岡山理科大学紀要 第40号 A pp. 111-120(2004)

デッドロックを含む環境下における強化学習の性能と評価

興石　尚宏・片山　謙吾*・成久　洋之*
岡山理科大学大学院工学研究科情報工学専攻
*岡山理科大学工学部情報工学科
(2004年9月30日受付、2004年11月5日受理)

1. まえがき

強化学習 (Reinforcement Learning)[1)2)] は，エージェントが自ら環境を認識し，環境と相互作用しながら行動を獲得する能動型学習の一つである．実世界において遭遇する問題は，未知であり複雑かつ動的な変化を伴う環境である．現実的な問題に対して，人間が設計した行動群に従うエージェントを予めプログラム化することには限界がある．そのような困難な問題に対して，試行錯誤しながら行動を獲得する強化学習を用いるエージェントが注目されている．

実世界に遭遇する問題の中には，エージェントが問題解決をする過程の中で問題解決が出来なくなる状態 (デッドロック) を含む問題もある．デッドロックとは，エージェントがどの行動を取っても目標状態へ至る可能性が無い状態のことを指す．目標状態に達することができないデッドロックが発生した場合，強化学習エージェントは目標状態に達成できないのにもかかわらず探索を継続することは，学習効率を低下させることになると考えられる．しかしながら，デッドロックを含む環境に対して強化学習エージェントがどのように学習を行うかを検証する研究の報告は我々が知る限り無く，デッドロックの存在が学習効率を低下させるかどうかの確証は得られていない．

本論文では，デッドロックを含む問題に対する強化学習アルゴリズムの有効性を検証する．デッドロックを含む問題としてよく知られている倉庫番を取り上げ，Profit Sharing 強化学習を適用したエージェントで倉庫番を解決する．実験結果から，倉庫番における強化学習アルゴリズムの収束性がどのような傾向を示すか報告する．さらに，その傾向から倉庫番を対象とする強化学習エージェントの問題点について考察する．

2. 倉庫番

倉庫番[3)4)5)] は，1982年に今村宏行氏によって開発されたゲームであり，シングルエージェントを対象とする有名なゲームの一つとして知られている．倉庫番は，Culberson によって PSPACE 完全であると証明されている問題でもある[6)]．本論文では，以下の設定に基づく倉庫番を対象とする．

図1のように $n \times n$ 格子状の環境を設定し，ここにエージェント，一つまたは複数の荷物，荷物の数と同等かそれより多い荷物を置くためのゴール，および柱を配置する．エージェントは，上下左右の方向に1マス進むかまたはその場に留まる行動を一つ選択する．エージェントの目標は全ての荷物を任意のゴールまで押すことである．エージェントは目的を果す上で，以下に示す3つの規則を守らなければならない．

1. エージェントは1個の荷物を押すことができる
2. エージェントは2個以上の荷物を押すことはできない
3. エージェントは荷物を引いて動かすことはできない

複数の物体が同一マスに存在することはできない．しかし，エージェントとゴール，荷物とゴールは同一マスを占めることができる．

エージェントの視界は，$m \times m$ とし，自らの周囲 $m^2 - 1$ マスが見える．エージェントは視野内において，荷物，荷物を置くためのゴール，柱を知覚できる．

多くのシングルエージェントの問題は，どのような状態からでも問題を解くことができる．しかしながら，倉庫番にはエージェントが問題を解決できなくなる状態がある．一般的に，この状態はデッドロック

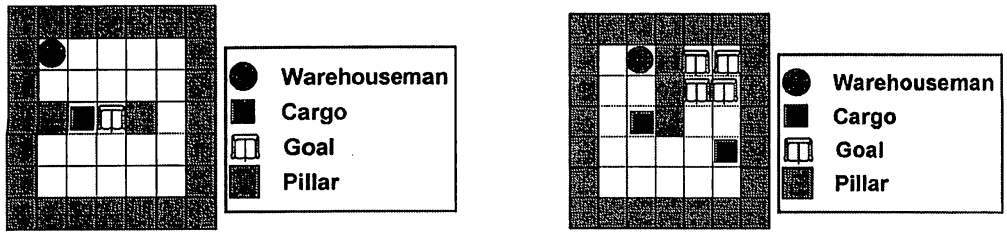

図 1　倉庫番の例

と呼ばれている．その例を図 2 に示す．図 2(a) では，エージェントは荷物を左右にしか動かせないため，エージェントは荷物を荷物の上にあるゴールまで運ぶことが不可能である．同様に，図 2(b) および (c) では，左右に荷物が二つ接触しており，かつ両方の荷物とも上側または下側で柱と接触しているため，エージェントは荷物を動かすことができない．

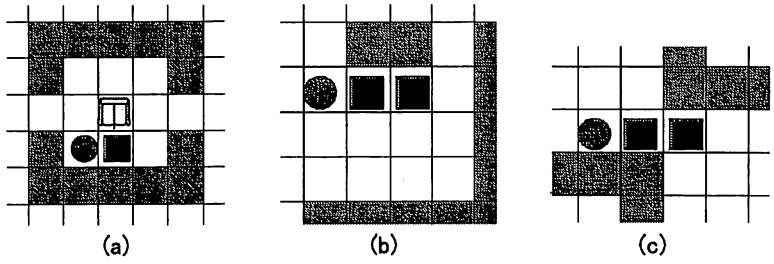

図 2　デッドロックの例

倉庫番において，デッドロックになる原因は荷物と柱，エージェントの位置が関係している．以下では，2 つ配置の例を挙げより詳しく述べる．

第 1 の配置は，一方通行の例である．図 3 に例を示す．一方通行の配置は，Steven Sabey によりエージェントが荷物を最小回数でゴールへ押す経路を求めるのは NP 困難であると示されている[7]．図 3 では，荷物はゴールにおかれている．この配置の特性を図 3 を用いて述べると，エージェントは A から B へ進むならば荷物を動かしたあと再び置きなおし，B へ進むことができる．しかし，エージェントは B から A へ進むならば荷物は動かすことが出来なくなり A へ進むことはできない．そのため，エージェントは A から B へ進むように行動を取らなければならない．

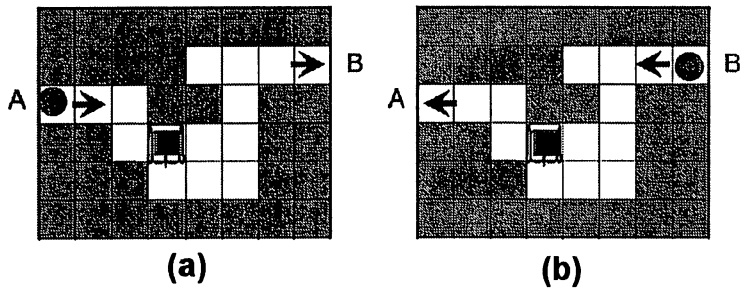

図 3　一方通行の例

第 2 の配置は，逆戻りの例である．図 4 に例を示す．図 4(a) では，荷物はゴールにおかれている．逆戻りとは，図 4(a) の状態でエージェントが B から A へ移動し，折り返し A から B へ移動しなければならないことを指す．図 4(a) で，エージェントは B から A へ進むだけならば容易である．しかし，A へ進むために通路をふさいでいる荷物を単純に動かすと二度とその荷物は動かすことが出来なくなる．例えば，エージェントが荷物を左側に動かしたなら荷物は角に入り動かせなくなる．エージェントが荷物を右側に動

かしたなら荷物はB側にある荷物に接触し動かせなくなる．そのため，図4(b)に示すように，エージェントはまずB側にある荷物を右に動かした後，A側にある荷物を右に動かさなければならない．図4(b)で，エージェントはAからBへ戻るだけならば同様に容易である．しかし，図4(a)に示すように荷物を再びゴールに配置しなければならない．図4(b)において，エージェントがA側にある荷物を先に戻さずにB側の荷物を動かすならば二度とその荷物は動かすことが出来なくなる．そのため，エージェントはまずA側にある荷物を左に動かした後，B側にある荷物を左に動かさなければならない．

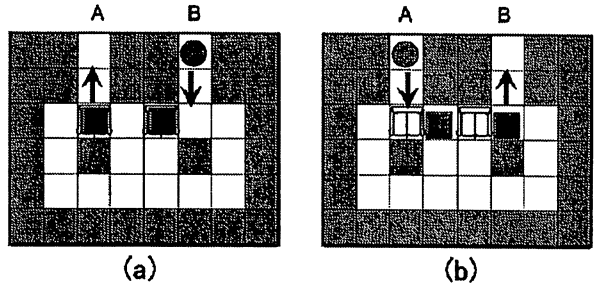

図4　逆戻りの例

3. Profit Sharing 強化学習

Profit Sharing[8]は，遺伝的アルゴリズム (genetic algrithm, GA) を併用するクラシファイアシステム (classifier system) での信用割当 (credit assignment) の方法として1980年代後半に提唱された．現在，Profit SharingはGAだけではなく強化学習の枠組みにおいても利用可能であり，さらに非MDPとなるようなマルチエージェント環境においても有用であると期待されている．Profit Sharingは他の強化学習手法より学習の立ち上がりが素早く，不完全知覚状態に対しても有効であることが示されている[9)10)]．そのため，本論文ではProfit Sharingを用いる．以下では，Profit Sharingについて記述する．

Profit Sharing (PS) は，報酬に至るまでのエピソードにおいて感覚入力の状態 s と実際に行った行動 a の対からなるルール系列を記憶しておき，報酬が得られたときにそれまで記憶した系列上のルールを一括して強化する学習方法である．ルール系列は次式を用いて強化する．

$$w(s_i, a_i) \leftarrow w(s_i, a_i) + f(r, i) \tag{1}$$

ここで，$w(s_i, a_i)$ はエピソード系列上の i 番目のルールの重み，r は報酬値，f は強化関数である．一般に強化関数 f は，報酬を獲得した時点からどれだけ過去であるかを引数として強化値を返す．

あるエピソードにおいて，同一状態が二回以上存在し，それぞれ別の行動を選択しているとき，その間のルール系列を迂回系列と呼ぶ．一般に迂回系列上のルールを無効ルール（ineffective rule）と呼び，それ以外のルールを有効ルール（effective rule）と呼ぶ．無効ルールは，報酬の獲得に貢献しない可能性があり，なるべくならば強化せず抑制したいと考えるべきである．我々は，政策の局所的合理性を保証する必要十分条件が証明されている[11)]の合理性定理にしたがい，無効ルールを抑制する次の強化関数を使用する．

$$f(r, j) = \frac{1}{S} f(r, j-1), \quad j = 1, \ldots, W-1. \tag{2}$$

ここで，W はエピソードの最大長，S は報酬割引率である．なお，報酬割引率は $S \geq L+1$ とする（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）．PSの合理性定理は，最適性を保証していないが，MDPの仮定を必要としないのでマルチエージェント系のような非MDP環境に対しても適用できる点に特徴がある．

PSの学習過程における行動選択法としては，ルーレット選択法が良い性能を示すことが経験的に知られている．ルーレット選択法は，ある状態 s において，各行動の重み $W(s, a_i)$ を全ての行動重みの合計 $\Sigma_a W(s, a)$ で割り，確率を求め，その確率により行動を選択する方法である．

$$P(a_i|s) = W(s, a_i) / \sum_a W(s, a) \tag{3}$$

また，非MDP環境における行動選択として有効である．以上の理由から本論文ではPSの行動選択としてルーレット選択を使用する．

4. 問題設定

本章では，デッドロックを含む問題が強化学習アルゴリズムの収束性にどのような影響を与えるのかを調べるために，2. 章で記述した倉庫番の基本設定と，以下に示す要素のもと，問題を設定する．すなわち，

- 荷物の数　　：単数または複数
- ゴールの数　　：単数または複数
- 荷物とゴールの数：荷物の数 $\leq$ ゴールの数

の組み合わせによって，以下の3つのタイプの問題を設定する．

Type1 [荷物の数：1，ゴールの数 1]
Type1は荷物の数とゴールの数がともに1である．エージェントは柱の位置のみに配慮してゴールまで荷物を運べばよい．

Type2 [荷物の数：2，ゴールの数 2]
Type2は荷物の数とゴールの数がともに2であり，Type1と同等に各数は等しい．しかしながら，エージェントは荷物の位置および柱の位置を配慮して荷物を運ばなくてはならない．

Type3 [荷物の数：2，ゴールの数 4]
Type3は，Type2のゴール数を倍に増やしたものである．そのため，エージェントは荷物をゴールの範囲内に納めればよいためType2よりも荷物の位置に配慮しなくても良い．

なお，問題の難しさの点からは，Type1が最も簡単であり，Type2が最も困難である．

5. 実験

本章では，4. 章で設定したType1〜3のそれぞれに対してエージェントの学習アルゴリズムとしてProfit Sharingを実装し，エージェントが荷物をゴールまで納める行動を学習するプロセスを実験により調べ，結果について考察する．実験は，二つ行う．一つ目は，4. 章で設定したType1〜3における問題の結果の比較をおこない，どのようになるか比較検討を行う．二つ目は，4. 章で設定したType1において問題サイズを変え，荷物の到達距離を増加させることにより，どのような傾向を示すか比較検討を行う．

全てのTypeにおいてエージェントの視覚サイズは $7 \times 7 (m = 7)$ とし，Profit Sharing強化学習の初期ルールの重みを0.1，公比を0.9の等比減少関数を強化関数とする．各Caseに対する学習の試行回数は5回である．1試行の学習は10万エピソードとする．2. 章で記述したように，倉庫番にはデッドロックが存在する．それにより，試行錯誤を繰り返す学習の初期段階では，エージェントが頻繁にデッドロックにおちいる．そのため、エージェントのステップ数が1万になった時点で次のエピソードに移る．

5.1 実験1の結果および考察

図5は，各Typeで検証する環境を示している．図5の(a)はType1を示し，(b)はType2を示し，そして(c)はType3を示す．

各Typeごとに，Profit Sharingによってそれぞれ5試行実験を行った．それぞれの2万エピソードまでの学習初期の結果を図6に示す．なお，縦軸は1エピソードごとに目標を達成するまでに要した行動（ステップ）回数，横軸はエピソード数である．図6で(a)はType1の結果を示し，(b)はType2の結果を示す．そして，(c)はType3の結果を示す．また，表1は，学習後に100エピソード分の行動（ステップ）回数を計測し，その行動回数の平均と標準偏差を求めたものである．

1. **Type1の結果**
   図6(a)から，エージェントは2千エピソードから収束がみられる．つまり，エージェントはデッドロックになることがあまりなく目標を達成していると考えられる．そして，表1からエージェントの移動回数は最小ではないが，荷物は最小の回数である．また，標準偏差が0なことから荷物が押され

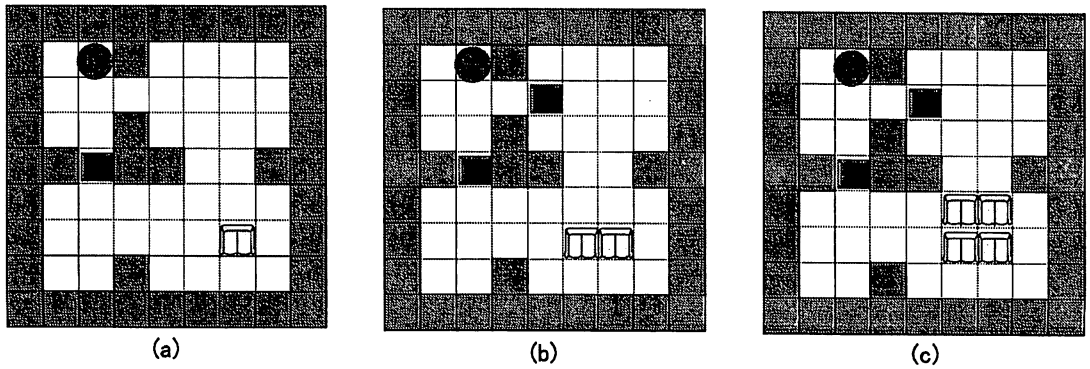

図 5　各設定における環境：Type1(a)，Type2(b)，および Type3(c)

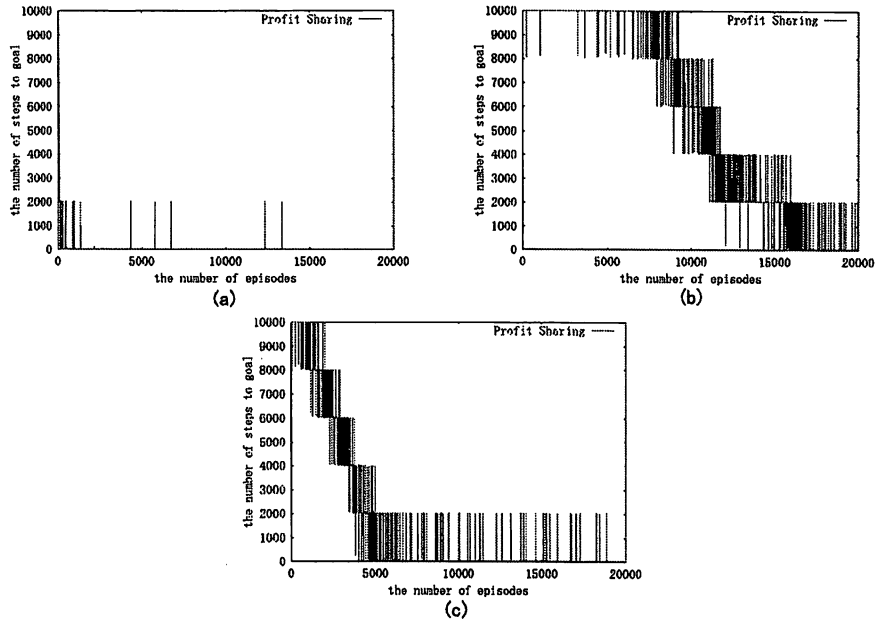

図 6　倉庫番の各 Type における学習初期の学習曲線

る経路が確立されたと判断できる．これは，以下に示す Type2 と Type3 においても同様の傾向が観察される．

2. **Type2 の結果**
図 6(b) から，エージェントは 2 万エピソードを経ても収束は見られるが，他の試行で失敗する影響を受け完全な収束は見られなかった．これは 10 万エピソード終了後でも同様な結果であった．エージェントは荷物をゴールにいれなければならない．そのため，図 7 で示すように左側のゴールに荷物を先に置くとエージェントは二つ以上の荷物を押すことはできないため目標を達成できにくくなる．

3. **Type3 の結果**
図 6(c) から，Type3 は Type2 と異なり 2 万エピソードあたりから収束が見られる．これは，Type3 は Type2 とゴール数が異なることからエージェントは図 7 の状態になり難いためである．それは，表 1 から判断できる．表 1 の Type3 の結果から，標準偏差は 0 とばらつきが無いことがわかるが，平均移動回数が 9.2 であることから，ある試行で異なる経路が獲得されたと考えられる．

表 1　倉庫番の各 Type における学習後期の結果

	Agent		Cargo	
	Avg.	Stdv.	Avg.	Stdv.
Type1	13.05	0.82	6	0
Type2	29.39	1.34	11	0
Type3	25.91	1.13	9.2	0

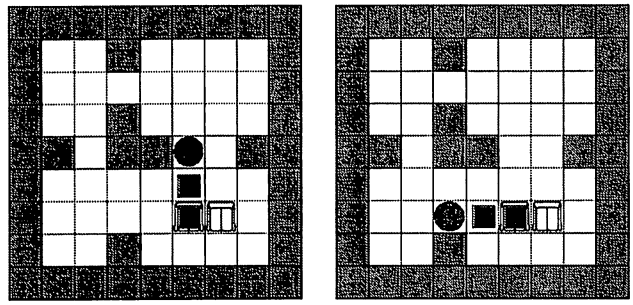

図 7　目標を達成しにくい状態の例

上記の実験と他に，荷物を 3 ゴールを 3 とした問題を行った．図 8 に環境を示す．この問題の場合，Profit Sharing 強化学習エージェントでは 10 万エピソード中数回しか目標を達成することができなかった．この問題は，Type2 と同 Type であるが Type2 よりデッドロックが発生しやすい．すなわち，より多くの荷物を取り扱う場合には従来の強化学習だけを用いるのは適切ではない．そのため，エージェントは強化学習を用いるとともに，自ら判断してデッドロックを回避するような能力が必要となる．

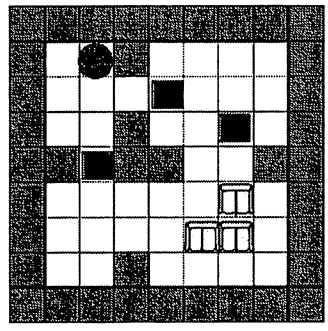

図 8　荷物の数 3，ゴールの数 3 の環境

5.2　実験 2 の結果および考察

図 9 に示す 8 つの Case の問題に対して実験を行う．Case1 を基準とし Case 番号が増えるごとに格子サイズ 2 ずつ拡張していく．Case1 は最小の格子サイズ 7 であり，Case8 は最大の格子サイズ 21 である．そして，荷物の移動回数も同様に Case1 を基準とし Case 番号が増えるごとに荷物の移動回数を 3 ずつ増加する．Case1 は荷物の最小の移動回数は 3 であり，Case8 は荷物の最小の移動回数は 24 である．

図 10(a) から (f) までは，各 Case に対する学習初期（2 万エピソードまで）の学習結果である．また，(g) と (h) は 10 万エピソードまで表示している．なお，縦軸は 1 エピソードごとに目標を達成するまでに要した行動（ステップ）回数，横軸はエピソード数である．

表 2 は，上述の各 Case の学習に対して，10 万エピソード学習を行った後に 100 エピソード分のエージェント行動（ステップ）回数と荷物の移動回数を計測し，その各回数の平均とその標準偏差の結果である．

図 10 から，倉庫番において問題のサイズや、荷物の移動回数が学習を行う上で色濃く影響を与えていることがわかる．問題のサイズが増加することは，エージェントにとって不完全知覚領域が拡大するとともに

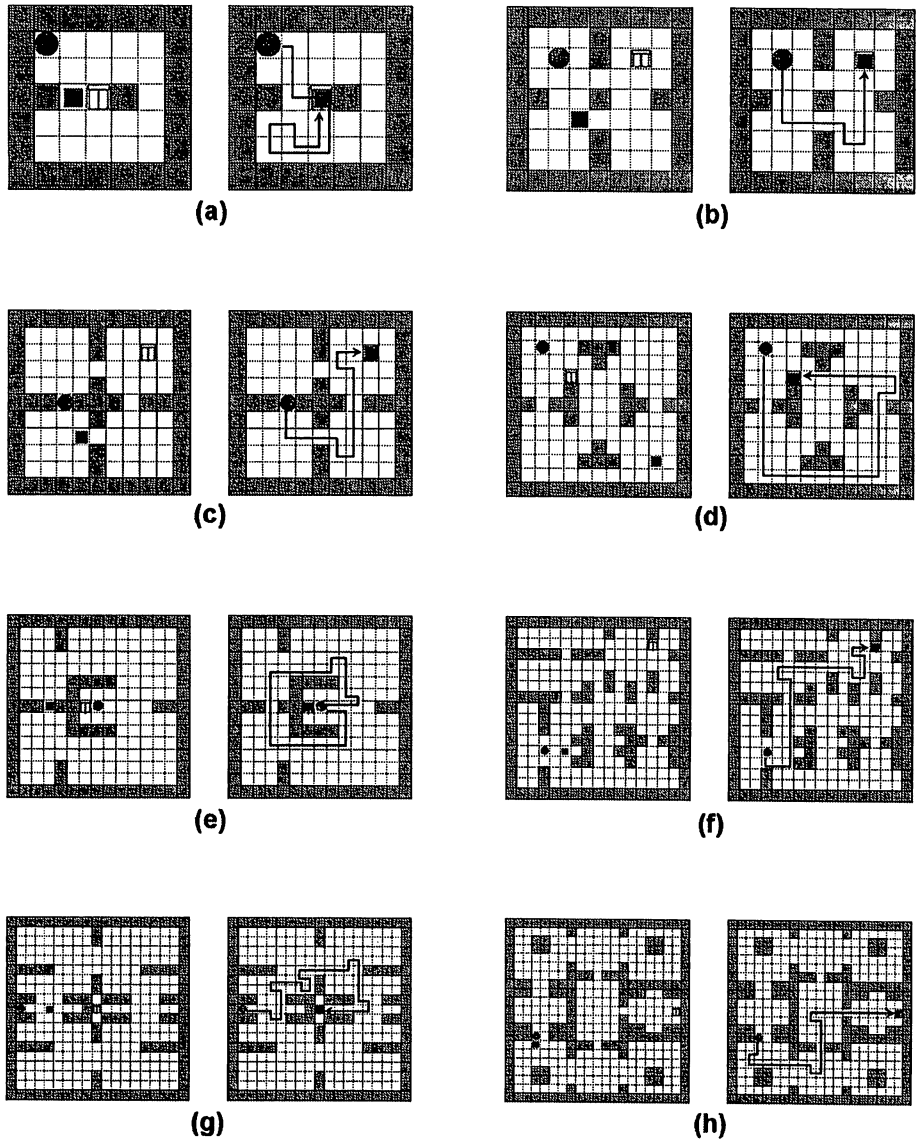

図 9　倉庫番の各 Case と解答例

多数の同一状態存在することを意味する．そして，荷物の移動回数が増加することにより，エージェントが荷物を動かす上でデッドロックがより引き起こされやすくなることが考えられる．Type1 において顕著にこの傾向が現れることから，Type2 ではより困難を極めると考えられる．そのことから，エージェントは不完全知覚状態や同一状態において適切な行動を獲得するとともに，5.1 節で述べたのと同様，自ら判断してデッドロックを回避するような能力が必要となる．

6. むすび

本論文では，デッドロックを含む問題において強化学習アルゴリズムの収束性がどのような傾向を示すか調査した．デッドロックを含む問題として有名な倉庫番を対象とし，Profit Sharing 強化学習を適用したエージェントにより実験を行った．各実験結果から，強化学習エージェントがゴールへ荷物を運ぶまでの移動回数が増加すること，そして問題内に複数の荷物が存在することにより強化学習アルゴリズムの学習性能が低下することがわかった．強化学習エージェントは，目標状態に到達した際に与えられる報酬を手掛りに学習を行うため，報酬が得られなければ学習は進行しない．つまり，問題が複雑になることによりエージェントはデッドロックになりやすく，目標を達成できなくなる可能性が非常に高くなることを意味する．

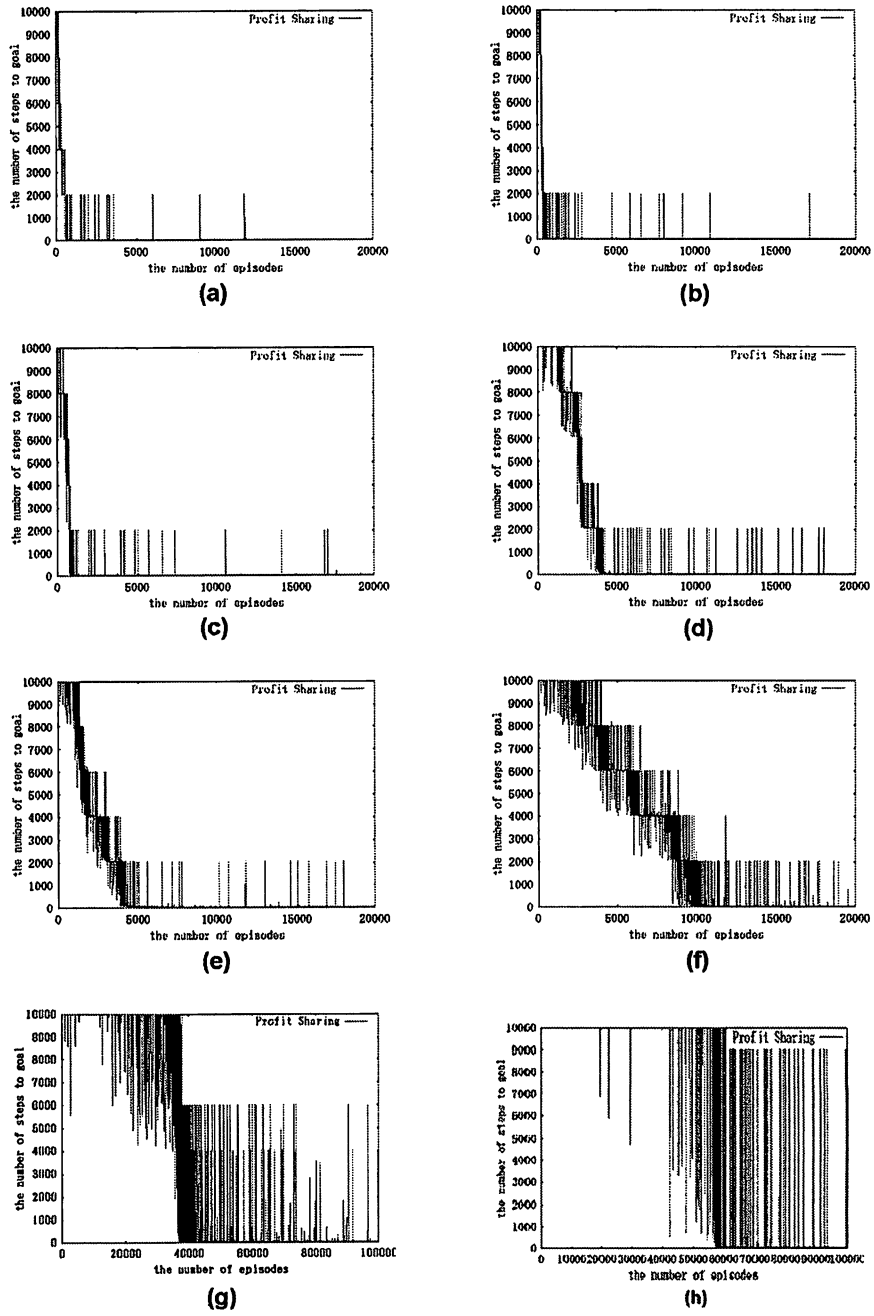

図 10　各 Case における学習初期の学習曲線

現実問題は上記の環境よりもとても複雑である．すなわち，現実で直面する問題を従来の強化学習だけで解決することは有効であると言いがたい．なぜならば，現実の場面では迅速な対応や安全性・確実性が求められるからである．そのため，強化学習エージェントは学習初期の段階から問題を解決する術をもたなければならない．デッドロックを含む問題であるならば，エージェントは自らデッドロックを判断して回避することが重要となる．倉庫番におけるデッドロックは，2. 章に示したようにエージェントが荷物を押した結果，荷物が壁や角に接触することである．5. 章で述べたように，荷物が壁や荷物に等の障害物に接触することに着目して考えると，エージェントは荷物の先に障害物があるか判断し障害物がある場合には荷物を押す行動を控え，荷物の押す方向を変えるような行動を取ればデッドロックの発生を抑制できると考えられる．このように，他の問題においても強化学習アルゴリズムを用いるとともに，人間が考えられる知識を予め

表 2 各 Case における学習後期の結果

	Agent		Cargo	
	Avg.	Stdv.	Avg.	Stdv.
Case1	18.10	0.94	3	0
Case2	22.74	1.22	7	0
Case3	19.47	1.00	9	0
Case4	39.09	1.55	12	0
Case5	43.53	1.30	15	0
Case6	31.14	1.23	18	0
Case7	45.74	2.39	21	0
Case8	37.31	3.55	24	0

エージェントに知らせておくことにより，強化学習エージェントは迅速な対応や安全性・確実性をもって問題を解決することが可能になると考えられる．我々は知識を用いる強化学習エージェントの有効性をマルチエージェントの問題である追跡問題で示している[12]．今後の課題として，デッドロックを含む環境においても文献[12]で提案したエージェントが有効であることを検証する．

参考文献

1) Kaelbing, L. P., Littman, M. L., and Moore, A. W. "Reinforcement Learning: A Survey," Jurnal of Artificial Intelligence Research, Vol.4, pp.237–285, 1996.

2) Richard S. Sutton, Andrew G. Barto, "Reinforcement Learning–An Introduction–," The MIT Press, 1998.

3) A. Junghanns, and J. Schaeffer, "Sokoban: A Challenging Single-Agent Search Problem," Workshop on Using Games as an Experimental Testbed for AI Research, Proceedings IJCAI-97, 1997.

4) A. Junghanns, and J. Schaeffer, "Single-Agent Search in the Presence of Deadlocks," Proceedings of AAAI-98, pp. 419-424, 1998.

5) A. Junghanns, and J. Schaeffer, "Sokoban: Evaluating Standard Single-Agent Search Techiniques in the Presence of Deadlock," In R.Mercer and E. Neufeld editors, Advances in Artificial Intelligence, pp.1–15, 1998.

6) J. Culberson, "Sokoban is PSPACE-complete," Techinical Report TR97-02, Dept. of Computing Science, University of Alberta, 1997.

7) Stephen Sabey. "On the complexity of Sokoban," Unpublished 1996.

8) Grefenstette, J. J. "Credit assignment in rule discovery systems based on genetic algorithms," Machine Learning, Vol.3, pp.225–245, 1988.

9) 荒井幸代, 宮崎和光, 小林重信, "マルチエージェント強化学習の方法論–Q-Learning と Profit Sharing による接近–," 人工知能学会誌, Vol.13, No.4, pp.609–618, 1998

10) 宮崎和光, 木村元, 小林重信, "Profit Sharing に基づく強化学習の理論と応用," 人工知能学会誌, Vol.14, No.5, pp.800-807, 1999.

11) 宮崎和光, 山村雅幸, 小林重信, "強化学習における報酬割り当ての理論的考察," 人工知能学会誌, Vol.9, No.4, pp.580–587, 1994

12) 片山謙吾, 輿石尚宏, 成久洋之, "強化学習エージェントへの階層化意思決定法の導入—追跡問題を例に—," 人工知能学会論文誌, Vol.19, No.4, pp.279–291, 2004.

Performance of Agent using Reinforcement Learning in the Environment Containing Deadlock

Takahiro KOSHIISHI, Kengo KATAYAMA* and Hiroyuki NARIHISA*

Graduate School of Engineering,
**Department of Information and Computer Engineering,*
Faculty of Engineering,
Okayama University of Science.
1 - 1 Ridai-cho, Okayama, 700-0005, Japan.
(Received September 30, 2004; accepted November 5, 2004)

Reinforcement Learning (RL) is a learning method by trial-and-error search and delayed reward, and RL is expected as a technology to apply to real world problems. The problems containing the state of DeadLock appear in real world problems. If agent is in the state of DeadLock, agent cannot solve a problem. In the problem containing DeadLock, it is important to verify the performance of RL-agent. However, as far as we know, there is no research report which verifies it so far.

In this paper, we investigate the performance of RL-agent using Profit Sharing, and evaluate it in the environment containing DeadLock. We take up the Sokoban problem as one of the environment containing DeadLock. To evaluate RL-agent in an environment containing DeadLock, we test various environments using RL-agent. We also describe an improving point when RL-agent solves Sokoban.