数理モデル化と問題解決 22−10
（1998. 11. 27）

特徴構成法を用いた Q 学習の効率改善

宮本 行庸　上原 邦昭
神戸大学 工学部 情報知能工学科

本稿では，特徴構成法を用いた強化学習システム FCQL について述べる．従来の強化学習では，対象とする環境の各状態を識別する適切な属性が，学習の前段階であらかじめ準備されていることを仮定している．現実には，学習システムが状態を識別するのに充分な入力系を持っているとは限らず，領域に固有の特徴を適宜構成していく機能が必要とされる．本稿では，構成的帰納学習に用いられる特徴構成法を，強化学習の一手法である Q 学習と統合し，有限離散時間環境における適切な内部表現と評価関数を学習する手法を提案する．結果として，単位時間における期待報酬値を最大化するのみでなく，収束までに費やす状態数の大幅な削減が実現できた．

Improving the Effectiveness of Q-Learning by Using Feature Construction

Yukinobu Miyamoto　Kuniaki Uehara
Department of Computer and Systems Engineering,
Faculty of Engineering, Kobe University

In this paper, we describe a new reinforcement learning system called FCQL (Feature Constructive Q-Learning). Usually, reinforcement learning methods assume that they can identify each state before learning. In a real-world domain, the learner only has limited sensors, so is required the ability to construct new features. This paper describes an approach integrating feature construction with Q-learning to learn efficient internal state representation and a decision policy simultaneously in a finite, deterministic environment. The result shows that FCQL can not only maximize the long-term discounted reward per unit time, but also reduce the number of status to converge.

1　はじめに

近年，実環境における移動ロボットのように，未知なる環境下での学習機能を持ったシステムに感心が高まりつつある．このようなシステムの設計において，**強化学習** [5] が注目されており，中でも特に注目されている手法に Q **学習** [6] があげられる．

Q 学習は，有限離散 Markov 環境下において，充分な試行の後では最適解への収束が保証されている一方，学習完了までに必要とする状態が膨大になるという欠点を持っている．これは，Q 学習が過去の状態を参照する際に完全一致を要求するため，環境の全状態を探索するまで学習が収束しないことが原因である．また，Q 学習が収束するための条件として，状態を識別するのに充分な属性が与えられなければならないという仮定がある．このため，最初に決定する属性を慎重に選ぶ必要がある．逆に，状態を記述する属性を詳細にとりすぎると，状態の一致条件が厳しくなり，さらに収束が遅れるという問題が生じる．このような問題の解決策として，得られた状態から新たな特徴を作り出し，それらの特徴を用いて状態を評価する機能を Q 学習に持たせることが考えられる．

構成的帰納学習 [1][4] で提案された**特徴構成法**は，対象領域に適切な特徴を新たに作り出し，状態の記述を更新する手法である．特徴構成法は，分類などの概念学習において蓄積される状態数の削減と学習精度の向上などの成果が報告されている．本稿では，特徴構成の機能を持つ強化学習システム FCQL（Feature Constructive Q-Learning）を提案する．FCQL は Q 学習の枠組に特徴構成法を統合したシステムで，報酬を用いて状態を分類し，各クラスごとに共通の空間的な特徴を構成して，最適解への収束に必要な状態数の削減と，評価関数の早期収束を達成している．

2 特徴構成法

2.1 構成的帰納学習における特徴構成法

対象領域の状態集合を分類し，クラスごとの概念を獲得する手法に帰納学習がある．一般的な帰納学習は，概念記述が与えられた属性から選ばれるために，選択的帰納学習とも呼ばれる．選択的帰納学習で学習できない問題の例が DNF 問題で，学習したい概念が DNF （選言標準形： Disjunctive Normal Form）で記述される．DNF 問題の例である三目並べ問題（図1）は,「三目並べの終了時における X の勝ち」という概念を学習する問題で，学習したい概念に含まれる例を正例，それ以外を負例と呼ぶ．

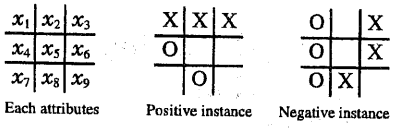

図 1: 三目並べ問題

ここで，正例の概念が $(x_1$=X) であると仮定して，選択的帰納学習による分割を考えると，この概念は正例中にも負例中にも一様に存在しているため，学習は失敗する．これは，三目並べ問題での目標概念が $(x_1$=X) $\wedge$ $(x_2$=X) $\wedge$ $(x_3$=X) $\vee$ $(x_4$=X) $\wedge$ $(x_5$=X) $\wedge$ $(x_6$=X) $\vee \cdots$ のような DNF で記述され，単項の属性では分割できないことが原因である．

構成的帰納学習では，状態から特徴が構成され，一定の基準を満たせば選択される．つまり，構成的帰納学習には，選択的帰納学習に加えて,「特徴の構成」と「特徴の選択」の機能が追加されている．特徴の構成は，学習した概念に矛盾が発生した場合[1]，選択できる概念の候補がなくなった場合などに行われる．

特徴構成法において議論される点に，特徴構成の手続きが組み込み型か前処理型かという点がある．組み込み型は学習中に特徴構成を行うことができるが，各アルゴリズムごとに特化した実装をする必要がある．一方，前処理型は学習前に特徴構成を行うため，他の学習アルゴリズムのフィルタとして用いることができるが，学習中には特徴を構成できない欠点がある．Q 学習では逐次的に状態に遭遇するため，FCQL では組み込み型を採用している．さらに，特徴に基づいた行動決定と，特徴の評価値の更新のために，FCQL では各特徴と行動の対に評価値を与える関数を定義している．本稿では，Q 学習では非効率な問題の例として DNF 問題を取り上げる．以下では属性—値対の連言のことを特徴と呼ぶ．

[1]正例ばかりで過剰一般化となる場合など．

2.2 FCQL における特徴構成法

2.2.1 特徴の構成

帰納学習では，状態は複数の属性—値対と，その状態の属するクラスで記述されている．FCQL では，クラスを報酬を用いて定義しており，時刻 t での状態を s_t ，報酬を r_t とすると，s_t は複数の属性—値対，r_t がクラスに相当する部分となる．各クラスの報酬が既知であるとき，s_t は報酬が r_t に最も近いクラスに分類される．また，FCQL で扱う特徴は，属性—値対の連言で記述され，2 値をとる．2.1 節の例では，状態 s_t に対し，特徴 $f_1 = (x_1$=X) $\wedge$ $(x_2$=X) = True となることは，s_t に $(x_1$=X) と $(x_2$=X) という属性—値対が含まれていることを意味している．

FCQL の特徴構成では，まず，得られた状態の属性—値対について，すべての 2 項連言が構成される．このとき，すでに構成された特徴が存在すれば，それらの特徴と属性との連言についても可能な組合せが構成される．三目並べの例では，特徴 f_1 が存在しているとき，特徴 $f_2 = (f_1 = \text{True}) \wedge (x_3 = \text{X}) = (x_1 = \text{X}) \wedge (x_2 = \text{X}) \wedge (x_3 = \text{X})$ という，属性の 3 項以上の連言も組み合せも可能となる．状態の属性数を n とすると，特徴を構成する項数は最大で n ，一つの状態から構成される特徴数は，最大で $\sum_{i=1}^{n} {}_n\mathrm{C}_i$ となり，全特徴数はこの状態数倍が上限となる．

2.2.2 特徴の選択

構成された特徴の候補は，何の評価も受けていないため，その中から適切な特徴を選択する必要がある．たとえば，2.2.1 項の三目並べでの特徴 f_1 を用いて，特徴 $f_3 = (f_1 = \text{True}) \wedge (x_4 = \text{O}) = (x_1 = \text{X}) \wedge (x_2 = \text{X}) \wedge (x_4 = \text{O})$ といった特徴も構成できるが，この特徴は三目並べにおける正例の条件[2] を満たさず，学習には不適切な特徴である．FCQL では，特徴の採用に評価指標を設け，特徴の候補の中から最適な特徴を選択している．特徴選択のための評価指標として，FCQL では**利得比基準** [3] を採用する．

FCQL では，構成された特徴の候補の中で利得比が最大の特徴を 1 つ選択し，新しい特徴としている．利得比が大きな特徴は，その特徴を用いて状態集合を分割したときに，同じクラスの状態が集まりやすいことを意味している．利得比が同じ特徴が複数個

[2]X が一列に並んでいる状態．

存在する場合は，連言を構成する項数が少ない特徴を選択している．最後に，この特徴がすでに採用された特徴と重複しなければ，新たな特徴として採用する．採用された特徴は**特徴集合**と呼ばれる領域に追加される．この特徴集合中にある j 番目の特徴 f_j に対し，状態 s_t において $f_j = \mathrm{True}$ となるとき，特徴 f_j は**状態 s_t に含まれる**といい，$f_j \in s_t$ と表記される．このとき採用された特徴を f_j，特徴構成の直前にとった行動を a_t，得られた報酬を r_t とすると，f_j と a_i の対に対して評価値を与える関数 F を**特徴関数**と呼び，$F(f_j, a_i)$ の初期値を

$$F(f_j, a_i) = \begin{cases} r_t & (i = t) \\ 0 & (i \neq t) \end{cases} \quad (1)$$

と定義する．特徴関数は，特徴の評価値の更新，および特徴の淘汰に用いられる．

2.2.3 特徴関数の更新

2.2.2 項で得られた特徴を追加していくのみでは，特徴が増えすぎてしまい，構成の時点ではよいと判断された特徴も，学習が進むにつれ不適切になっていく可能性もある．この現象を解消するために，選択された特徴の評価値を更新し，不要な特徴を淘汰する手続きが必要となる．FCQL では，この手続きを行う評価基準に特徴関数を用いており，ある特徴 f_j の特徴関数の値 $F(f_j, a_i)$ は，以下の式に基づいて更新される（α は学習率，γ は割引率）．

$$\begin{aligned} F(f_j, a_i) \leftarrow\ & F(f_j, a_i) + \alpha(r_t \\ & + \gamma \max_{g \in s_{t+1}, b \in A} F(g, b) \\ & - F(f_j, a_i)) \end{aligned} \quad (2)$$

すべての特徴は，0 でない報酬が得られた状態から構成されており，試行が進むにつれ F 値は実際の報酬に近づく．誤って構成された特徴は，以降の試行で報酬を得られることが少なく，F 値は 0 に近づく．更新の結果，すべての行動について報酬のクラスが 0 になった特徴は特徴集合から削除される．

2.2.4 強化学習から特徴構成へのフィードバック

2.2.3 項で更新された特徴関数を用いて，特徴構成の際に棄却された特徴の評価値を更新することを考える．2.2.1 項で構成された特徴のうち，すでに特徴集合中に存在するために棄却された特徴は，式 (2) に従って評価値が更新される．以上の操作を強化学習側から特徴構成へのフィードバックとし，強化された特徴関数を更新に用いた結果，既存の特徴の評価値をより早く正確な値に近づけることができる．

2.2.5 状態集合の大きさに関する考察

利得比基準による評価は，状態数に依存する．充分な数の状態集合が得られたとき，対象領域の表現に充分な数の特徴を得られるが，学習初期には経験した状態数が少ない．利得比基準による評価は小さな状態集合に対しては安定せず [3]，必ずしも正当な評価がなされるとは限らない．不適切な特徴が学習初期に不適切な行動を招き，適切な状態の蓄積による Q 関数拡張の妨げになっていると考えられる．この現象は，充分な量の状態を蓄積しないうちに特徴構成を始めた点に問題があり，蓄積された状態数に基づいて特徴構成を始める時点を考慮する必要がある．

解決策として，充分な量の状態を得てから特徴構成を始めることが考えられるが，「充分な」量を定量的に示す指標を作ることは困難である．また，充分な量の状態を得るまで待つと，試行全体の速度に影響が出る可能性もある．FCQL では，最初に特徴構成をしない Q 学習を行い，一定数の状態を蓄積した後，特徴構成を行う手続きに切り替える．このとき，学習アルゴリズムを Q 学習から FCQL に切替えるまでの状態を蓄積する過程をディレイと呼ぶ．

3 特徴構成法を用いた Q 学習

3.1 対象とする学習領域

FCQL が対象とする学習領域は，環境が有限離散 Markov 決定過程でモデル化される．時刻 t でのシステムの入出力は，入力を状態 s_t と報酬 r_t，出力を行動 a_t とする．状態 s_t は離散属性で記述され，報酬 r_t はあらかじめ定義されたクラスのいずれかに属するものとする．行動 a_t は，あらかじめ定義された行動集合 A のうち，s_t において可能な行動から選択される．学習の目的は，長期にわたる割引報酬和の最大化にあり，単位行動あたりの報酬で評価される．

また，FCQL が対象とする問題領域は，行動に伴う報酬が即時に得られると仮定している．このため，即時的な報酬に無関係な状態は学習の対象とされず，即時に報酬が得られる状態から学習が行われる．このような行動の獲得は**反射的行動獲得**と呼ばれ，FCQL では反射的行動獲得を行う．

3.2 FCQL アルゴリズム

FCQL では，現在の状態の入力，行動選択，行動の実行と報酬の獲得，学習と特徴構成による内部状態の更新までの一連の手続きを一周期とする．図2に FCQL アルゴリズムを示す．

MAIN LOOP:

1. 現在の状態 s_t を入力する．
2. $a_t \leftarrow$ **Policy**(s_t).
3. a_t を実行し，次の状態 s_{t+1} に遷移し，報酬 r_t を獲得する．
4. **Learn**(s_t, a_t, r_t, s_{t+1}).
5. 1. に戻る．

Policy(s_t):
以下に示す優先順位で，いずれかの処理が選択される．

1. s_t に含まれる特徴が存在すれば，特徴に基づく行動選択を行う．
2. s_t と同一の状態が Q 表に存在すれば，状態に基づく行動選択を行う．
3. ランダムに行動を選択する．

Learn(s_t, a_t, r_t, s_{t+1}):
以下に示す優先順位で，いずれかの処理が選択される．

1. 特徴に基づく行動選択を行った場合，特徴関数を更新する．
2. 状態に基づく行動選択を行った場合，Q 関数を更新する．
3. $r_t \neq 0$ の場合，新たに特徴を構成する．
4. 内部状態の更新を行わない．

図 2: FCQL アルゴリズム

行動選択手続き **Policy** では，まず s_t に含まれる特徴を判定する．特徴集合中の j 番目の特徴について，f_j = True となる特徴を検出し，この操作を特徴集合中のすべての特徴について行う．s_t に含まれる特徴が存在する場合は，経験済の状態と同様の特徴をもつ状態であると判断され，各行動の評価値によって Boltzmann 分布に基づく確率選択が行われる．s_t に含まれる特徴が複数のときは，各特徴の報酬和によって Boltzmann 分布に基づく確率選択が行われる．すなわち，状態 s_t に含まれる特徴 f_j に基づく行動 a_i を選択する確率 $p(a_i|f_j \cup f_j|s_t)$ は，

$$p(a_i|f_j \cup f_j|s_t) = \frac{e^{F(f_j,a_i)/T}}{\sum_{f_l \in s_t} \sum_{a_k \in A} e^{F(f_l,a_k)/T}} \quad (3)$$

となる（T は温度定数）．このような選択方式を**特徴に基づく行動選択**と呼ぶ．s_t に含まれる特徴がない場合は，学習者は s_t を Q 表と照合し，s_t と一致する状態が Q 表内にあれば，Boltzmann 分布に基づく確率選択を行う．すなわち，状態 s_t で行動 a_i を選択する確率 $p(a_i|s_t)$ は，

$$p(a_i|s_t) = \frac{e^{Q(s_t,a_i)/T}}{\sum_{a_k \in A} e^{Q(s_t,a_k)/T}} \quad (4)$$

となる．この選択方式を**状態に基づく行動選択**と呼ぶ．上記のいずれにも該当しない場合，s_t において可能な行動の中からランダムに行動が選択される．

学習手続き **Learn** では，a_t の実行で遷移した状態 s_{t+1} と得られた r_t を用いて内部状態を更新する．このとき，行動選択手続きでの戦略により処理が分かれる．特徴に基づく行動選択が行われた場合，式 (2) に基づいて特徴関数を更新する．状態に基づく行動選択が行われた場合は，以下の式に基づいて Q 関数を更新する．

$$\begin{aligned} Q(s_t,a_t) \leftarrow\ & Q(s_t,a_t) + \alpha(r_t \\ & +\gamma \max_{b \in A} Q(s_{t+1},b) \\ & -Q(s_t,a_t)) \end{aligned} \quad (5)$$

上記のいずれにも該当しない場合は，r_t によってさらに処理が分かれる．$r_t \neq 0$ のときは，報酬が得られる特徴があると判断できるため，2.2 節の手法で特徴の構成と選択を行う．また，s_t を Q 表に追加し，各行動に対する Q 値の初期値を

$$Q(s_t,a_i) = \begin{cases} r_t & (i = t) \\ 0 & (i \neq t) \end{cases} \quad (6)$$

と定義する．$r_t = 0$ のときには，関数の更新および特徴構成のいずれの処理も行われない．

4 FCQL による学習例

4.1 問題設定

本章では，人工的な迷路問題[2] に対し FCQL を用いてシミュレーションを行い，その結果について述べる．対象となる問題領域を採用した理由として，

- 有限離散 Markov 環境である．
- 構成したい特徴が DNF で記述される．

といった点があげられる．なお，対象とする問題領域は，図3 に示すような 2 次元の迷路を想定する．迷路は広さが 7 × 7 ブロック の格子状の環境で，この環境内には 5 種類の物体が存在する．図3 で矢頭型の物体が学習者の位置を表している．小さな点が

餌で，学習者が重なると +0.5 の報酬を得る．大きな円が敵で，出現地で静止しており，学習者が重なると −1.0 の報酬を得て，その試行は終了する．黒い矩形状の物体は障害物で，さらに先に障害物がある場合は，学習者はその方向へ進めない．それ以外の場合は，障害物の向うにある物体を押し潰しながら障害物とともに1ブロック進む．このとき，敵を押し潰すと +1.0 の報酬を得る．白い矩形状の領域は空白で，学習者はこの領域に進める．入力属性は，学習者の位置，および学習者の周囲 4 方向 × 距離 2 ブロック，合計で 9 属性である．各試行開始時の学習者，すべての敵，障害物，餌の出現地点は，順に空白の中からランダムに選ばれる．各物体の数は敵が3，壁が15，餌が10とする．

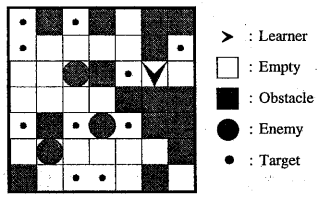

図 3: シミュレーション環境

学習者が行う状態の入力，行動選択，行動の実行と報酬の獲得，内部状態の更新までの一連の手続きを**1 ステップ**とし，学習者が敵に重なるか，あるいは100 ステップが経過するまでの一連のステップ群を**1 試行**とする．シミュレーションは最大 100 ステップの試行を 2000 回繰り返し，以上の作業 5 回の平均を取り，単位試行あたりの報酬値と蓄積されている状態数を評価する．アルゴリズム中の各定数は $\alpha = 0.9$，$\gamma = 0.9$，$T = 0.4$ とする．

この領域の特徴は，学習者の前後左右いずれか 1 ブロックに餌があるという単項の属性で記述される特徴が 4 種類，同様に 1 ブロックが敵である特徴が 4 種類，学習者の前後左右いずれかが壁で，壁の 1 ブロック先が敵であるという 2 項連言の特徴が 4 種類，計 12 種類である．図4 に，特徴の正解を示す．上段の f が対象領域の特徴で，学習者の周囲 4 方向にある物体を示している．下段の a/r が特徴 f に基づく行動と報酬で，矢印はその方向へ進むことを表している．各報酬ごとに一つのクラスとすると，これらの特徴は各クラスごとの DNF で記述できる．

4.2 シミュレーション結果

FCQL を用いたシミュレーション結果を示す．比較対象に，正解の特徴を最初から与えた FCQL，および Q 学習を採りあげ，学習効率の変化を評価する．本シミュレーションでは，正解の特徴を最初から与えた FCQL が収束する値を理論値と見なしている．評価対象は，学習中に蓄積された状態数と単位試行あたりの平均報酬値の推移とする．本シミュレーションでは特徴構成を始める状態数を 400 個と定める．なお，ディレイをかける状態数については，5 章の問題点で再び検討する．図5に状態数の推移，図6 に平均報酬値の推移を示す．

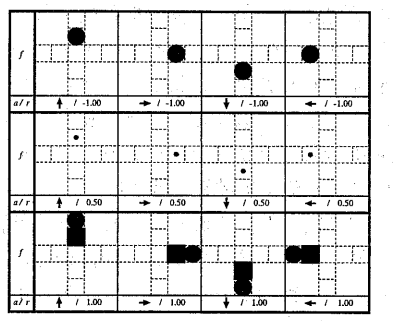

図 4: 問題領域に固有の特徴

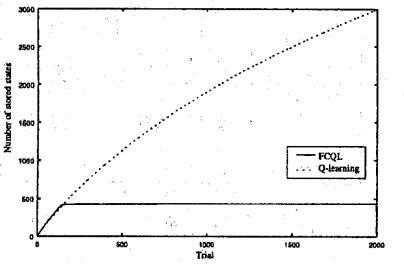

図 5: 蓄積状態数の推移

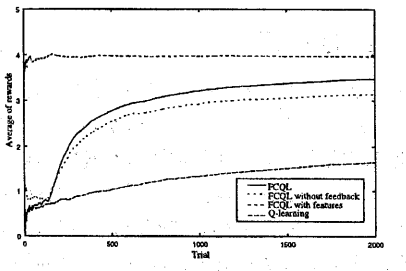

図 6: 平均報酬値の推移

図5より，蓄積状態数は Q 学習よりかなり少ない数で安定している．これは，一括して最初に蓄積した 400 個の状態が，対象領域の性質をよく反映したものであると考えられる．よって，ディレイをかけている間に蓄積された状態をもとに特徴を構成すると，

以降の学習効率の向上に多大な貢献があったものと推測される．図6より，FCQL は学習初期に急激な傾斜を描いて最適値へと収束し，その後も安定している．Q 学習と比較すると，学習初期の速度，および収束値の双方で改善が見られる．ディレイによる遅れは学習全体に影響を及ぼさなかったと考えられ，学習効率の向上が見られた．また，強化学習からのフィードバックを除いた場合，特徴関数の収束が遅れるため，収束値が若干劣る結果が表れている．

次に，学習中に構成された特徴の例を 図7 に示す．この例では，実際に構成された特徴は 28 個で，そのうち12 個を構成された順に示している．正解の特徴のうち，2 項連言の特徴は 4 つすべてが構成されていた．単項の特徴は，他の不要な属性が混在し，冗長な記述となっているが，報酬と無関係な特徴は一つしか構成されておらず（図7の斜線部），蓄積状態の分類には充分であったと考えられる．構成された特徴の連言の項数はほとんどが 2 項までで，最長で 3 項のものがごくまれに見られる程度であった．

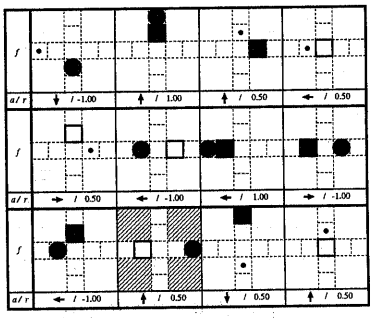

図 7: 構成された特徴の一部

5 おわりに

本稿では，FCQL を人工的な迷路問題に適用し，シミュレーションを行った．環境の例として採用した迷路問題は，ロボット学習の範疇で用いられる問題の一つであるが，現実のロボット学習は連続値を扱うため，FCQL の枠組に組み込むには適切な方法で離散化する必要がある．本稿での迷路問題はすでに離散化された状態にあり，連続空間を対象とする問題に対しては，適切な離散化を行ったのちに対応が可能であると考えられる．また FCQL では，特徴を構成する操作子として論理和を採用したが，他にも特徴を構成する操作子に発見的な手法の採用が考えられ，この点についても検討の余地がある．対象領域とした DNF 問題は，Q 学習のみでは非効率な問題の例として採用したが，最終的には入力属性の組合わせで特徴を構成できるという前提に立っている．状態を識別する要素が，センサからの情報に全く現れてこない，いわゆる隠れ状態を扱う問題では，得られた状態からの特徴構成ができないため，他の状態再構成の手法が有利となる場合がある．

FCQL の問題点として，ディレイをかける状態数を試行中に決定できない点があげられる．図8 は，状態数を 0, 100, 200, 300, 400, 500, 700 に設定した場合の報酬の推移を示している．

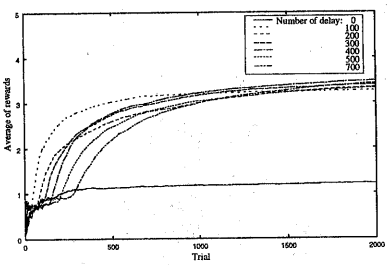

図 8: 各ディレイにおける報酬の推移

状態数が少ない場合は特徴の精度が低く，不安定な振る舞いを示している．逆に，状態数を増やすと，学習初期における報酬の向上が遅くなる．状態数が 400 のとき，2000 試行終了時の収束値が最も高いので，本稿では状態数を 400 個と決定した．今後は，特徴構成を始めるために充分な状態数を決定するための理論的な指標を検討する必要がある．

参考文献

[1] Aha, D. W.: Incremental Constructive Induction: An Instance-Based Approach, *Proc. of the Eighth International Workshop on Machine Learning*, pp. 117–121 (1991).

[2] Maclin, R. and Shavlik, J. W.: Incorporating Advice into Agents that Learn from Reinforcements, *Proc. of AAAI-1994*, pp. 694–699 (1994).

[3] Quinlan, J. R.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann (1993).

[4] 滝 寛和: 構成的帰納学習とバイアス, 人工知能学会誌, Vol. 9, No. 6, pp. 818–822 (1994).

[5] 畝見達夫: 強化学習, 人工知能学会誌, Vol. 9, No. 4, pp. 830–836 (1994).

[6] Watkins, C. J. C. H: Learning from Delayed Rewards, *Ph.D thesis, Cambridge University Psychology Department* (1989).