

摂動完全均衡に基づく マルチエージェント部分観測可能マルコフ決定過程のプラン構築

Multiagent Planning with Trembling-hand Perfect Equilibrium in Multiagent POMDP

藪悠一*1 横尾真*2 岩崎敦*2
Yuichi Yabu Makoto Yokoo Atsushi Iwasaki

*1九州大学大学院システム情報科学府
Faculty of ISEE, Kyushu University

*2九州大学大学院システム情報科学研究院
Graduate School of ISEE, Kyushu University

Multiagent Partially Observable Markov Decision Processes (POMDPs) are emerging as a popular approach for modeling multiagent system in the presence of uncertainty. Since finding the optimal solution for multiagent POMDPs has been shown to be NEXP-Complete, a popular approach is finding a locally optimal, best response policies for each agent. In an existing approach (Joint Equilibrium-based Search for Policy with Nash Equilibrium, JESP-NE), the best response is defined based on an equilibrium concept called Nash equilibrium. However, in JESP-NE, the possibility that agents unexpectedly make some mistakes or deviate from their policies is not considered. In this paper, we extend the JESP framework utilizing an equilibrium concept called Trembling-hand Perfect Equilibrium (TPE), i.e., a best response is calculated assuming other agents deviate with a small probability. The experimental results show that the JESP-TPE outperforms the JESP-NE in problem settings where an initial policy is far from the optimal policies.

1. はじめに

部分観測可能マルコフ決定過程 (Partially Observable Markov Decision Process, POMDP) とは、不確実が存在する状況下でマルチエージェントシステムの振る舞いをモデル化する手法である。例えば、分散センサネットワークのようなシステムでは個々のセンサが正常に機能しているかどうか分からないといった不確実性を考慮する必要がある。POMDP はこうした状況をモデル化し、状況に応じて適切なプラン設計を行う手法として注目されている [Lesser 03]。

しかし、POMDP を用いたマルチエージェントシステムにおけるプラン構築の大域的最適解を求める問題は NEXP-Complete であることが証明されている [Bernstein 00]。そのため、現実的な時間でポリシーを求めるには、適用領域の一般性、ポリシーの形式、大域的最適解のうち一つ以上をあきらめなければならない。

このため、大域的最適解を必ず求められるとは限らないが現実的な時間でポリシーを求めることができるアルゴリズムとして、Joint Equilibrium-based Search for Policy with Nash Equilibrium (JESP-NE) が提案されている [Nair 04]。JESP-NE はナッシュ均衡となるポリシーを動的計画法を用いて局所的最適解を求めるアルゴリズムである。しかし、この JESP-NE は各エージェントのエラーを考慮しないため、あるエージェントが求めたポリシーと異なる振る舞いをしたとき、全体の効率性が著しく低下する可能性がある。

そこで、本論文では摂動完全均衡にもとづくマルチエージェント POMDP のプラン構築アルゴリズム (JESP with Trembling-hand Perfect Equilibrium, JESP-TPE) を提案する。ここで摂動とはエージェントが想定しているポリシーと異なる振る舞いをする、すなわちエラーを起こすことを意味している。さらに計算機実験を用いて JESP-NE と JESP-TPE の性能を比較する。JESP-TPE ではエージェントのエラーを考慮したプラン構築を行うため、問題設定によっては JESP-NE

より JESP-TPE が高い性能をもつ可能性を示し、エージェントのエラーに頑健であることを示した。

本文の構成を以下に示す。2章でマルチエージェントシステムのモデル化としてマルチエージェント協調決定問題を示す。3章で POMDP における例題である Tiger Problem を概説する。4章で従来手法である JESP-NE について述べる。5章で本論文で提案する JESP-TPE を説明し、6章で Tiger Problem を対象とした比較実験を行う。最後に7章で結論を示す。

2. マルチエージェント協調決定問題

本論文で対象とするマルチエージェント POMDP の具体的なモデルについて述べる。マルチエージェント POMDP のモデルにはマルチエージェント協調決定問題 (Multiagent Team Decision Problem, MTDP) [Pynadath 02] やネットワーク分散型 POMDP (Networked Distributed POMDP) [Nair 05] などがあるが、本論文では MTDP を対象とする。

MTDP では n 人のエージェントからなるチームに対して $\langle S, A, P, \Omega, O, B, R \rangle$ を定義する。 $S = \{s_1, \dots, s_n\}$ は各エージェントの状態を表す有限集合である。 A_i をエージェント i のとりうる行動の集合とし、各エージェントの行動の集合の組合せ A は、 $A = \prod_{1 \leq i \leq n} A_i$ と表す。さらに遷移関数 $P(s_i, \langle a_1, \dots, a_n \rangle, s_f)$ はあるエージェント i の状態が s_i であるとき、全てのエージェントの行動の組合せ $\langle a_1, \dots, a_n \rangle$ に対して状態が s_f に遷移する確率を表す。次に各エージェントの観測の組合せの集合を Ω とし、エージェント i の観測の集合は Ω_i とする。

これらの状態 s 、観測の組合せ ω に対して、観測関数 $O(s, \langle a_1, \dots, a_n \rangle, \omega)$ は、その観測の組合せ $\omega \in \Omega$ が発生する確率を与える。すなわち、観測関数 O は $O(s, \langle a_1, \dots, a_n \rangle, \omega) = O_1(s, \langle a_1, \dots, a_n \rangle, \omega_1) \cdot \dots \cdot O_n(s, \langle a_1, \dots, a_n \rangle, \omega_n)$ と表される。観測において重要となるのは、あるエージェントの観測が自分自身を含むすべてのエージェントの1つ前のステップにおける行動に対して独立となることである。

次にエージェント i はステップ t までの観測履歴に対する

連絡先: 藪悠一, 九州大学大学院システム情報科学府, 812-8581 福岡県福岡市東区箱崎 6-10-1, (092)642-3882, yabu@agent.is.kyushu-u.ac.jp

表 1: 遷移関数 P : *は Listen 以外の任意行動を表す

Action/Transition	$SL \rightarrow SL$ ($SR \rightarrow SR$)	$SL \rightarrow SR$ ($SR \rightarrow SL$)
$\langle *, * \text{ or } Listen \rangle$	0.5	0.5
$\langle * \text{ or } Listen, * \rangle$	0.5	0.5
$\langle Listen, Listen \rangle$	1.0	0.0

表 2: 観測関数 O : *は Listen 以外の任意行動を表す

State	Action	HL	HR	Reset
SL	$\langle Listen, Listen \rangle$	0.85	0.15	0.0
SR	$\langle Listen, Listen \rangle$	0.15	0.85	0.0
*	$\langle *, * \text{ or } Listen \rangle$	0.0	0.0	1.0
*	$\langle * \text{ or } Listen, * \rangle$	0.0	0.0	1.0

状態の確率分布である信念状態 $b_i^t \in B_i$ をもつ。ただし B_i は i が到達可能な信念状態の集合である。観測を得たエージェントは状態更新関数を使って信念状態を更新する。すなわち、エージェント i のステップ t における信念状態 b_i^t を観測履歴 $\bar{\omega}_i^t = \omega_i^1, \dots, \omega_i^t$ で表す: $b_i^t = \bar{\omega}_i^t$ 。

$R(s, (a_1, \dots, a_n))$ は報酬関数であり、状態 s において各エージェントの行動の組合せが $\langle a_1, \dots, a_n \rangle$ であるときに獲得する報酬を返す。この報酬関数から得られるポリシー π_i にしたがって、エージェント i は行動を決定する。このポリシーは信念状態と行動をマッピングしており、 $\pi_i(\bar{\omega}_i^t)$ を i のステップ t での行動とする。また、 $\pi = \langle \pi_1, \dots, \pi_n \rangle$ を結合ポリシーと呼ぶ。JESP-NE および JESP-TPE では最大ステップ数 T に対するエージェント全体の期待利得が最大となる結合ポリシーを見つけることを目的とする。

3. Tiger Problem

本章では、マルチエージェント POMDP の適用例を示す。シングルエージェントでの POMDP の適用例として用いられる Tiger problem がある。本論文では、この Tiger Problem をマルチエージェント用に変更されたもの [Nair 04] を扱う。

2 人のエージェントが存在し、目の前に二つの部屋とその扉がある。片方の部屋にはお腹をすかせたトラがいて、もう片方の部屋にはお金がある。そしてエージェント等はどちらの部屋にトラがいるのかを知らない。状態 $S = \{SL, SR\}$ は SL がトラが左の部屋にいることを表し、 SR は右の部屋にトラがいることを表す。エージェントはどちらかの部屋に入るか、部屋の様子をうかがうことができる: $A_1 = A_2 = \{OpenLeft, OpenRight, Listen\}$ 。どちらかのエージェントが扉を開けると観測 *Reset*、全エージェントが行動 *Listen* を選択すると、観測 *HL* か *HR* を得る: $\Omega_1 = \Omega_2 = \{Reset, HL, HR\}$ 。

遷移関数 P 、観測関数 O 、報酬関数 R を表 1、表 2、表 3 に示す。報酬関数は、*Listen* の報酬を $-c$ とし、トラのいる部屋に一人で入ったときの報酬を $-d$ とし、二人でトラのいる部屋に入ったときの報酬を $-d/2$ とする。これは二人で一緒にトラのいる部屋に入った場合は、一人でトラのいる部屋に入ったときより傷を負わないで済むことを表す。 c, d は 6 章での実験において適宜変更する。

表 3: 報酬関数 R : c は *Listen* のコストを表し、 d はトラのいる部屋に一人で入ったときのコストを表す。 $d/2$ は、二人でトラのいる部屋に入ったときのコストを表す。

Action/State	SL	SR
$\langle OpenRight, OpenRight \rangle$	+20	$-d/2$
$\langle OpenLeft, OpenLeft \rangle$	$-d/2$	+20
$\langle OpenRight, OpenLeft \rangle$	$-d$	$-d$
$\langle OpenLeft, OpenRight \rangle$	$-d$	$-d$
$\langle Listen, Listen \rangle$	$-c$	$-c$
$\langle Listen, OpenRight \rangle$	+10	$-d$
$\langle OpenRight, Listen \rangle$	+10	$-d$
$\langle Listen, OpenLeft \rangle$	$-d$	+10
$\langle OpenLeft, Listen \rangle$	$-d$	+10

4. ナッシュ均衡に基づく従来のプラン構築

Bernstein 等によって、マルチエージェント POMDP において大域的最適解を求めることは、NEXP-Complete であることが示されている [Bernstein 00]。そこで、大域的最適解はあらかじめ局所的最適解を求めるアプローチを用いる。

本章では、マルチエージェント POMDP においてナッシュ均衡となるプランを構築するアルゴリズムである Joint Equilibrium-based Search for Policies with Nash Equilibrium (JESP-NE) を紹介する [Nair 04]。

ナッシュ均衡とは、どのエージェントも相手が戦略を変えない限り、自分の戦略を変える誘因がない戦略の組合せである。

JESP-NE の重要な点は、あるエージェントのポリシーを考えると、他の全てのエージェントのポリシーを固定し、それに対して期待利得が最大となるようなポリシーを見つけることである。この過程をポリシーが均衡に達するまで繰り返す。^{*1}

ここで、重要なことは、他のエージェントのポリシーを固定することで、エージェント i のシングル POMDP として扱うことができるということである。しかし、信念状態の確率分布はエージェント i の観測履歴だけではなく、他のエージェントの行動や観測履歴にも依存する。そのため新たに、信念状態を拡張した episode e とエージェント i のマルチエージェント信念状態 B が定義されている。^{*2}

動的計画法を用いた JESP-NE の重要な要素は最大ステップ数 T での価値関数である。価値関数 $V_i(B^t, b)$ は、初期状態 b から最終ステップ T までのそれぞれのステップで得られるエージェント全体の期待利得である。各ステップで選択可能な行動の中から、価値関数を最大化する行動を見つけることで、最適なポリシーを構築する。

5. 摂動完全均衡に基づくプラン構築

本章では、摂動完全均衡の概念に基づくマルチエージェント POMDP のプラン構築アルゴリズム (JESP with Trembling-hand Perfection Equilibrium, 以下 JESP-TPE) を提案する。

5.1 摂動完全均衡

摂動完全均衡とは、各エージェントの戦略が、他のエージェントが均衡から外れた戦略をとる小さな可能性があっても依然として最適反応となることである。

ここで、3 章の tiger problem を用いて例を示す。最大ステップ数を 2、 $c = 2$ 、 $d = 40$ として、表 4 のようなポリシーにつ

*1 アルゴリズムの詳細は参考文献 [Nair 04] を参照。

*2 紙面の都合上定義を省略する

いて考える．表 4 はある観測を得たときの，エージェント i, j のポリシーを示している．

表 4: 最大ステップ数 2 での各エージェントのポリシー

$t = 0$		$t = 1$	
観測	ポリシー	観測	ポリシー
(n/a)	(Listen, Listen)	HL	(OpenRight, OpenRight)
		HR	(OpenLeft, OpenLeft)
		Reset	(OpenLeft, OpenRight)

このとき，エージェントのエラーを考えなければ遷移し得る全ての状態において各エージェントのポリシーは最適反応となっているため，このポリシーはナッシュ均衡となっている．しかし，ここでエージェント j がエラーを起こしてステップ 1 で *OpenLeft* を選択すると仮定する．このとき，エージェントがエラーを起こさないときは遷移するはずのなかった状態，つまりステップ 1 で観測 *Reset* を得て遷移する状態でのポリシーも考慮しなければならないこのときのポリシーは明らかに互いに最適反応とはなっていない．以上より，表 4 はナッシュ均衡ではあるが摂動完全均衡とはなっていない．例えば，ステップ 1 で観測 *Reset* を得たときのポリシーを (Listen, Listen) とすると，摂動によって遷移する状態も含めて全ての状態におけるポリシーが互いに最適な反応となり，摂動完全均衡となる．

これらのことから分かるように，摂動完全均衡となるポリシーは必ずナッシュ均衡となるが，その逆は言えない．

5.2 アルゴリズムの詳細

JESP-NE では，ナッシュ均衡となる局所最適解を求めることで，高いパフォーマンスを実現した．しかし，頑健性の面ではエージェントのエラーを考慮しないという問題点がある．

本論文では，各エージェントのポリシーが摂動完全均衡となるプラン構築アルゴリズムである JESP-TPE を提案する．

JESP-TPE では，各エージェントのポリシーを決める際，他のエージェントが $\epsilon (> 0)$ の確率でポリシーに反する行動を選択すると考える．*GETVALUEACTION-TPE* はある行動に対する期待利得を計算し，*UPDATE-TPE* は状態更新関数である．両アルゴリズムは，JESP-NE にエージェントのエラーを考慮したアルゴリズムとなっている．具体的には，5, 13 行目のように，報酬関数や観測関数を用いるとき，他のエージェントがポリシーどおりに行動する確率を $1 - \epsilon$ ，ポリシーに反する行動をする確率を $\epsilon/|A_j| - 1$ として計算する．

その結果，JESP-NE では遷移しなかった信念状態にも遷移する可能性が生じる．図 1 にエージェントのエラーを考慮したマルチエージェント信念状態を示す．摂動を考慮しないときとの違いは， B_1'' で episode (*SL(Reset)*), (*SR(Reset)*) の確率が 0.5 となっていることである．これは摂動により，エージェント j がポリシーに反して *OpenLeft* あるいは *OpenRight* を選択する可能性が生じるためである．摂動を考慮しないときは， B_1'' に遷移しない．このため，JESP-TPE では，JESP-NE では考慮されなかった B_1'' 以降のポリシーも探索する．

価値関数 $V_i^{a_i}$ はステップ t でエージェント i が a_i という行動を取ったときの期待利得を表し，*GETVALUEACTION-TPE* で計算する． B_i^{t+1} は，ステップ t でエージェント i が行動 a_i を行い，観測 ω_i^{t+1} を得たときに更新された信念状態である． B_i^{t+1} は，*UPDATE-TPE* で計算される．

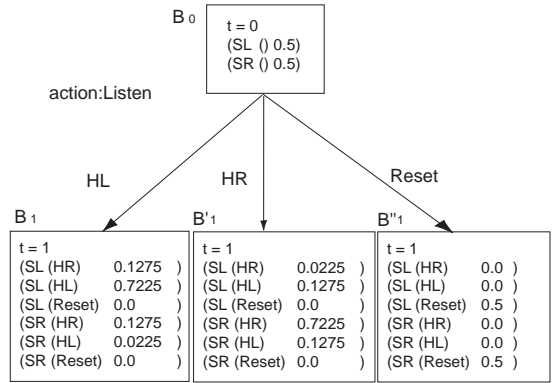


図 1: ステップ 1 までの信念状態：ポリシーの初期設定 Listen 摂動有り

Algorithm 1

GETVALUEACTION-TPE(B^t, a_i, b, π_j, t, T)

```

1:  $\delta_j \leftarrow \pi_j(b)$ 
2:  $value \leftarrow 0$ 
3: for all  $a_j \in A_j, e^t = \langle s^t, \bar{\omega}_j \rangle$  s.t.  $B^t(e^t) > 0$  do
4:   if  $a_j == \delta_j(\bar{\omega}_j, T - t)$  then
5:      $value \leftarrow B^t(s^t, \bar{\omega}_j) \cdot R(s^t, \langle a_i, a_j \rangle) \cdot (1 - \epsilon)$ 
6:   else
7:      $value \leftarrow B^t(s^t, \bar{\omega}_j) \cdot R(s^t, \langle a_i, a_j \rangle) \cdot \frac{1}{|A_j| - 1} \epsilon$ 
8: for all  $\omega_i \in \Omega_i$  do
9:    $B^{t+1} \leftarrow UPDATE(B^t, a, \omega_i, \delta_j)$ 
10:   $prob \leftarrow 0$ 
11:  for all  $a_j \in A_j, s^t, e^{t+1} = \langle s^{t+1}, \bar{\omega}_j \rangle$ 
12:    s.t.  $B^{t+1}(e^{t+1}) > 0$  do
13:    if  $a_j == \delta_j(\bar{\omega}_j, T - t)$ 
14:       $prob \leftarrow B^t(s^t, \bar{\omega}_j) \cdot P(s^t, \langle a_1, a_2 \rangle, s^{t+1}) \cdot$ 
15:         $O_1(s^{t+1}, \langle a_1, a_2 \rangle, \omega_1) \cdot$ 
16:         $O_2(s^{t+1}, \langle a_1, a_2 \rangle, \omega_2) \cdot (1 - \epsilon)$ 
17:    else
18:       $prob \leftarrow B^t(s^t, \bar{\omega}_j) \cdot P(s^t, \langle a_1, a_2 \rangle, s^{t+1}) \cdot$ 
19:         $O_1(s^{t+1}, \langle a_1, a_2 \rangle, \omega_1) \cdot$ 
20:         $O_2(s^{t+1}, \langle a_1, a_2 \rangle, \omega_2) \cdot \frac{1}{|A_j| - 1} \epsilon$ 
21:   $value \leftarrow prob \cdot GETVALUE(B^{t+1}, t + 1)$ 
22: return  $value$ 

```

6. 提案アルゴリズムの計算機実験による評価

本章では，3 章の Tiger Problem を用いてエージェント全体の期待利得の比較実験を行い，その結果を示し，考察する．

本実験では，初期状態に等確率で SL か SR を与え，初期ポリシーを *Listen* とする．次に摂動の確率である ϵ は十分小さな値として $1.0E - 13$ とする．これはエージェント全体の期待利得に影響を与えないようにするためである．また，状態の確率分布は摂動によって遷移する信念状態の中で確率分布を用いて正規化するため， ϵ の変化に依存しない．また， c, d, T は表 5 に示す 3 つの設定を用いた．

図 2 は表 5 で与えた設定に対して，エージェントが獲得したポリシーから得られるすべてのステップの期待利得を示している．設定 1 では JESP-NE および JESP-TPE の期待利得は約 2.25 となり，ほぼ等しい．設定 2 ではそれぞれ $-9.70, 10.0$ となり，また設定 3 では $-4.75, 15.0$ となり，JESP-TPE が与える期待利得は JESP-NE から大きく改善された．

ここで設定 2 での実験を見てみる．1 回目のポリシーの計算

Algorithm 2 UPDATE-TPE($B^t, a_i, \omega_i, \pi_j$)

```

1: for all  $e_i^{t+1}$  do
2:    $B^{t+1}(e^{t+1}) \leftarrow 0$ 
3:   for all  $a_j \in A_j, s^t \in S$  do
4:     if  $a_j = \delta_j(\bar{\omega}_j)$  then
5:        $B^{t+1}(e^{t+1}) \leftarrow B^t(e^t) \cdot P(s^t, \langle a_i, \dots, a_j \rangle, s^{t+1}) \cdot O(s^t, \langle a_i, \dots, a_j \rangle, \omega) \cdot 1 - \epsilon$ 
6:     else
7:        $B^{t+1}(e^{t+1}) \leftarrow B^t(e^t) \cdot P(s^t, \langle a_i, \dots, a_j \rangle, s^{t+1}) \cdot O(s^t, \langle a_i, \dots, a_j \rangle, \omega) \cdot \frac{1}{|A_j|-1} \epsilon$ 
8:   normalize  $B^{t+1}$ 
9:   return  $B^{t+1}$ 

```

表 5: Listen cost c , tiger cost d , 最大ステップ数 T の設定

	Listen cost c	tiger cost d	最大ステップ数 T
設定 1	8	20	4
設定 2	14	20	4
設定 3	14	20	5

を行うとき、ステップ 1 では両アルゴリズムとも互いに Listen を選択するのが最良であった。このとき 5.2 節で説明したように従来アルゴリズムでは、図 1 の状態 B_1'' は全て 0 になり、 B_1'' 以降のポリシーは計算されない。しかし提案アルゴリズムでは、エージェントのエラーにより信念状態は図 1 の B_1'' のようになり、それ以降のポリシーも計算され最適化される。このことにより、2 回目以降のポリシーの計算で、エージェントが *OpenLeft* や *OpenRight* の期待利得を計算するとき、状態 B_1'' 以降のポリシーがある程度最適化されている。そのため、ステップ 1 での *OpenLeft* や *OpenRight* によって得られる期待利得が増加し、Listen によって得られる期待利得を上回りポリシーが変化する。設定 3 で改善された理由も設定 2 の場合と同様である。

しかし、設定 1 の実験では期待利得は変化していなかった。設定 1 の実験も 1 回目のポリシーの計算では Listen を選択するのが最良となっていた。ここで状態 B_1'' 以降のポリシーは最適化されるが、そのことによる *OpenLeft* と *OpenRight* の期待利得の増加が Listen の期待利得を上回らなかったため、ポリシーは変化しなかった。

そして、どの設定の実験でも JESP-TPE は摂動によって遷

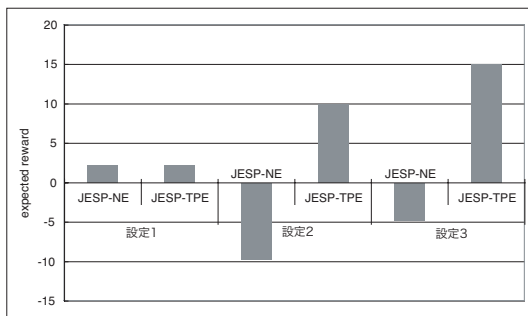


図 2: JESP-NE および TPE が与える期待利得 (設定 1,2,3 については表 5 を参照)。

移する信念状態以降の期待利得が増加していることが分かった。しかし、摂動する確率が微小であるため全体の期待利得にはほとんど影響はなかった。

以上の実験結果は、初期ポリシーと解となるポリシーがある程度異なる場合、JESP-TPE は JESP-NE が与えるポリシーより高い期待利得を与えるポリシーを与えることを示している。それゆえ JESP-TPE は、摂動完全均衡を用いてエージェントのエラーを考慮することで、エージェントのエラーに対して頑健となっていると言える。

7. おわりに

本論文では、マルチエージェント POMDP における摂動完全均衡となるポリシーの探索アルゴリズムである JESP-TPE を提案した。

既存の手法である JESP-NE は、ナッシュ均衡となるポリシーを探索するアルゴリズムで、高速かつ高パフォーマンスを実現した。しかし、JESP-NE はエージェントがエラーを起こしたとき、それ以降のポリシーは考慮しないという問題がある。

これに対して JESP-TPE はエージェントがエラーを起こすことを前提としたプラン構築を行った。

実験の結果、JESP-TPE では、JESP-NE で考慮されなかった摂動によって遷移する信念状態以降のポリシーも最適化することで、エージェントのエラーに頑健となり、さらには JESP-NE とは異なるより良い均衡に到達することが分かった。

一方で、摂動を許すことで、探索空間が指数関数的に増加し、実行時間が増大するという問題点がある。今後の研究では、JESP-TPE の実行時間を短縮するアルゴリズムの提案や、ND-POMDP などの他の POMDP [Nair 05] へ適用することで、提案アルゴリズムの応用可能な領域を広げていきたい。

参考文献

- [Bernstein 00] Bernstein, D. S., Zilberstein, S., and Immerman, N.: The Complexity of Decentralized Control of Markov Decision Processes, in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pp. 32–37 (2000)
- [Lesser 03] Lesser, V., Ortiz, C., and Tambe, M.: *Distributed Sensor Networks: A Multiagent Perspective*, Kluwer (2003)
- [Nair 04] Nair, R., Roth, M., Yokoo, M., and Tambe, M.: Communication for Improving Policy Computation in Distributed POMDPs, in *Proceedings of the Third International Joint Conference on Agents and Multiagent Systems (AAMAS-04)*, pp. 1098–1105 (2004)
- [Nair 05] Nair, R., Varakantham, P., Tambe, M., and Yokoo, M.: Networked Distributed POMDPs: A Synthesis of Distributed Constraint Optimization and POMDPs, in *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)* (2005)
- [Pynadath 02] Pynadath, D. V. and Tambe, M.: The Communicative Multiagent Team Decision Problem: Analyzing Teamwork Theories and Models, *Journal of Artificial Intelligence Research*, Vol. 16, pp. 389–423 (2002)