2J3-2

ネットワーク分散部分観測可能マルコフ決定過程における プラン構築への通信の導入

Introducing Communication in Networked Distributed POMDPs

岩成 祐樹 Yuki Iwanari　籔 悠一 Yuichi Yabu　田崎 誠 Makoto Tasaki　横尾 真 Makoto Yokoo

九州大学大学院システム情報科学府
Graduate School of ISEE, Kyushu University

Distributed Partially Observable Markov Decision Process (Dis-POMDP) is a popular approach for modeling multi-agent systems acting in uncertain domains. In particular, Networked Distributed POMDP (ND-POMDP) can handle large-scale problems by utilizing the locality in agents' interaction. However, the size of a local policy grows exponentially for the length of the policy. To overcome this problem, we introduce on-line communication among agents, i.e., agents periodically communicate their observation/action history with each other. After a communication phase, agents can start from a new synchronized belief state. Also, we introduce a technique similar to point-based value iteration to use a fixed number of synchronized belief belief states. As a result, we can use a set of small, fixed size policies instead of a single huge policy with an exponential size. Our experimental results show that we can obtain much longer policies than existing algorithms as long as the interval between communications is small.

1. はじめに

不確実性の下で協調的に動作するマルチエージェントの振舞いをモデル化する手法として，ネットワーク分散部分観測可能マルコフ決定過程 (Networked Distributed Partially Observable Markov Decision Process, ND-POMDP) が提案されている．例えば，大規模なセンサネットワークでは，ある目標を追跡するために直接協力する必要のあるエージェントは全体のごく一部である．ND-POMDP は，このような分散センサネットワークや分散無人機群などのエージェント間の相互作用による局所性が存在する問題をモデル化することができる．

ND-POMDP では，エージェントは行動をポリシーによって決定する．ポリシーとは，ある時点までのエージェント自身の行動履歴と，得られた観測履歴に対して，次に取るべき行動を指定したものであり，木構造で表現される．ポリシーを構築する従来アルゴリズムとして，LID-JESP (Locally Interacting Distributed Joint Equilibrium-based Search for Policies) [Nair 04]，SPIDER (Search for Policies In Distributed EnviRonments) [Varakantham 07] が提案されている．これらのアルゴリズムは局所性を利用しており，エージェント数の増加に対する計算量の増加を抑えている．しかしながら，エージェントの持つポリシーのサイズは行動回数（ステップ数）に対して指数オーダであるという課題があった．

本論文では，エージェントがプランの実行時に定期的な通信を行うことにより，ポリシーのサイズを削減する方法を示す．

具体的には，全てのエージェントは互いに自分の観測履歴と行動履歴を定期的に交換する．交換した情報に基づいて，全エージェントの信念状態を統一することで，得られた新しい信念状態において適切なポリシーを選択することが可能となる．つまり，指数的に増加する 1 つの巨大なポリシーを構築する代わりに，小さな定数サイズのポリシーを複数利用することが可能となる．

しかしながら，通信後に到達する可能性のある信念状態の数は，プランの長さに対して指数的に増加する．このため，定数サイズのポリシーを指数個準備しておく必要が生じる．

本論文では，この問題点を解決するため，Point-Based Value Iteration (PBVI) アルゴリズム [Pineau 06, Spaan 05] と類似した，定数個の信念状態に関してのみポリシーを準備しておき，他の信念状態に関しては，これらの定数個のポリシーから最も良いものを選ぶという近似手法を提案する．

我々は上記のアイデアを用いて ND-POMDP を拡張した ND-POMDP-Comm を提案する．さらに，従来アルゴリズムである LID-JESP 及び SPIDER を ND-POMDP-Comm に拡張した，LID-JESP-Comm，SPIDER-Comm を提案する．計算機実験により，提案アルゴリズムにより長期間のポリシーが構築可能となることを示す．

2. ND-POMDP

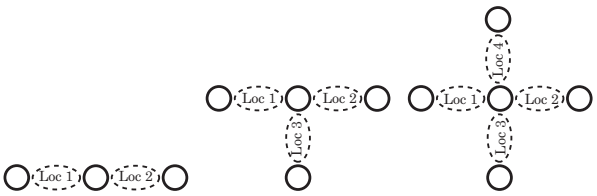
2.1 モデル

ND-POMDP は，マルチエージェントシステムにおけるエージェント間の相互作用の局所性を表現したものであり，$\langle S, A, P, \Omega, O, R, b\rangle$ で定義される．

$S = \times_{1\le i\le n} S_i \times S_u$ は状態集合である．S_i はエージェント i の内部状態の集合であり，S_u はエージェントの行動に影響を受けない状態の集合である．例えば天候やセンサネットワークにおけるターゲットの位置等を示す．A_i はエージェント i がとり得る行動の集合であり，$A = \times_{1\le i\le n} A_i$ は全てのエージェントの行動の組合せの集合を示す．

次に，$P(\vec{s}, \vec{a}, \vec{s'}) = P_u(s_u, s'_u) \cdot \prod_{1\le i\le n} P_i(s_i, s_u, a_i, s'_i)$ は状態の遷移関数を示す．具体的には，状態 $\vec{s} = \langle s_1, \ldots, s_n, s_u\rangle$ において $\vec{a} = \langle a_1, \ldots, a_n\rangle$ という行動をとった場合，状態 $\vec{s'} = \langle s'_1, \ldots, s'_n, s'_u\rangle$ に遷移する確率を返す．エージェント i が得る観測の集合を Ω_i としたとき，$\Omega = \times_{1\le i\le n}\Omega_i$ はエージェン

連絡先: 岩成 祐樹 九州大学大学院システム情報科学府 812-0395 福岡県福岡市西区元岡 744 番地，(092)802-3576, iwanari@agent.is.kyushu-u.ac.jp



(1) 3 センサネットワーク　(2) 4 センサネットワーク　(3) 5 センサネットワーク

図 1: センサネットワークの例

トが得る観測の組合せの集合である．このとき，$O(\vec{s}', \vec{a}, \vec{\omega}) = \prod_{1 \leq i \leq n} O_i(s_i', s_u', a_i, \omega_i)$ は観測関数であり，状態 $\vec{s}$ で行動 $\vec{a}$ をとったとき，観測 $\vec{\omega} = \langle \omega_1, \ldots, \omega_n \rangle \in \Omega$ を得る確率を返す．さらに，l は相互作用のあるエージェントの部分集合であり，$R(\vec{s}, \vec{a}) = \sum_l R_l(s_{l_1}, \ldots, s_{l_r}, s_u, \langle a_{l_1}, \ldots, a_{l_r} \rangle)$ は報酬関数を示す．すなわち，$R_l(\cdot)$ は局所的な報酬関数であり，l 中のエージェントの行動と状態を引数として，これらのエージェントに与えられる報酬を返す．この報酬関数に基づき，エージェント間の相互作用を表すグラフ $G = (A_g, E)$ が定義される．ノードの集合 A_g の要素は各エージェントであり，エージェント間の報酬関数による依存関係を表すハイパーエッジの集合 E の各要素は，相互作用のあるエージェントの部分集合 l に含まれるエージェントを連結する．

b は信念状態を表し, $b(\vec{s}) = b_u(s_u) \cdot \prod_{1 \leq i \leq n} b_i(s_i)$ と定義される．b_u は状態が s_u である確率を表す．b_i はエージェント i の内部状態が s_i である確率を表し, 各エージェントはそれぞれ独立した信念状態を持つ．エージェント i がステップ t にもつ信念状態 $b_i^t \in B_i$ は, t までの観測履歴 $\vec{\omega}_i^t = \langle \omega_i^1, \omega_i^2, \cdots, \omega_i^t \rangle$ から一意に定まる．B_i は i の到達しうる信念状態の集合である．

エージェント i は, 行動をポリシー π_i によって決定する．π_i は, 全ての b_i^t に対して, そこで取るべき行動をマッピングしたものである．本研究の目標は, 初期信念状態 b, 最大ステップ数 T に対してシステム全体の報酬を最大化するエージェント全体のポリシーの組合せ (結合ポリシー) $\pi = \langle \pi_1, \ldots, \pi_n \rangle$ を探索することである．

2.2 適用例

ND-POMDP により表現可能な応用事例として，図 1 に示すセンサネットワークを紹介する．図 1 において，それぞれのセンサネットワークで起こりうる状態は，(1) の場合，ターゲットが両方に存在する，どちらか片方にのみ存在する，どちらにも存在しない場合の 4 通りである．(2) の場合は 8 通り，(3) の場合には 16 通りである．本論文では以下，ターゲット発見時の利得は，Loc1, Loc2, Loc3, Loc4 それぞれで +35, +45，+30，+40，未発見時の利得は −5 と仮定する．

3. ND-POMDP-Comm

本章では以下，ND-POMDP にオンラインの通信を導入した ND-POMDP-Comm の概要を示す．ND-POMDP-Comm では，通信後の各信念状態に対して任意の k ステップの小さな結合ポリシーを構築し，ポリシーの大きさによる組合せ爆発を回避する．

ND-POMDP-Comm の基本的な通信プロセスを下記に示す．

- 全てのエージェントは，共通した初期状態から k ステップのポリシーを実行する．
- k ステップ経過後，エージェントは通信フェーズへと移行する．通信フェーズでは，全てのエージェント間で行動履歴と観測履歴の情報を交換する．
- 通信後，全エージェントの信念状態が統一される．以後，この信念状態に対応するポリシーを実行する．

結合ポリシーは得られる可能性のある全ての信念状態に対して準備する必要がある．例えば，初期状態から k ステップのポリシーを実行後に通信を行った場合を考える．n エージェントが存在し，各エージェントが 1 ステップごとに得る観測が $|\Omega|$ 通りである場合，通信結果となりうる信念状態の数は $O(|\Omega|^{n*k})$ となる．また，これらの全ての状態に対して，さらに k ステップ先のポリシーを実行した先の状態は同様に増加し，信念状態の数はポリシーの長さに対して指数的に増加する．

この問題に対応するため，以下に示す Point-Based Value Iteration (PBVI) アルゴリズム [Pineau 06, Spaan 05] と類似した方法により，定数個の信念状態に関するプランを用いる方法を導入する．

3.1 PBVI アルゴリズム

3.1.1 アルゴリズムの概要

PBVI アルゴリズムは，シングルエージェント POMDP における近似的なポリシー構築手法である．全ての到達しうる状態に対して最適なポリシーを構築する代わりに，状態空間のいくつかの代表点でのみ最適なポリシーを探索し，そのポリシーに応じて状態空間全体を覆うベクトル (α ベクトル) を作成する．これらのベクトルを用いることで，状態空間上における任意の状態に対して近似解となるポリシーを構築している．本論文では，通信後の信念状態に関して，同様に有限個の代表点に関してベクトルを作成することにより，通信後の状態数の組合せ爆発を回避する．

3.1.2 α ベクトルによるポリシーの近似手法

任意の結合ポリシーにおける期待利得は，初期信念状態の要素を用いて図 2 のようなベクトル (α ベクトル) として表現できる．例えば，図 2 の α は，初期信念状態 b における最適な結合ポリシー π を，任意の信念状態で実行した場合の期待利得を表すベクトルである．信念空間上における最適な期待利得は，複数の α ベクトルを用いた区分的に線形な凸関数 (Piece-Wise Linearity and Convexity, PWLC) として表現できる．

本論文では, 通信後に得る信念状態全てに対して最適な α ベクトルを探索する代わりに, 図 2 の右のように, 各代表点 (b', b'', b''') に対して最適解となる α ベクトル $(\alpha_0, \alpha_1, \alpha_2)$ を探索する．$(\alpha_0, \alpha_1, \alpha_2)$ から構築した PWLC 関数を, 通信後に得る信念状態に対する, 近似解として与える．例えば図 2 の左では, 任意の信念状態 b に対して最適解である α が与えられる．これに対して提案手法を用いた図 2 の右では, b に対して最適解である α の代わりに, 代表点を用いて構築した近似解 α_1 を与える．

図 3 に ND-POMDP-Comm によって構築する 5 ステップのポリシーの例を示す．ステップ 3 の黒丸は通信を表す．この例では，各代表点からステップ 4, 5 のポリシー $\pi_{10}, \cdots, \pi_{13}$ を構築している．これらのポリシーに対応する α ベクトルを構築し，ステップ 3 の通信によって得る信念状態に対して $\pi_{10}, \cdots, \pi_{13}$ のいずれかを与える．

3.2 詳細なアルゴリズム

ND-POMDP-Comm の擬似コードを Algorithm 1 に示す．π^* はエージェント全体の結合ポリシー，$\vec{\alpha}$ は結合ポリシーから

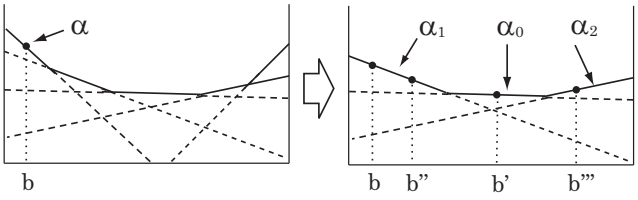


図 2: 代表点を用いた近似の例　左：最適解, 右：近似解

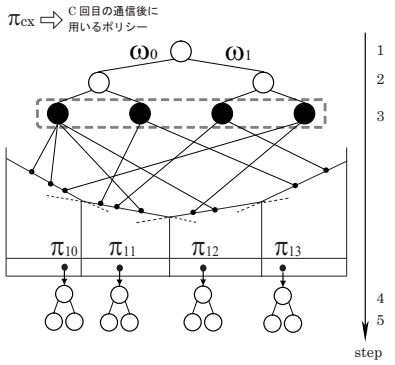


図 3: ND-POMDP-Comm によって構築されるポリシー

得る価値関数を表す．$CommPhase$ は通信回数を表す．ND-POMDP-Comm は動的計画法を用いており，最後の k ステップから計算する．まず，先述の代表点選択手法を利用し代表点の集合 B を作る (2 行目)．全ての信念状態 $b \in B$ に対して k ステップの最適な結合ポリシー，及びそのポリシーの α ベクトルを求め，結合ポリシーを $\pi^*[b, CommPhase]$ に格納する (5-7 行目)．各通信ごとに，後述の LID-JESP-Comm, SPIDER-Comm を利用した FINDPOLICY 関数を呼び出して結合ポリシーと α ベクトルを求める．

4. LID-JESP-Comm

本章では，LID-JESP [Nair 04] を ND-POMDP-Comm に拡張した LID-JESP-Comm について説明する．LID-JESP はナッシュ均衡に基づいた局所的最適解を求めるアルゴリズムであり，各エージェントのポリシーは，他のエージェントの持つポリシーに対する最適反応となっている．具体的には，最初に各エージェントにランダムなポリシーを割り当て，近隣のエージェントへの最適反応となるようにポリシーを改善していく．

(i) 各代表点で，LID-JESP を利用して，最後の通信以降の k ステップに関して，エージェント毎に他のエージェントのポリシーに対する最適反応を求め，均衡となる結合ポリシーを求める．

(ii) 各代表点において，最後から 2 回目の通信以降の k ステップに関して，均衡となる結合ポリシーを求める．現在の k ステップに関しては，各エージェントは期待利得を求めるために隣接したエージェントのポリシーのみを必要とする．一方，通信後の期待利得を計算するために全てのエージェントのポリシーを考慮し，新たな信念状態を得る．これらの各状態において，(i) で求めた結合ポリシーの期待利得を利用する．

(iii) 最後から 3 回目の通信以降の各代表点において，均衡となる結合ポリシーを求める．以下，同様の手順を繰り返す．

Algorithm 1 ND-POMDP-Comm($k, CommPhase$)

```
1: initialize α⃗*, π* ← null
2: B ←BeliefExpansion(b_init)
3: while CommPhase ≥ 0 do
4:    for all b ∈ B do
5:       ⟨π*[b, CommPhase], α⃗⟩ ←
         FINDPOLICY(b, root, null, −∞, k, α⃗*)
6:       α⃗*[CommPhase] ← α⃗*[CommPhase]||α⃗
7:    CommPhase = CommPhase − 1
8: return π*
```

5. SPIDER-Comm

本章では，SPIDER [Varakantham 07] を ND-POMDP-Comm に拡張した SPIDER-Comm を提案する．SPIDER は，ヒューリスティック関数を用いて大域的最適解を求めるアルゴリズムである．エージェント間のハイパーエッジ E から深さ優先探索木を構築し，分枝限定法を利用した探索を行う．未決定のポリシーから得る期待利得に対して，マルコフ決定過程を用いたヒューリスティック値 (MDP ヒューリスティック) を求めることで，計算量を削減する．

5.1 アルゴリズムの概要

SPIDER では相互作用による局所性を利用して，計算量を削減している．しかし，通信を行う場合，通信後の信念状態は全エージェントのポリシーに依存するため，局所性を利用できない．そこで，SPIDER-Comm では貪欲法を利用し，SPIDER とは異なり，全ての結合ポリシーの組合せのチェックを行わない．このため，SPIDER-Comm は大域的最適解を得ることを保証できないが，相互作用の局所性を利用することで計算量の削減が可能となる．

(i) 各代表点で，SPIDER を利用して，最後の通信以降の k ステップに関して，エージェント全体における大域的最適となる結合ポリシーを求める．

(ii) 各代表点において，最後から 2 回目の通信以降の k ステップに関するポリシーを求める．まず，エージェント i の全てのポリシーを列挙する．i が葉ノードであれば，ポリシーが未決定のエージェントにランダムな初期ポリシーを与え，k ステップの期待利得と通信後の期待利得を計算する．そして，期待利得の合計が最大となるポリシーを選択する．一方，i が葉ノードではなかった場合，各ポリシーのヒューリスティック値を計算し，その降順にポリシーをソートする．そして，ソートした各ポリシーにおいて，子エージェントに対して再帰的に，エージェント i のポリシーに対する最適反応となるポリシーを探索し，最後にこれまでの計算したポリシーと比べ，より良いポリシーであればそれを保存する．

(iii) 最後から 3 回目の通信以降の各代表点において，(ii) と同様に結合ポリシーを求める．以下，同様の手順を繰り返す．

5.2 ヒューリスティック関数

SPIDER-Comm では，通信間の k ステップに対する期待利得と通信後の期待利得の見積りを求めるためのヒューリスティック関数を構築する必要がある．従来の SPIDER では，MDP に基づくヒューリスティック関数，すなわち，すべてのエージェントが単一の MDP だと仮定した場合の期待利得を用いていた．一方，SPIDER-Comm では，通信後の信念状態

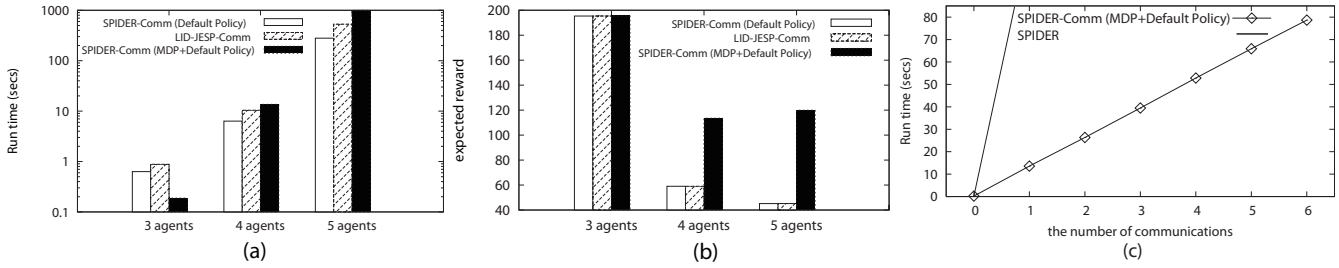


図 4: (a) SPIDER-Comm と LID-JESP-Comm の実行時間，(b) 期待利得，(c) 通信回数に対する SPIDER-Comm の実行時間

を推定する必要があり，MDP ヒューリスティック関数をそのまま利用することができない．本論文では，(1) ポリシーが未定のエージェントに関しては予め定められた初期ポリシーを用いると仮定して，通信間の k ステップに対する期待利得と通信後の期待利得を求める方法 (Default policy)，(2) 通信間の k ステップに対しては MDP ヒューリスティック関数を用い，通信後の期待利得に対しては default policy と同様に求める方法 (MDP+Default policy) の二種類の方法を開発した．これらのヒューリスティック関数は適格性を満たさないが，探索アルゴリズム自体が貪欲法に基づく，最適性を保証しないものであるため，適格性を満たすことは必須ではない．

6. 提案アルゴリズムの計算量評価

最大ステップ数を T，通信間のステップ数を k，通信回数を c，行動数を $|A|$，観測の数を $|\Omega|$，代表点の個数を $|B|$，そして DFS ツリーの深さを d とする．このとき, SPIDER と SPIDER-Comm における各エージェントのポリシーのサイズ，可能なポリシーの数，そしてエージェント全体の結合ポリシーに対する計算量を表 1 に示す．ポリシーのサイズは，SPIDER では最大ステップ数 T に対して指数的であるのに対して，k を定数と考えれば，SPIDER-Comm のポリシーのサイズは，T に対して線型に増加する．

一方，LID-JESP および LID-JESP-Comm に関しては，ポリシーのサイズ，可能なポリシーの数に関しては，それぞれ SPIDER, SPIDER-Comm と同じであるが，これらのアルゴリズムは反復完全型のアルゴリズムであり，アルゴリズムの計算量を見積もることは難しい．

表 1: SPIDER, SPIDER-Comm の計算量評価

	SPIDER	SPIDER-Comm
ポリシーのサイズ	$O(\|\Omega\|^T)$	$O((1+c\|B\|)\|\Omega\|^k)$
可能なポリシーの数	$O(\|A\|^{\|\Omega\|^T})$	$O(\|A\|^{\|\Omega\|^k})$
計算量	$O((\|A\|^{\|\Omega\|^T})^d)$	$O((1+c\|B\|)(\|A\|^{\|\Omega\|^k})^d)$

7. 評価実験

本章では，図 1 に示すセンサネットワークを用いて提案アルゴリズムに対する評価実験を行う．図 4 に実験結果を示す．

図 4 (a) 及び図 4 (b) に，$k = 2$，通信回数 $c = 1$ のときの, SPIDER-Comm と LID-JESP-Comm の実行時間と期待利得の比較を示す．計算時間は，3 エージェントのとき，MDP+Default policy が最も短く，4，5 エージェントのときは Default policy が最も短い．また，期待利得は 4 エージェントと 5 エージェントでは，MDP+Default policy が Default policy と LID-JESP-Comm を上回った．MDP+Default policy は，実行時間は他のアルゴリズムと比較して若干長いものの，得られる解の品質は大幅に向上している．

次に，図 4 (c) に，$k = 2$，4 センサネットワークでの通信回数の増加に対する SPIDER-Comm (MDP+Default policy) の実行時間の推移を示す．通信回数 6 のとき最大ステップ数は 20 である．SPIDER は最大 4 ステップまで，LID-JESP は 6 ステップ程度のポリシーしか得られなかった．一方，提案アルゴリズムで最も実行時間の長い SPIDER-Comm (MDP+Default policy) でさえも，20 ステップのポリシーを得るのに高々80 秒しか必要としない．これらの結果は，通信の導入によりステップ数に関する計算量が劇的に減少したことを示している．

8. おわりに

本論文では ND-POMDP に対してエージェント間のオンライン通信を導入し，2 つの新しいアルゴリズム (LID-JESP-Comm と SPIDER-Comm) の提案を行った．通信後の新しい信念状態の数が指数的に増加するという課題に対して Point-Based Value Iteration (PBVI) アルゴリズムに基づくアイデアを適用した．これらのアルゴリズムは，既存のアルゴリズムと比較して，扱えるポリシーの長さが大幅に増加していることを計算機実験によって示した．今後の課題として，通信のモデルを一般化すること，例えば，任意のタイミングでの通信や，局所的な通信を可能とすることが考えられる．

参考文献

[Nair 04] Nair, R., Roth, M., Yokoo, M., and Tambe, M.: Communication for Improving Policy Computation in Distributed POMDPs, in *AAMAS-04*, pp. 1096–1103 (2004)

[Pineau 06] Pineau, J., Gordon, G., and Thrun, S.: Anytime Point-Based Approximations for Large POMDPs, *JAIR*, Vol. 227, pp. 335–380 (2006)

[Spaan 05] Spaan, M. T. J. and Vlassis, N.: Perseus: Randomized Point-based Value Iteration for POMDPs, *JAIR*, Vol. 24, pp. 195–220 (2005)

[Varakantham 07] Varakantham, P., Marecki, J., Yabu, Y., Tambe, M., and Yokoo, M.: Letting Loose a SPIDER on a Network of POMDPs: Generating Quality Guaranteed Policies, in *AAMAS-07*, pp. 822–829 (2007)