特　集　「強化学習」

離散マルコフ決定過程下での強化学習

Reinforcement Learning Systems for Discrete Markov Decision Processes

宮崎　和光*　　小林　重信*
Kazuteru Miyazaki　Shigenobu Kobayashi

*　東京工業大学大学院総合理工学研究科
Graduate School of Interdisciplinary Science and Engineering, Tokyo Institute of Technology.

1997年9月29日　受理

Keywords: reinforcement learning, Markov decision processes, profit sharing, k-certainty exploration method, MarcoPolo.

1. はじめに

宇宙や深海など人間にとってまったく未知の環境において，自律的に動き回り，知的なタスクを実行するロボットを実現することは人類の永年の夢の1つである．未知の環境では，観測される状態に対しどういう行動を取ったらよいか，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しなければならない．そのような情報を総称して報酬と呼ぶ．強化学習とは，報酬という特別な入力を手がかりに環境に適応した行動決定戦略を追求する教師なし機械学習である．

強化学習の目的は，できるだけ多くの報酬をできるだけ素早く獲得することにある．より多くの報酬を得るためには，環境を広く探査（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は軽視されがちになる．このように強化学習には，環境同定と報酬獲得の間にトレードオフの関係が存在する．

強化学習についての理論的な成果が最も多く蓄積されているクラスは，離散マルコフ決定過程 (MDPs) である．本解説の目的は，MDPs を対象とした強化学習の最近の研究を概観し，主要な手法の特徴と限界を考察することにある．

以下，2章では，MDPs を対象とする強化学習システムの一般的な枠組みを紹介した後，古典的強化学習，Q-learning とその発展形，環境同定の手法を紹介する．3章～5章では，著者らが MDPs を対象に行ったオリジナルな研究成果を紹介する．3章では，古典的学習法である Profit Sharing を取り上げ，学習結果に一定の合理性を保証するための必要十分条件を紹介する．4章では，MDPs 環境を効率よく同定することを目的に考案された k-確実探査法を紹介する．5章では，3章および4章で紹介した方法をベースに，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である MarcoPolo を紹介する．6章では，非 MDPs 環境への対応を論じる．7章は結論であり，今後の研究課題をとりまとめる．

2. MDPs を対象とする学習システム

2・1　離散マルコフ決定過程

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．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．一般に，エージェントの行動に対して環境の状態遷移は決定的であ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．

本稿では取り扱う問題領域を**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

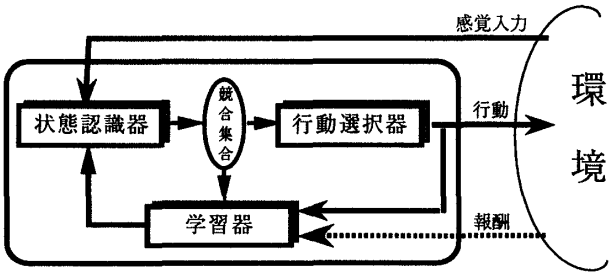

図1　強化学習システムの一般的な枠組み

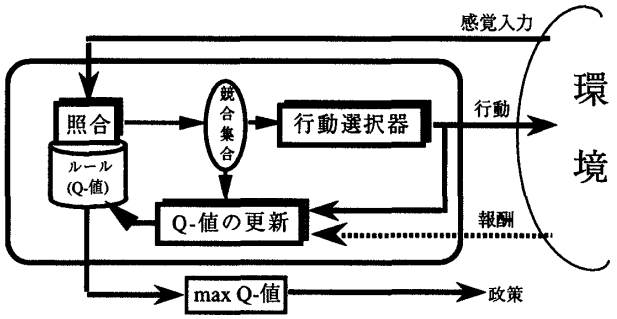

図2　Q-learning の構成図

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 を対象とする強化学習システムは状態認識器，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できる．図 1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．一般にここでは感覚入力とルールの前提部との照合が行われる．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できるだけ素早く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．本稿では [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**，最適政策を得るために環境同定を重視する接近を**環境同定型**と呼ぶ．

2・2　古典的強化学習

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 87] の枠組みの中で研究されてきた **Bucket Brigade** [Holland 86] と **Profit Sharing** [Holland 87, Grefenstette 88] がよく知られている．

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．報酬の有無にかか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．

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，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かった．

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．本稿では，Profit Sharing の合理性について **3** 章で議論する．

2・3　Q-learning とその発展形

MDPs を対象とした代表的手法に **Q-learning** [Watkins 92] がある．Q-learning の原型は [Sutton 88] の **Temporal Difference 法** (TD 法) にある．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のに対し，Q-learning は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．

Q-learning の構成を 図 2 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．

$$Q(x,a)=(1-\alpha)Q(x,a)+\alpha(r+\gamma\max_b Q(y,b)) \quad (1)$$

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にある．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．

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．

〔1〕 **Dyna**

[Sutton 90]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-値の収束を加速するための枠組みとして**Dyna**を提案している．Dynaの欠点は非決定性を適切に取り扱えないことにある．

〔2〕 **TPQ-learning**

[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**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．本手法は,Dynaと同様,非決定性を正しく取り扱えない欠点を持つ．

〔3〕 **教示の導入**

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 91]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 91]とは異なり教師が時々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1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．

〔4〕 **CQ-learning**

[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**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．CQ-learnigでは，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に問題がある．

〔5〕 **Q(λ)-learning**

[Peng 95]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[Dayan 92,Sutton 88]のTD(λ)を発展させた手法であるが，MDPs下での最適性が保証されているわけではない．

〔6〕 **R-learning**

Q-learningでは割引期待報酬の最大化が学習の目的とされる．それに対し[Schwartz 93]は，平均報酬の最大化を目指す**R-learning**と呼ばれる手法を提案している．[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られているに過ぎない．

2・4 環境同定の手法

環境を効率よく同定する基本的考えは，まだ十分に試されていない行動を優先して試すことにある．

〔1〕 **IE法**

[Kaelbling 91]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行う**IE法**を提案している．IE法は移動ロボットへの適用やk-DNFと呼ばれる問題クラス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．

〔2〕 **Prioritized Sweeping**

[Moore 94]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により，効率のよい環境同定を行う**Prioritized Sweeping**を提案している．本手法では，連立1次方程式の解法としてよく知られるGauss-Siedelの反復法を利用して政策を求め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．

〔3〕 **Fiechterの方法**

[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

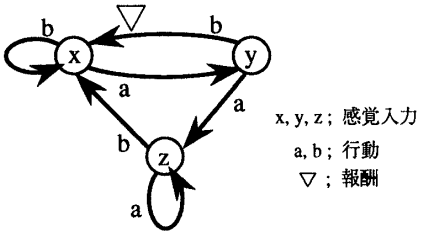

図3 例で用いた環境

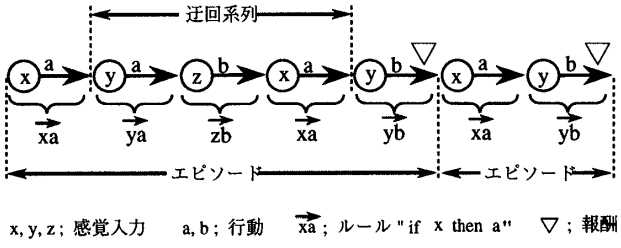

図4 エピソードおよび迂回系列の例

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．本手法では，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．

〔4〕 **Thrun の方法**

[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．この方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強い仮定が導入されている．

〔5〕 **その他の方法**

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 1 つである非決定性が正しく取り扱われていない．[Dean 92, Rouvellou 95, Shen 93] 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不自然な仮定を導入している．

3. Profit Sharing の合理性定理

3・1 準　　備

Profit Sharing (PS) とは報酬を得たときにそれまでに使用されたルール系列を一括的に強化する手法である．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**という．例えば図 3の環境でロボットが$\overrightarrow{xa},\overrightarrow{ya},\overrightarrow{zb},\overrightarrow{xa},\overrightarrow{yb},\overrightarrow{xa},\overrightarrow{yb}$と行動したとすると，このなかには$(\overrightarrow{xa},\overrightarrow{ya},\overrightarrow{zb},\overrightarrow{xa},\overrightarrow{yb})$,$(\overrightarrow{xa},\overrightarrow{yb})$ の 2 つのエピソードが含まれている (図 4)．PS ではエピソード単位でルールに付加された評価値を強化する．報酬からどれだけ過去かを引き数とし，強化値を返す関数を**強化関数**と呼ぶ．

あるエピソードで，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**という．例えば図 3の環境で，エピソード$(\overrightarrow{xa},\overrightarrow{ya},\overrightarrow{zb},\overrightarrow{xa},\overrightarrow{yb})$ には，迂回系列 $(\overrightarrow{ya},\overrightarrow{zb},\overrightarrow{xa})$ 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**と呼び，それ以外を**有効ルール**と呼ぶ．

3・2 合 理 性 定 理

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．著者らは，任意の無効ルールが抑制されるための強化関数の必要十分条件が以下の式であることを証明している [宮崎 94]．

［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

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

□

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 (2) を**無効ルール抑制条件**と呼ぶ．

定理 1 は,1 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．次に，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得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得ることである．

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**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**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 5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．

[宮崎 94] 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証明した．

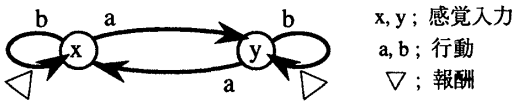

図5 例で用いた環境

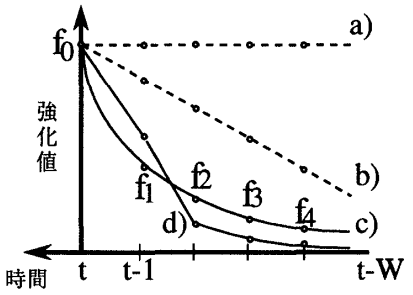

図6 強化関数の例. c) と d) が定理を満たす関数

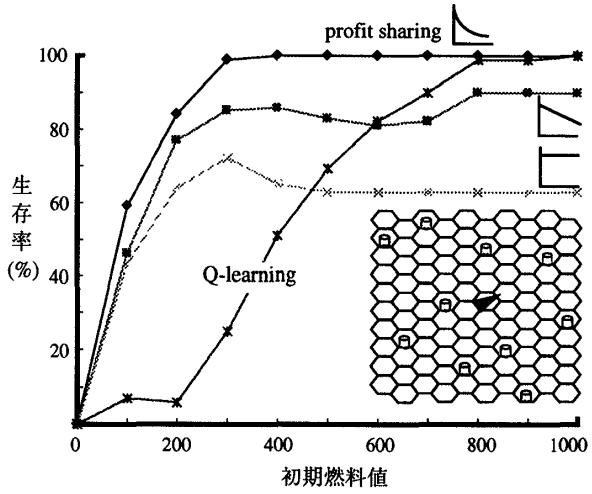

図7 Profit Sharing の挙動を理解するための数値例

［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. □

定理 2 を PS における合理性定理と呼ぶ.

3・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. また定理 2 は, PS 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.

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合理的なルールの選択が保証される. 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 6-a) や等差減少関数 (図 6-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考えられる (図 6-c).

$$f_n = \frac{1}{S} f_{n-1}, \quad n = 1, 2, \cdots, W-1. \qquad (3)$$
$$\text{ただし, } S \geq L+1$$

この他にも定理を満たす関数には図 6-d など様々なものが考えられる.

3・4 数値例

10×10 のハニカム状の環境に置かれたロボットが燃料を捕獲する問題を用い合理性定理の有効性を示す. ロボットは燃料が切れると死んでしまう. 図 7にロボットの初期燃料値を変化させたときの生存率を示す. 生存率は乱数の種を変えた 100 回の実験中, 何回までが燃料切れになるまでに報酬プランを学習できていたかを表す.

定理を満たす関数を用いれば少ない燃料でも安定して学習できているが, 定理を満たさない関数では十分な燃料を有していても学習できていないことがみてとれる. この問題は MDPs で記述可能なので,Q-learning を用いれば最適性が保証される. しかしそのためには多くの試行錯誤, すなわちこの場合, 多くの初期燃料を要することが実験結果からもよくわかる.

合理性定理を満たす強化関数を用いれば, 学習結果に一定の合理性を保証することができる. また, 合理性定理では環境側にマルコフ性を仮定していない. したがって PS は, 非 MDPs 環境への適用が大いに期待されている手法でもある. しかし PS では, 環境が MDPs であっても最適性が保証されるとは限らない点には注意を要する.

4. k-確実探査法による環境同定

4・1 基本方針

MDPs の環境では, 環境すなわち各ルールの状態遷移確率および得られる報酬の期待値が既知であれば **Policy Iteration Algorithm** (PIA)[ワグナー 78] により最適政策を決定することができる [Singh 92]. したがって, 環境をより少ない行動選択回数で同定できる手法が重要となる. 著者らは,MDPs の環境を効率よく同定する手法として k-確実探査法を提案している [宮崎 95].

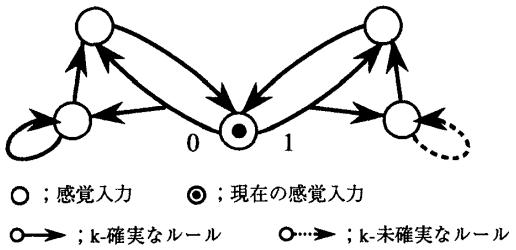

図8　k-確実なループに至るルールの例

```
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9　k-確実探査法のアルゴリズム

k-確実探査法の主要部分は，効率のよい環境同定を実現するための行動選択器にある．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**，k-確実でないルールを**k-未確実**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8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，環境の同定効率が悪化する．k-確実探査法の基本的アイデアは，そのようなルールを選択候補から除外することにある．

4・2　k-確実探査法のアルゴリズム

k-確実探査法のアルゴリズムを図9に示す．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．さらに,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．

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1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1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1つをランダムに選択する．

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．

4・3　k-確実探査法の特徴

k-確実探査法は次のような特徴を持つ．

(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

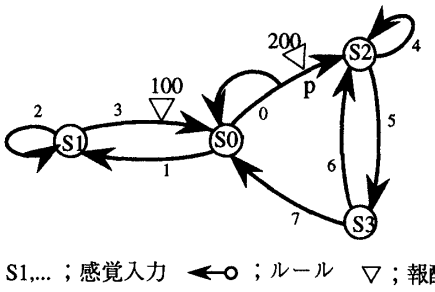

図10　k-確実探査法と Q-learning の比較で用いた環境

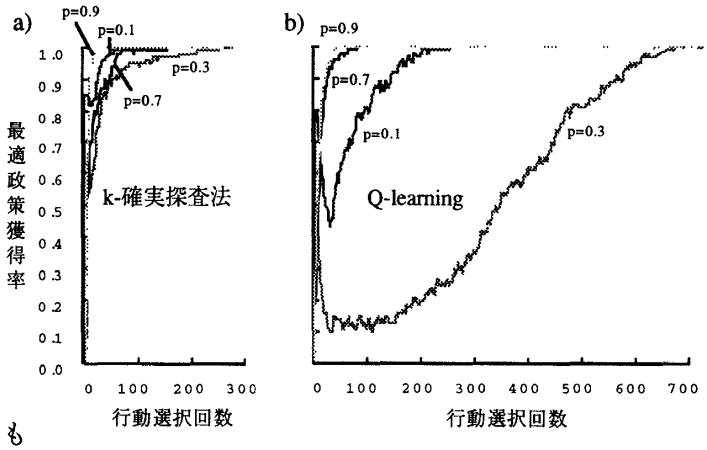

図11　k-確実探査法の挙動を理解するための数値例

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.

(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.

4・4 数　値　例

図 10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. 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. $p < 0.5$ 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最適政策の獲得される速さを Q-learning と k-確実探査法とで比較する.

結果を図 11に示す. 横軸は行動選択回数, 縦軸は最適政策の獲得率である. 最適政策の獲得率は, 乱数の種を変えて行った 100 回の実験中, 何回までが, その行動選択回数の時点で最適政策を獲得していたかを表す. 図 11からもわかる通り, k-確実探査法では環境の構造によらず安定して素早く最適政策が求まっているのに対し, Q-learning では, p の値によって大きく性能がばらつくことがみてとれる.

4・5 k-確実探査法の拡張

k-確実探査法では, ルールの選択回数の多寡によって, 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, 必ずしも効率的に環境が同定されるとは限らない.

著者らは,k-確実探査法を拡張し, 確率的な状態遷移下でも効率的に振る舞う手法として l-確実探査法を提案している [宮崎 96]. 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.

$$\text{達成率 (\%)} = \frac{\text{現時点までの選択回数}}{\text{同定に要する選択回数}} \times 100 \tag{4}$$

ここで, 同定に要する選択回数は信頼区間の考えを応用して計算される. 具体的計算方法については [宮崎 96] を参照されたい.

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. 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効率的な環境同定を実現している.

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有効である.

k-確実探査法や l-確実探査法を利用すれば, MDPs の環境を効率よく同定することができる. しかしこれらの手法は環境の同定のみを考慮した手法なので, 環境同定途上の報酬獲得は一般に非常に低いレベルに留まる. 特に PIA が利用できない学習の初期段階では, 通常, ほとんど報酬の獲得は期待できない.

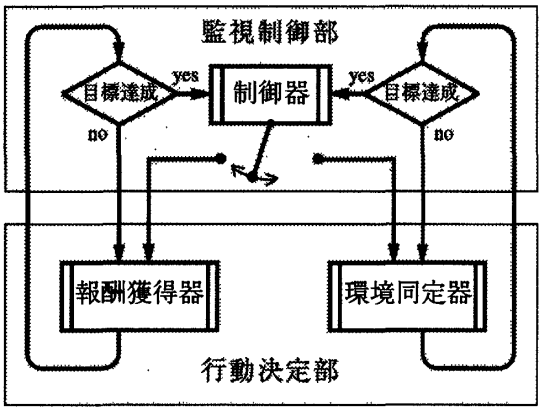

図12 MarcoPoloの基本的枠組み

5. MarcoPolo：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

5・1 基本的枠組み

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．著者らは，このトレードオフを考慮した手法としてMarcoPoloを提案している [宮崎 97]．

MarcoPoloでは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12に示すような**監視制御部**と**行動決定部**からなる枠組みを採用している．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．以下では，各構成要素について詳しく説明する．

5・2 MarcoPoloの各構成要素

〔1〕 報酬獲得器の設計

報酬獲得の手法としては，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) が採用されている．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，現時点までに同定された環境下での最適政策を得ることができる．PIAが利用できない場合PSを利用する．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．

報酬獲得器は報酬が得られた時点で打ち切られる．報酬獲得器の目的は報酬の獲得なので十分妥当な基準である．

〔2〕 環境同定器の設計

環境同定の手法としてはk-確実探査法が採用されている．すべてのルールが1-確実以上になれば，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．

環境探査器は少なくとも1つのルールがk-確実になり，かつ，現状態で,k-確実でないルールが選べなくなった時点で打ち切られる．1つのルールがk-確実になった時点ですぐに打ち切ってしまう場合に比べ，より意味のある単位での環境同定が期待できる．

〔3〕 制御器の設計

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** (E_R) と**環境同定コスト** (E_I) が利用される．これら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．MarcoPolo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している．

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 $100 * T\%$ ($0 \leq T \leq 1$) 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．

$$\frac{E_R * x}{E_R * x + E_I * (1.0 - x)} = T \tag{5}$$

したがって，確率

$$x = \frac{T * E_I}{(1 - T) * E_R + T * E_I} \tag{6}$$

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．これにより，報酬獲得と環境同定のトレードオフを考慮した挙動が実現されている．

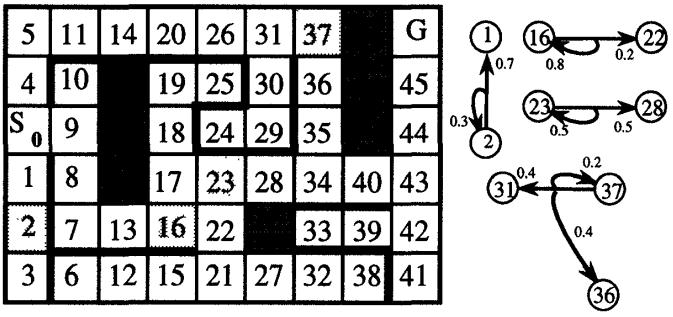

図 13　迷路走行問題

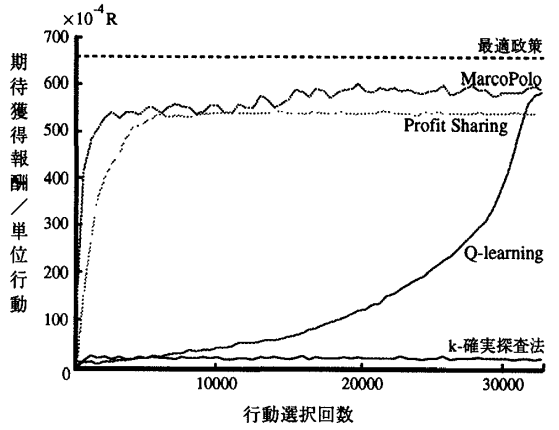

図 14　非決定的迷路環境における単位行動当たりの期待獲得報酬の時間的変化

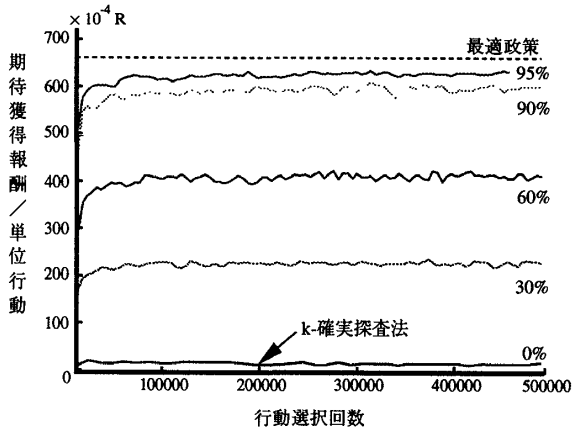

図 15　非決定的迷路環境における報酬獲得と環境同定のトレードオフの関係．数字は報酬獲得に従事する割合

5・3 MarcoPolo の特徴

MarcoPolo は以下のような特徴を持つ．

(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．

5・4 数　値　例

図 13に示すような迷路走行問題に MarcoPolo を適用することを考える．図 14に図 13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．

MarcoPolo は素早く期待される値に収束していることがわかる．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まれであった．

図 15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．それぞれの要求に応じ，素早く正しい値に収束していることがみてとれる．

6. 非 MDPs 環境への対応

本稿では MDPs を対象とする強化学習を中心に概要を紹介してきたが，現在，強化学習研究のトピックスは非 MDPs 環境へ移行しつつある．MDPs を越えたクラスのなかでも Semi Markov Decision Processes (SMDPs) や Partially Observable Markov Decision Processes (POMDPs) などの MDPs を拡

張したクラスに対しては, 理論的基板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. これらのクラスでは最適性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 [Bradtke 94, 北川 67], 本稿で述べた k-確実探査法などをもと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られる.

POMDPs に関する接近法としては, 現在, モデルベース型と確率的政策を利用する手法とが有名である. モデルベース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. 確率的政策はメモリー量や計算量は少なくて済むが, 確率的に行動を出力するため, 決定的政策に比べ, 報酬を得るために必要以上に多くの行動を要する場合がある.

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であると考える. 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, 有効な手法を提案したいと考えている.

非 MDPs 環境の中でも, 最近, 人工生命やマルチエージェント系など, 学習器に何らかの創発を期待するクラスへの関心も高まっている. このクラスでは最適性の定義そのものが困難なため, 環境同定型からのアプローチは難しいと考える. これらの問題に対しても, 今後は, Profit Sharing などの経験強化型をもとにした接近が重要になると考える.

7. おわりに

本解説では, 取り扱う環境の性質を MDPs に限定した強化学習について各種手法の特徴と限界を概観した. MDPs を対象とする代表的手法として知られる Q-learning は, 収束性が保証されているという安心感ゆえに, 各種の応用が試みられると同時に, 欠点を克服するためのさまざまな拡張が行われてきた. Q-learning は, 経験強化型側面と環境同定型側面の両方を併せ持った手法といえるが, 見方を変えれば, 中途半端な手法ともいえる.

Profit Sharing は経験強化型に徹した手法であり, 学習初期における報酬獲得という意味での立ち上がりの早さは注目に値する. Profit Sharing の合理性定理は, 無効ルールの強化を抑制するとともに, 報酬が継続して得られることを保証している. 著者らの最近の研究では, Profit Sharing は, POMDPs 環境下でも予期した以上に頑健であること, またマルチエージェント強化学習の手法としても極めて有望であるとの知見を得ている.

k-確実探査法は, Profit Sharing の対極に位置する環境同定に徹した手法である. k-確実という単純明快な指標を手がかりに環境同定を行う方法は, Policy Iteration Algorithm と組み合わせることにより, MDPs 環境下で最適政策を効率よく導くことが可能である.

1 章で述べたように, 強化学習には, 環境同定と報酬獲得の間にトレードオフの関係が存在する. 言い換えれば, 最適性と効率性のトレードオフを考慮して強化学習システムを設計することが重要である. MarcoPolo は, ユーザが設定した環境同定コストと報酬獲得コストのトレードオフ比に基づいて, 環境同定と報酬獲得を交互に繰り返す統合型学習システムであり, 工学的に意味のある挙動を示す点で興味深い.

今後の研究課題として, 1) MDPs 環境を対象に考案された経験強化または環境同定の手法の POMDPs への適用可能性とその限界を明らかにすること, 2) マルチエージェント強化学習に適用可能な頑健な手法を確立すること, 3) 報酬がベクトルで与えられる場合の強化学習, 4) 強化学習手法の性能評価を行うためのベンチマークを整備すること, 5) 強化学習の応用領域を広げること, 6) 強化学習と進化的計算の融合を図ること, 7) 強化学習に基づく新しい制御論を構築すること, などを指摘して結びとする.

◇ 参 考 文 献 ◇

[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).

[Bradtke 94] Bradtke, S. J. and Duff, M. O. *Reinforcement Learning Method for Continuous-Time Markov Decision Problems*, Advances in Neural Information Processing Systems 7 (NIPS-94), pp.393–400 (1994).

[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).

[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*, Proc. of 10th National Conference on Artificial Intelligence, pp.208–214 (1992).

[Fiechter 94] Fiechter, C. N. *Efficient Reinforcement Learning*, Proc. of 7th Annual ACM Conference on Computationl Learning Theory, pp.88–97 (1994).

[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 artificial intelligence approach, Volume II. R. S. Michalski, J. G. Carbonell and T. M. Mitchell, eds., 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. eds., Academic Press (1987).

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

[北川 67] 北川 敏男編. 情報科学講座 $A \cdot 5 \cdot 1$「マルコフ過程」, 共立出版 (1967).

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

[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).

[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).

[宮崎 96] 宮崎 和光, 山村 雅幸, 小林 重信. ℓ-確実探査法:エージェントによる環境同定のための行動選択戦略～k-確実探査法の不確実性下への拡張～, 人工知能学会誌, Vol.11, No.5, pp.128–132 (1996).

[宮崎 97] 宮崎 和光, 山村 雅幸, 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).

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

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

[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).

[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).

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

[Sutton 90] 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).

[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).

[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).

著 者 紹 介

宮崎 和光(正会員), **小林 重信** (正会員) は, 前掲(Vol.12, No.1, p.89)参照.