPs を対象とする強化学習システムについて考察する. 強化学習システムを報酬獲得および環境同定という観点で捉え, まず始めに学習途中での報酬獲得を重視した接近である経験強化型に注目する. 特に,profit sharing に注目し, 従来, 場当たり的に設定されていた強化関数に対し解析的な考察を加える. 具体的には, 明らかに無駄なルールを強化しないという局所的な合理性と必ずいくらかの報酬を継続的に得るという大局的な合理性を満足するための強化関数の必要十分条件を求める.

経験強化型は, 環境同定に要する行動を犠牲にし, その分, 継続的な報酬の獲得を実現した手法である. そのため, 環境全体が完全に同定されることは希であり, 学習終了時に最適政策が得られる保証もない. ここに経験強化型の限界がある.

そこで本論文では次に, 環境同定に注目し, より効率的に環境を同定することができる行動選択器である **k-確実探査法** (k-Certainty Exploration Method) およびその発展形である **ℓ-確実探査法** (ℓ-Certainty Exploration Method) を提案する. これらは, まだ十分試されていない行動を優先して選択するための手法である. さらに, 環境同定の確からしさを考慮し, 同定の精度を徐々に向上させることも行う. これにより PIA と組み合わせることによって, 環境が同定される確からしさを考慮した政策を得ることができ, 最適政策の獲得に要する行動回数の短縮も期待される.

ところで, 強化学習の応用, 例えば自律走行ロボットへの適用を考えた場合, 最適政策を得るために無限の試行を繰り返すことは不可能であり, 学習の初期段階からそこそこの報酬を獲得しつつ, 同時に環境の同定を行い, 報酬獲得の水準を段階的に向上させるような挙動が求められる.

そこで本論文では, 最後に, 報酬獲得と環境同定のトレードオフを陽に考慮し, 学習の初期段階から終了に至るまで一貫した挙動を示すことのできる強化学習システムである **MarcoPolo** を提案する. MarcoPolo は, 経験強化型と環境同定型の相補的性質に着目して, 両者を巧みに統合した強化学習システムである. そのため MDPs における強化学習システムとして非常に完成度の高いものになることが期待される.

3　経験強化型学習における報酬割当の理論的考察

3.1　はじめに

本章では経験強化型学習の代表である profit sharing を取り上げ, 報酬の割当を決定する強化関数について解析的に考察する. 既に述べているように, この強化関数は学習結果に多大な影響を及ぼすことが予想されるにも関わらず, 従来, つねに一定や, 徐々に減少させたりと, 場当たり的に定められてきた. 本章では, 明らかに無駄なルールを強化しないという局所的な合理性と必ずいくらかの報酬を継続的に得るという大局的な合理性を満足するための強化関数に関する必要十分条件を求める.

以下 2.2 節では, 以降で使用される用語を定義した後, 強化学習に要求される局所的な合理性を保証するための強化関数の必要十分条件を求める. そして 2.3 節では,2.2 節の条件が大局的な合理性を満たす強化関数の必要十分条件と一致することを示す.

3.2　無効ルールの抑制

3.2.1　準備

以降の議論で使われる用語を定義する. ロボットの挙動は, 感覚入力とそのとき選択した行動の対, すなわちルールの系列からなる. 初期状態あるいは報酬を得た直後から次の報酬までのルール系列を**エピソード** (episode) という. 例えば図 7 の環境でロボットが $\overrightarrow{x\grave{a}} \cdot \overrightarrow{y\grave{a}} \cdot \overrightarrow{z\grave{b}} \cdot \overrightarrow{x\grave{a}} \cdot \overrightarrow{y\grave{b}} \cdot \overrightarrow{x\grave{a}} \cdot \overrightarrow{y\grave{b}}$ と行動したとすると, この中には $(\overrightarrow{x\grave{a}} \cdot \overrightarrow{y\grave{a}} \cdot \overrightarrow{z\grave{b}} \cdot \overrightarrow{x\grave{a}} \cdot \overrightarrow{y\grave{b}})$, $(\overrightarrow{x\grave{a}} \cdot \overrightarrow{y\grave{b}})$ の 2 つのエピソードが含まれている (図 8 参照).

profit sharing ではエピソード単位で強化を行う. **強化関数** (reinforcement function) とは, 報酬からどれだけ過去かを引き数とし, 強化値 (reinforcement) を返す関数である. 時点は離散なので f_i によって報酬から i ステップ前の強化値を参照する. 長さ l のエピソード $(r_l \cdots r_i \cdots r_2 \cdot r_1)$ に対して, ここでは $\omega_{r_i} = \omega_{r_i} + f_i$ によってルールの重みを強化する場合について考える.

ある episode で, 同一の感覚入力に対して異なるルールが選択されているとき, その間のルール系列を**迂回系列** (detour) という. 例えば図 7 の環境で, エピソード $(\overrightarrow{x\grave{a}} \cdot \overrightarrow{y\grave{a}} \cdot \overrightarrow{z\grave{b}} \cdot \overrightarrow{x\grave{a}} \cdot \overrightarrow{y\grave{b}})$ には, 迂回系列 $(\overrightarrow{y\grave{a}} \cdot \overrightarrow{z\grave{b}} \cdot \overrightarrow{x\grave{a}})$ がある (図 8 参照). 迂回系列上のルールは, 報酬の獲得には貢献しない可能性がある. 現在までのすべてのエピソードで, つねに迂回系列上にあるルールを**無効ルール** (ineffective rule) と呼び, それ以外を**有効ルール** (effective rule) と呼ぶ. 無効ルールと有効ルールとが競合するならば, 明らかに無効ルールを強化すべきではない. 以下

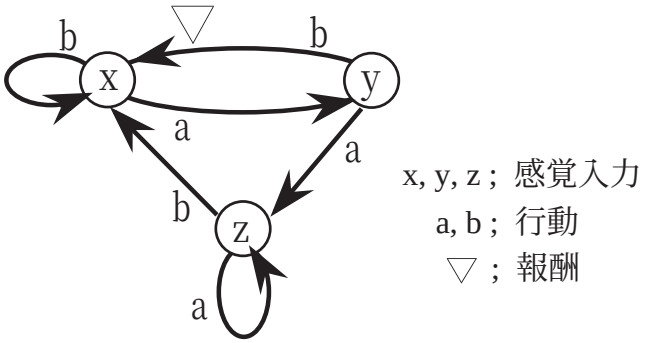


図 7: 例で用いた環境

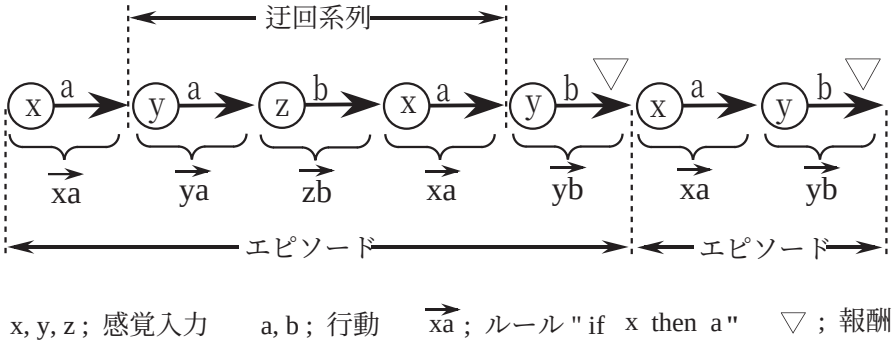


図 8: エピソードおよび迂回系列の例

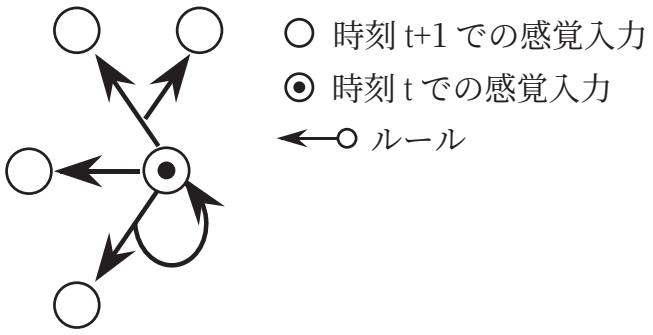


図 9: ルールの競合構造の一例

では図 9 のようにひとつの感覚入力に注目した状態遷移図の一部を特に**ルールの競合構造** (conflict structure) という.

3.2.2 無効ルール抑制定理

本節では無効ルールの抑制を保証する定理を導出する. 無効ルールの抑制とは, 無効ルールがそれと競合する有効ルールを差し置いて一番には強化されないことである. まず, 無効ルールを抑制することが最も困難となるルールの競合構造を選ぶ. ここで 2 つの競合構造 A と B について, A において無効ルールを抑制できる強化関数のクラスが B のそれに包含されるとき, A は B よりも困難であるという. 次に, 最も困難な構造に対して, 無効ルールを抑制するための強化関数の必要十分条件を求める. 最後にそれを任意のルールの競合構造に対して拡張する.

補題 1 (最も困難な構造)
唯一の回帰的無効ルールの抑制が最も困難である. □

証明は付録 A に示す. 図 10 に最も困難な競合構造を示す. L 本の有効ルールと唯一の回帰的無効ルールが競合している. ここで, 行動をとった結果感覚入力の変化が生じないルールは**回帰的**であるという.

補題 2 (唯一の回帰的無効ルールの抑制)
唯一の回帰的無効ルールが抑制される必要十分条件は

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

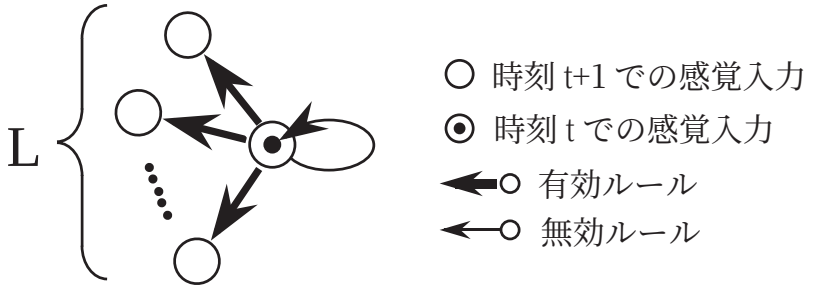


図 10: 最も困難な競合構造

□

証明は付録 B に示す. ここで, W はエピソードの最大長, L は同一感覚入力下に存在する有効ルールの最大個数である.

補題 1 と 2 から推移律により直ちに次の定理が得られる.

定理 1 (無効ルールの抑制)
任意の無効ルールが抑制される必要十分条件は

$$\forall i = 1, 2, ..., W. \quad L \sum_{j=i}^{W} f_j < f_{i-1} \tag{3}$$

□

ここで,W はエピソードの最大長, L は同一感覚入力下に存在する有効ルールの最大個数である. 以後式 3 を**無効ルール抑制条件**と呼ぶ.

3.2.3 定理の意味

定理 1 は, 無効ルールが有効ルールを差し置いて一番に強化されることがないという局所的合理性を保証している. したがって, 各感覚入力では最も大きな重みを持つルールを選択すればよい. 特に $L = 1$, すなわち同一感覚入力下に存在する有効ルールの個数が最大 1 個の場合は, つねに最適なルールの選択が保証される. 一般にはこの L の値は学習以前には知ることができないが, 実装にあたっては, L を可能な行動出力の種類引く 1 とすれば十分である.

従来の定数関数 (図 11-a) や等差的減少関数 (図 11-b) は定理を満たさず, 非合理的な学習をする場合がある. 定理を満たす最も簡単な強化関数としては, 次に示す等比減少関数が

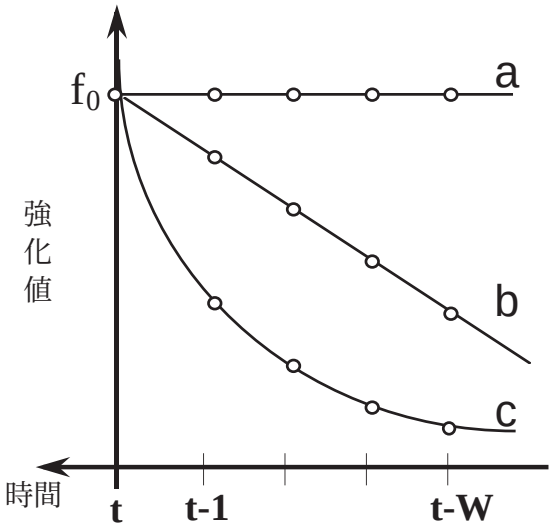


図 11: 強化関数の例. a:定数関数, b:等差的減少関数, c:等比減少関数

考えられる (図 11-c).

$$f_n = \frac{1}{S} f_{n-1}, \quad n = 1, 2, \cdots, W-1. \tag{4}$$
$$\text{ただし, } S \geq L+1$$

ここで, S を**強化減少比**と呼ぶ. この関数が定理を満たすことは以下のように確認される.

$$\begin{aligned} L\sum_{j=i}^{W} f_j &= \frac{L}{S}\sum_{j=i-1}^{W-1} f_j \qquad (5) \\ &= \frac{L}{S} f_{i-1} + \frac{L}{S}\sum_{j=i}^{W} f_j - \frac{L}{S} f_W \end{aligned}$$

よって,

$$\begin{aligned} L\sum_{j=i}^{W} f_j &= \frac{L}{S-1}(f_{i-1} - f_W) \qquad (6) \\ &\leq f_{i-1} - f_W \\ &< f_{i-1} \end{aligned}$$

この他にも定理を満たす関数には様々なものが考えられる.

数値例

次に定理の意味を数値的に例示する. 環境を図 12 に示す. 感覚入力 x, y, z では, それぞれ, ルール $\overline{x}\hat{a}, \overline{y}\hat{a}, \overline{z}\hat{a}$ が有効ルールである. ルールの選択は重みに比例した確率に基づくいわ

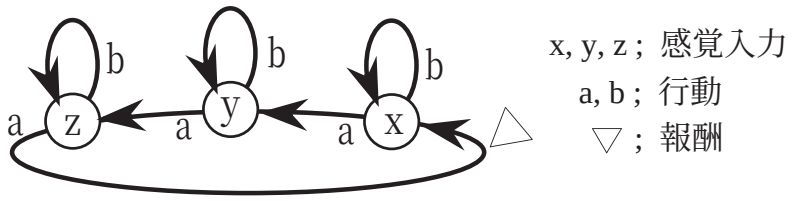


図 12: 数値例で用いた環境

ゆるルーレット選択によって行った. 強化関数としては a,b,c の 3 種類の関数を用いた. 関数 a は一定である. 関数 b は公差 -5.0 の等差減少関数である. 関数 c は公比 $1/2$ の等比減少関数である. 報酬値, すなわち f_0 は 100.0 で, ルールの重みの初期値はすべて等しく 10.0 とした. エピソード長は実装の都合上, 最大 15 とした.

実験は強化関数のそれぞれについて, 乱数の種を変えて 20 例ずつ行い, $\omega_{\overline{x}a}/\omega_{\overline{x}b}$ の時間的変化を見た. 図 13a に関数 a, 図 13b に関数 b, 図 13c に関数 c を用いた結果を示す. 縦軸は $\omega_{\overline{x}a}/\omega_{\overline{x}b}$ の対数軸でとってある. 横軸は時間で, ルールがひとつ選ばれるごとに 1 ずつ加算してある.

関数 a や b では重みの比が 1.0 以下となり無効ルールが強化される例が多く観察された. さらにルールの重みの比が振動を繰り返す例も多く観察された. 関数 c でもごく希に振動的な挙動が観察されるが, それは一時的なもので最終的には重みの比は皆増加していった. これらの結果は定理の意味を鮮明に示している.

3.3 報酬プランの獲得

3.3.1 準備

定理 1 は, ひとつの感覚入力における無効ルールの抑制という強化学習に要求される局所的な合理性を保証している. 本章ではより大局的な合理性に焦点をあてる.

いかなる方法を用いても最適な学習がなしえない環境を排除するために, 環境は無限かつ可達と仮定する. 可達とは, 適当な行動を選択すれば任意の感覚入力から任意の感覚入力へ到達しうることである.

各感覚入力に対して高々1 個の有効ルールを選択する部分関数を考える. このうち与えられた環境において無限にルールを選択し続けられるものを**プラン** (plan) という. 単位行動当たりの報酬の期待値が 0 でないプランを**報酬プラン** (rewardfull plan) といい, それを最大化するものを**最適プラン**という. プランは有効ルールのみから構成されるので局所的な

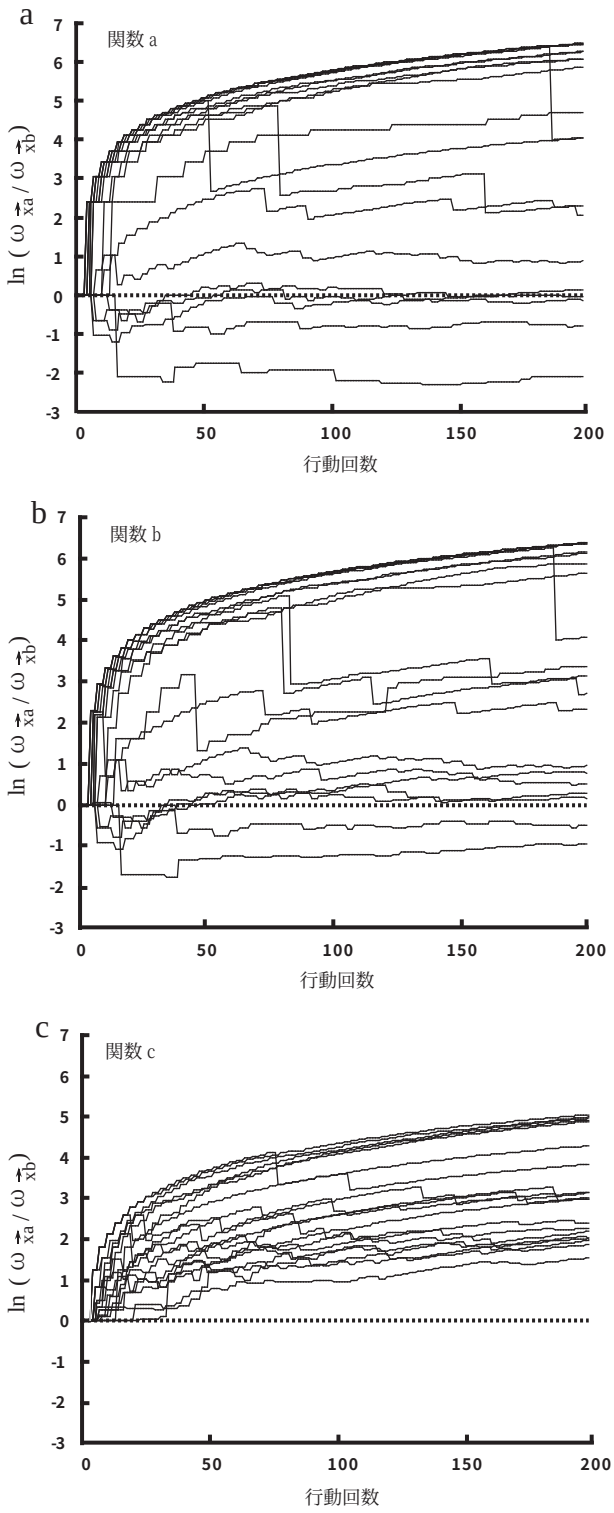


図 13: ルール xa とルール xb の重みの比の時間的変化. a; 関数 a, b; 関数 b, c; 関数 c

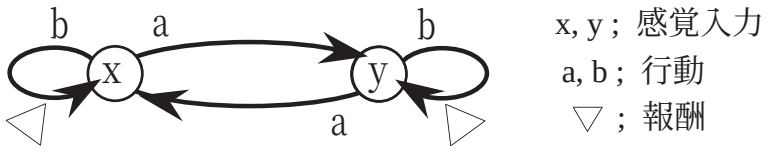


図 14: 例で用いた環境

合理性は保たれているが, 大局的に見てそのプランによって継続的に報酬が得られるとは限らない. 例えば, 図 14 の環境ですべてのルールが有効ルールであるが, プラン $\{\overrightarrow{xa}, \overrightarrow{ya}\}$ は報酬を得られない.

本章では, 定理 1 を満たす強化関数が報酬プランを学習できることを示す.

3.3.2 報酬プラン獲得定理

プランは有効ルールのみから構成されるので, 定理 1 すなわち無効ルールの抑制が必要である. しかし, 単なる有効ルールの組み合わせが継続的に報酬を得られるわけではないので, その十分性について調べる.

補題 3 (無効ルール抑制条件の十分性) 無効ルール抑制条件を満たす強化関数は報酬プランを獲得する. □

証明は付録 C に示す.

補題 3 より直ちに次の定理が得られる.

定理 2 (報酬プランの獲得)
無効ルール抑制条件は報酬プランを獲得するための必要十分条件である. □

3.3.3 定理の意味

定理 2 は, profit sharing における局所的な無効ルールの抑制条件が, 大局的な報酬プランの獲得条件に一致することを意味する. 局所的な無効ルールの抑制は, 例えば, 迂回系列の共通部分をとるような記号的方法によっても実現できる. しかし, それだけでは大局的な報酬プランが獲得されるとは限らないので, profit sharing にとっての定理 2 の意義は大きい.

数値例
定理の意味を数値的に例示する. 環境を図 15 に示す. 初期状態は S_0 である. 太線は有効

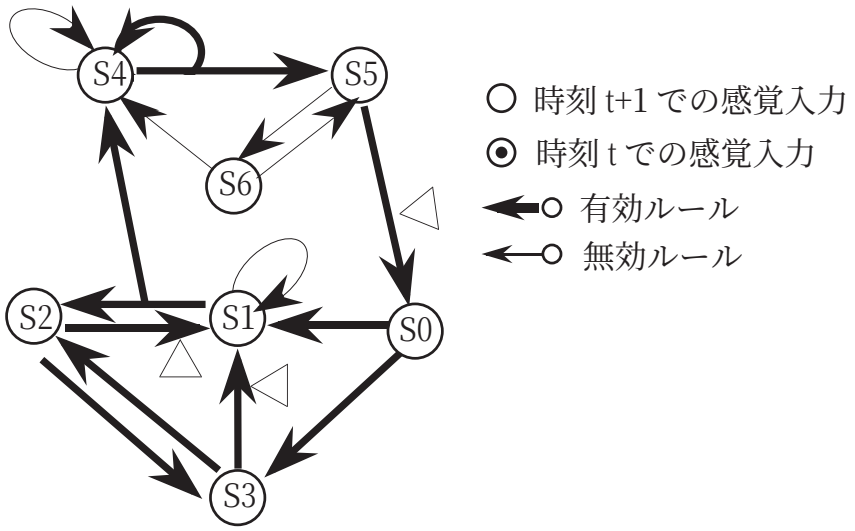


図 15: 数値例で用いた環境

ルール, 細線は無効ルールである. 可能なプランは図 16 に示す 7 種類である. このうち報酬プランは図 16-a から図 16-e までの 5 種類である. エピソードの始点は S_0 と S_1 だけなので, 図 16-e のように特定の感覚入力 (ここでは S_3) を含まない報酬プランも存在することに注意されたい.

この環境に対して乱数の種を変えて 100 回実験を行った. 学習は 200 ステップで打ち切った. 強化関数は次の 3 つを用いた. 関数 a は一定である. 関数 b は公差 -5.0 の等差減少関数である. 関数 c は公比 $1/3$ の等比減少関数である. 他の条件は前節での実験と同一である.

報酬プランの獲得された回数および失敗の内訳を表 1 にまとめる. 第 1 行は図 16-a ～図 16-e といった報酬プランの獲得された回数. 第 2,3 行は報酬プランが獲得されなかった場合の失敗の内訳である. 無報酬プラン (rewardless plan) とは, 図 16-f や図 16-g といった報酬プラン以外のプランが獲得された回数である. 無効ルールを含む (with ineffective rule) とは, プランが構成されない, すなわち無効ルールが抑制しきれていない試行の回数である.

表 1 より関数 c を用いた場合は, つねに報酬プランが獲得されていることがわかる. 関数 a や b では, 報酬プランが獲得されないばかりか, 無効ルールが抑制しきれていない場合が多々あることがわかる. 特に関数 a は関数 b よりも性能が悪い. これは関数 b の減少の効果が, 学習が誤った方向へ進むのをいくらかは防いでいるためと思われる. このように定理 2 を満たす関数 c のみが報酬プランを確実に獲得していることがわかる.

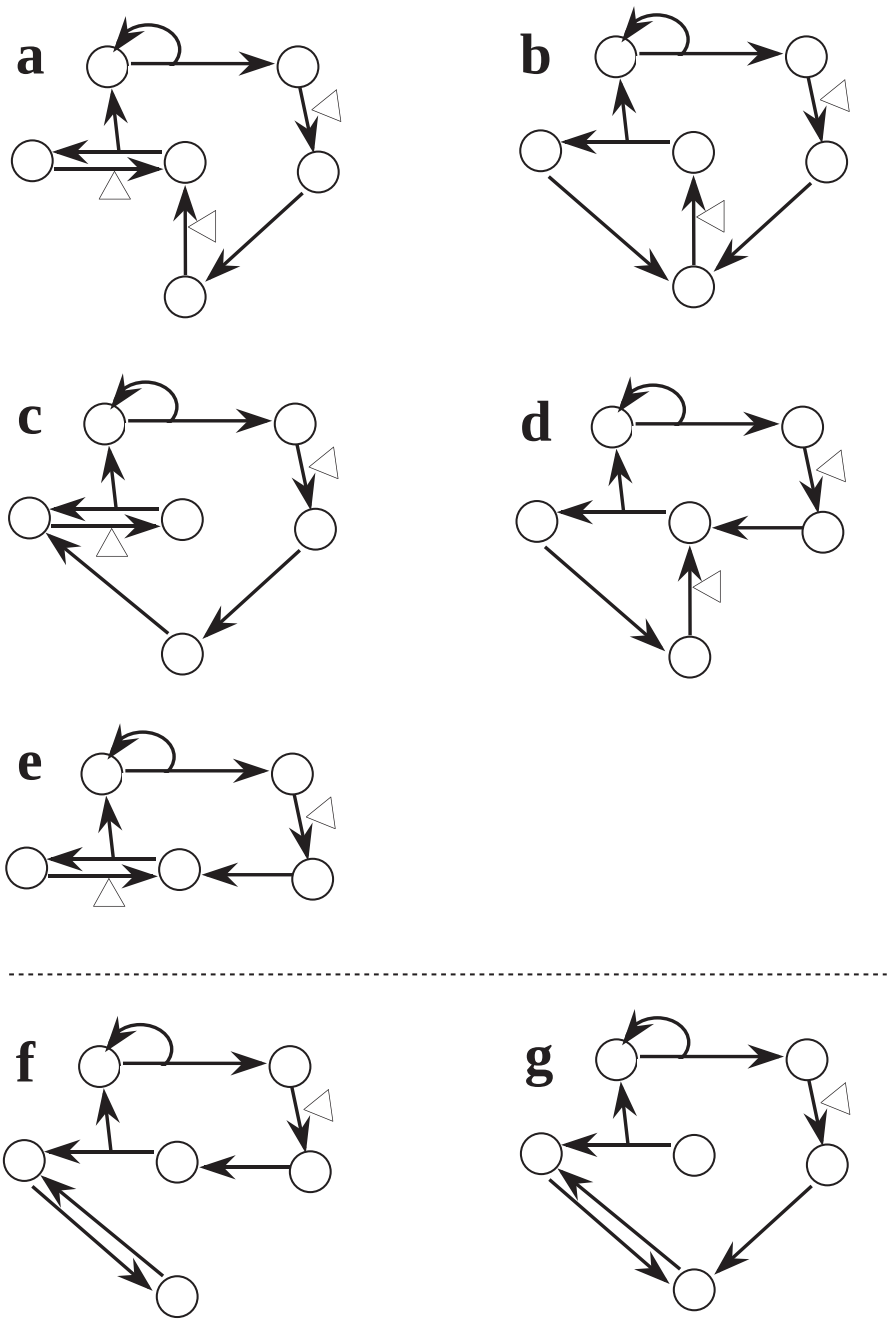


図 16: 数値例で用いた環境におけるすべてのプラン．a〜e が報酬プランである．

	関数 a	関数 b	関数 c
報酬プラン	38	60	100
無報酬プラン	2	3	0
無効ルールを含む	60	37	0

表 1: 強化関数を変えたときの報酬プランの獲得回数 (上段), 報酬プランが獲得されなかった場合の失敗の内訳 (下段).

3.4 他手法との比較

定理 2 を満たす強化関数を用いた profit sharing が Q-learning や bucket brigade と比べどの程度定量的に優れているかを数値例により示す.

図 17 に示すようなハニカム状の環境に置かれたエージェントを考える. 周囲はトーラスである. 環境中には 20 l の容積を持つ燃料タンクがつねに 10 個置かれている. エージェントはひとつの行動を出力するたびに 1 l の燃料を消費し, 所有する燃料が 0 になると死を意味する. エージェントが燃料タンクを摂取すれば, その燃料が 20 l だけ増加し, 報酬が与えられる. 但し, エージェントが保有できる燃料の上限値は初期燃料値を越えることはできない. 燃料タンクはエージェントに摂取されると, その場から消失し, 同一垂直軸上の任意の場所にランダムに発生する.

エージェントは以下の 3 種類の感覚入力を知覚することができる; 1)**SF**: 前方に何かが存在する, 2)**NF**: 前方に何も存在しない, 3)**ON**: 何かの上に乗っている.

エージェントは以下の 4 種類の行動を出力することができる; 1)**MOVE**: 前に一歩動く, 2)**TURN**: 右に 60 度回転する, 3)**RIGHT**: 燃料タンクを右手で摂取する, 4)**LEFT**: 燃料タンクを左手で摂取する. 但し,LEFT は確率 50%で失敗する.

図 18 にこの問題の状態遷移図を示す. 図 18 からもわかる通り, この問題には以下の 2 種類の報酬プランが存在する; 1)**plan 1**: {NF → TURN, SF → MOVE, ON → RIGHT}, 2)**plan 2**: {NF → TURN, SF → MOVE, ON → LEFT}. plan 1 が最適プランである.

この問題に対し, エージェントの初期燃料値を変化させたときの報酬プランの獲得率を 3 手法で比較した. 報酬プランの獲得率は, 乱数の種を変えて行った 100 回の実験中何回までが燃料切れで死ぬまでに報酬プランを獲得することができたかを表す. 結果を図 19 に示す.

bucket brigade は大局的な合理性はもとより局所的な合理性すら保証していない. したがって図 19 に示すように, 十分な初期燃料値を保有していたとしても, 報酬プランが学習

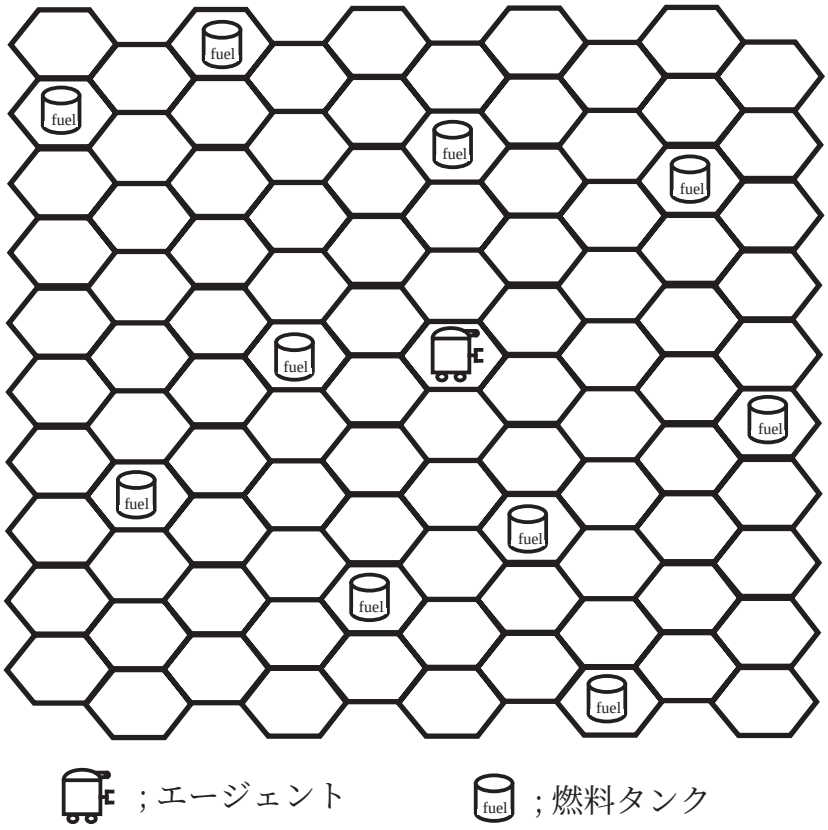


図 17: 他手法との比較に用いた環境

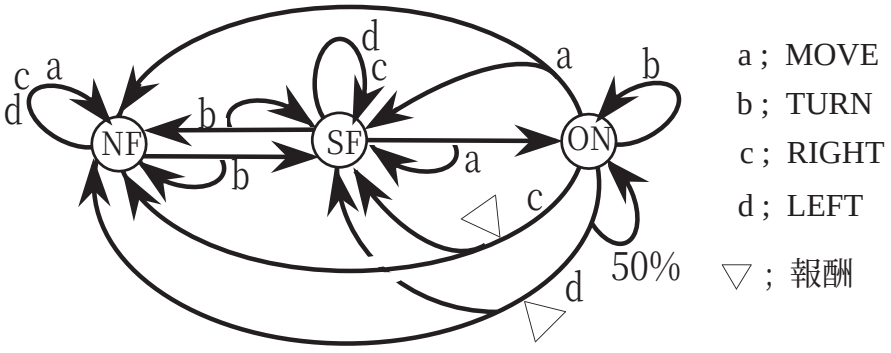


図 18: 実験で用いた環境における状態遷移図

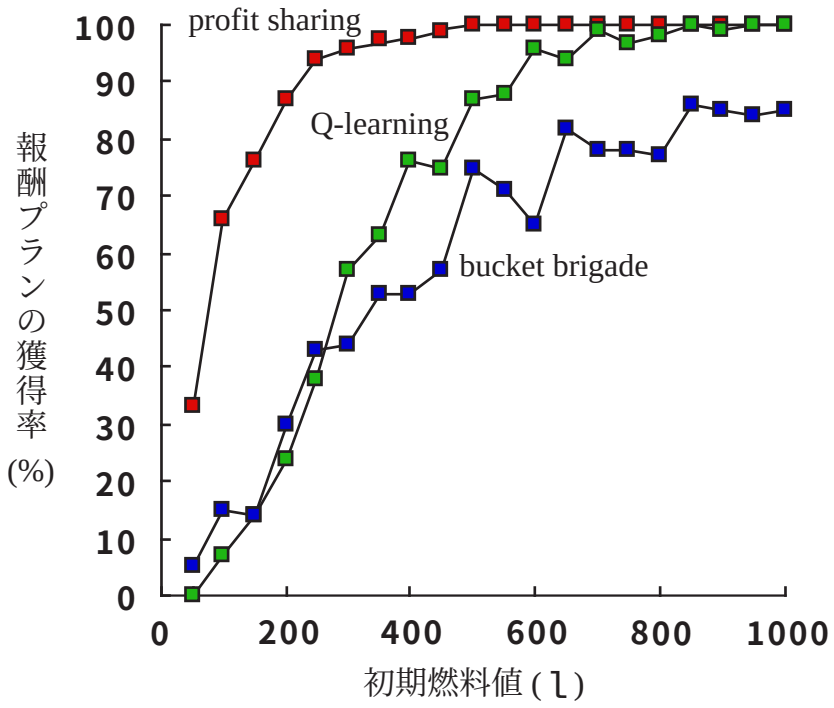


図 19: 初期燃料値を変化させたときの報酬プランの獲得率

しきれず死に至る場合が多々観察された.

定理 2 を満たす profit sharing は報酬プランの獲得を保証している. profit sharing は報酬を得た経験により直ちにルール系列が強化されるので, ひとたび燃料タンクを摂取すれば, 素早く報酬プランが獲得される. したがって十分な初期燃料値がなくても高い確率で生き残ることができる.

Q-learning は最適性が保証されているので, この問題でもつねに最適プランが獲得されるはずである. しかし学習には膨大な経験を要するため, 少ない初期燃料値では報酬プランが学習しきれず死に至る場合が多い. Q-learning の最適性は profit sharing にはない魅力的な点ではあるが, その学習速度の遅さから, このように実用に耐えない結果を生む場合が多い.

3.5 おわりに

本研究では, profit sharing をとりあげ, 従来場当たり的に設定されてきた強化関数について解析的に考察した. まず, 局所的に見て報酬を得ることに貢献しない無効ルールが, それと競合する有効ルールよりも強化されてしまわないための必要十分条件を求めた. 次に, 大局的に見て学習されたルールの選択プランによって報酬を得るための必要十分条件が, 先に求めた条件と同値であることを示した. 本研究によって, profit sharing は局所的な合理性と同時に大局的な合理性も満足する有効な強化学習法となりうることがわかった.

今後の課題は大きく分けてふたつある. ひとつは学習速度に関する解析である. 定理を満たす関数には様々なものがある. 特により単純な等比減少関数であっても, 強化減少比を変えることによって関数の形が変わる. そしてそれが, 学習の早さに影響を及ぼすことが考えられる. 強化減少比と学習速度との関係について早急に明らかにしたいと思う.

もうひとつは得られた条件の緩和である. 本定理は, 行動選択器については何も規定していないが, この点を固定すれば, 条件が緩和される可能性がある. また目的を, ある確率 p で学習されるための十分条件という形にすれば, 更に緩い条件が得られる可能性がある. 定理として美しいのは, 本章で求めたような必要十分条件かもしれないが, 実際に使うことを考えると, そのような十分条件にも大いに存在意義があると考える.

ところで profit sharing は報酬により経験を強化していくタイプの学習手法なので, 素早い学習が可能であるという利点を持つが, その反面, Q-learning では保証されている最適性が保証されないという欠点を持つ. そこで, 次章では, より効率よく最適政策を得るための手法について考察する.

付録

A　補題 1 の証明

以下の証明のために次の言葉を定義する. ひとつの感覚入力に対して, 適用可能なルールの数を**競合数**, 可能な状態遷移の総数を**枝分かれ数**という.

簡単のため $L = 1$ とする. それ以外も全く同様である.

明らかに, 無効ルールが強化される回数が多いほど, 無効ルールを抑制できる強化関数の集合は小さくなり, 無効ルールの抑制はより困難となる. よって, 強化される回数の大小のみを考えれば十分である. 枝分かれ数の小さい順に可能な競合構造を数え上げる.

1) 枝分かれ数が 1 の場合

競合数 1 なので, 明らかに困難ではない (図 20-a).

2) 枝分かれ数が 2 の場合

競合数が 1 ならば 1) と同様 (図 20-b). 競合数が 2 の場合を考える. 回帰ルールを含む場合 (図 20-c) とそうでない場合 (図 20-d) とに分けられる. ここで A を有効ルール, B を無効ルールとする. 任意のエピソードで, 1 回の A につき B は繰り返し使われた可能性がある. 図 20-c の方が図 20-d よりも 1 回の A につき B を選ぶ回数を多くできる. したがって, 図 20-d よりも図 20-c の方がより困難な構造である.

3) 枝分かれ数が 3 の場合

競合数が 1 の場合は 1) と同様 (図 20-e). 競合数が 2 の場合 (図 20-f) は 2) と同様に, 唯一の回帰的無効ルールと競合する場合が最も困難となる. 競合数が 3 の場合 (図 20-g) 無効ルールが全部で 2 個なので, 無効ルール 1 個あたりの強化回数は 1 個の場合よりも減少する. よって, 競合数 2 のときが最も困難である.

同様に, 枝分かれ数 n のときも, 唯一の回帰的無効ルールと競合する場合が最も困難となる.

ゆえに, 無効ルールを抑制することが最も困難となる構造は, 有効ルールが唯一の回帰的無効ルールと競合する構造である. (証明終り)

B　補題 2 の証明

簡単のため $L = 1$ とする. それ以外も全く同様である.

エピソードの長さを W とし, 唯一の回帰的無効ルールと競合する有効ルールが, 報酬か

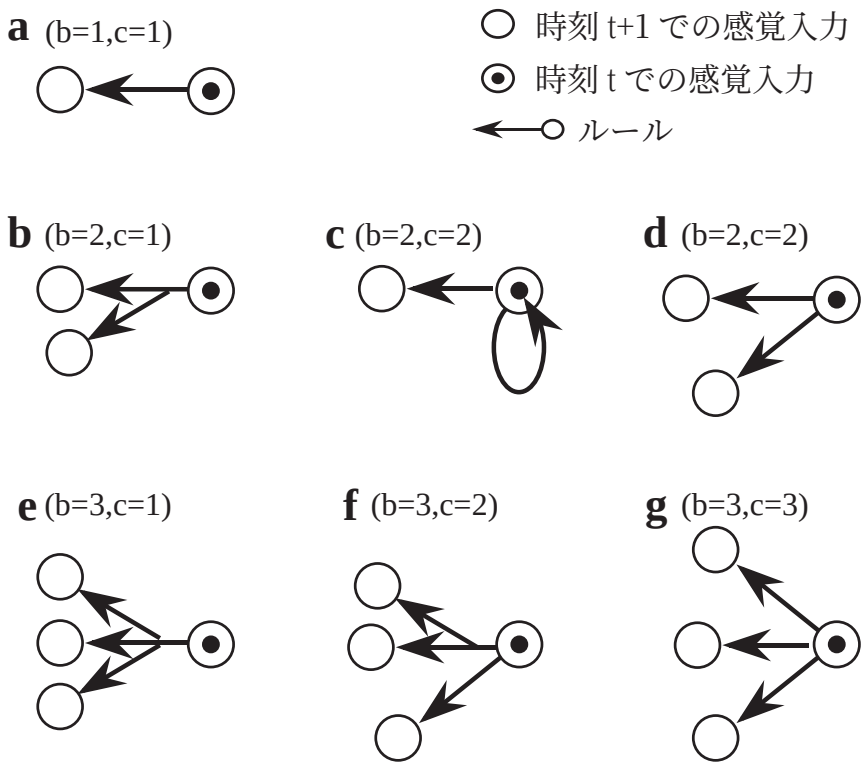


図 20: 証明で用いた競合構造

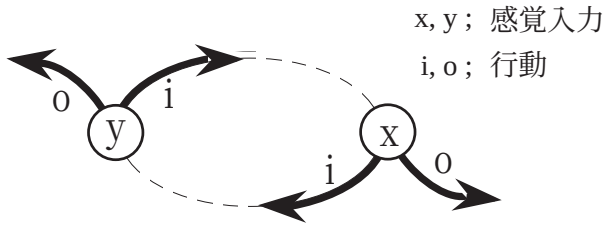


図 21: 証明で用いた環境

ら N ステップ以前に選ばれていたとする. 回帰的無効ルールの強化値が最大となるのは, 有効ルールより以前の $(W-N)$ ステップにおいて, つねに回帰的無効ルールが選ばれた場合である. このとき, 回帰的無効ルールが, 有効ルールを差し置いて強化されないためには, 次の不等式が必要である.

$$\sum_{j=N+1}^{W} f_j < f_N, \quad N = 0, 1...W-1. \tag{7}$$

十分性は明らかである. (証明終り)

C 補題 3 の証明

あるプランが報酬プランとならないためには, 報酬を伴わないループを含む必要がある. このようなループを構成するためには, 少なくとも 2 個のエピソードが必要で, ループの出口となる感覚入力において次の不等式群が成り立つ必要がある (図 21 参照). ここで簡単のため $L = 2$ かつ出口は x, y の 2ヶ所であるとするが, それ以外の場合も全く同様である.

$$\Delta\omega_{\overrightarrow{xo}} < \Delta\omega_{\overrightarrow{xi}} \tag{8}$$

$$\Delta\omega_{\overrightarrow{yo}} < \Delta\omega_{\overrightarrow{yi}} \tag{9}$$

$\overrightarrow{xo}, \overrightarrow{yo}$ はそのループから出ていくルール, $\overrightarrow{xi}, \overrightarrow{yi}$ はそのループ内に戻るルールである. また Δ はそのルールに加算される強化値の総和を表す. 無効ルール抑制条件を満足し, かつ, $\overrightarrow{xi}$ を含むエピソードが $\overrightarrow{xo}$ を含んでいたとすると, 定理 1 より,

$$\Delta\omega_{\overrightarrow{xo}} > \Delta\omega_{\overrightarrow{xi}} \tag{10}$$

よって, $\overrightarrow{xi}$ を含むエピソードは $\overrightarrow{xo}$ 以外のルールを使ってループの外へ出る必要がある. $\overrightarrow{yi}$ に関しても同様である. このとき定理 1 より次の不等式群が成り立つ.

$$\Delta\omega_{\overrightarrow{yo}} > \Delta\omega_{\overrightarrow{xi}} + \Delta\omega_{\overrightarrow{yi}} \tag{11}$$

$$\Delta\omega_{\overrightarrow{xo}} > \Delta\omega_{\overrightarrow{yi}} + \Delta\omega_{\overrightarrow{xi}} \tag{12}$$

式 8,9 を辺々加え, 式 11,12 を辺々加えると, 次の不等式が得られる.

$$\Delta\omega_{\overrightarrow{xi}} + \Delta\omega_{\overrightarrow{yi}} > \Delta\omega_{\overrightarrow{xo}} + \Delta\omega_{\overrightarrow{yo}} > 2(\Delta\omega_{\overrightarrow{xi}} + \Delta\omega_{\overrightarrow{yi}}) \tag{13}$$

$\Delta\omega$ は正なので, この不等式を満たす解は存在しない. したがって, 報酬を含まないループは構成できない. ゆえに, 無効ルール抑制条件を満たす強化関数では, 必ず報酬プランが獲得される. (証明終り)

4　強化学習のための環境同定戦略：k-確実探査法

4.1　はじめに

前章では経験強化型の代表として profit sharing を取り上げ, 学習結果の合理性を保証するための定理を求めた. 一般に経験強化型では, 学習終了時に最適政策が得られるとは限らない. 本論文で対象とする MDPs の環境では, 環境が正確に同定されていれば,PIA により最適政策を求めることができる. そのため効率よく環境を同定することができる行動選択器が重要となる. 本章では, 最適政策を追求するという観点から強化学習システムを捉え, 効率的な環境同定戦略を提案する.

以下 4.2 節では, 環境の同定に焦点をあてた行動選択器である **k-確実探査法** (k-Certainty Exploration Method) を提案する. k-確実探査法と PIA を組み合わせることにより最適政策の獲得を実現する. そして 4.3 節では, 他の手法との比較を通じ,k-確実探査法の特徴を明らかにする.

4.2　k-確実探査法の提案

4.2.1　学習システムの基本的な考え

環境同定を重視した強化学習システムの主要部分は, 効率よく環境を同定するための行動選択器にある. 一般に, 確率過程を同定する際には誤差が生じる. 誤差はルールすなわち行動の選択回数に応じて小さくなる. ある確実さで環境が同定されるためには, すべてのルールは一定回数以上選択されなければならない.

各ルールの選択回数のバラツキを極力小さくしながら, すべてのルールを最低 k 回選択することを考える. ここで, 選択回数が k 回以上になっているルールを **k-確実**と呼び, k-確実でないルールを **k-未確実**と呼ぶ. さらに, 感覚入力を単に**状態**と呼び, 特に, その時点で知覚されている状態を**現状態**, その時点までに知覚された状態を**既知状態**と呼ぶ.

現状態で k-確実なルールを選択し, その後再び現状態に戻ることを考える. この際, 現状態で選択可能なルールの中で, そのルールを選ぶと, 以後選択可能なルールがすべて k-確実となるとき, そのルールを **k-確実なループに至るルール**と呼ぶ. 例えば図 22 では現状態で選択可能なルールは 0 と 1 であるが, ルール 0 のみが k-確実なループに至るルールである. k-確実なループに至るルールを選ぶと, 以後 k-確実なルールばかりが選択され, 効率が悪化する. 本研究のアイデアは, そのようなルールを選択候補から除外することにある.

次節では, まず始めに, k-確実なループに至るルールを見いだし, 効率よく環境を同定す

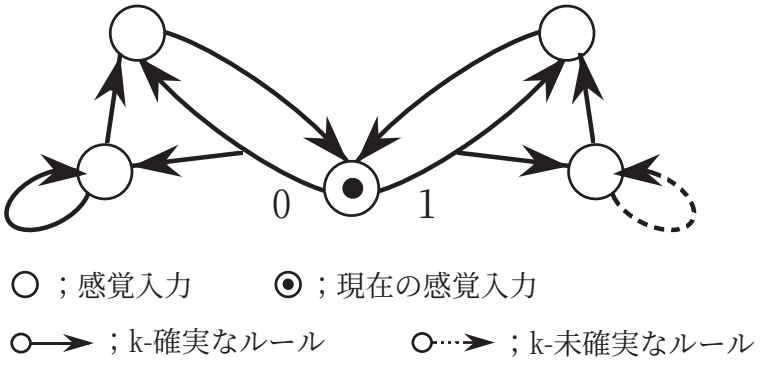


図 22: k-確実なループに至るルールの例

procedure k-確実探査法

begin
if 現状態に対し1-未確実なルールが存在する then k:=1.
else if すべての既知ルールがk-確実である then k:=k+1;

begin
if 現状態に対しk-未確実なルールが存在する then
その中のひとつをランダムに選ぶ.
else すべての既知状態に対しフラグを立てる
for 現状態以外のすべての既知状態 do
if k-未確実なルールが存在する または
フラグの降りた状態に遷移し得るルールが存在する then
その状態のフラグを降ろす
while 新たにフラグを降ろした状態が存在する;
現状態よりフラグの降りた状態に遷移し得るルールの
ひとつをランダムに選択する.
end;

end.

図 23: k-確実探査法のアルゴリズム

る行動選択器である k-確実探査法を提案する. その後, 最適政策を獲得するための k-確実探査法に基づく学習システムを提案する.

4.2.2 k-確実探査法

k-確実探査法は, 図 23 および以下に示すように, 確実さ k の更新および k-確実探査に基づく行動の選択からなる.

k-確実探査法の手順

k-確実探査法はルールを選択した直後に生じる状態遷移の確率および得られる報酬の期待値を同定するために最尤推定を行う. すなわち, 状態 i でルール a を選んだときの状態 j への遷移確率 $\overline{q_{ij}^a}$ および状態 i でルール a を選んだときに得られる報酬の期待値 $\overline{r_i^a}$ は

$$\overline{q_{ij}^a} = \frac{\text{ルール } a \text{ を選び状態 } i \text{ から } j \text{ へ遷移した回数}}{\text{ルール } a \text{ の選択回数}} \tag{14}$$

$$\overline{r_i^a} = \frac{\text{ルール } a \text{ を選択した直後に得た報酬値の総和}}{\text{ルール } a \text{ の選択回数}} \tag{15}$$

で与えられる.

さらに k-確実探査では, k-確実なループに至るルールを選択しないためにフラグを用いる. フラグは各状態ごとに割り当てられる. 行動の種類は予め既知であるが, 現在同定を試みている環境において, 以後生じる状態の種類は前もってはわからない. したがって, フラグの個数はつねに増加する可能性がある. 以下に k-確実探査の手順を示す.

もし現状態で選択可能なルールの中に k-未確実なルールが存在すれば, その中のひとつをランダムに選ぶ. しかしそのようなルールが存在しない場合には, k-確実なループに至るルールを選ばないための処理を行う. 具体的には, まずすべての既知状態にフラグを立てる. そして現状態以外の状態の中で, k-未確実なルールを選択し得る状態, およびフラグの降りている状態に遷移可能なルールを選択し得る状態のフラグを降ろす. このフラグを降ろすための処理は, フラグが全く降ろされなくなるまで繰り返す. 少なくともひとつ k-未確実なルールが存在すれば, その状態のフラグが降ろされ, その降ろされた状態へ遷移し得る状態のフラグも次々降ろされる. したがって, 現状態で選択可能なルールの中で, フラグの降りている状態に遷移することのないルールは k-確実なループに至るルールである. そこで, それ以外の, すなわちフラグの降りた状態に遷移し得るルールのひとつをランダムに選択する.

確実さ k の更新手順

k-確実探査法は k によって確実さを更新する. そこでは k の初期値は 1 とし, 今まで知覚したことのない新たな状態を知覚した場合にも 1 とする. そしてすべての**既知ルール**が k-確実となった時点で k に 1 を加算する. ここで既知ルールとは, 遷移の仕方が既知である状態において, 選択可能なルールである. 既知ルールの総数は, 状態の種類やフラグの個数同様, 前もってはわからないことに注意されたい.

k-確実探査法に基づく学習システム

k-確実探査法に基づく学習システムの構成を図 24 に示す. 本学習システムは, 環境の同定を行う部分と政策を決定する部分とに大きく分かれる. 環境を同定する部分は k-確実探査法である. そこでは k-確実探査により行動を選択し, k により確実さを更新する.

全ルールが k-確実となったとき Policy Iteration Algorithm により政策を求める. その政

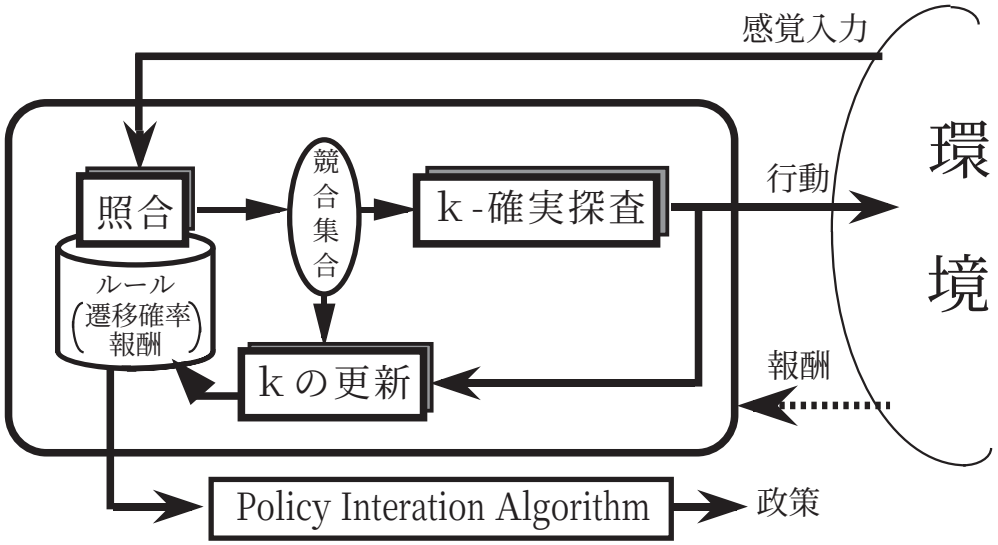


図 24: k-確実探査法に基づく学習システムの枠組み

策は, 全ルールは最低 k 回以上選択されているという意味での確実さを保証している.

4.2.3 動作例

k-確実探査法の挙動を理解するための動作例を図 25 に示す. エージェントは 2 種類の行動を実行でき, 最初 S0 を知覚していたとする. したがって学習以前には図 25-a の形で環境を認識している.

まず k=1 とする. どのルールも 1-未確実なので, ランダムに, 例えばルール 0 が選ばれたとする. その結果, 新たに S3 が知覚される (図 25-b). S3 ではどのルールも 1-未確実なので, ランダムに, 例えばルール 2 が選ばれたとする. その結果, 再び S3 が知覚される (図 25-c). S3 ではルール 2 は 1-確実なので必ずルール 3 が選ばれる. このとき報酬が得られる. 以後, 同様に, ルール 1,5 が選ばれ (図 25-d,e), 再び S0 が知覚される (図 25-f).

現在 S0 で選択可能なルールは共に 1-確実なので, この情報のみではいずれのルールを選択すべきかは判断できない. そこで 1-確実なループに至るルールを選択しないための処理が行われる. まず, 既知状態である S0,S1,S3 にフラグを立てる. S1 には 1-未確実なルールが存在するので S1 のフラグが降ろされる. その結果, ルール 1 が選ばれ, S1 が知覚される (図 25-g). ここで, 状態遷移は確率的なので S1 が知覚されない場合も有り得ることに注意されたい. S1 ではルール 5 は 1-確実なので必ずルール 6 が選ばれる. その結果, 新たに S2 が知覚される (図 25-h). S2 ではどのルールも 1-未確実なので, ランダムに, 例えばルール

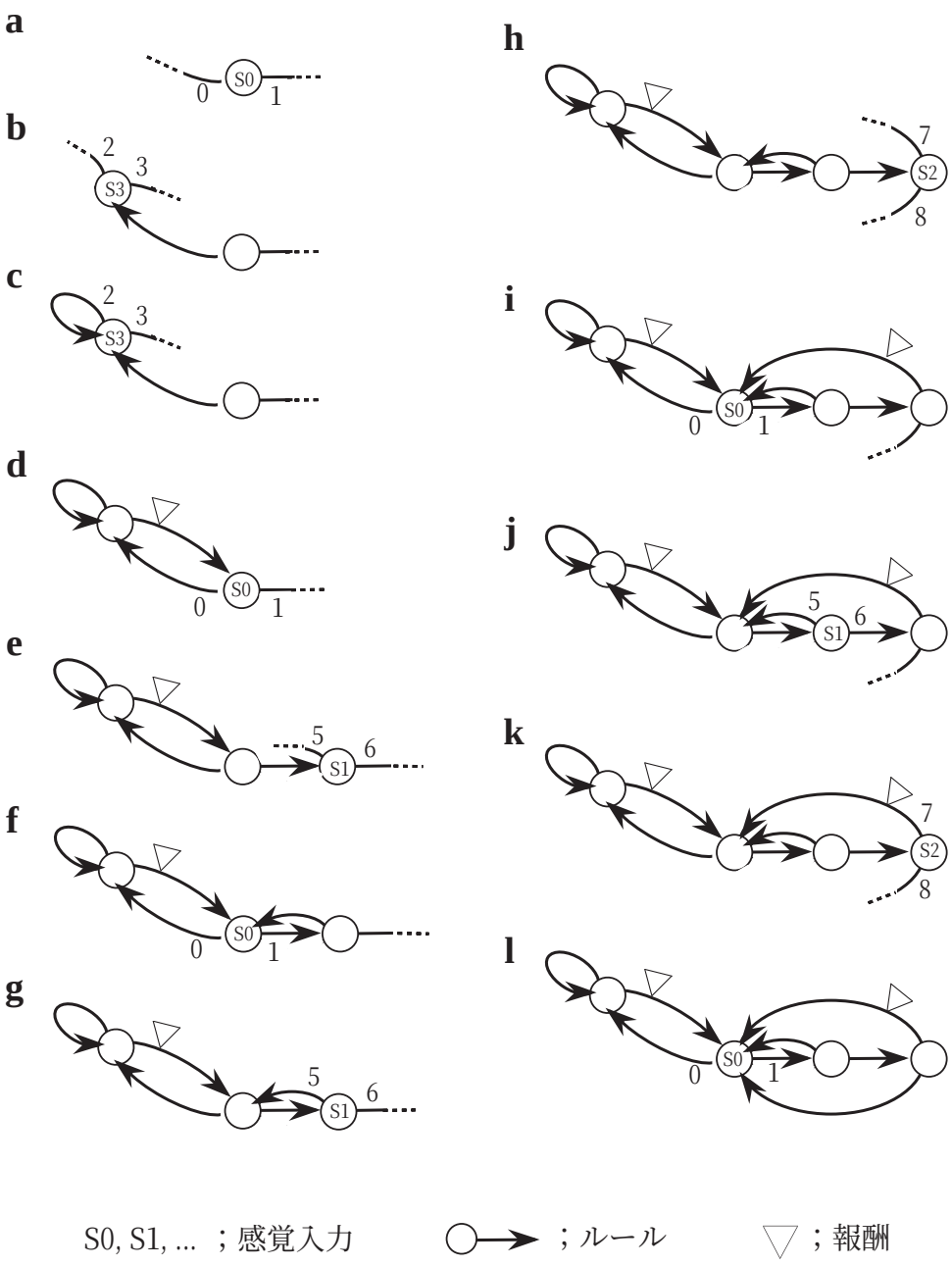


図 25: k-確実探査法の動作例

7 が選ばれたとする. このとき報酬が得られる. そして再び S0 が知覚される (図 25-i).

S0 で選択可能なルールは共に 1-確実なので, 1-確実なループに至るルールを選択しないための処理が行われる. S0,S1,S2,S3 にフラグが立てられ, 1-未確実なルールが存在する S2, および S2 へ遷移し得る S1 のフラグが降ろされる. その結果, ルール 1 が選ばれ, S1 が知覚される (図 25-j). S1 でも同様に 1-確実なループに至るルールを選択しないための処理が行わる. その結果, ルール 6 が選ばれ, S2 が知覚される (図 25-k). S2 ではルール 7 は 1-確実なので必ずルール 8 選ばれる. そして再び S0 が知覚される (図 25-l).

以上の結果, 全ルールが 1-確実となったので, Policy Iteration Algorithm を適用すれば, k=1 という意味での最適政策として, S0 に対してルール 0 を選択し, その結果 S3 に遷移し, そこでルール 3 を選択して S0 に戻る政策が獲得される. なお, Policy Iteration Algorithm については付録 D を参照されたい. さらに確実さを向上させるためには k=2 として同様の処理を繰り返せばよい.

4.2.4 k-確実探査法の特徴

k-確実探査法は次のような特徴をもつ.

(1) k-確実なループに至るルールの抑制

k-確実なループに至るルールの発火を抑制することにより, k-確実なルールが繰り返し選択されることを回避できる.

(2) 決定的環境下では 1-確実で正しく同定可能

決定的状態遷移下では, すべてのルールを 1-確実とすることにより環境が正しく同定される.

(3) 確率的環境下では同定精度を段階的に向上可能

確率的状態遷移下では, k の値を徐々に向上させることにより環境同定の精度を段階的に向上させることができる.

(4) 報酬の獲得場所の影響を受けない

k-確実探査法は, 行動決定時に報酬の影響を一切受けない. これは報酬による強化の影響が必ずしも環境の同定に貢献するとは限らないという我々の考えを反映したものである.

(5) 計算量は多項式オーダーで抑えられる

k-確実探査法が要する計算量は, 行動の種類を m, 感覚入力の種類を n とすると, 空間的には $O(mn^2)$, 1 行動に要する時間量としては $O(mn^3)$ である. 状態を走査する処理があるので, 時間量は多いが, 多項式オーダーで抑えられている. また, Policy Iteration Algorothm に関しては多項式時間で解けることが知られている [Papadimitriou 87].

k-確実探査法には以上のような特徴がある．次章では他の手法と比べて定量的にどの程度優れているかを示す．

4.3 k-確実探査法の性能評価

4.3.1 多重後戻り環境での行動回数の見積り

k-確実探査法には, k-確実なループに至るルールの発火を抑制するという特徴がある．本節では, この特徴の有効性を確認する．

強化学習の目的は単位行動当たりの期待獲得報酬を最大化する政策を見いだすことにあるが, 学習システムの評価基準としては, 最適政策をできるだけ少ない行動回数で見いだすことおよび次の行動を決定するための計算量をできるだけ少なくすることが考えられる．前者は, エージェントの感覚あるいは動作の回数をできるだけ減らしたいこと, 後者は, 次の行動を決定するための考える時間をできるだけ減らしたいことにそれぞれ対応する．一般に, 強化学習の研究では, 前者が重要とされ, 後者は多項式オーダーであれば実際上あまり問題とされない．なぜなら, 強化学習は未知なる環境への適応を取り扱っているので, 行動は予測し難いコストやリスクを本質的に伴うが, 内部の計算量は, 比較的予測や制御が容易であると考えられるからである．したがって以下の比較では, 行動回数の多寡により学習システムを評価する．

k-確実探査法は状態を走査する処理を含むが, これに対し, 現状態に対して選択回数が最も少ないルールを選択する方法が考えらる．ここでは, これを**最少選択優先法**と呼ぶ．k-確実探査法が一見複雑な処理を必要とするのに対し, 最少選択優先法は処理が単純であり, 直感的にわかりやすい．しかしこの方法では環境が少し複雑になると行動回数が指数的に増大してしまうことが予想される．

図 26 のような多重に後戻りが存在する環境を考える．この環境に対し, 各ルールを k-確実から (k+1)-確実にすることを考えると, k-確実探査法では

$$\frac{1}{2}(n+1)(n+2)+n+1 \tag{16}$$

最少選択優先法では

$$3\cdot 2^n+\frac{1}{2}n-1 \tag{17}$$

といった行動回数を要する．

ここで示したような多重後戻り環境は, 例えば正しい順番で機械を操作しなければならない際に, 誤った操作を行うと全部やり直しになるようなことを考えるとごく普通に起こ

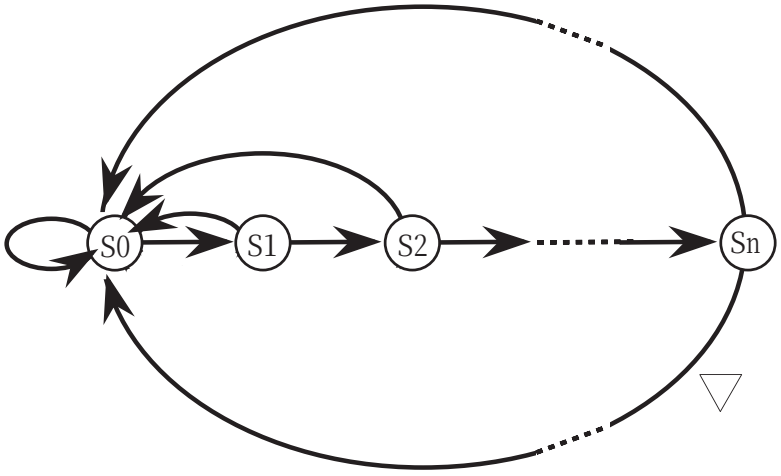


図 26: 多重後戻り環境

り得る. したがって, 最少選択優先法のこのような指数的爆発は好ましい性質ではない.

4.3.2 迷路走行タスクにおける Q-learning との比較

k-確実探査法は, 決定的状態遷移下では, 1-確実で正しく環境を同定することができる. 本節では [Sutton 90] の迷路走行タスクを使用し, Q-learning との比較を通じ, この特徴の有効性を確認する.

実験問題

[Sutton 90] の迷路走行タスクは, 図 27 に示すような 6×9 の迷路的な環境におかれたエージェントが, 始点 (S) から終点 (G) までの最短パスを学習する問題である.
エージェントは各枡目をそれぞれ別々の状態として認識でき, 各状態では上下左右の中からひとつの移動を選択できる. 終点に到達すると報酬が得られ, 始点に戻される. 黒い壁には進入できない. 最短パスは 14 ステップであり, 最適政策は全部で 6 種類ある. この問題において, 最適政策が獲得されるまでの行動回数を比較することとする.

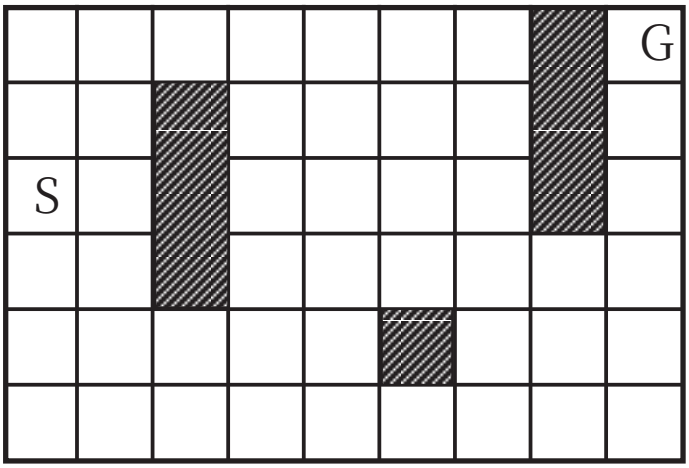


図 27: 迷路走行タスクに用いた環境 [Sutton 90]

k-確実探査法の評価

k-確実探査法に基づく学習システムを用いた場合の結果を表 2 の上段に示す. 実験は乱数の種を変えて 100 回行った.

この環境には確率的な状態遷移が存在しないので, すべてのルールを k=1 とした時点で必ず最適政策が得られる. したがって, この結果は, 全ルールを k=1 とするまでに要した行動回数と同一である. ちなみに k=2 とするまでに要した行動回数の平均は 1503.5, 標準偏差は 397.7 である.

	平均	標準偏差
k-確実探査法	598.62	397.50
Q-learning	4780.42	2124.46

表 2: 迷路走行タスクにおける行動回数の比較

k-確実探査法は, k-確実なループに至るルールを選択しないので, この問題では, 壁に進入しようとする行動は各状態で 1 回しか選ばれない. そのためこのように状態数が多い場合でも, 無意味に行動回数が増大することはない.

Q-learning との比較

Q-learning の結果を表 2 の下段に示す. Q-learning での行動選択器には, ごく一般的な, Q-値に比例したルーレット選択器を採用した. 報酬は 100.0, Q-値の初期値は 10.0, Q-learning の学習定数は 0.5, 割引率は 0.9 である. これらのパラメータは, 予備的実験の後, 行動回数を最少とするものを選んだ.

Q-learning は, 報酬が正しく伝播されなければ学習されない. この問題のように, 46 種類もの状態が存在すると, 始点に報酬が伝播されるためにはかなりの行動を要する. そのため表に示したように, k-確実探査法に基づく学習システムの約 6 倍もの行動回数を要する結果となる.

以上のように数値例からも k-確実探査法に基づく学習システムの有効性は確認された.

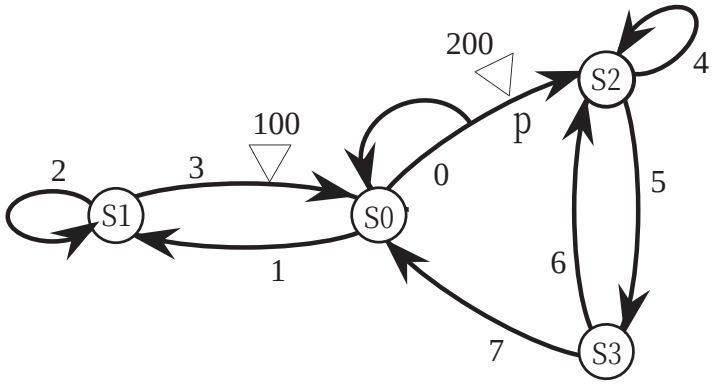
4.3.3 確率的状態遷移下での Q-learning との比較

k-確実探査法は, 確率的状態遷移下では, k の値を徐々に向上させることにより環境同定の精度を段階的に向上させることができる. また, 行動選択時には報酬の影響を受けないので, 環境の状態遷移の確率が変化し, 最適政策が変化したとしても, 学習効率に影響が及ぼされることはない. 本節では, これらの特徴を Q-learning との比較を通して確認する.

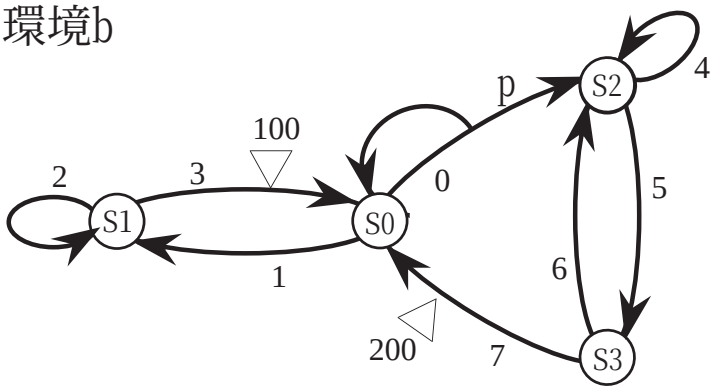
実験問題

図 28 のような環境を考える. ここで p は S0 でルール 0 を選んだときの S2 への遷移確率である. 環境 a では, ルール 3 を実行すれば 100.0, ルール 0 を実行し S2 へ遷移すれば 200.0 の報酬が得られる. 一方, 環境 b では, ルール 3 で 100.0, ルール 7 で 200.0 である. 環境 a,b 共に $p < 0.5$ のときは, ルール 1, 3 を選べば最適政策が得られるが, $p > 0.5$ ではルール 0, 5, 7 を選ぶべきである. この環境で p を $0.1, 0.3, 0.7, 0.9$ と変化させたときの k-確実探査法に基づく学習システムと Q-learning との性能を比較することとする.

環境a



環境b



S0, S1, ... ；感覚入力　⟶ ；ルール　▽ ；報酬

図 28: 確率的状態遷移を持つ環境

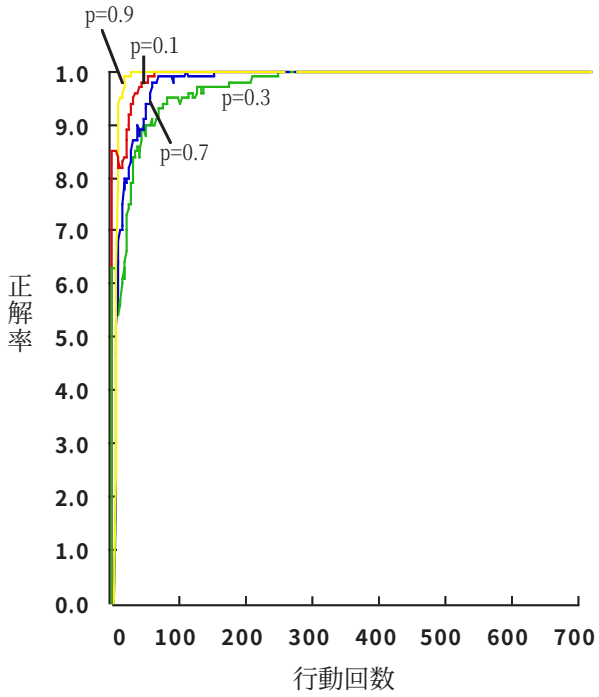


図 29: 環境 a および b の下での k-確実探査法に基づく学習システムの結果

k-確実探査法の評価

k-確実探査法の結果を図 29 に示す. 横軸は行動回数, 縦軸はその行動回数での正解率である. 正解率は, 乱数の種を変えて行った 100 回の実験中, 何回までが, その行動回数の時点で最適政策を獲得していたかを表す. また, 図 30 には正解率が 100%となった時点での k の度数分布を示す.

k-確実探査法は, 行動選択時に報酬の影響を受けない. したがって, 図 29 に示したように環境 a,b に関わらず同一の結果となる. 学習に影響するのは状態遷移の確率のみである. この問題では, $p = 0.5$ を境にして最適政策が変化するので, p が 0.5 に近ければ近いほど, 準最適な政策を誤って最適とみなしやすくなる. k-確実探査法では, そのような場合, k を大きくし, 環境同定の精度を向上させることにより対応する. したがって, 図 30 に示したように, p が 0.3 や 0.7 のように 0.5 に近いときは, 0.1 や 0.9 のときに比べ, より大きな k が要求され, k の分布がばらつく.

Q-learning との比較

Q-learning の結果を図 31a,31b に示す. 図 31a,31b はそれぞれ環境 a,b に対応するもので

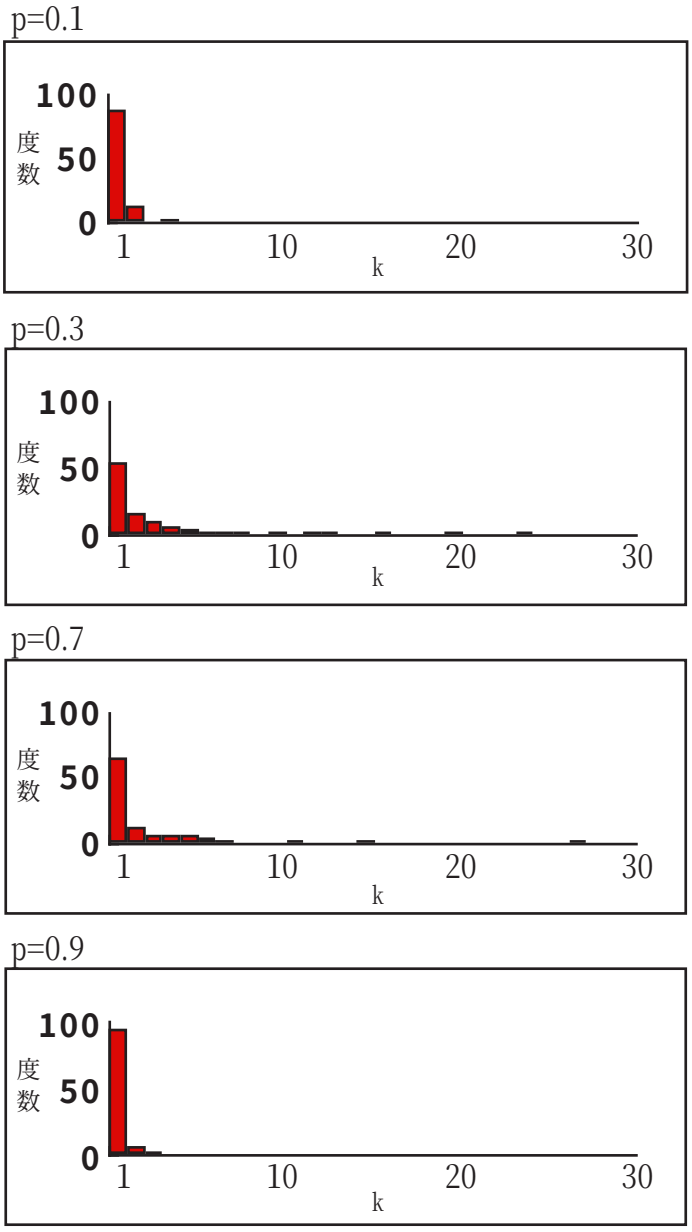


図 30: 学習終了時の k の度数分布

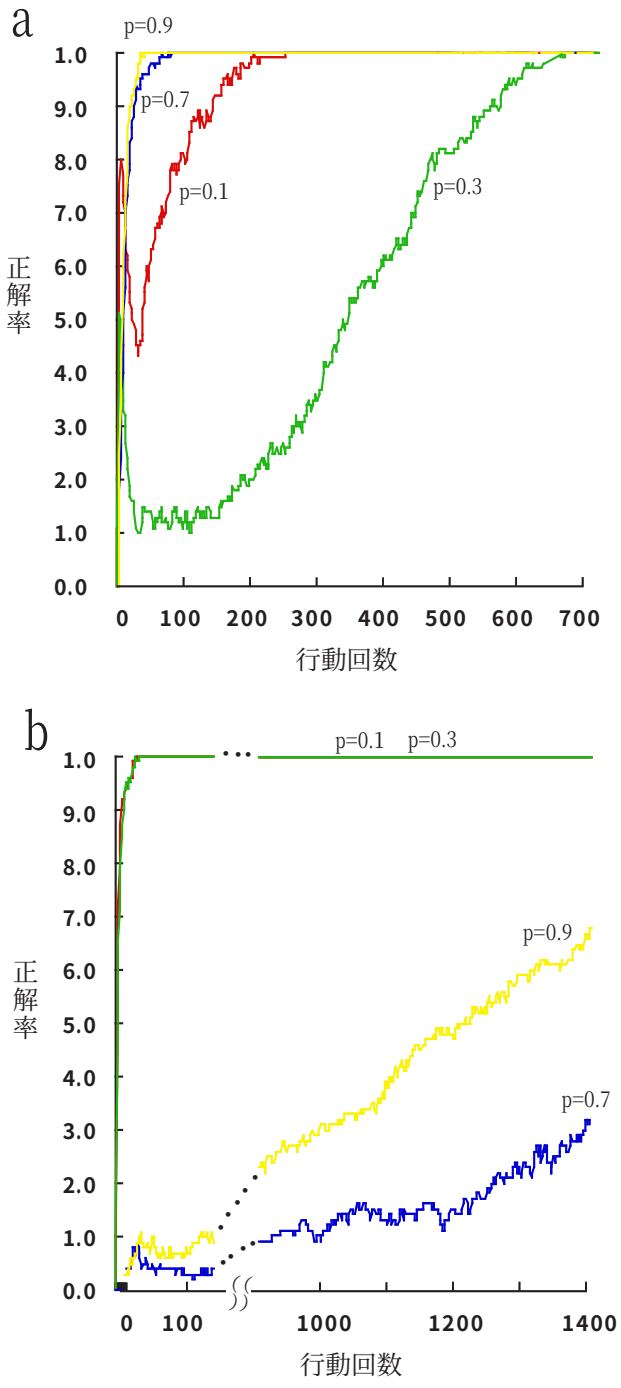


図 31: Q-learning の結果. a; 環境 a, b; 環境 b

ある．4.3.2 節同様, Q-learning での行動選択器には, Q-値に比例したルーレット選択器を採用した．Q-値の初期値は 10.0, Q-learning の学習定数は 0.5, 割引率は 0.9 である．これらのパラメータは, 予備的実験の後, 行動回数を最少とするものを選んだ．

Q-learning は Q-値を通じて最適政策を求めている．Q-値には報酬の期待値が反映されている．そのため Q-learning では報酬が最適政策を求める際の効率に大きく影響する．最適政策に含まれるルールが, 報酬により正しく強化されれば効率は向上するが, それ以外のルールが強化されてしまうと, それが最適政策の獲得を妨げ, 効率は低下する．

本問題の環境 a では, ルール 0 の Q-値はルール 1 の Q-値に比べ増加しやすい．これはルール 0 では確率 p で報酬が直接得られるので, その場合にルール 0 の Q-値が大きく上昇するためである．したがって, $p > 0.5$ の範囲では高速に最適政策を獲得できるが, $p < 0.5$ では p が 0.5 に近づけば近づくほど効率は悪化する．また, 環境 b では逆に, ルール 1 の Q-値の方が増加しやすい．これは Q-learning では報酬に近いルールから強化されていくためである．したがって, 環境 a と全く正反対の結果となる．このように Q-learning では報酬の与えられる場所が学習効率に大きく影響する．

Q-learning は報酬による強化の側面も併せ持つので, 環境を完全に同定できなくても, 最適政策を発見できる場合がある．しかし報酬による強化の影響が最適政策の獲得に対してつねに有効であるとは限らない．k-確実探査法に基づく学習システムはそのような報酬の影響を受けず, この種の環境変化に対し頑健である．

4.4 おわりに

本章では, 環境同定を重視する立場から強化学習に接近し, 効率よく環境を同定することができる行動選択器である k-確実探査法を提案した．k-確実探査法と Policy Iteration Algorithm を統合した学習システムを構築することにより, 最適政策を効率よく獲得できる手法を実現した．環境同定型の代表的手法である Q-learning との比較を通じ, 本手法の有効性を明らかにした．

環境同定型の強化学習は, profit sharing に代表される経験強化型と比べると行動回数の点では不利である．しかし本手法は, 学習に要する行動を極力減らしつつ, 経験強化型では保証されない学習結果の最適性を保証している点に特徴がある．

ところで k-確実探査法では, 行動選択時に, 環境側に存在する非決定性は考慮されない．したがって, 決定的な状態遷移下ではつねに効率的に環境を同定することができるが, 確率的な状態遷移下では必ずしも効率的とはいえない．次章では,k-確実探査法を拡張し, 確率的

な状態遷移下でもつねに効率的に環境を同定できる行動選択器を提案する.

付録

D Policy Iteration Algorithmの概要

参考のためにPolicy Iteration Algorithmの概要を示す. Policy Iteration Algorithmは, マルコフ決定過程において, 各状態の遷移確率および得られる報酬の期待値が既知の場合, 最適性の原理 [Bertsakas 76] に従って, 最適政策を反復手続きによって求める方法であり, アルゴリズムは以下のとおりである [ワグナー 78].

ステップ1

適当な政策を選ぶ.

ステップ2

与えられた政策に対して, $w_i, i = 1, 2, ..., m$ についての連立一次方程式

$$w_i = r_i^{k_i} + \gamma \sum_{j=1}^{m} q_{ij}^{k_i} w_j, i = 1, 2, ...m \tag{18}$$

を解く. ここで m は状態数, k_i は状態 i での行動, γ は割引率.

ステップ3

$$W_i = \max_{1 \le k \le n} \{r_i^k + \gamma \sum_{j=1}^{m} q_{ij}^k w_j\}, i = 1, 2, ...m \tag{19}$$

を計算する. ここで n は行動の種類.

ステップ4

すべての i について $W_i = w_i$ ならば, そのときの政策が最適な政策である. もし $W_i > w_i$ となる i が存在するときは, そのようなすべての i についてステップ3で最大値を与える行動 k を k_i としてステップ2へ戻る.

5　k-確実探査法の不確実性下への拡張：ℓ-確実探査法

5.1　はじめに

前章で述べた k-確実探査法は選択回数最少のルールを優先的に選択する手法であるが, 選択回数の多少のみで, 行動を実行した際の状態遷移確率は考慮していない. そのため状態遷移が決定的な場合には, 環境を効率よく同定できるが, 確率的な状態遷移下では必ずしも効率的とはいえない. 本章では,k-確実探査法を拡張して, 確率的な状態遷移下でもつねに効率的な環境同定が可能な行動選択器を提案する.

以下 5.2 節では, 行動の結果生じる非決定性の評価方法について考察する. 5.3 節では,5.2 節の結果に基づき, 非決定性を考慮した行動選択器である ℓ-確実探査法を提案する. 5.4 節では, 数値実験により ℓ-確実探査法の有効性を評価する.

5.2　ルール構造の同定について

5.2.1　準備

以降の議論で使われる用語を定義する. 感覚入力 i の下でルール a を実行した後, 感覚入力 j へ遷移する確率を**枝分かれ確率**と呼び q_{ij}^a と書く. ひとつのルールに対する可能な状態遷移の総数を**枝分かれ数**, 枝分かれ確率を並べたものを**ルール構造**と呼ぶ. ルール a に関して, 枝分かれ数が n ならば, ルール構造は $(q_{ij_1}^a, q_{ij_2}^a, \cdots, q_{ij_n}^a)$ と記述される. ひとつのルール構造を構成している枝分かれ確率の総和は 1 である. 図 32 を用いて, これらの用語の具体例を示す. ルール a では, 感覚入力 S_0 から S_1,S_2,S_3 への枝分かれ確率は 0.3,0.4,0.3 であるので, 枝分かれ数は 3, ルール構造は (0.3,0.4,0.3) となる. 同様に, ルール b の枝分かれ数は 2, ルール構造は (0.5,0.5) である.

本章では, マルコフ性だけが知られている未知の環境に対して, エージェントが行動しながらサンプルを体系的に収集することにより環境を過不足なく記述するルール集合および各ルールに付随するルール構造を同定する問題を対象とする.

一般に, 個々の枝分かれ確率の推定値に**誤差**が伴う. ここでは誤差 e を次式で定義する.

$$e = |(\text{真の枝分かれ確率}) - (\text{枝分かれ確率の推定値})| \tag{20}$$

ルール構造中のすべての枝分かれ確率が, 信頼度 ℓ で, 誤差 e 以内にあると推定されるとき, そのルールは**誤差 e, 信頼度 ℓ で同定された**と呼ぶ.

ひとつのルールを誤差 e, 信頼度 ℓ で同定するためには, ある必要回数だけそのルールを

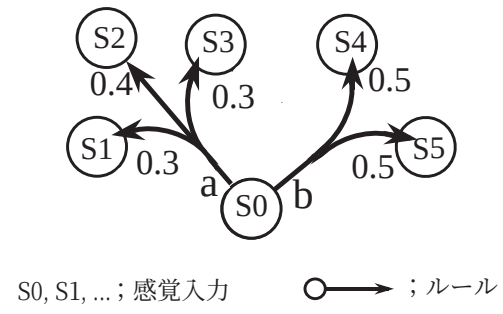


図 32: 枝分かれ確率およびルールの構造

選択しなければならない. 各ルールごとに記録された各感覚入力への遷移回数のベクトルを**観測情報**と呼ぶ. ルール a の観測情報を A_a で表す. 例えば, 感覚入力 S_0 の下でルール a が 10 回選択され,S_1, S_2, S_3 へそれぞれ 4,3,3 回遷移したとき A_a は (4,3,3) で表される. 本章では, 観測情報を用いて, 各ルール構造の候補への帰属確率を計算し, その確率からルールを誤差 e, 信頼度 ℓ で同定するために必要となる選択回数の期待値を算出することを考える.

ルール構造の候補は, 枝分かれ数および枝分かれ確率の**刻み幅**によって決まる. ここで刻み幅とは任意の枝分かれ確率間の差の絶対値の最小値である. 例えば, 枝分かれ数が 2 で, 枝分かれ確率の刻み幅が 0.1 のとき, B_1=(0.1,0.9),B_2=(0.2,0.8),$\cdots$,B_9=(0.1,0.9) の 9 つがルール構造の候補とされる. 以後, ルール構造の候補を B_i で表す.

次節では, まず, 枝分かれ確率の刻み幅が固定され, かつ枝分かれ数が既知の場合に, ルールを誤差 e, 信頼度 ℓ で同定するために必要な選択回数を見積もる. 次に, 枝分かれ数が未知の場合の考察を行う. 最後に, 枝分かれ確率の刻み幅に関する感度解析を行う.

5.2.2 枝分かれ数が既知の場合のルール構造の同定

枝分かれ数が既知の場合に, ルールを誤差 e, 信頼度 ℓ で同定するために必要な選択回数を見積もることを考える. あるルール a のルール構造を同定するためには, まず, 各ルール構造の候補 B_i に対する帰属確率 $P(B_i|A_a)$ を維持, 管理する必要がある. 枝分かれ数が既知の場合には, 枝分かれ確率の刻み幅を設定すればルール構造の候補数が定まるので, 有限個の $P(B_i|A_a)$ を管理すればよい.

ところで, 各 B_i はそれぞれ枝分かれ数に応じた多項分布を構成するので, 各 B_i に対する $P(A_a|B_i)$ は, 付録 E に示した同時確率分布 [Feller 60] から容易に計算される. また自律的

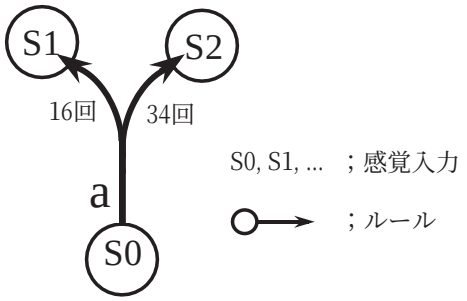


図 33: 必要選択回数の計算例に用いた環境

学習システムでは, 予め環境に関する知識は持たせないので, 事前確率 $P(B_i)$ はすべて等しいものと仮定する. よって $P(B_i|A_a)$ は, 次式から求められる.

$$P(B_i|A_a) = \frac{P(A_a|B_i)}{\sum_j P(A_a|B_j)} \tag{21}$$

ここで $\sum_j$ は, ルール構造の候補全体についての総和を表す.

帰属確率 $P(B_i|A_a)$ から同定に要する選択回数を算出する方法は, 最大の $P(B_i|A_a)$ のみを利用するなど種々考えられるが, ここではすべての $P(B_i|A_a)$ を利用して, 期待値として同定に要する選択回数を算出することとする. すなわち, ルール a を誤差 e, 信頼度 ℓ で同定するために要する選択回数の期待値 E_a は次式で与えられる.

$$E_a = \sum_i P(B_i|A_a)m(B_i) \tag{22}$$

他の戦略, 例えば, 最大の $P(B_i|A_a)$ のみを利用する場合にも同様な計算は可能だが, 本論文の内容は, この戦略の選択に影響されるものではない. 式 22 中の $m(B_i)$ は, ルール構造が B_i であるとき, そのルールを誤差 e, 信頼度 ℓ で同定するために必要とされる選択回数である. 誤差 0.05, 信頼度 50%,60%,$\cdots$,90%,95%,99%の場合の $m(B_i)$ を付録 F に, 誤差 0.01, 信頼度 50%,60%,$\cdots$,90%,95%,99%の場合の $m(B_i)$ を付録 G に示す. 表中で枝分かれ数が 1 の場合, すなわち (1.0) というルール構造に対応する $m(B_i)$ は, そのルールに枝分かれが無いと判断するために必要な選択回数であることに注意せよ.

図 33 を用いて式 22 の計算例を示す. ルール a を 50 回実行し,S_1 へ 16 回,S_2 へ 34 回遷移したときを考える. ルール構造の候補としては, 枝分かれ数を 2, 枝分かれ確率の刻み幅を 0.1 とし, B_1=(0.1,0.9),B_2=(0.2,0.8),$\cdots$,B_9=(0.9,0.1) の 9 つを想定する. 同時確率分布より

	B1	B2	B3	B4	B5	B6	B7	B8	B9
P(Aa|Bi)	1.37×10^{-5}	1.64×10^{-2}	1.15×10^{-1}	6.06×10^{-2}	4.37×10^{-3}	4.10×10^{-5}	2.73×10^{-8}	2.38×10^{-13}	9.12×10^{-23}
P(Bi|Aa)	6.99×10^{-5}	8.37×10^{-2}	5.87×10^{-1}	3.09×10^{-1}	2.23×10^{-2}	2.09×10^{-4}	1.39×10^{-7}	1.21×10^{-12}	4.65×10^{-22}

表 3: 上段;$A_a(16.34)$ に対する $P(A_a|B_i)$ の値, 下段;$A_a(16.34)$ に対する $P(B_i|A_a)$ の値

可能なすべての $P(A_a|B_i)$ は表 3 上段のように計算されるので, 式 21 を用いて,$P(B_i|A_a)$ は表 3 下段のように求まる. したがって, 誤差 0.05, 信頼度 70%でルール a を同定するために要する選択回数の期待値 E_a は

$$E_a = 6.99 \times 10^{-5} \times 35.0 + 8.37 \times 10^{-2} \times 63.6 + \cdots = 87.5 \tag{23}$$

となる. ルール a は既に 50 回選択されているので, あと 38 回選択すれば, 信頼度 70%で同定されることが見込まれる.

5.2.3 枝分かれ数が未知の場合のルール構造の同定

枝分かれ数が未知の場合には, 以下に示すインクリメンタル戦略に基づいて, ルール構造の候補を決定する方法を採用する.

まず, すべてのルールは枝分かれはないと仮定して同定作業を開始する. その後, ルールの選択回数の増加と共に新たな枝分かれが見い出されれば, そのときの枝分かれ数をもってルールの枝分かれ数とみなし, ルール構造の候補を決定する. 例えば, 枝分かれ確率の刻み幅が 0.1 で, あるルール a の現在観測されている枝分かれ数が 2 の場合には, B_1=(0.1,0.9),B_2=(0.2,0.8),$\cdots$,B_9=(0.9,0.1) の 9 つがルール a に対するルール構造の候補とされるが, 新たな枝分かれが見い出され, 枝分かれ数が 3 となれば, ルール a に関して B_1=(0.1,0.1,0.8),B_2=(0.1,0.2,0.7),$\cdots$,B_{36}=(0.8,0.1,0.1) の 36 個をルール構造の候補と考える.

5.2.4 枝分かれ確率の刻み幅について

ルール構造の候補数は枝分かれ確率の刻み幅に依存して定まるので, この値の設定も重要な課題である. ところで式 22 中の $m(B_i)$ の値は, ルール構造同士の**類似度**が高い場合には大きな差がなく, 枝分かれ確率の刻み幅を上げて,B_i の種類を増加させたとしても, 結果

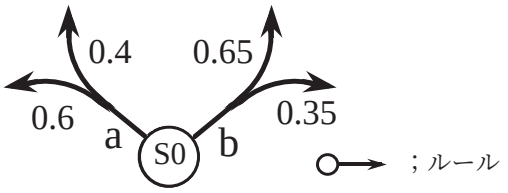


図 34: 刻み幅に関する感度解析で用いた環境

	ルールa	ルールb
刻み幅0.1	340.4	324.9
刻み幅0.05	340.5	324.7

表 4: 刻み幅を変化させたときの必要選択回数の変化. (各ルールを 50 回選択した後の値)

として式 22 の値には大きな変化は現れないものと予想される. ここで, 類似度はルール構造を構成している枝分かれ確率同士の差の総和として定義する. 以下では数値例でこの現象を確認する.

図 34 に示すような枝分かれ数 2 のルール a および b に対し, 刻み幅を 0.1 に設定した場合と 0.05 に設定した場合の相違を調べる実験を行った. 表 4 に図 34 のルールを誤差 0.05, 信頼度 95%で同定するために必要な選択回数の期待値を示す. 表の値は, それぞれのルールを各々50 回ずつ選択する実験を乱数の種を変えて 100 回行ったときの平均である. 環境が有する枝分かれ確率の刻み幅は, ルール a では 0.1 であるが, ルール b では 0.05 である. したがって, ルール b に対しては, 刻み幅を 0.1 に設定した場合と 0.05 に設定した場合とでは必要選択回数の期待値に差が生じる可能性がある. しかし表 4 より明らかなように, それらの間に大差は見られない.

したがって枝分かれ確率の刻み幅の設定は, 枝分かれ構造の同定において, クリティカルな問題ではないと帰結される.

5.3 ℓ-確実探査法の提案

5.3.1 k-確実探査法の拡張

環境を効率よく同定するためには, 同定され具合の低いルールを優先的に選択するような行動選択器が必要である. 第 4 章で述べた k-確実探査法では, ルールの選択回数の多少によって, ルールの同定され具合を評価している. しかしこの方法は素朴であり, 状態遷移確率をルールの選択回数に反映してはいない. そのため確率的な状態遷移下では, k-確実探査法は必ずしも効率的に環境を同定することは保証されない. 本論文では k-確実探査法を拡張して, 確率的な状態遷移下でも効率的に振る舞う ℓ-確実探査法を提案する.

ℓ-確実探査法ではルールを選択した直後に生じる状態遷移の確率および得られる報酬の期待値を同定するために最尤推定を行う. 状態 i でルール a を選んだときの状態 j への遷移確率 $\overline{q_{ij}^a}$ および状態 i でルール a を選んだときに得られる報酬の期待値 $\overline{r_i^a}$ は

$$\overline{q_{ij}^a} = \frac{\text{ルール } a \text{ を選び状態 } i \text{ から } j \text{ へ遷移した回数}}{\text{ルール } a \text{ の選択回数}} \tag{24}$$

$$\overline{r_i^a} = \frac{\text{ルール } a \text{ を選択した直後に得た報酬値の総和}}{\text{ルール } a \text{ の選択回数}} \tag{25}$$

となる.

ℓ-確実探査法ではルールの同定され具合の評価基準として**達成率**を用いる. 達成率とは, そのルールの現時点までの選択回数が, そのルールを誤差 e, 信頼度 ℓ で同定するために要する選択回数からどれだけ離れているかを表す量であり, 次式で定義される.

$$\text{達成率 } (\%) = \frac{\text{現時点までの選択回数}}{\text{同定に要する選択回数}} \times 100 \tag{26}$$

達成率が低いということは, 同定され具合が不十分であることを意味する. そこで,ℓ-確実探査法では, 達成率最低のルールを優先的に選択することにより環境同定を行うものとする.

ここで予め信頼度に目標値を設定できる場合には, その下で達成率が 100%になった時点で同定を打ち切ればよい. また未知環境においては予め信頼度の適切な目標値を設定することは困難なので, 最初は仮の信頼度を設定し, 逐次, 信頼度を向上させることにより, 環境同定の精度を段階的に向上させる方法が考えられる.

5.3.2 ℓ-確実探査法

ℓ-確実探査法のアルゴリズムを図 35 に示す. 図 35 に示すように本アルゴリズムは信頼度 ℓ の更新および ℓ-確実探査に基づく行動の選択からなる.

```
procedure  ℓ-確実探査法
  誤差eおよび枝分かれ確率の刻み幅を設定する.
  信頼度ℓを初期化する.

  begin ;信頼度ℓの更新
   if 今まで知覚したことのない新たな状態を知覚した then
     信頼度ℓを初期化する.
   すべてのルールの達成率を計算し最小値を知る.
   if すべてのルールの達成率が100%である then
     信頼度ℓを増加させる.

    begin ; ℓ-確実探査
     if 現状態に達成率最小のルールが存在する then
       その中のひとつをランダムに選ぶ.
     else すべての状態に対しフラグを立てる
      for 現状態以外のすべての状態 do
       if 達成率最小のルールが存在する または
          フラグの降りた状態に遷移し得るルールが存在する then
          その状態のフラグを降ろす
      while 新たにフラグを降ろした状態が存在する;
      現状態よりフラグの降りた状態に遷移し得るルールの
      ひとつをランダムに選択する.
    end.

  end;
```

図 35: ℓ-確実探査法のアルゴリズム

ここでフラグは, 現状態に達成率最低のルールが存在しない場合に, 以後達成率が最低でないルールばかりを選び続けてしまわないための処理に用いている. このフラグによる具体的な処理方法は k-確実探査法と同一である. また, 全ルールの達成率が 100%となった時点で Policy Iteration Algorithm を利用すれば, 全ルールが少なくとも誤差 e, 信頼度 ℓ で同定されているという確実さの下での政策が, 適宜, 得られる.

5.3.3 ℓ-確実探査法の特徴

ℓ-確実探査法は次のような特徴をもつ.

(1) 行動の結果生じる非決定性を考慮

k-確実探査法では, 単に選択回数の多少のみで行動選択をしていたが, ℓ-確実探査法では, 行動の結果生じる非決定性も陽に考慮している. そのため k-確実探査法に比べて, より効率のよい環境同定が行われると期待できる.

(2) 学習途中の解の確実さが明確

学習途中の解は,k-確実探査法では, すべてのルールが最低 k 回以上選択されているという確実さでの解でしかないが, ℓ-確実探査法では, 各ルールの同定され具合を表す信頼度付きの確実さの下で解が評価される.

(3) 学習の打ち切りが可能

k-確実探査法では, ルールの選択回数をどんなに増加させても, 学習を打ち切るための手がかりは得られない. しかし ℓ-確実探査法では, 達成率が 100%となった時点で学習を打ち切れば, 信頼度 ℓ という意味での確実さをもった解が得られる.

(4) 報酬の獲得場所の影響を受けない

ℓ-確実探査法は k-確実探査法同様, 行動決定時に報酬の影響を一切受けない. これは報酬による強化の影響が必ずしも環境の同定に貢献するとは限らないという考えを反映したものである.

次節では,ℓ-確実探査法を定量的に把握するために他の手法との比較評価を行う.

5.4 ℓ-確実探査法の性能評価

環境同定率の評価

ℓ-確実探査法はルール構造の違いによって生じる必要選択回数の差異を利用した手法であるので, 環境中に存在するルール構造の種類が多ければ多い程, より効率的に環境を同定で

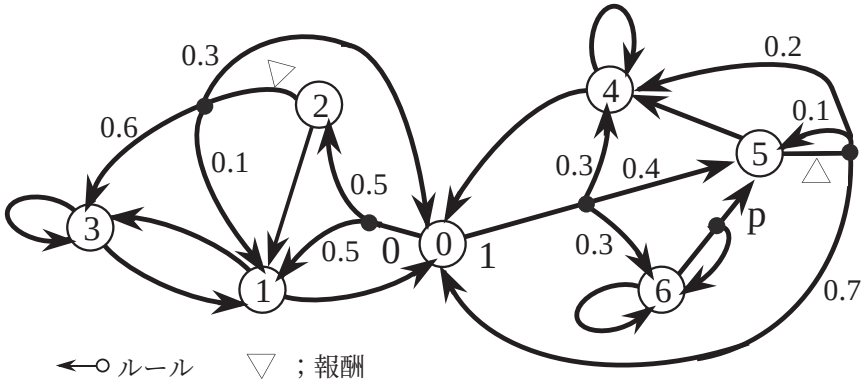


図 36: ℓ-確実探査法の特徴を確認するための実験に用いた環境

きるという特徴を持つ. ここでは, 図 36 に示す環境を用いて, ℓ-確実探査法の環境の同定され具合の早さを k-確実探査法とランダム選択との比較によって評価する.

図 36 の環境では, 多くの種類のルール構造が存在しているので, ℓ-確実探査法の優位性が特に示されることが予想される. また, 確率 p が変化し, 環境の状態遷移の構造が変わったとしてもルール構造の種類自体は変化しないので, ℓ-確実探査法の優位性は維持されるものと考える.

結果を図 37 に示す. 図 37 の横軸は行動回数, 縦軸はその行動回数の時点での環境同定率を表す. 環境同定率は, 乱数の種を変えて行った 100 回の実験中, 何回までが, その行動回数の時点で, すべてのルールのすべての枝分かれ確率が誤差 0.05 で同定されていたかを表す. 予想通り,$p = 0.2, 0.3, 0.4$ のすべてにおいて ℓ-確実探査法は他の手法よりも効率よく環境を同定していることが確認された.

最適政策獲得率の評価

次に, 最適政策の獲得され具合の早さを調べる. 図 36 の環境では, 状態 0 でルール 1 を選択すれば最適政策が得られるが, ルール 0 を選択すると準最適政策しか得られない. 準最適政策による単位行動当たりの期待獲得報酬は 0.19R であるが, 最適政策によるこの値は表 5 に示すように確率 p により変化する.

最適政策と準最適政策との差が少ない環境では, より厳密に各ルールの枝分かれ確率を同定しなければならない. その結果, より効率的に環境が同定できる手法が重要となる. すなわち図 36 の環境では,$p = 0.4$ のときよりも $p = 0.2$ のときの方がより ℓ-確実探査法の優位性が示されるものと期待される. ここでは,ℓ-確実探査法の最適政策の獲得され具合の早

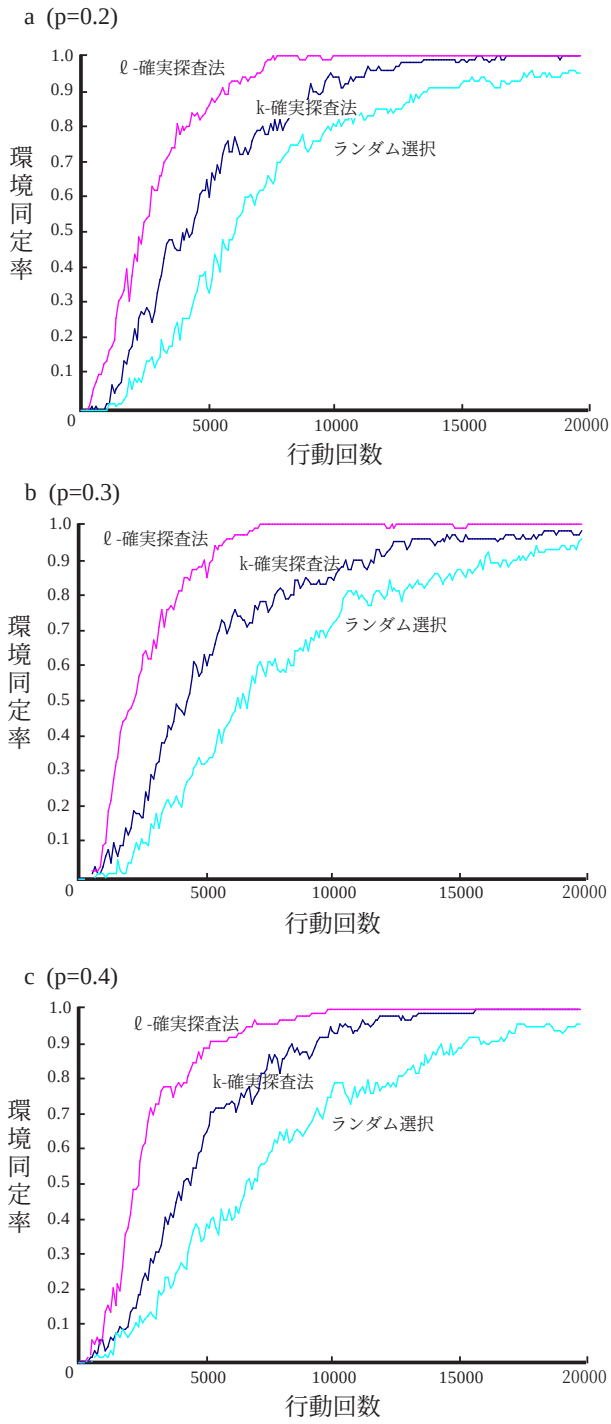


図 37: p を変化させたときの環境同定率の時間的変化, a は p=0.2, b は p=0.3, c は p=0.4 の場合.

p	0.2	0.3	0.4
単位行動当たりの期待報酬	0.20	0.24	0.26

表 5: p を変化させたときの最適政策による単位行動当たりの期待獲得報酬

さを k-確実探査法とランダム選択との比較によって評価する.

結果を図 38 に示す. 図 38 の横軸は行動回数, 縦軸はその行動回数の時点での最適政策獲得率を表す. 最適政策獲得率は, 乱数の種を変えて行った 100 回の実験中, 何回までが, その行動回数の時点で,Policy Iteration Algorithm により最適政策を求めることができたかを表す. 予想通り,ℓ-確実探査法の優位性は $p = 0.2$ に近いほど強く発揮された. また,$p = 0.4$ の場合には,3 手法の間に性能の差はほとんど見られない. このように環境の構造により, 環境同定が完全に行われなくても最適政策が求まる場合には, 各種の同定手法の間には性能の差異が生じないこともあり得ることが併せて確認された.

5.5 おわりに

本章では,k-確実探査法を拡張し, 確率的な状態遷移下でもつねに効率的な環境同定が可能な行動選択器である ℓ-確実探査法を提案した.

本手法は,k-確実探査法の不確実性下への自然な拡張であり, k-確実探査法では不十分であった確率論に基づく理論的裏付けを有するという特徴を持つ. さらに, 従来, あまり意識されていなかった学習途中の解の意味付けや, 学習の打ち切りを可能としている点など優れた部分が多い.

k-確実探査法,ℓ-確実探査法共に環境同定を中心に考えた手法なので, 学習途中での報酬獲得は完全に無視されている. ところが強化学習の工学的な応用を考えた場合などでは, 特に, 学習の初期段階からそこそこの報酬を獲得しつつ, 同時に環境の同定を行い, 報酬獲得の水準を段階的に向上させるような挙動が求められる. そこで次章では, 経験強化型と環境同定型の相補的性質に着目して, 両者を巧みに統合した強化学習システムを提案する.

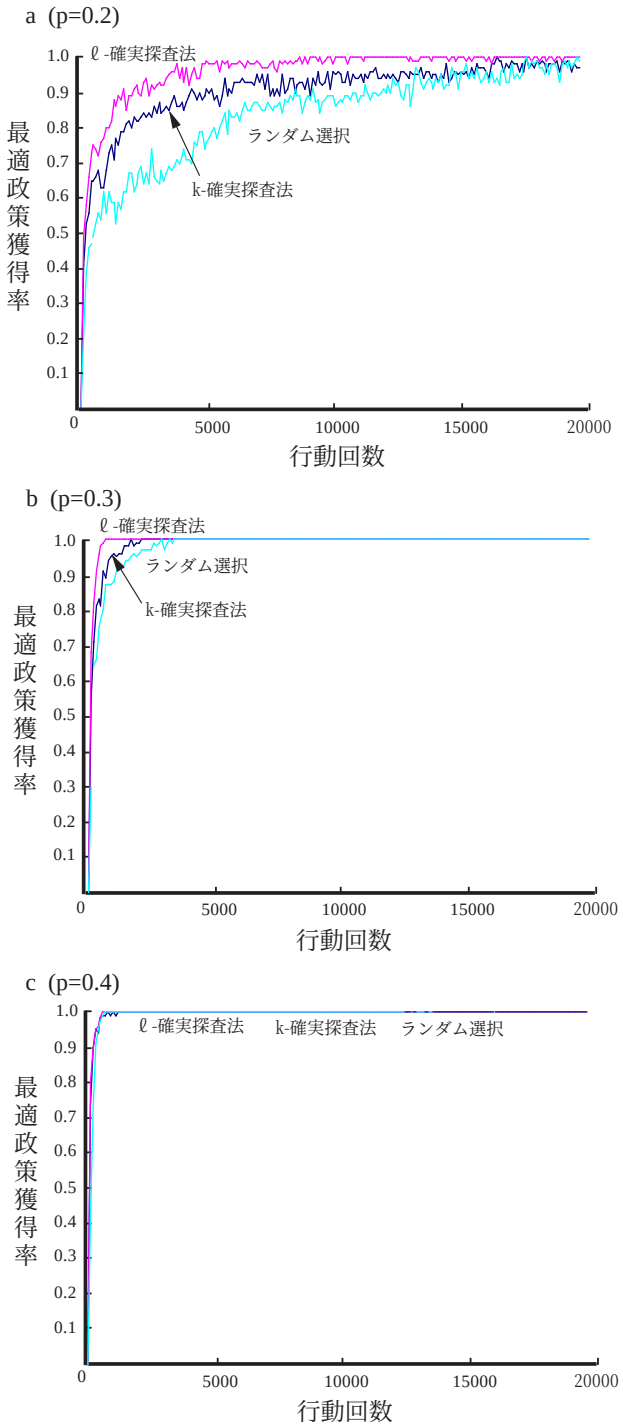


図 38: p を変化させたときの最適政策獲得率の時間的変化, a は p=0.2, b は p=0.3, c は p=0.4 の場合.

付録

E　同時確率分布

参考のために同時確率分布について記す.

おのおのの試みが数個の結果の中のひとつを取り得るような独立に繰り返される n 回の試みを考える. おのおのの試みで可能な結果を $E_1, \cdots, E_r$ で表わし, おのおのの試みで E_i が実現する確率を p_i $(i = 1, \cdots, r)$ とする. 一般に p_i は条件

$$p_i + \cdots + p_r = 1, \tag{27}$$

に従う. n 回の試みで E_1 が k_1 回,E_2 が k_2 回,$\cdots$,E_r が k_r 回実現する確率は

$$\frac{n!}{k_1!k_2!\cdots k_r!} p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r} \tag{28}$$

である. ここで k_i は任意の負でない整数であり,

$$k_1 + k_2 + \cdots + k_r = n \tag{29}$$

を満たす.

F　誤差 0.05 の場合の $m(B_i)$

	50%	60%	70%	80%	90%	95%	99%
(1.0)	14.0	18.3	24.0	31.9	45.3	58.7	89.9
(0.1,0.9)	15.0	15.0	35.0	55.0	94.0	132.6	228.8
(0.2,0.8)	14.0	34.0	63.6	94.6	165.9	237.2	397.0
(0.3,0.7)	25.2	51.8	84.0	125.2	202.2	310.6	516.8
(0.4,0.6)	35.0	61.1	98.9	152.8	255.0	357.1	566.1
(0.5,0.5)	11.7	57.0	100.8	150.7	260.5	362.4	609.1
(0.1,0.1,0.8)	54.1	71.7	97.2	130.4	192.6	258.0	408.4
(0.1,0.2,0.7)	72.1	96.0	131.9	175.4	261.7	345.4	536.6
(0.1,0.3,0.6)	82.5	108.4	152.6	206.7	307.3	405.0	621.4
(0.1,0.4,0.5)	89.0	117.1	162.7	217.0	326.4	431.2	671.6
(0.2,0.2,0.6)	86.0	117.8	164.0	211.6	310.5	400.7	592.7
(0.2,0.3,0.5)	101.2	130.8	182.4	238.1	339.9	438.3	667.3
(0.2,0.4,0.4)	100.9	136.6	182.1	242.1	347.1	446.2	645.1
(0.3,0.3,0.4)	106.6	144.2	197.7	255.3	363.4	464.2	685.5
(0.1,0.1,0.1,0.7)	75.3	95.9	128.7	165.4	242.8	325.4	517.8
(0.1,0.1,0.2,0.6)	94.8	116.5	156.2	203.9	295.3	383.6	579.1
(0.1,0.1,0.3,0.5)	96.0	132.3	173.6	224.7	323.4	423.1	660.0
(0.1,0.1,0.4,0.4)	98.1	134.3	175.6	229.4	329.8	429.1	636.0
(0.1,0.2,0.2,0.5)	108.9	134.5	177.1	230.8	322.3	411.1	629.4
(0.1,0.2,0.3,0.4)	118.6	151.0	197.4	249.7	347.7	442.1	643.9
(0.1,0.3,0.3,0.3)	116.0	150.0	196.8	249.4	351.4	444.4	671.3
(0.2,0.2,0.2,0.4)	123.2	151.4	197.6	247.2	340.4	425.6	604.9
(0.2,0.2,0.3,0.3)	126.1	160.5	204.8	257.1	348.6	436.5	631.4

G　誤差 0.01 の場合の $m(B_i)$

	50%	60%	70%	80%	90%	95%	99%
(1.0)	68	89	119	164	235	292	417
(0.1,0.9)	269	500	821	1368	2225	2872	4083
(0.2,0.8)	588	1022	1622	2467	4098	5573	8117
(0.3,0.7)	798	1269	1933	3112	5377	7353	10078
(0.4,0.6)	923	1703	2488	3757	6090	7574	10200
(0.5,0.5)	434	1728	2752	4049	6271	7390	8766
(0.1,0.1,0.8)	1238	1707	2269	3146	4594	5957	8225
(0.1,0.2,0.7)	1681	2212	2938	4026	6193	8373	10538
(0.1,0.3,0.6)	2038	2675	3380	4671	7107	8618	11018
(0.1,0.4,0.5)	2366	3006	3703	5079	7270	8354	10236
(0.2,0.2,0.6)	2335	3028	3854	5317	7587	8995	11558
(0.2,0.3,0.5)	2610	3283	4239	5844	7949	8869	10952
(0.2,0.4,0.4)	2743	3348	4224	5989	8306	9580	11551
(0.3,0.3,0.4)	2628	3265	4248	5863	8615	9779	11674
(0.1,0.1,0.1,0.7)	1879	2386	2996	3950	5839	7673	10086
(0.1,0.1,0.2,0.6)	2393	2972	3650	4754	7066	8357	11055
(0.1,0.1,0.3,0.5)	2669	3223	3927	5166	7439	8506	10497
(0.1,0.1,0.4,0.4)	2769	3345	4079	5435	7834	9265	11275
(0.1,0.2,0.2,0.5)	2803	3317	4055	5472	7383	8360	9856
(0.1,0.2,0.3,0.4)	2867	3450	4295	5718	8034	9279	11486
(0.1,0.3,0.3,0.3)	2836	3467	4336	6044	8500	9761	11674
(0.2,0.2,0.2,0.4)	3127	2686	4656	6484	8098	9343	11777
(0.2,0.2,0.3,0.3)	3175	3810	4754	6634	8566	9855	11968

6 報酬獲得と環境同定のトレードオフを考慮した学習システム：MarcoPolo

6.1 はじめに

前章までは, 報酬獲得, 環境同定それぞれを個別に追求する手法を検討してきた. ところで, 強化学習の応用, 例えば自律走行ロボットへの適用を考えた場合, 最適政策を得るために無限の試行を繰り返すことは不可能であり, 学習の初期段階からそこそこの報酬を獲得しつつ, 同時に環境の同定を行い, 報酬獲得の水準を段階的に向上させるような挙動が求められる. そこで本章では, 報酬獲得と環境同定のトレードオフを陽に考慮して, 学習の初期段階から終了に至るまで一貫した挙動を示す強化学習システムの提案を行う.

以下,6.2 節では, 報酬獲得と環境同定のトレードオフを考慮する必要性を論じ, 6.3 節で, その実現例として強化学習システム MarcoPolo を提案する. そして 6.4 節では,MarcoPolo の基本性能を確認した後, 迷路走行タスクへの応用により有用性を評価する.

6.2 報酬獲得と環境同定のトレードオフ

既に述べているように, 報酬獲得と環境同定の間にはトレードオフの関係が存在する. 図 39 は, 代表的な強化学習手法の典型的な挙動について, 学習の進行に伴い獲得報酬と環境同定がどのように推移するかをイメージ的に示したものである. 図 39 の横軸は環境同定の割合を表し, 縦軸は単位行動当たりの獲得報酬の期待値を表す.

Q-learning は, 学習終了時には最適政策の獲得が保証されるが, 最適政策を獲得するまでに膨大な行動回数を要することに加えて, 学習途中での報酬獲得が考慮されていないため, 図 39 に示すように報酬獲得の立ち上がりが遅れる傾向にある.

profit sharing (PS) は, 学習終了時に最適政策が得られる保証はないが, 図 39 に示すように報酬獲得については立ち上がりが早く, Q-learning とは対照的な挙動を示す. したがって, 学習の初期段階において報酬獲得が要請される場合, PS は極めて有望な手法と考えられる.

離散マルコフ決定過程の領域では, 枝分かれ確率および報酬の期待値が既知であれば, 動的計画法の一手法である Policy Iteration Algorithm (PIA) を適用することにより最適政策を速やかに決定することができる. しかし, 強化学習では学習以前には環境は未知の存在なので, 環境既知が前提とされる PIA を学習の初期段階で利用することはできない.

一方,k-確実探査法や ℓ-確実探査法は, 環境の同定を目的に設計されており, 報酬獲得については何も考慮されていない. 実際, これらが作動しているときの期待獲得報酬は, ランダ

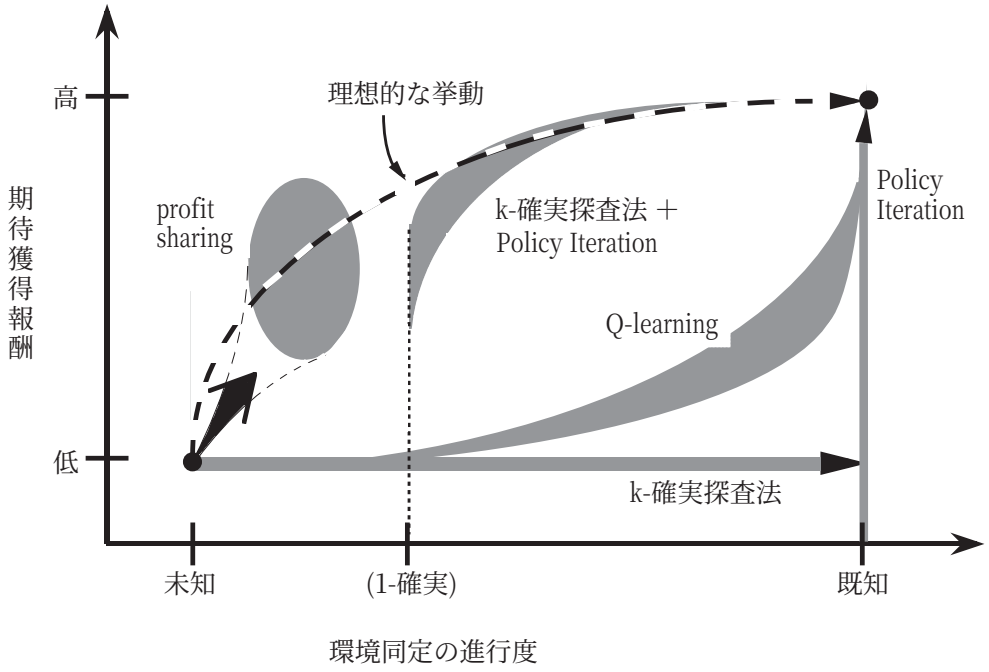


図 39: 強化学習システムの挙動

ムウオークによる期待獲得報酬と同程度であり, 単独で用いた場合には, 図 39 に示すように, 低いレベルに留まる. しかし,k-確実探査法などによってすべてのルールが 1-確実以上になれば, 4 章で示したように PIA を適用し政策を求めることができる. しかし,4 章では, 環境同定のフェーズと報酬獲得のフェーズは分離されており, いつ k-確実探査法を打ち切り,PIA に切り替えるをべきかについては明確には考察されていない.

一般に, 強化学習システムに対しては, 学習の初期段階であってもそこそこの報酬獲得が期待され, 環境同定が進むにつれて単位行動当りの期待獲得報酬が確実に増大していくことが求められる. 図 39 の破線は強化学習システムの理想的な挙動を示している.

次節以下では, 図 39 に示される理想的な挙動を実現するために, 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の設計を行い, 実験により性能を確認する.

6.3 強化学習システムの設計

6.3.1 基本的枠組

本節では, 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の設計を行う. 報酬獲得と環境同定のトレードオフを考慮するために, 図 40 に示すような**監視制御部**

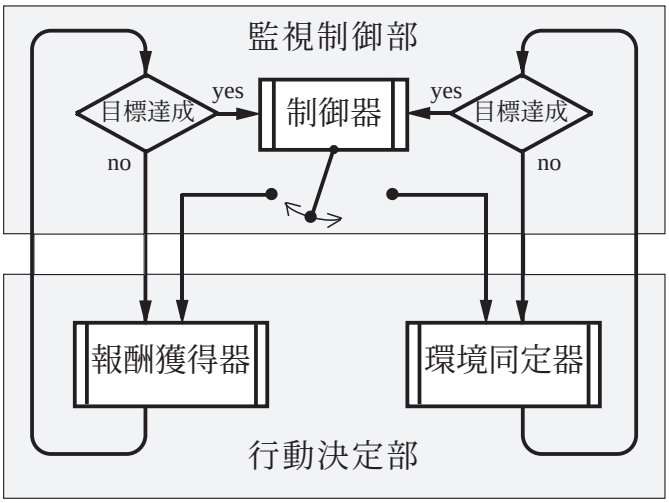


図 40: MarcoPolo の基本的枠組み

と**行動決定部**からなる枠組みを採用する．行動決定部は環境に働きかける行動を決定する部分であり, 報酬獲得を目的とした行動を決定する報酬獲得器と環境同定を目的とした行動を決定する環境同定器からなる．監視制御部は, 行動決定部の挙動を監視して, 制御モードの切り替えを決定する部分である．

6.3.2 行動決定部の設計

報酬獲得器の設計

報酬獲得器の手法として, 最適性と効率性について相補的関係にある Policy Iteration Algorithm (PIA) と profit sharing (PS) を採用する．その理由は, 環境同定の進み具合に応じて, PIA と PS を切り替えて使うことにより, 学習途中において継続的な報酬獲得が期待できるからである．

現時点で同定されているすべてのルールが 1-確実以上のとき, PIA を利用することができる．PIA は現時点で同定された環境下での最適政策を得ることができるので, 利用可能であれば PIA を使うべきである．よって, 本学習システムではすべてのルールが 1-確実以上のとき, PIA を使うものとする．

PIA は, 現時点で同定された環境中に 1-確実に満たないルールが存在する限り, 利用することはできない．特に, 学習の初期においては環境同定が不十分なことから, 多くの場合,PIA は利用不可能である．PS は, 環境同定の進み具合にかかわらず, いつでも利用可能である．すなわち,PS は PIA に比べて即応性と効率性において優れている．しかし,PS に依存して報酬獲得を続けることは最適性の観点からみれば好ましいことではない．本学習システムでは,PIA が利用できない学習の初期段階および新しいルールが見いだされて PIA が利用できなくなったときに限って, PS を使うものとする．このように PIA と PS を相補的に使うことにより, 学習の初期段階から終了まで報酬を継続的に獲得することが保証され, 加えて獲得報酬を段階的に増大させていくことが期待できる．

環境同定器の設計

環境同定器の目的は未知の環境を効率よく同定することであり, 行動決定のための探査戦略 (explroration strategy) を必要とする．ランダム探査や最少選択優先法などは素朴な探査戦略であり, 小規模かつ単純な環境では十分通用するが, 環境が大規模または複雑になるにつれて合理的な探査戦略が要請される．MDPs 環境下での合理的な探査戦略として, 第 4 章では k-確実探査法, 第 5 章では ℓ-確実探査法を提案した．本学習システムでは環境同定器

として k-確実探査法または ℓ-確実探査法を採用する.

k-確実探査法や ℓ-確実探査法によって, すべてのルールが 1-確実以上になれば, 前述したように,PIA を適用することにより, その時点での最適政策を求めることができる. k-確実探査法や ℓ-確実探査法は, 優れた環境同定手法であるが, 報酬獲得についてはランダムウオークと同程度しか期待できないことには注意しなければならない.

6.3.3 監視制御部の設計

基本的方針

報酬獲得と環境同定の間にはトレードオフの関係が存在する. 学習途中での報酬獲得を重視するのか, 環境同定を重視するのか, どちらを重視するかにより, 強化学習システムに期待される挙動はまったく異なったものになる. 報酬獲得と環境同定のトレードオフ比は学習システムのユーザが陽に設定できることが望ましい. 監視制御部の目的は, 第 1 に行動決定部の報酬獲得器と環境同定器の行動を監視して, それぞれの目標を達成したかどうかを判定すること, 第 2 に報酬獲得と環境同定のトレードオフ比を考慮して, 報酬獲得器と環境同定器の切り替えを適切に制御することにある.

以下では, まず行動決定部の監視, すなわち報酬獲得器と環境同定器の目標達成の判定について述べる. つぎに, 報酬獲得と環境同定のトレードオフ比を実現するための制御器による切り替えについて述べる.

行動決定部の監視

まず行動決定部の実行主体が報酬獲得器にある場合を考える. 報酬獲得器の目標は報酬を獲得することにある. したがって, 報酬獲得器が報酬を獲得した時点で当座の目標が達成されたと判断して, 報酬獲得器の実行をいったん打ち切り, 制御部においてつぎの実行主体を決定するものとする. 報酬獲得器が目標を達成できないと判断されるときも, 報酬獲得器の実行を打ち切る必要がある. 学習が不十分なとき, 報酬獲得器によって行動を選択できない場合が生じる. 同様のことは, ルールの枝分かれにより, 未知の新しい感覚入力を知覚したときにも生じる. このような場合, 報酬獲得器の実行を打ち切り, 実行主体を環境同定器に切り替える必要がある.

つぎに行動決定部の実行主体が環境同定器にある場合を考える. 環境同定器の目標は環境を同定することにある. 本学習システムでは環境同定器に k-確実探査法を採用している. これまで k-確実でなかったルールのひとつが k-確実になったとき, 目標の一部が達成され

たと判断される. このとき,k-確実でない他のルールを続けて選択することにより, k-確実とすることができるならば, そうすべきであろう. したがって, 環境同定器が k-確実でなかったルールを少なくともひとつ k-確実にして, かつ現在知覚している感覚入力において k-確実でないルールが選べなくなった時点で当座の目標が達成されたと判断して, 環境同定器の実行をいったん打ち切り, 制御部においてつぎの実行主体を決定するものとする.

行動決定部の制御

監視制御部の第 2 の目的は, 前述したように, 報酬獲得と環境同定のトレードオフ比を考慮して, 行動決定部の実行主体を動的に制御することにある. 報酬獲得と環境同定のトレードオフを考慮するために, **報酬獲得コスト**と**環境同定コスト**を導入する. 報酬獲得コスト (E_R) と環境同定コスト (E_I) とは, 実行主体が報酬獲得器または環境同定器であるときの開始から打ち切りまでの期待行動回数をいう.

本学習システムでは, 報酬獲得コストと環境同定コストの比率をあらかじめ設定されたトレードオフ比に近づくように, 行動決定部の実行主体を動的に制御することにより, 報酬獲得と環境同定のトレードオフを実現する.

具体的な制御方法は, ユーザが報酬獲得と環境同定のトレードオフをどのように指示するかにより異なる. 例として, 全行動のうち $100 * T\%$（$0 \leqq \mathrm{T} \leqq 1$）は報酬獲得のために行動して欲しいという要求がユーザによって与えられた場合を考える. この場合,x を報酬獲得器が選ばれる確率とすれば以下の式が成り立つ.

$$\frac{E_R * x}{E_R * x + E_I * (1.0 - x)} = T \tag{30}$$

したがって, 確率

$$x = \frac{T * E_I}{(1 - T) * E_R + T * E_I} \tag{31}$$

で報酬獲得器を選択すれば, ユーザの要求は満たされる.

このように, 制御器は, 与えられた要求にしたがって, その時点での環境同定コストおよび報酬獲得コストを参照して, 報酬獲得器と環境同定器の間の切り替えを動的に行うことにより, 報酬獲得と環境同定のトレードオフを考慮した学習システムの挙動を実現することができる.

6.3.4 強化学習システム：MarcoPolo の特徴

以上の考え方に基づき設計された強化学習システムを **MarcoPolo** (Reinforcement Learning System under **Mar**kovian Environment **co**nsidering tradeoff between **Po**licy Iteration, Profit Sharing and k-Certainty Exp**lo**ration) と命名する. MarcoPolo は以下のような特徴をもつ.

(1) 報酬獲得の持続性

報酬獲得器として PS と PIA を相補的に利用しているので, 学習の初期段階から終了に至るまで継続して報酬を獲得することが期待できる.

(2) 環境同定の信頼性

環境同定器として k-確実探査法や ℓ-確実探査法を利用しているので, 環境同定において不必要な探査が排除され, 信頼性が向上する方向に探査が選択的に進められる.

(3) 任意のトレードオフ比の実現

報酬獲得と環境同定のトレードオフを考慮して, 報酬獲得器と環境同定器の実行が動的に制御できるので, ユーザが指定する任意のトレードオフ比を実現することができる.

(4) MDPs での理想的挙動を実現する枠組み

MarcoPolo は,MDPs を対象に, 報酬獲得と環境同定の相補的性質に着目して, 両者を個別に追求した手法を巧みに統合した学習システムになっており, 強化学習に求められる理想的な挙動を実現し得る枠組みを提供している. 上記 (1)〜(3) の特徴と併せて, MarcoPolo は MDPs における強化学習システムとして完成度の高いものと考えられる.

6.4 MarcoPolo の性能評価

MarcoPolo の性能を評価するために,3 通りの実験を行った. 第 1 の実験では,MarcoPolo の基本性能を明らかにするために, 人為的な問題を設定し,MarcoPolo の特徴を確認するとともに, 他の手法との比較を行うことを目的としている. 第 2 の実験では, 強化学習の世界でベンチマーク的に使われている迷路走行タスク問題を取り上げ, 他の手法との比較評価を行うとともに, 非決定性が存在する場合に対しての適応を調べることを目的としている. 第 3 の実験では, 報酬獲得と環境同定のトレードオフの解決が, より切実な意味を持つ問題を設定し, そのような問題における MarcoPolo の有効性を調べることを目的としている.

なお本節では, 環境同定器にはすべて k-確実探査法を採用したが, ℓ-確実探査法を用いたとしても結果に大きな差はみられないことを付記する.

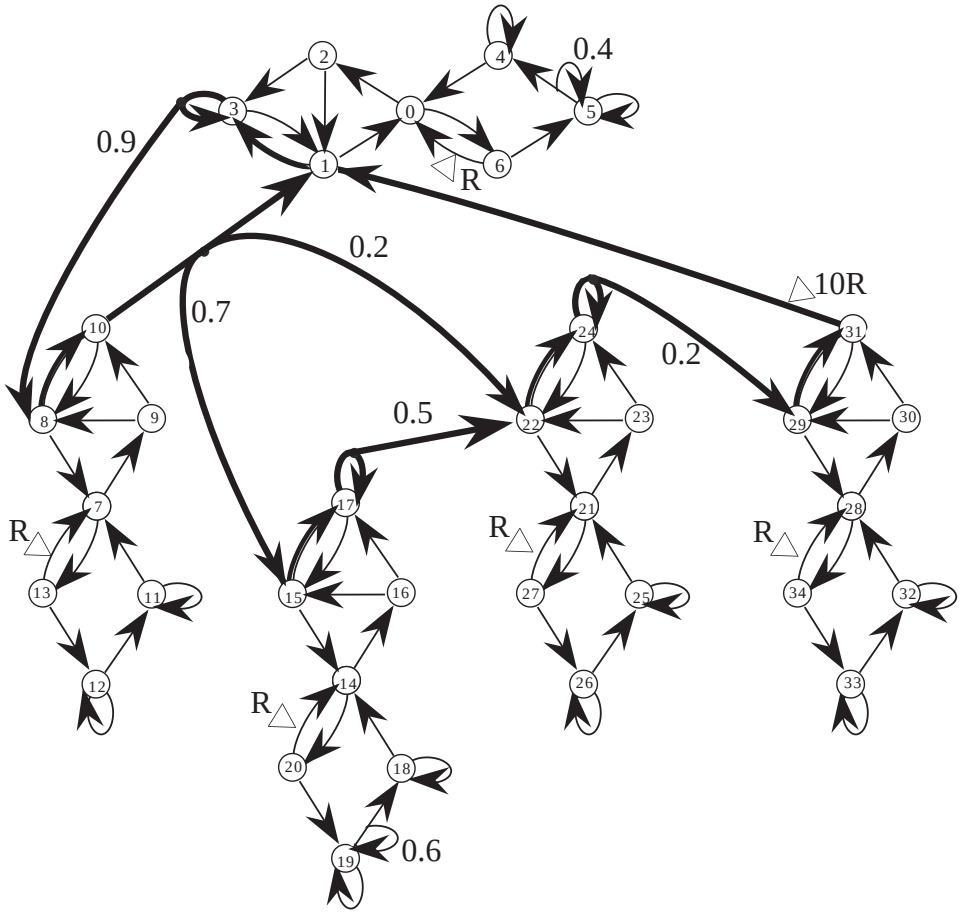


図 41: 基本性能評価に用いた環境

6.4.1 基本性能評価

実験問題

MarcoPolo の基本性能を確認するために, 図 41 に示す環境を設定した. この環境は 5 つのブロックから構成されており, 各ブロックにはそれぞれ 7 つの状態が存在する. 最適政策は太線で示されるアークからなるネットワークであり, すべてのブロックを横断していることに注意されたい. この環境は主に以下のふたつのことを評価するために設計した. 第 1 は MarcoPolo の報酬獲得器における PS と PIA の相補性を確認することを目的として,MarcoPolo から PS を除外した (Marco-PS) との比較実験を行い, 性能の評価を行う. 第 2 は他の強化学習手法（PS および Q-learning）との性能比較を行う.

第 1 の評価に当たっては, 学習の進行とともに状態が次第に増加していく環境が望ましい. なぜなら状態が次第に増える環境下では, すべてのルールを 1-確実にするのは容易では

ないことから, PIA のみに依存して報酬を獲得する (Marco-PS) では, 学習途中での報酬獲得が困難と予想されるためである. この環境では, あるブロックから他のブロックへ遷移するためにはある特定のルールを選択しなければならず, さらにそのルールを選択したとしても望むブロックへ遷移できるとも限らない.

第 2 の評価に当たっては, 準最適政策が非常に多い環境が望ましい. なぜなら準最適政策が多ければ多いほど最適政策が獲得されにくくなるためである. PS では最適政策の獲得が保証されていないので, 偶然発見した準最適政策から抜け出せなくなる恐れがある. 一方,Q-learning では最適政策の獲得が保証されているので, そのような心配はないが, 報酬に近いルールから強化されていくという Q-learning の特徴故に, この環境のように, 報酬から遠いルールが最適政策に含まれる場合には, それが最適であると判断するのに非常に多くの行動回数を要することが予想される.

本問題に対して, 全行動のうち,90%を報酬獲得のために, 残り 10%を環境同定のために, それぞれ割り当てたいという要求が与えられたとする. MarcoPolo および Marco-PS では, 式 31 のパラメータ T を 0.9 に設定すればよい. 一方,PS や Q-learning では, 全行動のうち,90%を最大の重みを持つルールの選択に, 残り 10%をランダムなルールの選択に, それぞれ割り当てることにより, 同様の効果を期待することができる.

MarcoPolo の動作確認

図 42 に実験結果を示す. 図 42-a に環境同定率の時間的変化を示す. ここで, 環境同定率とは, 乱数の種を変えて行った 100 回の試行中, 各時点でルールの枝分かれ確率を $\pm$ 0.05 の誤差で同定できた割合をいう. 環境同定率については k-確実探査法が優れていることは自明である. MarcoPolo よりも Marco-PS の方が若干優れているのは, Marco-PS では PIA が利用できないとき, 環境同定器に制御が移され, 環境の同定がより進行するためである.

図 42-b に単位行動当たりの期待獲得報酬の時間的変化を示す. この図は乱数の種を変えて行った 100 回の試行を平均したものである. 図 42-b より学習の初期では MarcoPolo のが Marco-PS よりも多くの報酬を獲得することに成功していることがわかる. これは,MarcoPolo では, 環境の同定が不十分であっても, PS が有効に機能して, 報酬が確実に獲得されているためである. MarcoPolo と Marco-PS は, 共に最適政策（破線で示される）の期待獲得報酬に対して, 約 93 %の水準に収束している. これは,PIA により求められる最適政策の報酬獲得と k-確実探査法よる報酬獲得を 9:1 に内分する水準にほぼ相当する. このことは, 全行動のうち,90%を報酬獲得のために, 残り 10%を環境同定のために, それぞれ割り当てた

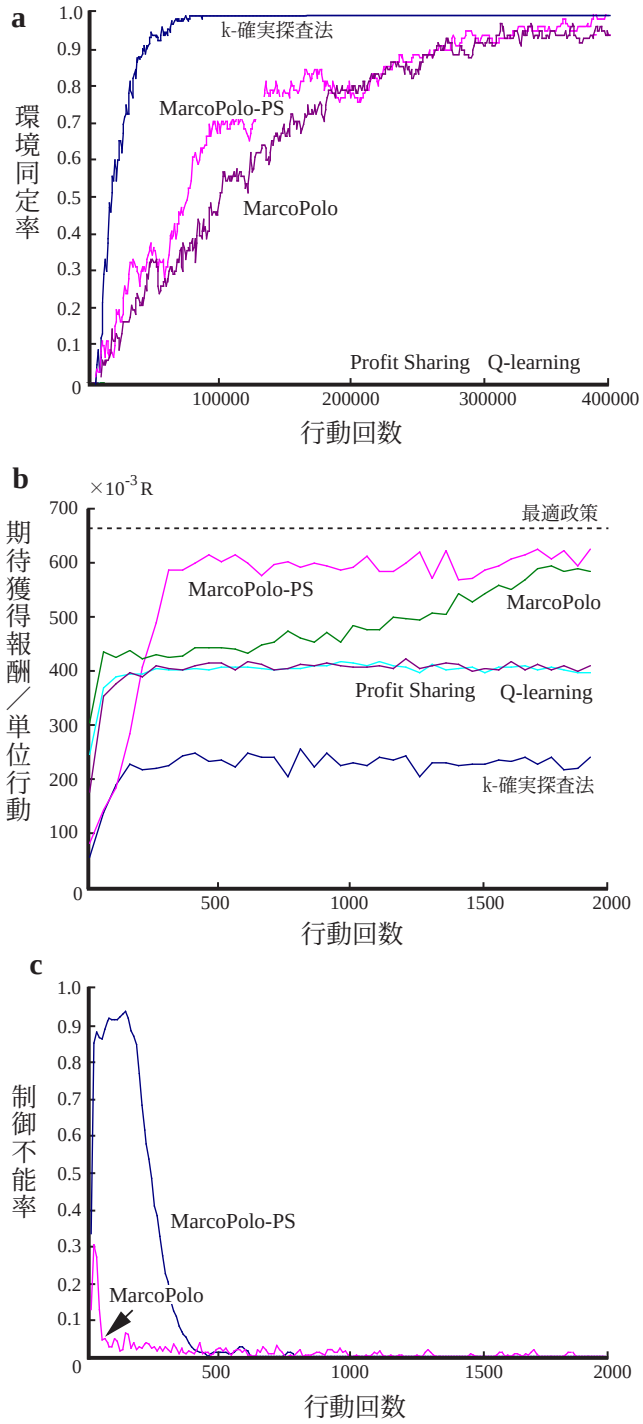


図 42: 基本性能評価実験の結果. a:環境同定率の時間的変化, b:単位行動当たりの期待獲得報酬の時間的変化, c:制御不能率の時間的変化

いという当初の要求通りに, 学習システムが作動したことを意味する.

Marco-PS と MarcoPolo の違いは, 図 42-c に示した制御不能率のグラフからもわかる. ここで, 制御不能率とは, 乱数の種を変えて行った 100 回の試行中, 報酬獲得器に制御が移されたにもかかわらず, 行動を選択することができずに, 環境同定器に切り替わった割合をいう. 図 42-c より Marco-PS は MarcoPolo に比べて制御不能率が非常に高いことがわかる. 制御不能率が高いことは, 報酬を得たくとも, 報酬を得るための行動がみつからないことを意味する. すなわち,Marco-PS は, 学習の初期段階において, 本来の設計指針に背いた挙動を示すことを意味する.

他手法との比較

PS と Q-learning は, 環境同定についてはランダム探査に依存しているので, 予想された通り, 図 42-a に示すように環境同定率の立ち上がりはほとんど期待できない.

図 42-b に示す単位行動当たりの期待獲得報酬については, 学習初期における PS の素早い立ち上がりが特徴的である. しかし PS では一般に最適政策の獲得は保証されないので, 収束時には低い水準に留まる. PS では第一ブロック内から抜け出せない例が多々観察された.

Q-learning は最適性の原理に基づいているので, 最終的には最適政策が得られるはずであるが, 第一ブロック内で与えられる報酬が最適政策の獲得を妨害するため, 100 万回の行動を繰り返しても, 最適政策を見いだすことはできなかった. 一方,MarcoPolo では, このような失敗はなく, 速やかに最適政策に収束することが確認されている.

6.4.2 迷路走行タスクへの応用

実験問題

強化学習でよく使われる問題として [Sutton 90] の迷路走行タスクを考える. これは図 43 に示す 6×9 の迷路環境におかれたエージェントが, 始点 (S) から終点 (G) までの最短パスを学習する問題である. エージェントは各マス目をそれぞれ別々の状態として認識し, 各状態では上下左右の中からひとつの移動を選択できる. 終点に到達すると報酬が得られ, 始点に戻される. 黒い壁には進入できない.

図 43 の迷路走行タスクは強化学習が対象とする環境の特徴のひとつである非決定性を含んでいない. そこで本研究では, 図 44 のように非決定性を含んだ迷路走行タスクも考える. さらにこのタスクには壁を多く付加することにより,k-確実探査法の特徴である k-確実

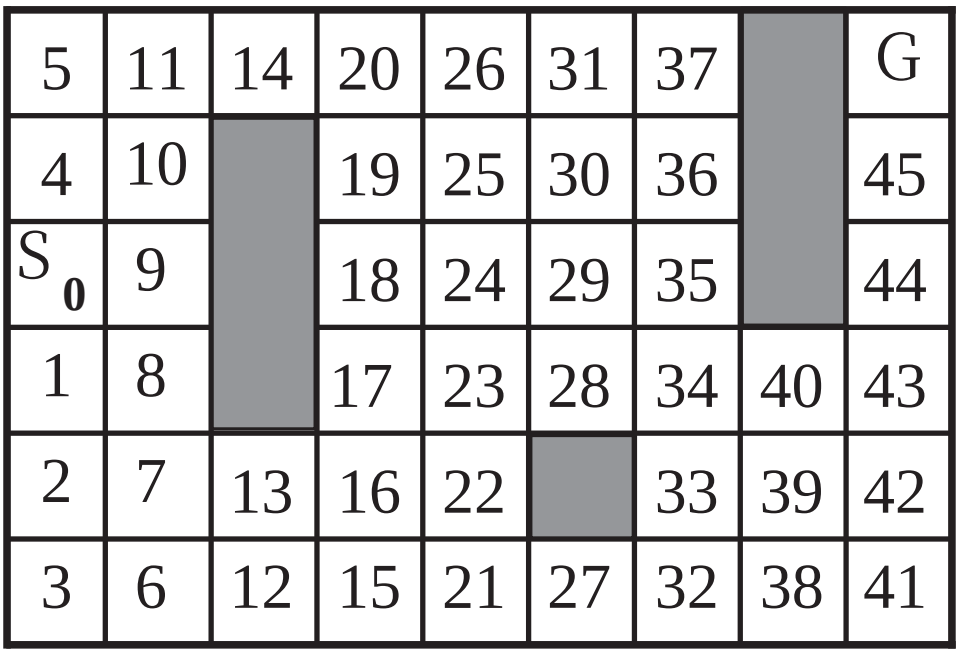


図 43: 迷路走行タスクで用いた環境

なルールのみから構成されるルール群の排除が有効に働くような問題にもなっていることを付記する.

図 43 の最短パスは 14 ステップで, 最適政策は 6 種類ある. 一方, 図 44 では 15 ステップが最短で 1 種類, 準最適政策は 19,20 ステップの 2 種類がある.

実験結果

図 45 に図 43 の環境において, 全行動のうち,90%を報酬獲得のために, 残り 10%を環境同定のために, それぞれ割り当てたいという要求が与えられた場合の実験結果を示す. MarcoPolo の比較対象として,Q-learning,PS,k-確実探査法を取り上げた. 図 43 の環境には非決定性が存在しないので, 全ルールを 1-確実にすれば, 環境は完全に同定されたことになる.

MarcoPolo は環境同定率および期待獲得報酬のどちらも期待された通りの挙動を示している. Q-learning は MarcoPolo に比べて収束するまでに 10 倍の行動回数を要した. PS については, 準最適政策が多く存在するため, 準最適政策から脱出できない場合がしばしば観察され, 最適政策の学習がなされるのは希であった.

同様に, 図 46 には図 44 の環境に対する実験結果を示す. 非決定性が存在する図 44 の環境でも,MarcoPolo は準最適政策に陥ることはなく, つねに最適政策が速やかに得られて

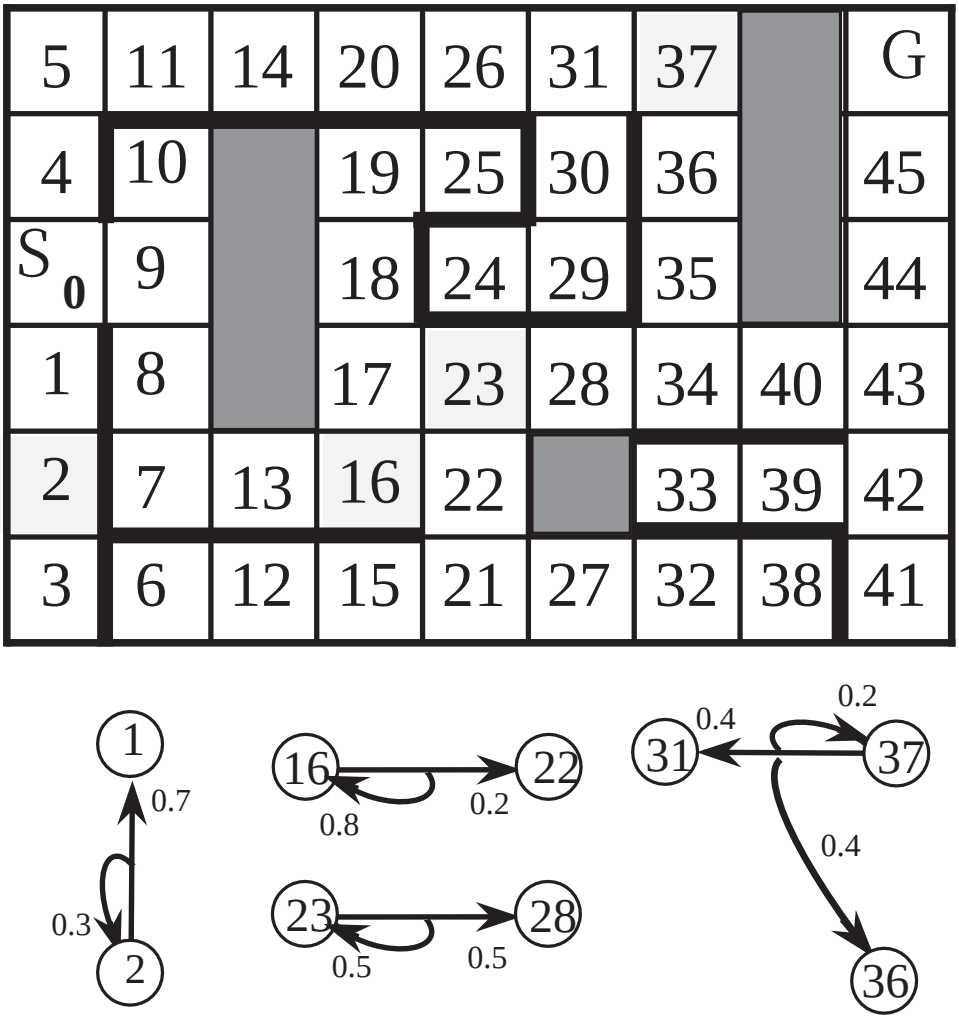


図 44: 非決定性を含む迷路環境

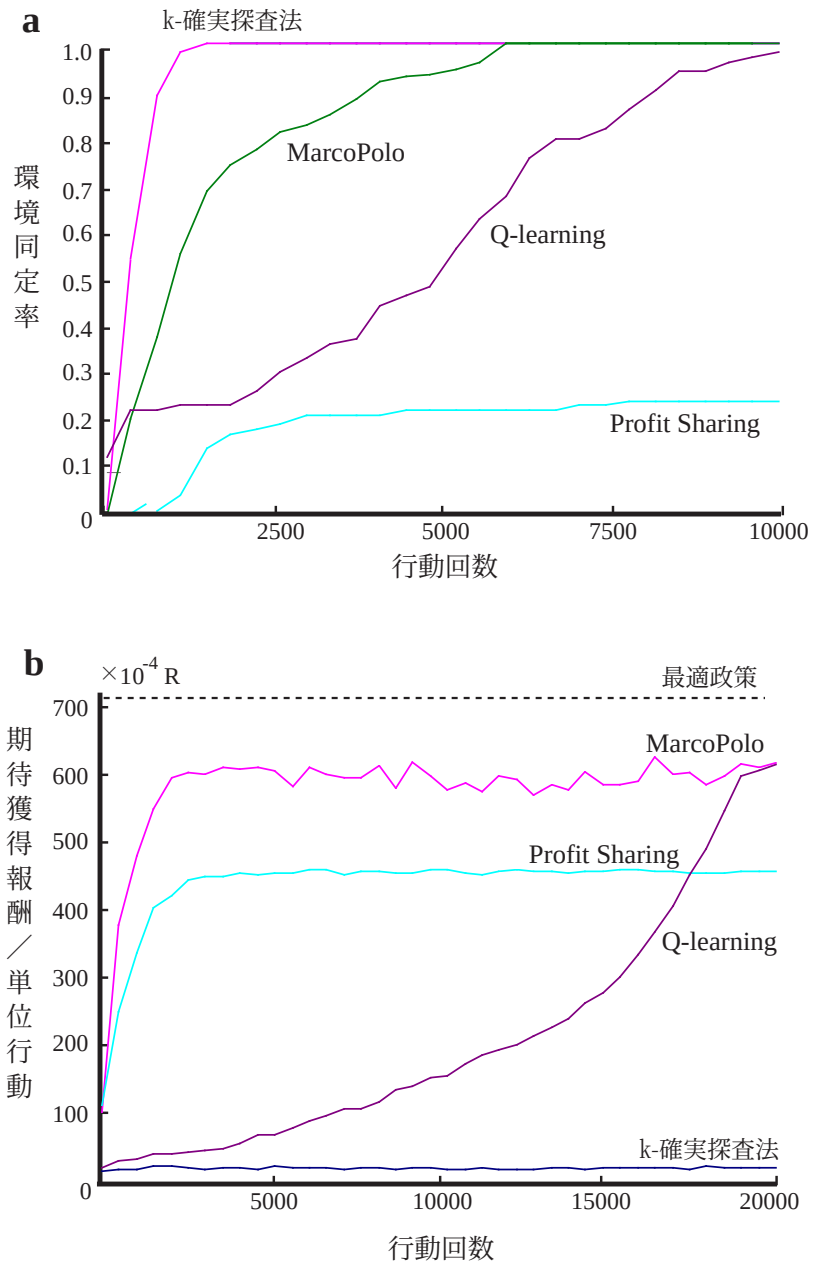


図 45: 決定的迷路環境に対する実験結果．a:環境同定率の時間的変化,b:単位行動当たりの期待獲得報酬の時間的変化

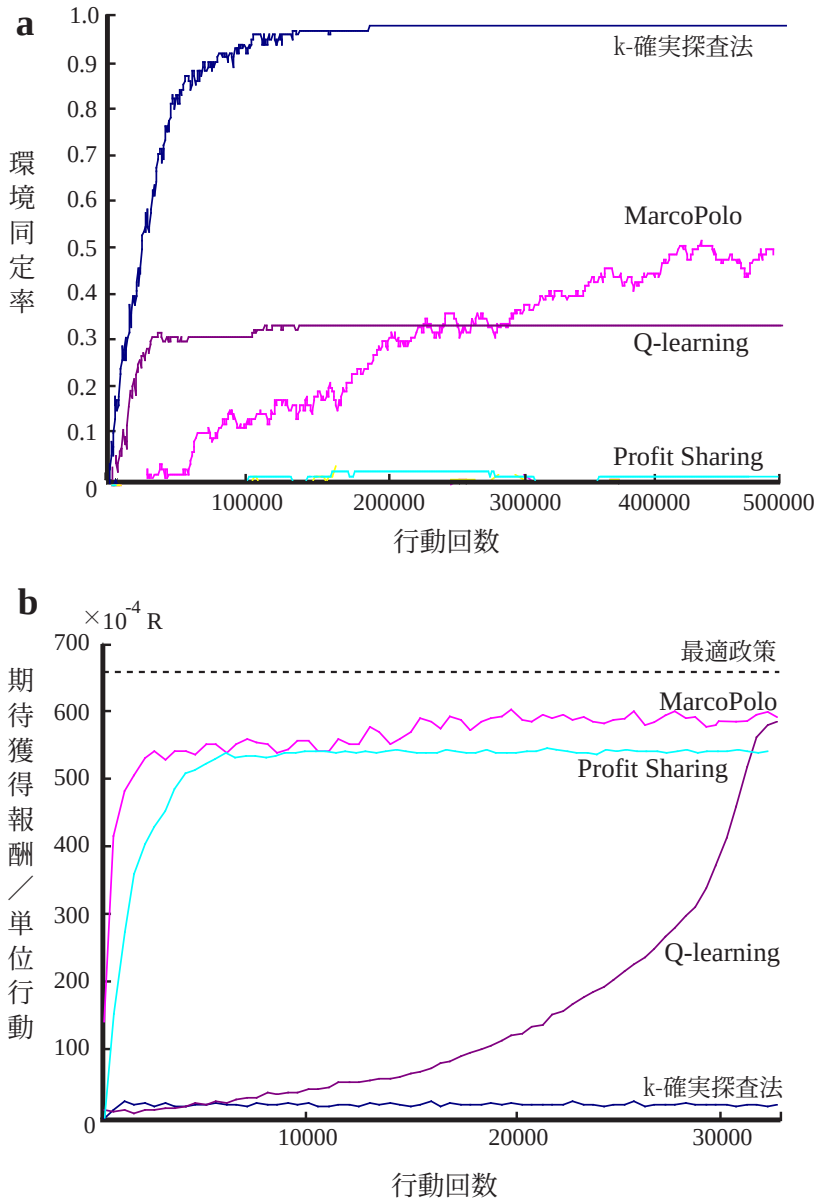


図 46: 非決定的迷路環境に対する実験結果. a:環境同定率の時間的変化,b:単位行動当たりの期待獲得報酬の時間的変化

いる．Q-learning も,MarcoPolo 同様, つねに最適政策が得られた．しかし収束に要する行動回数は MarcoPolo と比べると非常に多くなっている．これは Q-learning では報酬に近いルールから強化されていくが, 図 44 のように 1 箇所でしか報酬が与えられない場合には, 始点付近のルールの収束が特に遅れ, ランダムウォーク同然の動きが長く続くためである．MarcoPolo では PIA で最適政策を求めているため, このような報酬の与えられる場所による影響はなく, 環境の同定の進行と共に, 速やかに最適政策が獲得されていることがわかる．

図 47, 図 48 に MarcoPolo における報酬獲得と環境同定のトレードオフ関係を示す．図 47 は図 43 の環境に対する結果であり, 図 48 は図 44 の環境に対する結果である．図中の数字（%）は全行動中で報酬獲得のために費やされた行動の割合を示す．報酬獲得のために費やされる行動の割合が高くなるにつれて, 環境同定は遅れるが, 期待獲得報酬は確実に向上することが確認される．

6.4.3 燃料を有するエージェントによる学習

実験問題

報酬獲得と環境同定のトレードオフの解決が, より切実な意味をもつ問題を考える．そのためにここでは燃料を有するエージェントによる [Sutton 90] の迷路走行タスク (図 43) を考える．燃料は, エージェントが 1 回ルールを実行するごとに,1 単位だけ消費される．終点 (G) に到達し, 報酬を得た時点で,100 単位の燃料がエージェントに与えられる．学習の目的は, 燃料をゼロにしないこと, すなわち, 生き延びることである．

以下では,MarcoPolo には,2 通りの制御器を用意した．ひとつは前節までと同様の, 全行動のうち 90%を報酬獲得のために従事させるものである (Marco90)．もうひとつは,Marco90 に, 所有する燃料が E_R 以下の場合には, 必ず報酬獲得器を選択する機能を付加したものである (Marco90 ＋)．これは, 所有する燃料が少ない場合には, 冒険はしないということを単純に定式化したものである．

実験結果

この問題に対し, エージェントが初期に持っている燃料値を変えたときの生存率を調べた．生存率は, 乱数の種を変えた 100 回の実験中, 何回までが, 燃料をゼロにせずに,1 万回ルールを実行し得たかを表す．結果を図 49 に示す．図 49 の横軸はエージェントの初期燃料値, 縦軸は生存率である．MarcoPolo の比較対象として,Q-learning,PS,k-確実探査法を取り上げた．

図 49 より, k-確実探査法や Q-learning では, 全く学習できていないことがわかる．これ

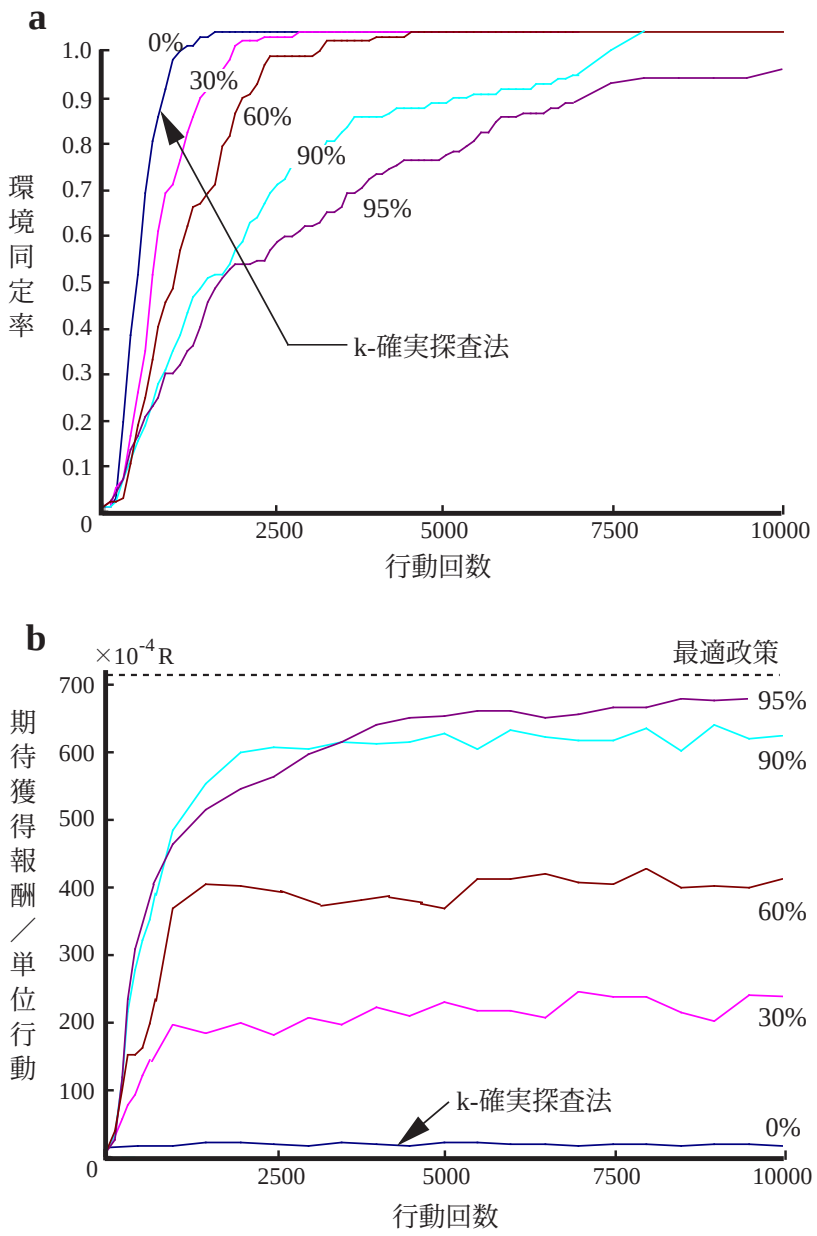


図 47: 決定的迷路環境における報酬獲得と環境同定のトレードオフの関係. a:環境同定率の時間的変化, b:単位行動当たりの期待獲得報酬の時間的変化, 数字は報酬獲得に従事する割合.

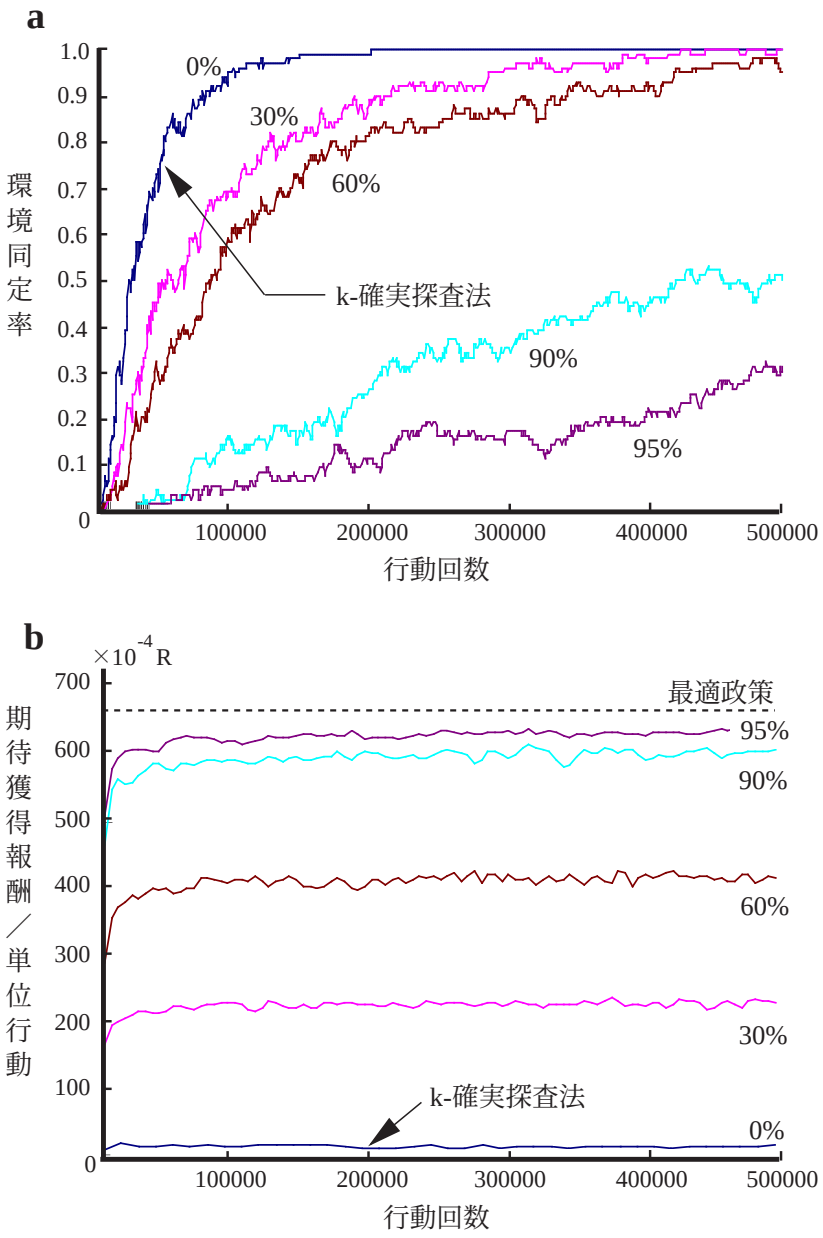


図 48: 非決定的迷路環境における報酬獲得と環境同定のトレードオフの関係. a:環境同定率の時間的変化, b:単位行動当たりの期待獲得報酬の時間的変化, 数字は報酬獲得に従事する割合.

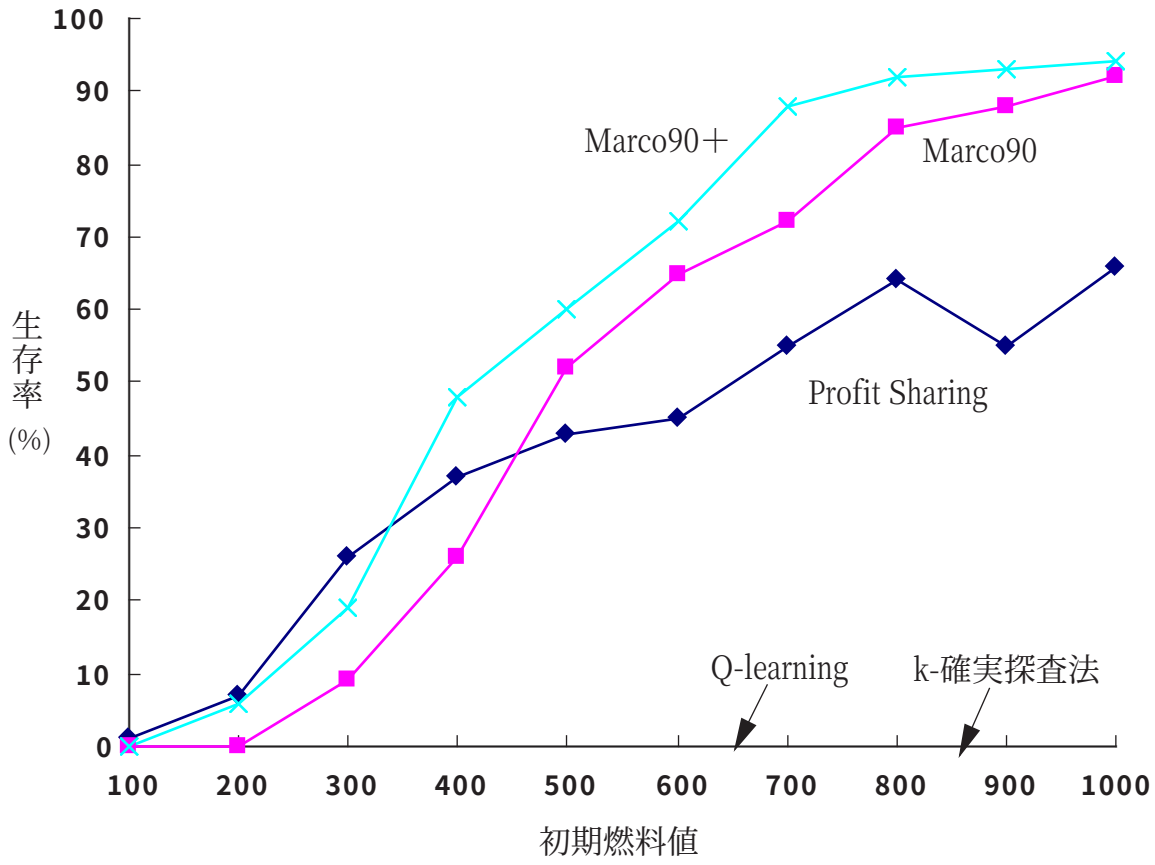


図 49: 初期燃料値を変化させたときの生存率の変化.

は, これらの手法では, 学習途中での報酬獲得のレベルが低く, 生存のために必要な燃料を十分には確保できないためである. 一方,MarcoPolo は低初期燃料値からでも, かなり高い割合で学習できていることがわかる. 特に, 本問題により特化した制御器を持つ Marco90 ＋の優位性が特徴的である. PS の低初期燃料値での健闘には特筆すべきものがある. しかし PS では必ずしも最短パスで終点 (G) に到達するとは限らず, 十分な量の初期燃料値を持っていても, 死に至る場合が多い.

このように MarcoPolo は, 報酬獲得と環境同定のトレードオフの解決が, より切実な意味を持つ問題においても十分有効であることが確認された.

6.5 おわりに

本章では, 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 MarcoPolo を提案した. MarcoPolo では, 報酬獲得時には PIA と profit sharing を相補的に利用し, 環境同定時には k-確実探査法を利用している. そのため学習の初期段階から終了に至るまで継続して報酬を得ることができ, なおかつ, 環境の完全で正確な同定も可能としている. さらに MarcoPolo では, 報酬獲得と環境同定のトレードオフを考慮して, 報酬獲得器と環境同定器の切り替えを動的に制御できるので, ユーザが指定する任意のトレードオフ比を実現することができる. 以上より MarcoPolo は,MDPs 環境下での強化学習システムとして非常に完成度の高いものと考えられる.

7　結論

7.1　まとめ

本論文では, 取り扱う環境の性質を離散マルコフ決定過程に限定した強化学習システムについて考察した. そこでは, 学習途中での報酬獲得を重視する接近を経験強化型, 最適政策の獲得を保証し得る接近を環境同定型と呼び, まず始めにそれらを個別に追求する手法に関し考察した. そしてその後, 経験強化型と環境同定型の相補的性質に着目して, 両者を巧みに統合した強化学習システムの提案を行った. 各章で述べられた成果および各手法の特徴と限界は以下のようにまとめられる.

経験強化型学習における合理性の保証

第 3 章では, 経験強化型学習の代表として profit sharing をとりあげ, 従来場当たり的に設定されてきた強化関数について解析的に考察した. まず, 局所的に見て報酬を得ることに貢献しない無効ルールが, それと競合する有効ルールよりも強化されてしまわないための必要十分条件を求めた. 次に, 大局的に見て学習されたルールの選択プランによって報酬を得るための必要十分条件が, 先に求めた条件と同値であることを示した. この結果により,profit sharing は局所的な合理性と同時に大局的な合理性も満足する有効な強化学習法となりうることが明らかにされた.

経験強化型は, 環境同定に要する行動を犠牲にし, その分, 継続的な報酬の獲得を実現した手法である. そのため, 環境全体が完全に同定されることは希であり, 学習終了時に最適政策が得られる保証もない. ここに経験強化型の限界がある.

強化学習のための効率的な環境同定戦略の提案

次に第 4 章では, 環境同定を重視する立場から強化学習に接近し, 効率よく環境を同定する行動選択器である k-確実探査法を提案した. k-確実探査法はランダム探査や最少選択優先法などに比べ, 効率的に環境が同定できる探査手法であることを示した. また,k-確実探査法と Policy Iteration Algorithm を統合した学習システムを構築することにより, 最適政策を効率よく獲得できる手法を実現した. 最適性を保証している強化学習システムの中の代表的手法である Q-learning との比較を通じ, 本手法の有効性を明らかにした.

k-確実探査法は選択回数最少のルールを優先的に選択する手法であるが, 選択回数の多少のみで, 行動を実行した際の状態遷移確率は考慮していない. そのため状態遷移が決定的な場合には, 環境を効率よく同定できるが, 確率的な状態遷移下では必ずしも効率的とはい

えない. そこで本論文では次に第 5 章において, 確率的な状態遷移下でもつねに効率的な環境同定が可能な行動選択器である ℓ-確実探査法を提案した. ℓ-確実探査法は k-確実探査法の不確実性下への自然な拡張であり, k-確実探査法では不十分であった確率論に基づく理論的裏付けを有するという特徴を持つ. さらに, 従来, あまり意識されていなかった学習途中の解の意味付けや, 学習の打ち切りを可能としている点など優れた部分が多い.

環境同定型は, 学習途中での報酬獲得に要する行動を犠牲にし, その分, 効率的な環境の同定を重視した手法である. そのため, 学習途中でもそこそこの報酬が要求される場合には, 有効な手法とは言えない. ここに環境同定型の限界がある.

報酬獲得と環境同定のトレードオフを考慮した手法の提案

ところで強化学習では, 一般に, 報酬獲得と環境同定という 2 つの目的が要求され, 両者の間にはトレードオフの関係が存在する. にもかかわらず, 従来提案されている手法の多くは, 報酬獲得または環境同定のいずれかを重視してつくられており, 強化学習の本来の要請に応えるものにはなっていない. 強化学習の代表的手法である Q-learning では収束後の最適性は保証されるものの, 学習途中の報酬獲得についてはなんら考慮されていない. 経験強化型の profit sharing は学習初期での報酬獲得に優れているものの, 準最適政策からの脱却が困難である. 環境同定手法の k-確実探査法や ℓ-確実探査法は環境同定に優れているものの, 学習途中の報酬獲得はランダムウォークの水準に留まる.

そこで本論文では, 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 MarcoPolo を最後の第 6 章において提案した. MarcoPolo では, 報酬獲得時には Policy Iteration Algorithm と profit sharing を相補的に利用し, 環境同定時には k-確実探査法や ℓ-確実探査法を利用している. 報酬獲得器と環境同定器の切り替えは, それぞれ意味のある単位で行われるため, 切り替え時のロスは排除されている. MarcoPolo は, ユーザが指定する報酬獲得と環境同定の間の任意のトレードオフ比を実現することができる. 以上のことより, MarcoPolo は,MDPs 環境下での強化学習システムとしては完成度の高いものと判断できる.

7.2 今後の展望

本論文により MDPs を対象とする強化学習システムは十分な完成度に達したものと考える. そこで今後は, MDPs を超えたより広いクラスへの拡張が急務と言える.

MDPs を越えたクラスの中でも SMDPs や POMDPs などの MDPs を拡張したクラスに対しては, 理論的土台がしっかりしていることから比較的取扱いが容易であると思われる. これらのクラスでは最適解というものが予め定義できるため, それを追求することが学習の目的とされる. 特に,SMDPs においては MDPs における Policy Iteration Algorithm に相当するアルゴリズムが存在するので, 本論文で述べた k-確実探査法などを基にした環境同定型からのアプローチが有望であると考える.

POMDPs に関しては, 現在, モデル構築型とモデルフリー型とが提案されているが, それぞれ一長一短がある. モデル構築型は決定的政策の範囲内で, 最適性が保証されるものの膨大なメモリーと計算量を要する. モデルフリー型はメモリー量や計算量は少なくて済むが, 非現実的な確率的政策をとらねばならない. これらの問題は最適性を追求する限り避けて通ることはできない. そこで今後は, profit sharing などの経験強化型からのアプローチが重要となると考える. 現状でも,profit sharing は一部の非マルコフ性を取り扱えることが知られているが, 一般にどの程度のクラスまで適応可能かは明らかにされていない. POMDPs における profit sharing の挙動を早急に解析したいと考えている.

また, 報酬の関数が vector で与えられる場合の強化学習も非常に興味深い課題のひとつである. 現状では多くの強化学習において報酬は単一種類しか考えられていないが, 複数目標が与えられ, それらの間のパレート解が要求される場合は十分考えられる. 例えば, 餌と水をバランスよく摂取しなければならないような問題である. 現在, この問題に関しては [Tenenberg 92] の先駆的な研究があるに過ぎない. そこでは Q-learning を用いたシミュレーションが行われているが, 理論的に有効なクラスが明らかにされているわけではない. 今後は, 本研究で得られた MDPs 下での知見を複数種類の目標が与えられるクラスへも拡張したいと考えている.

一方,MDPs とは全く異なる問題クラスも存在する. それは例えば, 人工生命系やマルチエージェント系などのように学習器に何らかの創発を期待するクラスである. このクラスでは何が最適かを規定することが困難なため, 環境同定型からのアプローチは難しいと考える. そのため現状で最も有望なのは, 経験強化型からのアプローチであると思われる. 特に, 本論文で示した profit sharing の合理性の定理を利用した手法は何らかの解決策を与えてくれるものと期待する. 今後は, マルチエージェント系における問題の性質を明らかにし, 有効な手法を早急に提案したいと考えている.

さらに忘れてはならないのは実問題への応用である. 強化学習はエージェントによる学習を全面に打ち出しているためロボットへの適用が真っ先に考えられがちであるが, それ以外の問題にも十分適用可能な魅力的な枠組みであると考える. 強化学習が真に有望な手法であると認知されるためには, 人工生命的なアプローチの中でよく行われているようないわゆる Toy problem ではない実際的な問題に適用されることが望ましい. 今後はそのような有望なアプリケーション先を見つけることも強化学習の発展にとって重要な課題のひとつであると考える.

謝辞

本研究を行うにあたって終始多大なる御指導，御配慮を頂きました小林重信教授，をはじめ，山村雅幸助手，そのほか手伝って頂いた研究室の皆様に深く感謝いたします.

参考文献

[Barto 83] Barto, A. G., Sutton, R. S. and Anderson, C. W. *Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems*, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, no.5, pp.834-846, (1983).

[Basye 95] Basye, K., Dean, T. and Kaelbling, L. P. *Learning dynamics: system identification for perceptually challenged agents*, Artificial Intelligence 72, pp.139-171, (1995).

[Baird 95] Baird, L. *Residual Algorithms: Reinforcement Learning with Function Approximation*, Proc. of 12th International Conference on Machine Learning, pp.30-37, (1995).

[Bersini 94] Bersini, H. *Reinforcement Learning for Homeostatic Endogenous Variables*, Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior, pp.325-333, (1994).

[Bertsekas 76] Bertsekas, D. P. *Dynamic Programing and Stochastic Control*, Mathematics in Science and Engineering. Volume 125. Richard Bellman ed., Academic Press, (1976).

[Bradtke 93] Bradtke, S. J. *Reinforcement Learning Applied to Linear Quadratic Regulation*, Advances in Neural Information Processing Systems 5, (1993).

[Bradtke 94] Bradtke, S. J. and Duff, M. O. *Reinforcement Learning Methods for Continuous-Time Markov Decision Problems*, Advances in Neural Information Processing Systems 7, pp.393-400, (1995).

[Brooks 86] Brooks, R. *A robust layered control system for a mobile robot*, IEEE Journal of Robotics and Automation, vol 2, No.1, (1986).

[Cassandra 94] Cassandra, A. R., Kaelbling, L. P. and Littman, M. L. *Acting Optimally in Partially Observable Stochastic Domains*, Proceedings of the 12th National Conference on Artificial Intelligence, Vol. 2, pp.1023-1028, (1994).

[Chrisman 92] Chrisman, L. *Reinforcement learning with perceptual aliasing: The Perceptual Distinctions Approach*, Proceedings of the 10th National Conference on Artificial Intelligence, pp.183-188, (1992).

[Clouse 92] Clouse, J. A., and Utogoff, P. E. *A Teaching Method for Reinforcement Learning*, Proc. of 9th International Conference on Machine Learning, pp.92-101, (1992).

[Colorni 91] Colorni, A., Dorigo, M. and Maniezzo, V. *A Distributed Optimization by Ant Colonies*, Proc. of 1st European Conference on Artificial Life, pp.134-142, (1991).

[Dayan 92] Dayan, P. *The convergence of TD(**λ**) for general **λ***, Machine Learning 8, pp.341-362, (1992).

[Dean 92] Dean, T., Angluin, D., Basye, K., Engelson, S., Kaelbling, L., Kokkevis, E. and Maron, O. *Inferring Finite Automata with Stochastic Output Function and an Application to Map Learning*, Proceedings of the 10th National Conference on Artificial Intelligence, pp.208-214, (1992).

[Feller 60] Feller, W. *An Introduction to Probability Theory and Its Applications*, Modern Asia Editions, (1960).

[Gambardella 95] Gambardella, L. M., and Dorigo, M. *Ant-Q: A Reinforcement Learning approach to the traveling salesman problem*, Proc. of 12th International Conference on Machine Learning, pp.252-260, (1995).

[Grefenstette 88] Grefenstette, J. J. *Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms*, Machine Learning 3, pp.225-245, (1988).

[Holland 86] Holland, J. H. *Escaping brittleness*, Machine Learning, an artificialintelligence approach. Volume II. R. S. Michalski, J. G. Carbonell and T. M. Mitchell ed., Morgan Kaufmann, pp.593-623, (1986).

[Holland 87] Holland, J. H., and Reightman. J. S. *Cognitive Systems Based on Adaptive Algorithms*, Pattern-Directed Inference Systems. Waterman, D. A., and Hayes-Roth, F. ed., Academic Press, (1987).

[Jaakkola 94] Jaakkola, T., Singh, S. P. and Jordan, M. I. *Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems*, Advances in Neural Information Processing Systems 7, (1995).

[Kaelbling 91] Kaelbling, L. P. *An Adaptable Mobile Robot*, Proc. of 1st Europian Conference on Artificial Life, pp.41-47, (1991).

[Kaelbling 94] Kaelbling, L. P. *Associative Reinforcement Learning: Function in k-DNF*, Machine Learning 15, pp.279-298, (1994).

[Kimura 95] Kimura, H., Yamamura, M. and Kobayashi, S. *Reinforcement Learning by Stochastic Hill Climbing on Discounted Reward*, Proc. of 12th International Conference on Machine Learning, pp.295-303, (1995).

[Liepins 89] Liepins, G. E., Hilliard, M. R., Palmer, M., and Rangarajan, G. *Alternatives for Classifier System Credit Assignment*, Eleventh International Joint Conference on Artificial Intelligent, pp.756-761, (1989).

[Lin 90] Lin, L. *Self-improving Reactive Agents : Case Studies of Reinforcement Learning Frameworks*, Proc. of 1st International Conference on Simulation of Adaptive Behavior, pp.297-305, (1990).

[Lin 91a] Lin, L. *Programming Robot Using Reinforcement Learning and Teaching*, Proc. of 9th National Conference on Artificial Intelligent, pp.781-786, (1991).

[Lin 91b] Lin, L. *Self-improvement Based On Reinforcement Learning, Planning and Teaching*, Proc. of 8th International Workshop on Machine Learning, pp.323-327, (1991).

[Lin 92] Lin, L. and Mitchell, T. M. *Reinforcement Learning With Hidden States*, Proc. of 2nd International Conference on Simulation of Adaptive Behavior, pp.271-280, (1992).

[Lin 93] Lin, L. *Scaling Up Reinforcement Learning for Robot Control*, Proc. of 10th International Conference on Machine Learning, pp.182-189, (1993).

[Littman 94] Littman, M. L. *Markov games as a framework for multi-agent reinforcement learning*, Proceedings of the 11th International Conference on Machine Learning, pp.157-163, (1994).

[Littman 95] Littman, M. L. *Learning policies for partially observable environments: Scaling up*, Proceedings of the 12th International Conference on Machine Learning, pp.362-370, (1995).

[Lovejoy 91] Lovejoy, W. S. *A Survey of Algorithmic Methods for Partially Observed Markov Decision Processes*, Annals of Operations Research 28, pp.47-65, (1991).

[McCallum 92] McCallum, R. A. *Using Transitional Proximity for Faster Reinforcement Learning*, Proc. of 9th International Conference on Machine Learning, pp.316-321, (1992).

[McCallum 93] McCallum, R. A. *Overcoming Incomplete Perception with Utile Distinction Memory*, Proc. of 10th International Conference on Machine Learning, pp.190-196, (1993).

[McCallum 95] McCallum, R. A. *Instance-Based Utile Distinctions for Reinforcement Learning with Hidden State*, Proceedings of the 12th International Conference on Machine Learning, pp.387-395, (1995).

[Mahadevan 91a] Mahadevan, S., and Connell, J. *Automatic Programming of Behavior-based Robots using Reinforcement Learning*, Proc. of 9th National Conference on Artificial Intelligent, pp.774-780, (1991).

[Mahadevan 91b] Mahadevan, S., and Connell, J. *Scaling Reinforcement Learning to Robotics by Exploiting the Subsumption Architecture*, Proc. of 8th International Workshop on Machine Learning, pp.328-332, (1991).

[Mahadevan 94] Mahadevan, S. *To Discount or not to Discount in Reinforcement Learning: A Case Study Comparing R Learning and Q Learning*, Proc. of 11th International Conference on Machine Learning, pp.164-172, (1994).

[Moore 94] Moore A. W. *Prioritized Sweeping: Reinforcement Learning With Less Data and Less Time*, Machine Learning 13, pp.103-129, (1994).

[Munos 94] Munos, R. and Patinel, J. *Reinforcement learning with dynamic covering of state-action space: Partitioning Q-learning*, Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior, pp.354-363, (1994).

[Papadimitriou 87] Papadimitriou, C. H. and Tsitsiklis, J. N. *The Complexity of Markov Dicision Processes*, Mathmatics of Operations Research, pp.441-450, (1987).

[Peng 95] Peng, J. *Efficient Memory-Based Dynamic Programing*, Proceedings of the 12th International Conference on Machine Learning, pp.438-446, (1995).

[Riolo 87] Riolo, R. L. *Bucket Brigade Performance : I. Long Sequences of Classifiers*, Proceedings of the 2nd International Conference on Genetic Algorithms, pp.184-195, (1987).

[Rouvellou 95] Rouvellou, I. and Hart, W. H. *Inference of a Probabilistic Finite State Machine from its Output*, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-25, no.3, pp.424-437, (1995).

[Samuel 59] Samuel, A. L. *Some Studies in Machine Learning Using the Game of Checkers*, IBM Journal on Research and Development 3. pp.210-229, (1959).

[Schwartz 93] Schwartz, A. *A Reinforcement Learning Method for Maximizing Undiscounted Rewards*, Proc. of 10th International Conference on Machine Learning, pp.298-305, (1993).

[Shen 93] Shen, W. *Learning Finite Automata Using Local Distinguishing Experiments*, 13th International Joint Conference on Artificial Intelligent, pp.1088-1093, (1993).

[Singh 92] Singh, S. P. *Transfer of learning by Composing Solutions of Elemental Sequential Tasks*, Machine Learning 8, pp.323-339, (1992).

[Singh 94a] Singh, S. P. *Learning Without State-Estimation in Partially observable Markovian Decision Processes*, Proc. of 11th International Conference on Machine Learning, pp.285-292, (1994).

[Singh 94b] Singh, S. P. *Reinforcement Learning with Soft State Aggregation*, Advances in Neural Information Processing Systems 6, pp.361-368, (1994).

[Smith 91] Smith, R. E., and Goldberg, D. E. *Variable Default Hierarchy Separation in a classifier system*, Foundation of Genetic Algorithms, Rawlins ed., Morgan Kaufmann, 148-167 (1991).

[Sutton 88] Sutton, R. S. *Learning to Predict by the Methods of Temporal Differences*, Machine Learning 3, pp.9-44, (1988).

[Sutton 90a] Sutton, R. S. *Integrated Architecture for Learning, Planning, and Reacting Based on Approximating Dynamic Programing*, Proc. of 7th International Conference on Machine Learning, pp.216-224, (1990).

[Sutton 90b] Sutton, R. S. *Reinforcement Learning Architectures for Animats*, Proc. of 1st International Conference on Simulation of Adaptive Behavior, pp.337-343, (1990).

[Tan 93] Tan, M. *Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents*, Proc. of 10th International Conference on Machine Learning, pp.330-337, (1993).

[Tenenberg 92] Tenenberg, J., Karlsson, J., and Whitehead, S. *Learning via Task Decomposition*, Proc. of 2nd International Conference on Simulation of Adaptive Behavior, pp.337-343, (1992).

[Thrun 92] Thrun, S. B. *Active Exploration in Dynamic Environment*, Advances in Neural Information Processing Systems 4, pp.531-538, (1992).

[山村 95] 山村 雅幸, 宮崎 和光, 小林 重信. **エージェントの学習**, 人工知能学会誌, vol 10, No 5, pp.23-29, (1995).

[Yanco 92] Yanco, H. and Stein, L. A. *An Adaptive Communication Protocol for Cooperatin Mobile Robots*, Proc. of 2nd International Conference on Simulation of Adaptive Behavior, pp.478-485, (1992).

[ワグナー 78] ワグナー (高橋 幸雄, 森 雅夫, 山田 尭 訳). **「オペレーションズ・リサーチ入門** *5*=**確率的計画法」**, 培風館, (1978).

[Watkins 92] Watkins, C.J.C.H., and Dayan, P. *Technical Note:Q-Learning*, Machine Learning 8, pp.55-68, (1992).

[Whitehead 90] Whitehead, S. D., and Ballrd, D. H. *Active Perception and Reinforcement Learning*, Proc. of 7th International Conference on Machine Learning, pp.179-188, (1990).

[宮崎 94] 宮崎 和光, 山村 雅幸, 小林 重信. **強化学習における報酬割当の理論的考察**, 人工知能学会誌, vol 9, No 4, pp.104-111, (1994).

[Miyazaki 94] Miyazaki, K., Yamamura, M. and Kobayashi, S. *On the Rationality of Profit Sharing in Reinforcement Learning*, Proceedings of the 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing, pp. 285-288, (1994).

[宮崎 95] 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).