Undiscounted Prioritized Sweeping：最適政策の優先順序付け強化学習の効率化手法

山口 智浩*　　天正 新二郎†

Undiscounted Prioritized Sweeping：
An efficient reinforcement learning method added priority for optimal policy

Tomohiro YAMAGUCHI and Shinjiro TENSYO

This paper presents a new reinforcement learning method that of an average reward method based Prioritized Sweeping. Most reinforcement learning methods optimize the discounted total reward received in each rule by an agent, called the discounted optimization method. Prioritized sweeping that calculates utility to priority is one of efficient discounted methods. However, it is hard to decide optimal discount factor, and calculation cost increases when the number of rules of an environment increases. To solve them, LC-learning has been presented as one of the undiscounted optimal method. This method's utility is calculated as the sum of rewards per distance from the reward, so it is also called average reward method. To improve LC-learning, we present Undiscounted Prioritized Sweeping. We investigate efficiency of our method by experiment in MDP model environment.

1　はじめに

人工知能の主要な研究分野のひとつに強化学習(Reinforcement Learning: RLと略す)[1]がある．強化学習は，環境からの知覚入力を得て状態(State)を同定し，エージェントの取り得る行動(Action)を決定し，行動を起こすことによって環境中から得られる報酬(Reward：Rwと略す)を基に各状態での適切な行動を強化する，という動作を行う．強化学習の目的は，各状態における最良の行動の集合である最適政策(optimal policy)を決定することである．

多くの強化学習では，最適政策決定の際に効用値(utility)を学習基準として用いる．従来手法では，効用値の算出に，将来獲得する報酬を割り引いて見積もる割引報酬和を用いた割引型最適化手法による政策反復アルゴリズム(Policy Iteration Algorithm: PIA)[3]が代表手法である．PIAは，環境中の全ての状態の効用値を同期的に更新する．しかし，取り扱う環境が大きくなるほど計算コストが増加するという問題点がある．

この問題を，効用値の計算順序という切り口から改善する手法として，優先掃き出し法(Prioritized Sweeping: PriSwpと略す)[4]がある．PriSwpは，各状態の効用値を計算する際に，状態間の依存関係を考慮した順序で非同期的に効用値を更新する．これにより，PIAと比べ計算コストが削減される．しかしながら，効用値の計算に用いる割引率の合理的な値決定方法がない，という割引型最適化手法の問題点が残る．

この割引型最適化手法の問題点を改善する手法として，我々はLC-learning[7, 8, 9, 10]を提案した．LC-learningは，効用値として環境から得られる報酬の期待値と，報酬からの期待距離とを用いて平均報酬を最大化する非割引型最適化手法である．LC-learningは，環境の状態遷移の依存関係を利用して探索することで，最適政策の検出を容易にする．しかしながら，LC-learningは探索の優先順序を考慮せず，横型探索を行うので改善の余地がある．

そこで本稿では，PriSwpとLC-learningの2つの手法の特徴を組み合わせた，非割引型優先掃き出し法(Undiscounted Prioritized Sweeping: U-PriSwpと略す)を提案[11]し，実験により提案手法の有効性を検討する．

* 情報工学科
† 専攻科 電子情報工学専攻

注1) 第16回人工知能学会全国大会
2002年5月30日，口頭発表論文を加筆修正

2　本研究の基礎理論

本稿で取り扱うモデルに基づく強化学習法(model based RL)は，最適政策の学習を，1)観測による環境モデルの同定，2)モデル上での最適政策の探索，の2段階に分割して行う点が特徴である．今回は，環境モデルの情報は既知である，と仮定し，上記2)に焦点を絞る．

本研究で用いる環境モデルは，マルコフ決定過程(Malkov Decision Process: MDP)[1, 2]である．MDPは，1)(単純)マルコフ性(現在の行動決定が過去の過程に依存しない)，2)定常性(各状態間の遷移確率が時間経過によって変動しない)，という2点を満たすモデルである．本稿は，対象問題を決定的環境のMDPモデルに限定して行う．

3　割引型最適化手法

3.1　政策反復アルゴリズム (PIA)

従来の強化学習において，最適政策を求める方法のひとつの切り口として動的計画法に基づく割引型最適化手法があり，代表的な手法が(割引型の)政策反復アルゴリズム(PIA)[1, 3]である．PIAは各状態の効用値を同期的に更新し，これを網羅的に繰り返して最大値に収束させる最適政策を求める手法である．

割引型のPIAは，各状態の割引期待報酬和を効用値として更新を行う．状態iにおける効用値$U(i)$，及び状態iにおける最適政策$\pi^*(i)$は以下の式(1)，(2)で表される[1].

$$U(i) \leftarrow \max_a \sum_j P_{ij}^a \cdot (Rw_{ij}^a + \gamma \cdot U(j)) \tag{1}$$

$$\pi^*(i) = arg\max_a \sum_j P_{ij}^a \cdot (Rw_{ij}^a + \gamma \cdot U(j)) \tag{2}$$

P_{ij}^a：現状態iで行動aを行ったときの次状態jへの遷移確率，
Rw_{ij}^a：状態iで行動aを行い状態jへの遷移で得られる直接報酬，
γ：割引率，U(j)：次状態jの効用値

割引型PIAの学習のオーダは，以下の式(3)で表される．

$$O = n \cdot |S| \cdot |A| \tag{3}$$

n：効用値収束までの(各ルールの)繰り返し計算回数，
$|S|$：全状態数，$|A|$：全行動数

割引型のPIAは全状態の効用値を同期的に繰り返し更新するため，以下2点の問題が起こる．1)学習環境の規模が大きくなるほど，$|S| \cdot |A|$(全ルール数)が大きくなり，計算量が増加する．2)$|S| \cdot |A|$が増加すると，それに伴って収束までの繰り返し計算回数nも増加する．上記2点の問題は次元の呪いと呼ばれる．次節では，上記2)の問題点を抑制する割引型最適化手法として，優先掃き出し法を説明する．

3.2　優先掃き出し法 (PriSwp)

優先掃き出し法(PriSwp)[4]は，各状態の効用値を非同期的に更新し，各状態間の効用値算出の依存関係を利用した計算の優先順序付けによって，PIAの計算コストの削減を図る割引型最適化手法である．ここでPriSwpは，効用値更新の変化量が大きい状態に遷移する状態の更新の優先度を大きくし，順序付けて更新を行う．これにより，前述のPIAの問題点である，効用値の収束に要する計算コストを近似的に下げ，最適政策の決定に要する反復計算回数の減少を図る．以下に優先度$Pri(i')$を求める式を示す．

$$Pri(i') \leftarrow P_{i'i}^{a'} \cdot \Delta U(i) \tag{4}$$

$$\Delta U(i) \leftarrow |U_t(i) - U_{t-1}(i)| \tag{5}$$

$Pri(i')$：(現状態iに遷移可能な)前状態i'の優先度，
$P_{i'i}^{a'}$：前状態i'で行動a'を取ったときの現状態iへの遷移確率，
$\Delta U(i)$：状態iにおける更新前後の新旧効用値の差分の絶対値，
$U_t(i)$：更新後の現状態iの効用値，
$U_{t-1}(i)$：更新前の現状態iの効用値

図1にPriSwpのアルゴリズムを示す．

PriSwpの学習のオーダは，PIAと同じだが，PriSwpは優先順序を付けて非同期に効用値を更新することから，PIAと比べて効用値収束までの繰り返し計算回数nを抑制出来る．

3.3　割引型最適化手法の問題点

割引型最適化手法の特徴は，効用値の定義に割引率γを用いることである．割引率は，将来に渡る無限の報酬和を現在から遠ざかる報酬ほど割引くことで，報酬の有限和(割引期待報酬和)へと効用値を収束させる．

しかしながら，割引型PIAやPriSwpなどで用いる割引率には合理的な値決定法がなく，しかも学習の質と速度が相反する[6]ので，取り扱う問題環境ごとに設計者がその都度最適な値を決定するという第1の問題点がある．第2の問題点として，取り扱う環境の状態行動集合が大きくなるほど，収束までの各状態の効用値計算量が増加する，という次元の呪いが発生する．

これらを改善するために，1)割引率を用いずに，環境

Prioritized Sweeping Algorithm

step1 現在の State を知覚し，i に代入

step2 効用値の更新と政策の改善

$U(i) \leftarrow \max_a \sum_j P_{ij}^a \cdot (Rw_{ij}^a + \gamma \cdot U(j))$
$\pi^*(i) \leftarrow arg\max_a U(i)$

step3 優先度の決定

$Pri(i') \leftarrow P_{i'i}^{a'} \cdot \Delta U(i)$
$\Delta U(i) \leftarrow |U_t(i) - U_{t-1}(i)|$

step4 優先キューへの挿入・再構成

- 優先度 $Pri(i')$ がある閾値以下の場合，優先キューには挿入しない.
- 同じ状態の優先度がキュー内に存在する場合，優先度の大きい方のみをキューに残す.
- 優先度の大きい順にソートする.

step5 次に計算する State の選択

- キューの先頭の State を i に代入し，キューの先頭の優先度をキューから省き step2 へ戻る.
- 優先キューが空になるまで，step2 に戻る.
- 優先キューが空になればアルゴリズム終了.

図1：PriSwp アルゴリズム

から得られる値だけを用いる．2)次元の呪いを解消するため，各状態の効用値計算を有限回に抑制し，かつ最適政策の算出を保証する．以上2点を満たす手法として，本稿では非割引型最適化手法に着目する．

4 非割引型（平均報酬）最適化手法

4.1 効用値と最適政策の定義

割引率を用いない非割引型最適化手法の代表手法として，効用値を平均報酬 ρ で表す平均報酬法[5]がある．平均報酬 ρ の定義を式(6)に示す．

$$\rho = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{t-1} R_t \qquad (6)$$

N：総step数，R_t：時間tで得られる報酬値

本稿では，環境からの獲得報酬値と，報酬からの距離(step数)の2つのパラメータのみを基に ρ を決定する[6].

PIAに基づく手法は，全ての状態における最良な行動を反復計算で決定する．これに対し，次節で説明するLC-learningでは，対象となるMDPモデル上で，1)最適cycleの決定，2)最適cycleに合流するpathの最適化，と2段階に分けて効率良く最適政策を求めるのが特徴である．ここで，1)はgain-optimal，2)はbias-optimalと呼ばれる[5]．最適cycle，最適pathに関しては，次節以降で詳しく述べる．

4.2 LC-learning

LC-learning[7, 8, 9, 10]は，割引型最適化手法の問題点を改善し，解の最適性を保証する平均報酬手法である．LC-learningの動作の手順は，1)最適cycleの決定，2)最適cycleに合流するpathの最適化，の2つに分けられる．前者は，環境中の報酬獲得cycleを全て検出し，最も平均報酬値の大きいcycleを最適cycleとして決定する．これにより，LC-learningは式(6)における平均報酬 ρ のlimを取る必要がなくなり，各行動ルールの平均報酬値が有限回の計算で最適政策が決定できる．後者は，最適cycle中に含まれない状態においての最適cycleへの最適pathを求める．この2つの手順により環境中の最適政策を効率良く決定する．LC-learningは，MDPモデルにおける行動ルールを順方向(前向き)にtree状に展開(以下，tree展開と記述)し，(平均報酬値最大になる)cycle及びpathの検出を行う．図2にMDPモデルのtree展開の例を示す．

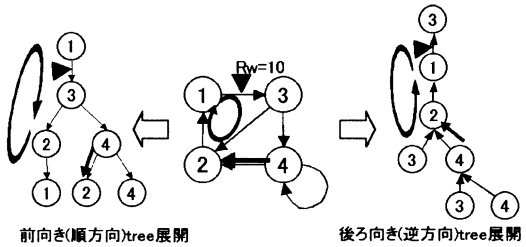


図2：MDPモデルのtree展開：最適cycleと最適path

LC-learningは，MDPモデルのtree展開において，各Rwの存在する行動ルール(以下，報酬ルールと記述)をtree展開の根として用いる．このとき，cycleに含まれる報酬ルールの数だけ同じcycleを再度検出してしまうという問題が生じる．これを防ぐために，LC-learningは第1の枝刈りとして，報酬ルールの順序付けによるcycleの多重検知防止(一度検出した報酬ルールや根の報酬ルールを検出した場合，以降を展開しない)を行い，計算コストの削減を図る．LC-learningは第2の枝刈りとして，展開するpathの平均報酬の上限値予測を行い，検出済みのcycleの最大平均報酬を下回るpathの展開を打ち切る．この2点のtree展開の枝刈りを行うことにより，計算コストの削減を図る．LC-learningの学習のオーダについて以下に述べる．1)報酬数(rとする)だけtreeを

展開する．2)割引型最適化手法と比べ，各状態における行動ルールを1度展開すればよいことから，繰り返し回数nは考慮しなくてもよい．3)2種類の枝刈りを行うことから，最適なcycleに含まれない行動ルールはtree展開されずに展開を省略されることがある．以上3点より，LC-learningの学習のオーダは，最悪の場合で，$O = r \cdot |S| \cdot |A|$であるといえる．但し，このオーダは対象を決定的環境に限定した場合のものである．

しかしながら，LC-learningの問題点として，最適cycleの検出stepにおいて，tree展開の順序を考慮していないので，取り組む環境によっては最適cycleの検出が遅くなることがある．この問題を踏まえて，より効率的な手法として，次節では非割引型優先掃き出し法を提案する．

4.3 非割引型優先掃き出し法 (U-PriSwp)

本稿で提案する，非割引型優先掃き出し法(U-PriSwp)[11]は，PriSwpとLC-learningの2つのアルゴリズムを基にした，非割引型最適化手法である．U-PriSwpの特徴は，1)平均報酬を用いて最適政策(cycle)を検出すること，2)tree展開を幅優先で行うこと，3)tree展開に優先順序を付けること，4)環境のtree展開はRwを根として逆方向(後ろ向き)に展開すること，以上4点である．1)，2)はLC-learningと共通，3)，4)はPriSwpと共通である．

図3にU-PriSwpのアルゴリズムを示す．U-PriSwpの学習のオーダは，基本的にはLC-learningと同等である．但し，現在のU-PriSwpはLC-learningにおける2種類の枝刈りを実装していないこと，優先順序を計算するコストがかかること，を考慮する必要がある．

5 実験

PriSwp，LC-learningとU-PriSwpの3手法について，2種類の比較学習実験を行い，提案手法の有効な点，また不利な点などの特徴を明らかにする．以下の実験において，PriSwpにおける割引率γは，一般的かつ(今回対象とする環境で)最適解の保証される$\gamma = 0.9$を用いた．

5.1 実験1：大きさの異なる環境での比較学習実験

実験1の目的は，格子状態のMDPモデルを対象問題とし，各手法の基本的な特徴を明らかにするとともに，非割引型最適化手法，及び提案手法の有用性を実験的に検証することである．

実験1では，図4に示すMDPモデルを拡張した，5種類のサイズ(状態・行動集合)の異なる環境を対象とし，各手法の効用値の総更新回数，及び最適政策(cycle)検出時期の比較を行う．各環境中に存在するRw数は2である．ここで効用値の総更新回数とは，PriSwpでは全状態の効用値が収束するまでに計算する各状態における

Undiscounted Prioritized Sweeping Algorithm

step1 環境中のRwの存在するState/Actionの探索

step2 平均報酬値の更新

$$\rho(start\_Rw, state, action, comb\_of\_Rw) = \frac{sum\_of\_Rw(succs\_s, succs\_a) + Rw(state, action)}{dist(succs\_s, succs\_a) + 1}$$

$\rho()$：行動ルールの平均報酬値，
$start\_Rw$：tree展開の根 Rw，
$comb\_of\_Rw$：獲得したRwの組み合わせ，
$succs\_$：遷移先の(現状態から遷移可能な)，
$dist()$：$start_Rw$からの距離(行動ルール数)

- $\rho()$の4つのパラメータが等しく，過去の$\rho()$よりも大きい場合，$\rho()$を更新し，step3へ進む．それ以外は$\rho()$の更新を行わずにstep5へ．

step3 優先度の決定

$$Pri(start\_Rw, preds\_s, preds\_a, comb\_of\_Rw) = \rho(start\_Rw, state, action, comb\_of\_Rw)$$

$Pri()$：ルール(行動)の優先度，
$preds\_$：遷移元の(現状態へ遷移可能な)

step4 優先キューへの挿入・再構成

- $start_Rw$の存在するルールの優先度が計算された場合，優先キューへ挿入しない．ここでcycle検出とし，ρをcycleの平均報酬値として保存する．
- $Pri()$の4つのパラメータが等しいルールの優先度がキュー内に存在する場合，優先度の大きい方をキューに残す．
- 優先度の大きい順にソートする．

step5 次に計算するState/Actionの選択

- キューの先頭のState/Actionを$state, action$に代入し，step2へ戻る．このとき，キューの先頭の優先度をキューから省く．
- 優先キューが空になるまで，step2に戻る．
- 優先キューが空になれば，次のRwを$start_Rw$としてstep2から繰り返す．
- すべてのRwについて展開が終わった時点でアルゴリズム終了．

図3：U-PriSwpアルゴリズム

る効用値の総更新回数，LC-learningとU-PriSwpではtree展開回数，すなわち各行動ルールにおける平均報酬値の総更新回数，をそれぞれ表す．最適政策(cycle)検出時期とは，PriSwpでは全状態の最適政策が変動しなくなった時点での効用値更新回数，LC-learningとU-PriSwpでは最適cycleが検出された時点でのtree展開回数，をそれぞれ表す．

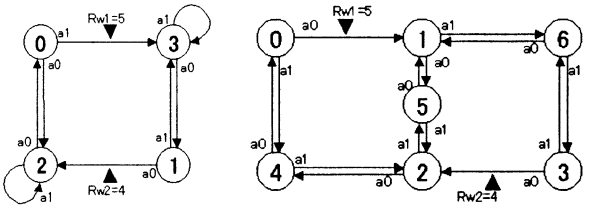


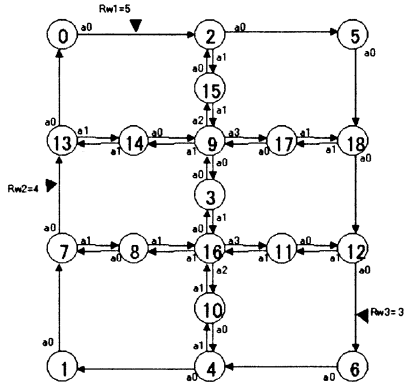
図4：実験1のMDPモデル

5.2 実験2：報酬位置のみ異なる環境での比較学習実験

実験2では，状態・行動集合は同じで，Rwの位置関係(Rw間の距離)の異なる3種類の環境を用いて最適cycle検出の時期を調べる．但し，全ての問題において最適政策(cycle)は同一である．実験2の目的は，問題となるMDPモデル中のRwの位置(距離)が異なる場合に，最適政策(cycle)の検出時期に違いがでるかどうかを明らかにすることである．図5に実験2で用いるMDPモデルの1つを示す．本実験では，報酬値の大きさが$Rw1 > Rw2 > Rw3$という条件において，

(env.a)＝Rw1とRw2は近くRw2とRw3は遠い，

(env.b)＝3つのRw間の距離はほぼ均一，



(env.c)＝Rw1とRw2は遠くRw2とRw3は近い，

の3種類の環境を用いる．

図5：実験2のMDPモデル

なお，実験2では実験1と同様に，効用値総更新回数と最適政策(cycle)検出時期の2項目を用いて比較する．

6 実験結果

6.1 実験1について

実験1の結果を図6，7に示す．グラフの縦軸は，図6では効用値の総更新回数，図7では最適政策(cycle)の検出された時点での効用値更新回数を指す．グラフの横軸は，共に環境の大きさ(総ルール数)を指す．

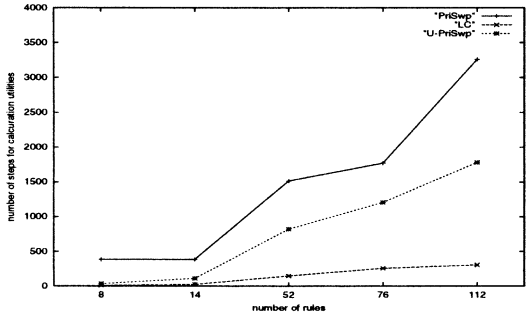


図6：実験結果1　効用値の総更新回数

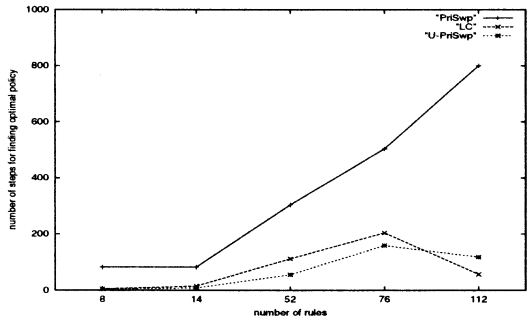


図7：実験結果1　最適政策(cycle) 検出step数

図6の効用値の総更新回数は，各手法の最適政策決定に要する学習コストを表す．この比較を以下に示す．

PriSwp ＞ *U-PriSwp* ＞ *LC-learning*

ここで，LC-learningのほうがU-PriSwpよりも学習コストが小さい理由は，LC-learningは2種類の枝刈りによるコスト削減を実現しているが，U-PriSwpは現段階ではこの枝刈りを実装しておらず，冗長なcycleも含めて検出していたからである，と考えられる．次に，図7の最適政策(cycle)検出step数は，最適政策の優先順序付き探索の効果の度合いを表す．この比較を以下に示す．

PriSwp ≫ *LC-learning* ＞ *U-PriSwp*

U-PriSwpが他の2手法よりも最適政策を早期に探索しており，潜在能力が大きいといえる．LC-learningのほうが早期検出している場合があるが，これはLC-learningの探索順序が環境(状態・行動の番号付けなど)に依存するためである．これに対しU-PriSwpでは展開済みpathの平均報酬値の大きい順に展開順序の優先順序付けを行うため，LC-learningと比べて最適cycle検出

時期が環境に依存しにくいと言える．

6.2　実験 2 について

実験2の結果を図8，9に示す．グラフの縦軸，横軸は実験1と同様である．図8の効用値の総更新回数の比較を以下に示す．

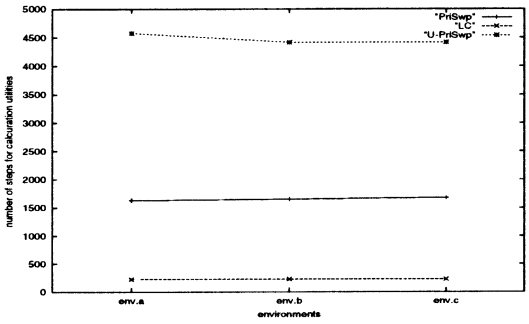


図8：実験結果2　効用値の総更新回数

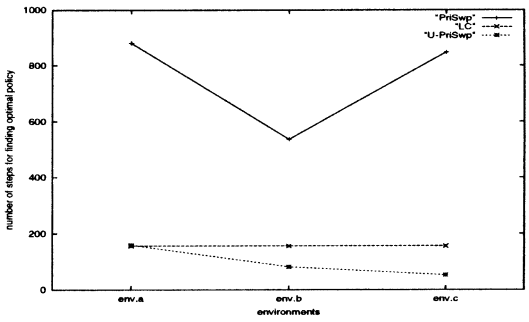


図9：実験結果2　最適政策(cycle)検出step数

U-PriSwp ≫ *PriSwp* ＞ *LC-learning*

U-PriSwpが最も悪いのは，LC-learningが行う枝刈りを実装していないため，Rw数が増えるほど重複して展開するcycleが増えるからである．図9の最適政策(cycle)検出step数の比較を以下に示す．

PriSwp ≫ *LC-learning* ≧ *U-PriSwp*

また図9より，環境中のRw間の距離に関して，PriSwpは環境の違いに左右される(env.bのRw間距離が均等の場合が最も速く最適政策を検出)のに対し，LC-learningとU-PriSwpは環境の違いにあまり左右されないことがわかる．

7　まとめと今後の課題

本稿では，強化学習における最適政策を効率良く算出する手法としてU-PriSwpを提案し，その有用性と特徴をPriSwpとLC-learningとの比較学習実験によって検証した．その結果，非割引型最適化手法にtree展開の概念を加えることにより繰り返し計算の回数を抑制できること，現状におけるU-PriSwpはLC-learningと比べて総コストがかかること，最適cycleの早期検出はU-PriSwpのほうがLC-learningよりも環境から受ける影響は小さい，ということが明らかになった．

今後の課題は，第一にU-PriSwpに枝刈りの機能を実装すること，そして，より効率的に学習できるように適切な優先度の定義を改良すること，扱える対象問題の範囲を確率的環境に拡張していくことである．

参考文献

[1] R. Sutton, A. Barto: 強化学習, 森北出版, 2000

[2] 馬場口 登，山田 誠二: 人工知能の基礎, pp.145-151, 昭晃堂, 1999

[3] S. J. Russell, P. Norvig: Artificial Intelligence-A Modern Approach, pp.598-624, Prentice Hall Inc., 1995

[4] A. W. Moore, C.G. Atkeson: Prioritized Sweeping - Reinforcement Learning With Less Data and Less Time, J.of Machine Learning 13, pp.103-130, 1993

[5] S. Mahadevan: An average-reward reinforcement learning algorithm for computing bias-optimal policies, In Proc. of the 13th AAAI (AAAI-96), pp.875-880, 1996

[6] 石村 健二，天正 新二郎，山口 智浩: RAE-PIA: 複数報酬環境下における最適政策の効率的強化学習, 第15回 人工知能学会全国大会, 2C1-05, 2001

[7] 誉田 太朗，天正 新二郎，山口 智浩: LC学習:モデルに基づく段階的平均報酬強化学習手法，第29回 知能システムシンポジウム論文集，pp.105-108, 2002

[8] Taro Konda, Tomohiro Yamaguchi: LC-Learning: Phased Method for Average Reward Reinforcement Learning - Analysis of Optimal Criteria -, PRICAI2002, Trends in Artificial Intelligence, M.Ishizuka and A.Sattar (Eds.), Lecture notes in Artificial Intelligence 2417, pp.198-207, 2002

[9] Taro Konda, Shinjiro Tensyo, Tomohiro Yamaguchi: LC-Learning: Phased Method for Average Reward Reinforcement Learning - Preliminary Results -, PRICAI2002, Trends in Artificial Intelligence, M. Ishizuka and A. Sattar (Eds.), Lecture notes in Artificial Intelligence 2417, pp.208-217, 2002

[10] 誉田 太朗，天正 新二郎，山口 智浩: モデルに基づく段階的平均報酬強化学習手法，第16回 人工知能学会全国大会論文集，2D3-04, 2002

[11] 天正 新二郎，山口 智浩: Undiscounted Prioritized Sweeping - 最適政策の優先順序付き強化学習の効率化手法 -，第16回 人工知能学会全国大会論文集，2D3-05, 2002