RF-004

マルチエージェントにより問題空間の分割を行う階層化複素強化学習の検討

A Study of the Hierarchical Complex-Valued Reinforcement Learning Partitioning Problem Space with Multi Agent

山崎 惇広 †　　濱上 知樹 †
Atsuhiro YAMAZAKI　Tomoki HAMAGAMI

1 はじめに

強化学習とは，学習主体であるエージェントが環境との相互作用を通じて適切な振る舞いを学習する手法である [1]。実環境において強化学習を応用する場合，センサなどの制約により，観測に必要な情報が不十分となることが考えられる。そのため，異なる環境の状態を同一の状態と観測してしまい，学習が困難になることがある。これを不完全知覚問題と呼び，実環境へ強化学習を応用する際のボトルネックとなっている。

不完全知覚問題に対して，従来提案された手法として，Utile Suffix Memory に代表されるメモリベース法 [2] や，階層的にタスクを分割する HQ-learning(Hierarchical Q-learning)[3], belief states を用いて現在の状態を確率的に表現する手法 [4], そして複素化された行動価値を用いる複素強化学習 [5] などが挙げられる。特に複素強化学習は，学習に環境の事前知識を必要とせず，またメモリの使用量も少なくすむという利点を持つ。

複素強化学習では，複素化された行動価値である複素行動価値と，内部参照値と呼ぶ複素変数によって，文脈依存な行動選択を実現しており，その行動選択は複素行動価値と内部参照値の双方の位相が大きく関わる。しかし，複素行動価値の位相は，不完全知覚の現れる周期に依存するため，学習の可否は問題の規模に依らず，不完全知覚の分布に依存することとなる [6]。すなわち，解決できる問題の規模は不完全知覚の分布に依存することとなる。

そこで本研究では，複素強化学習に HQ-learning で用いられる階層構造を取り入れることで，様々な不完全知覚の分布への対応を実現し，複素強化学習で扱うことのできる問題クラスの拡張を目標とする。

2 複素強化学習

複素強化学習とは，複素化された価値である複素価値と，内部参照値と呼ぶ複素変数によって，文脈依存な行動選択を実現した手法である。内部参照値はエージェントが行動するたびに変化する値であり，エージェントの文脈情報を表す。複素行動価値と内部参照値との相互作用によって，次のルールに従って行動選択をする。

- 複素行動価値の絶対値が大きいほど，その行動は選ばれやすい。
- 複素行動価値の位相と内部参照値の位相が近いほど，その行動は選ばれやすい。

これらのルールより，複素行動価値と内部参照値の内積の実部が，従来の価値の大きさに相当する。不完全知覚によって同一の観測となる状態においても，内部参照値の位相によって選択する行動が変化するため，不完全知覚問題が発生する問題でも有効に働く可能性を有する。

2.1 Q̇-learning

Q̇-learning とは，複素強化学習の一手法であり，Q-learning の価値を複素化したものである。時刻 t における環境の状態を s_t，その観測を o_t，選択した行動を a_t とすると，複素行動価値 $\dot{Q}(o_t, a_t)$ は，次式により更新される。

$$\dot{Q}(o_t, a_t) \leftarrow (1-\alpha)\dot{Q}(o_t, a_t) + \alpha(r_{t+1} + \gamma \dot{Q}^{(t)}_{\max})\dot{\beta} \quad (1)$$

$$\dot{Q}^{(t)}_{\max} = \dot{Q}(o_{t+1}, a) \quad (2)$$

$$a = \arg\max_b (Re[\dot{Q}(o_{t+1}, b)\bar{\dot{I}}_t]) \quad (3)$$

ここで，α は学習率，γ は割引率，$\dot{\beta}$ は行動価値の位相変化量，$R(s_t, a_t)$ は状態 s_t において行動 a_t をとったときの報酬を表す。$\dot{\beta}$ の値の設定には，文献 [5] の経験的な設定方法を用いる。

$\dot{I}_t$ は時刻 t における内部参照値であり，次のように定める。

$$\dot{I}_t = \begin{cases} \dot{Q}(o_t, a_t)/\dot{\beta} & t \geq 0 \\ \dot{Q}(o_0, a) \ \ a = \arg\max_b |\dot{Q}(o_0, b)| & t = -1 \end{cases} \quad (4)$$

$\arg\max_b (Re[\dot{Q}(s_{t+1}, b)\bar{\dot{I}}_t])$ は行動価値と内部参照値の内積の実部がもっとも大きい行動を表しており，先に示した行動選択のルールに基づいてもっとも適切とされる行動である。

Q̇-learning では，学習速度の向上と文脈情報の獲得のために，次のオンライン更新の適格度トレースを導入する。

$$\dot{Q}(o_{t-k}, a_{t-k}) \leftarrow (1-\alpha)\dot{Q}(o_{t-k}, a_{t-k}) + \alpha(r_{t+1} + \gamma \dot{Q}^{(t)}_{\max})\dot{u}(k) \quad (5)$$

$$\dot{u}(k) = \dot{\beta}^{k+1} \quad (6)$$

ここで，$k = 0, 1, \cdots, N_e - 1$ で，N_e をトレース数と呼ぶ。$N_e - 1$ ステップ前までの状態・行動に対する価値を更新することで，より多くの文脈情報の獲得が期待できる。

2.2 問題点

複素強化学習では，行動選択に複素行動価値と内部参照値 $\dot{I}$ の双方の位相が大きく関わる。複素行動価値の位相によっては，内部参照値 $\dot{I}$ の位相を適切な位相へ変化させることができないため，適切な行動選択が困難となる。また，複素行動価値の位相は，不完全知覚の現れる周期に依存するため，学習の可否は問題の規模に依らず，不完全知覚の分布に依存することとなる。すなわち，解決できる問題の規模は不完全知覚の分布に依存することとなる。そこで，複素強化学習をより幅広く利用できるアルゴリズムにするために，様々な分布で生じる不完全知覚への対応が必要となる。

3 階層構造を有する複素強化学習

提案手法では，HQ-learning の階層構造を複素強化学習に取り入れ，問題空間を分割することにより，様々な分布で生じる不完全知覚への対応を実現し，複素強化学習で解決できる問題のクラスを拡張することを目指す。

3.1 階層化の構造

提案手法では，HQ-learning の階層構造を取り入れる。つまり，エージェントは $C_1, C_2, \cdots, C_M$ の M 個の順序付けられたサブエージェントから構成され，それぞれのサブエージェントが選択するサブゴールにしたがって，サブタスクへの分割を行う。各時刻で行動決定権を有するサブエージェントはひとつ

† 横浜国立大学大学院工学府

であり，そのサブエージェントが持つ方策に従ってエージェントは行動選択をする。各サブエージェントは，それぞれ独立した方策を学習するため，サブエージェントごとに異なる文脈を学習することができる。

提案手法において，サブエージェントは，共通の内部参照値と，それぞれ独立した $\dot{Q}$-table と，HQ-table を持つ。$\dot{Q}$-table は，HQ-learning における Q-table を複素数に拡張した行列であり，その要素数は $|O|\times|\mathcal{A}|$ となる。ここで，O は観測全体の集合を表し，$\mathcal{A}$ は行動全体の集合を表す。$\dot{Q}$-table の各要素は，ひとつの観測行動対に対する複素行動価値を示す。HQ-table については，HQ-learning と同様であり，要素数 $|O|$ のベクトルで表される。そして，各要素はひとつの観測に対する HQ 値を示す。HQ 値とは，各観測に対してサブエージェントが持つ値であり，その観測のサブゴールとしての適格度を表す。

提案手法では，複素強化学習と同様に，$\dot{Q}$-table に保存される複素行動価値と内部参照値の相互作用によって行動選択を行う。このため，単一のサブエージェントが，複素強化学習と同等の不完全知覚対応能力を有する。したがって，各サブタスクを Q-learning によって学習する HQ-learning とは異なり，各サブタスクに不完全知覚問題が含まれていても有効である。その結果，学習が行えるサブゴールの組み合わせは HQ-learning よりも多くなり，学習の高速化が期待できる。

3.2 アルゴリズム

提案手法は，複素強化学習の一手法である $\dot{Q}$-learning を，先に述べた階層構造に適応できるよう拡張したアルゴリズムとなる。

3.2.1 行動価値の更新

提案手法では，$\dot{Q}$-learning の更新式を，サブエージェント間で価値の伝搬がされるよう拡張した更新式を用いる。時刻 t で行動決定権を持つサブエージェントの番号を $A(t)$ とすると，$C_{A(t)}$ の行動価値は，次のように更新される。

$$\dot{Q}_{A(t)}(o_t,a_t) \leftarrow (1-\alpha_Q)\dot{Q}_{A(t)}(o_t,a_t) + \alpha_Q(r_{t+1}+\gamma\dot{Q}^{(t)}_{A(t+1)\max})\dot{\beta} \quad (7)$$

$$\dot{Q}^{(t)}_{A(t+1)\max} = \dot{Q}_{A(t+1)}(o_{t+1},a) \quad (8)$$

$$a = \arg\max_b(Re[\dot{Q}_{A(t+1)}(o_{t+1},b)\bar{\dot{I}}_t]) \quad (9)$$

$\dot{Q}^{(t)}_{A(t+1)\max}$ は，次の時刻において行動決定権を持つサブエージェントのもっとも文脈に適した行動となるため，時刻 t と時刻 $t+1$ で行動決定権を持つサブエージェントが切り替わる場合にも，価値の伝搬が行われる。また，学習速度の向上と文脈情報の獲得のために，次のオンライン更新の適格度トレースを導入する。

$$\dot{Q}_{A(t-k)}(o_{t-k},a_{t-k}) \leftarrow (1-\alpha)\dot{Q}_{A(t-k)}(o_{t-k},a_{t-k}) + \alpha(r_{t+1}+\gamma\dot{Q}^{(t)}_{A(t+1)\max})\dot{u}(k) \quad (10)$$

$$\dot{u}(k) = \dot{\beta}^{k+1} \quad (11)$$

ここで，k は $k=0,1,\cdots,N_e-1$ であり，N_e をトレース数と呼ぶ。これは，$\dot{Q}$-learning で取り入れられている適格度トレースの手法を，階層化のために拡張したものである。

3.2.2 HQ 値の更新

HQ 値の更新は，エピソードの終了時に，そのエピソードで用いられたサブエージェントすべてについて行われる。C_1 から C_M のサブエージェントが用いられた場合には，C_m の HQ 値は次のように更新される。ただし，$m=M, M-1,\cdots,1$ である。

$$HQ_m(\hat{o_m}) = (1-\alpha_{HQ})HQ_m(\hat{o_m}) + \alpha_{HQ}(R_m+\gamma^{t_{m+1}-t_m}HV_{m+1}) \quad (12)$$

ここで，$\hat{o_m}$ はサブエージェント C_m がそのエピソードで選択したサブゴール，α_{HQ} は HQ 値のための学習率，t_m はサブエージェント C_m に切り替わった時刻，$R_m=\sum_{t=t_m}^{t_{m+1}}R(s_t,a_t)$，$HV_m=\max_{o_l\in O}\{HQ_m(o_l)\}$ である。

3.2.3 内部参照値の変更

内部参照値は，基本的に $\dot{Q}$-learning と同様の変更を行う。しかし，サブエージェントの切り替えが起きた場合，切り替え前のサブエージェントと，切り替え後のサブエージェントでは異なる文脈を学習しているために，文脈の整合性がとれなくなることが考えられる。そのため，サブエージェントの切り替えが起きる場合には，特別な内部参照値の変更を行う必要がある。ここでは，サブエージェントの切り替えが起きる場合は，切り替え後のサブエージェントの，次の時刻の観測においてもっとも絶対値の大きい複素行動価値に変更する。これらをまとめて，次のように定式化する。

$$\dot{I}_t = \begin{cases} \dot{Q}_1(o_0,a)\ a=\arg\max_b|\dot{Q}(o_0,b)| & t=-1 \\ \dot{Q}_{A(t)}(o_t,a_t)/\dot{\beta} & A(t)=A(t+1) \\ \dot{Q}_{A(t+1)}(o_{t+1},a)\ a=\arg\max_b|\dot{Q}_{A(t+1)}(o_{t+1},b)| & A(t)\neq A(t+1) \end{cases} \quad (13)$$

3.2.4 サブゴールの選択方法

サブゴールの選択は，サブエージェントの切り替わりが発生したときに行われる。ϵ-greedy 選択を用いると，確率 ϵ でランダムな観測を選択し，確率 $1-\epsilon$ でもっとも HQ 値の大きい観測が選択される。これは，HQ-learning で用いられているサブゴールの選択方法と同一の方法である。しかし，行動価値の更新方法の違いから，提案手法においては適切に機能しない。そのため，ここでは常にもっとも HQ 値の大きい観測をサブゴールとして選択する，greedy 選択についてのみ考える。

3.2.5 動的なサブエージェントの追加

サブゴールの選択に greedy 選択を用いた場合，サブゴールの組み合わせは固定されるため，より良いサブゴールを探索するために，サブゴールを変更する仕組みが必要となる。多くのサブエージェントが学習に必要となる場合，固定されたサブゴールでは，ひとつのサブエージェントでは対応できないサブタスクが生じてしまう可能性が高い。そこで，そのようなサブタスクに対しサブエージェントを追加して，さらに小さなサブタスクへと分割することを考える。具体的には，次のような処理を行う。

1. m エピソード毎に各サブエージェントの行動したステップ数をチェックする。
2. n ステップ以上行動したサブエージェントのうち，最後に切り替わったサブエージェントのコピーを作成し，その次のサブエージェントとして追加する。該当するサブエージェントがなければ，追加は行わない。
3. コピー元となったサブエージェントのサブゴールをランダムに設定する。ただし，その前のサブエージェントのサブゴールとは別の観測とする。

ここで，m, n は，事前にシステムの設計者が定める必要のあるパラメータである。

このような動的にサブエージェントを追加する仕組みによって，徐々にサブタスクを改善できる。そして，エージェントが学習可能なサブタスクへ，タスクを分割することが可能となる。

4 実験および考察

シミュレーション実験により，提案手法の性能を確認する。まず実験 1 の迷路タスクにより，提案手法である，階層化による効果と，動的なサブエージェント追加を行うことの効果の検証を行う。次に実験 2 の迷路タスクにより，提案手法の適用限界について検証を行う。各迷路タスクにおいて，エージェントは四方の壁の有無しか観測できないとするため，不完全知覚問

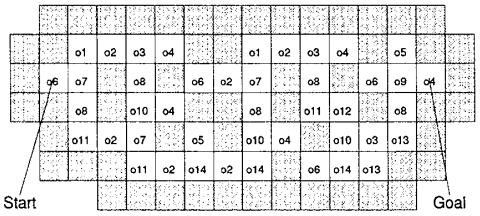

図 1　実験 1 で用いる迷路

表 1　実験 1 パラメータ

Q̇-learning	
学習率 α	0.25
位相変化量 $\dot{\beta}$	$\exp(j\pi/12)$
割引率 γ	0.9
Boltzmann 温度 T	20
トレース数 N_e	4
提案手法	
行動価値の学習率 α_Q	0.25
HQ 値の学習率 α_{HQ}	0.1
位相変化量 $\dot{\beta}$	$\exp(j\pi/12)$
割引率 γ	0.9
Boltzmann 温度 T	20
トレース数 N_e	4
サブエージェント数 M	12
サブエージェント	
追加周期 m	50 episodes
サブタスク学習不可判定	
ステップ数 n	50 steps

題が発生するタスクとなる。

4.1　実験 1: 提案手法の有効性についての検証

4.1.1　実験方法

図 1 の迷路について，Q̇-learning，動的なサブエージェントの追加を行う提案手法，動的なサブエージェントの追加を行わない提案手法の 3 つのアルゴリズムを用いて学習を行った。どちらもサブゴールの選択方法には，greedy 選択を用いた。実験に用いたパラメータを表 1 に示す。報酬はエージェントがゴールにたどり着いたときに限り与え，その値は $r = 100$ とした。エージェントの 1 回の行動を 1 ステップ，スタートからゴールにたどり着くまでを 1 エピソード，4000 エピソードを 1 学習として，1000 学習行った。

4.1.2　実験結果と考察

得られた実験結果を図 2 に示す。グラフの横軸はエピソード数を示し，縦軸はゴールに到達するまでにかかった平均ステップ数を示す。

Q̇-learning では，ゴールに到達するまでにかかった平均ステップ数は，900 ステップ前後で振動した。これより，Q̇-learning では，この迷路タスクについての学習は困難であることがわかる。

動的なサブエージェントの追加を行わない提案手法では，Q̇-learning よりも少ない平均ステップ数である，700 ステップ前後で振動した。これは，階層化の効果により，適切なサブゴールの組み合わせが選択された場合には，学習が成功するためである。しかし，サブゴールの組み合わせを変更する仕組みを持たないため，非常に低い確率でしか学習は成功しない。動的なサブエージェントの追加を行う提案手法では，ゴールに到達するまでにかかった平均ステップ数は，学習が進むにつれて，徐々に減少していき，最終的には 50 ステップほどに

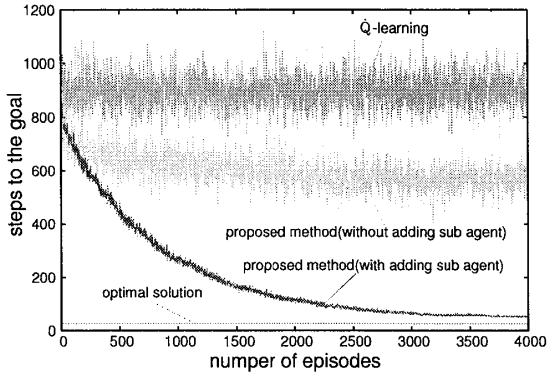

図 2　実験 1 エピソード毎の平均ステップ数

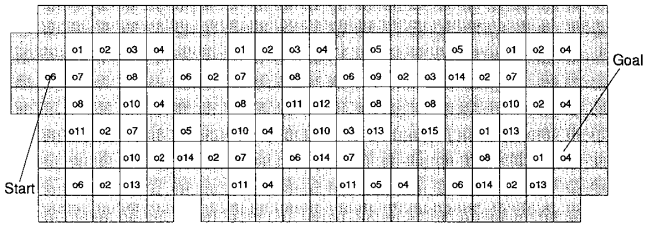

図 3　実験 2 で用いる迷路

収束した。これより，動的なサブエージェントの追加を行うことで，学習可能なサブゴールの組み合わせに改善できることがわかった。しかし，平均ステップ数の収束値は，最適経路で必要なステップ数である 25 ステップの約 2 倍と，大きな値となっている。これは，動的なサブエージェントの追加を行うことで，使用するサブエージェントの数が増加することが原因である。使用するサブエージェントの数が増加すると，ゴールに到達するには不必要なサブタスクが生じてしまう。このため，ゴールに到達するのには余分な経路が多く発生し，最終的に収束する平均ステップ数は大きくなる。

4.2　実験 2: 提案手法の問題の規模に対する適用限界についての検証

4.2.1　実験方法

図 3 の迷路について，動的なサブエージェント追加を行う提案手法，動的なサブエージェント追加を行わず事前知識として適切なサブゴールの組み合わせを与えた提案手法の 2 つのアルゴリズムを用いて学習を行った。実験に用いたパラメータを表 2 に示す。報酬はエージェントがゴールにたどり着いたときに限り与え，その値は $r = 100$ とした。エージェントの 1 回の行動を 1 ステップ，スタートからゴールにたどり着くまでを 1 エピソード，8000 エピソードを 1 学習として，100 学習行った。

4.2.2　実験結果と考察

得られた実験結果を図 4 に示す。グラフの横軸はエピソード数を示し，縦軸はゴールに到達するまでにかかった平均ステップ数を示す。

動的なサブエージェントの追加を行う提案手法では，ゴールに到達するまでにかかった平均ステップ数は，学習が進むにつれてわずかに減少するが，2000 ステップ前後の大きな値で振動した。これより，動的なサブエージェントの追加を行う提案手法では，この迷路タスクについての学習は困難であることがわかる。この理由として，扱っている迷路タスクの広大さが挙げられる。問題空間が大きくなればなるほど，その適切な分割の組み合わせは少なくなると考えられる。このため，現在のサ

表 2　実験 2 パラメータ

提案手法	
行動価値の学習率 α_Q	0.25
HQ 値の学習率 α_{HQ}	0.1
位相変化量 β	$\exp(j\pi/12)$
割引率 γ	0.9
Boltzmann 温度 T	20
トレース数 N_e	4
サブエージェント数 M	12
サブエージェント追加周期 m	200 episodes
サブタスク学習不可判定ステップ数 n	50 steps

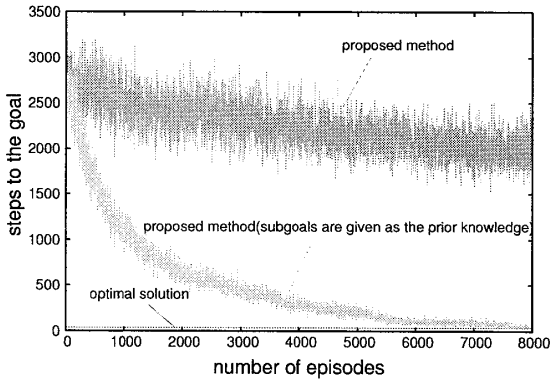

図 4　実験 2 エピソード毎の平均ステップ数

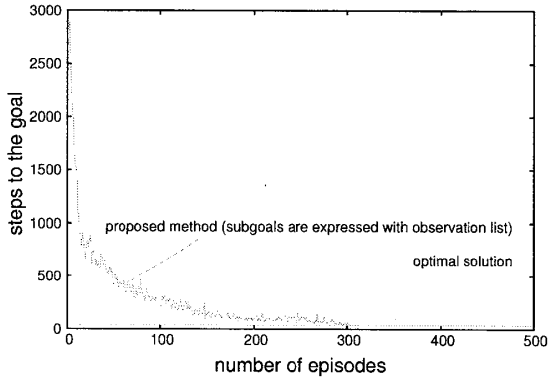

図 5　観測の系列をサブゴールとした場合のエピソード毎の平均ステップ数

ブゴール探索手法では，この迷路タスクの適切なサブゴールの組み合わせを発見することが困難となる。

次に，適切なサブゴールの組み合わせを事前知識として与えた場合について検証する。以上の方法で，上記に示したサブゴール探索可能性の問題とは切り離して考える。適切なサブゴールの組み合わせを事前知識として与えた場合には，ゴールに到達するまでにかかった平均ステップ数は，学習が進むにつれて徐々に減少していき，最終的には 45 ステップほどに収束した。しかし，収束するまでにかかるエピソード数は約 8000 エピソードと，サブゴールの組み合わせを事前知識として与えたのにも関わらず，学習に多大な時間を要していることがわかる。これより，事前知識としてサブゴールの組み合わせを与えた場合であっても，学習の成否は確率的に定まってしまうことがわかる。この理由として，サブゴールの表現方法の問題が挙げられる。現在の手法では，サブゴールをひとつの観測として表現している。このため，目標とする状態とは異なる状態をサブゴールとしてみなしてしまう，サブゴールの不完全知覚が生じうる。このため，元の問題空間を細かく分割する，一見適切なサブゴールの組み合わせであっても，必ずしも学習に成功することにはならない。また，このことが，適切なサブゴールの組み合わせをさらに制限することとなる。

そこで，観測の系列をサブゴールとして用いた場合について検証する。観測の系列をサブゴールとして用いた場合の学習結果を，図 5 に示す。図 4 と同様，グラフの横軸はエピソード数を示し，縦軸はゴールに到達するまでにかかった平均ステップ数を示す。図 5 より，約 300 エピソードと学習の早い段階でステップ数が最適解の 38 ステップに収束しており，適切な振る舞いを獲得できていることがわかる。このように，観測の系列をサブゴールとして用いることで，提案手法の性能が向上することが示された。

5　おわりに

本稿では，複素強化学習で解決できる問題のクラスの拡張を目的として，階層的に問題空間を分割する複素強化学習を提案した。

提案手法の有効性と問題点を明らかにするために，2 つの迷路タスクによる実験を行った。実験 1 では，従来の複素強化学習では学習が困難なタスクについて，提案手法では適切な振る舞いを獲得できることが確認された。しかし，動的なサブエージェントの追加を行うことで，使用するサブエージェント数が増加すると，ゴールに到達するには不必要なサブタスクが生じる問題がある。実験 2 では，サブゴールの探索手法に改善が必要であることや，サブゴールの不完全知覚の問題が明らかとなった。サブゴールの不完全知覚の問題については，観測の系列をサブゴールとして用いることで改善がみられた。しかし，観測の系列をサブゴールとして用いることは，メモリ消費の爆発的な増加を引き起こす可能性がある。このことは，省メモリ環境での動作を目標とする複素強化学習の理念にそぐわないため，観測の系列を用いることを安易に採用することはできない。したがって，系列を用いずにサブゴールの不完全知覚を解決する手法を考える必要がある。

今後は，観測の系列を用いずにサブゴールの不完全知覚を解決する手法の模索，サブゴールの探索手法の改善，階層化することと観測系列を用いたサブゴール表現がそれぞれどのような問題の特性に効果を与えるかの検証を行っていく予定である。

参考文献

[1] Richard S. Sutton and Andrew G. Barto(三上貞芳, 皆川雅章共訳): 強化学習, 森北出版株式会社, 2003.

[2] A. McCallum: Instance-based utile distinctions for reinforcement learning with hidden state, International Conference on Machine Learning, pp.387-395, 1995.

[3] M. Wiering and J. Schmidhuber: HQ-learning, Adaptive Behavior, Vol. 6, No. 2, pp. 219-246, 1998.

[4] A. Rodriguez, R. Parr, and D. Koller: Reinforcement learning using approximate belief states, In NIPS-12, 1999.

[5] 澁谷長史, 濱上知樹: 複素数で表現された行動価値を用いる Q-learning, 電子情報通信学会論文誌, vol.J91-D, no.5, pp1286-1295, 2008.

[6] T.Shibuya, T.Hamagami: Multiplied Action Values for Complex-valued Reinforcement Learning, Proc. of the International Conference on Electrical Engineering, I9FP0491, 2009pp.5-8, 2008.