RF-004

ND-POMDP の解法 CL におけるポリシの組み合わせ数と通信量の削減手法
Reducing the Size of Policies and Communication in Cooperative Solution Method for ND-POMDPs

川畑佑記 † 松井俊浩 † 松尾啓志 †
Yuki Kawabata Toshihiro Matsui Hiroshi Matsuo

1. はじめに

複数の自律的に振る舞う要素である「エージェント」が協調して仕事を達成したり問題を解決するシステムをマルチエージェントシステムと呼ぶ．マルチエージェントシステムは分散ミーティングスケジューリング，電源施設や分散センサ網の資源割り当てなどへの応用が期待される [1, 3, 5]．このようなシステムでは複数のエージェントが地理的あるいは通信網上の距離において離れて配置されているため，各エージェントは付近のエージェントを除く他のエージェントがどの程度の能力を持ち，どのような情報を保持しているかが分からない．そのような状況のもとで問題を解決するためには，分散アルゴリズムを考慮しつつ問題解決手法を検討する必要がある．

分散センサ網のような多くの応用システムでは，不確かな状況下で複数のエージェントが協調して行動しなければならない．このようなエージェント間の協調をモデル化した意思決定過程としてネットワーク分散型部分観測可能マルコフ決定過程 (Networked Distributed Markov Decision Process, ND-POMDP) [8] が提案されている．また，ND-POMDP において，複数エージェントの最適なポリシの組み合わせ (結合ポリシ) を求める手法 [2, 6, 10] として協調強化学習 (Coordinated Multi-Agent Reinforcement Learning, CL) [10] が提案されている．CL では強化学習を用いることによりエージェントに各結合観測履歴における行動を学習させる．

しかし，CL には学習する結合観測履歴の系列長に伴い学習結果の表に含まれる結合観測履歴が指数的に増加するため，ポリシの組み合わせ数が膨大となることが問題となる．また，CL はエージェント間の合意形成のために Max-Sum アルゴリズム [1] を用いる．この Max-Sum アルゴリズムの通信回数は CL を行うセンサの数に伴って増加する．本研究では，CL に通信を介さない学習手法である独立強化学習 (Independent Multi-Agent Reinforcement Learning, IL) [10] を部分的に導入することによりポリシの組み合わせ数と系全体における通信回数を削減する手法を提案する。

本論文の構成を以下に示す．まず，2 章で本研究の背景として ND-POMDP について述べ，その従来解法である CL と IL の概略を示す．3 章では，提案手法として CL に通信を介さない IL を部分的に導入することによりポリシの組み合わせ数と系全体における通信回数を削減する方法について述べる．4 章では，提案手法を実験により評価する．最後に 5 章で本研究のまとめと将来的な展望を示す．

†名古屋工業大学, Nagoya Institute of Technology

2. 研究背景

2.1. ネットワーク分散型部分観測可能マルコフ決定過程

不確実性が存在するマルチエージェントシステムをモデル化する手法として，部分観測可能マルコフ決定過程 (POMDP) を用いる研究がされている [4]．POMDP の拡張として，分散センサ網や分散無人機群などにおけるエージェント間の相互作用の局所性をモデル化したネットワーク分散型部分観測可能マルコフ決定過程 (ND-POMDP) が提案されている [8]．例えば，複数のセンサで構成される分散センサ網の場合，センサは近傍センサと協調して観測することにより特定のターゲットを補足する．ND-POMDP ではこのようなエージェント間の相互作用の局所性を扱う．

本研究で用いる ND-POMDP のモデルを説明する．ND-POMDP は，POMDP におけるエージェント間の相互作用の局所性を陽に表現したものであり，$\langle I, S, A, P, \Omega, O, R, b\rangle$ により定義される．

$I = \{1, \ldots, n\}$ はエージェントの集合である．

$S = \times_{i\in I} S_i \times S_u$ は状態集合である．S_i はエージェント i の局所状態の集合を表し，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を表す．

$A = \times_{i\in I} A_i$ はエージェントの行動の組み合わせの集合である．A_i はエージェント i の取りうる行動の集合を表す．

$P(s^{'} \mid s, a) = P_u(s_u^{'} \mid s_u) \cdot \prod_{i\in I} P_i(s_i^{'} \mid s_i, s_u, a_i)$ は状態の遷移関数である．$P(s^{'} \mid s, a)$ は結合状態 $s = \langle s_u, s_1, \ldots, s_n\rangle$ において結合行動 $a = \langle a_1, \ldots, a_n\rangle$ をとった場合に，次の結合状態 $s^{'} = \langle s_u^{'}, s_1^{'}, \ldots, s_n^{'}\rangle$ に遷移する確率を返す．また，遷移は他のエージェントの行動の影響を受けない．

$\Omega = \times_{i\in I} \Omega_i$ はエージェントが得る観測情報の組み合わせの集合である．Ω_i はエージェント i が得る観測情報の集合を表す．

$O(\omega \mid s, a) = \prod_{i\in I} O_i(\omega_i \mid s_i, s_u, a_i)$ は観測関数である．$O(\omega \mid s, a)$ は結合状態 $s = \langle s_u, s_1, \ldots, s_n\rangle$ において各エージェントが結合行動 $a = \langle a_1, \ldots, a_n\rangle$ をとった場合に，結合観測情報 $\omega = \langle \omega_1, \ldots, \omega_n\rangle$ を得る確率を返す．O_i は Ω_i に関する観測関数を表し，各エージェントが得る観測情報は確率的に観測誤差が生じ，それぞれ独立して発生する．

$R(s, a) = \sum_l R_l(s_l, s_u, a_l)$ は報酬関数である．l は相互作用のあるエージェントの部分集合を表している．R_l 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る報酬を返す．報酬関数に基づいて，エージェント間の相互作用を表すグラフ $G = (I, E)$ が定義される．

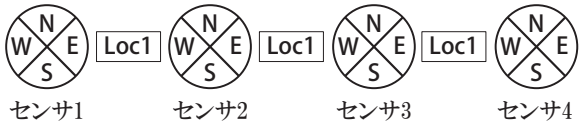

図 1: 分散センサ網の例

ノードの集合 I の要素は各エージェントであり，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，$b(s) = b_u(s_u) \cdot \prod_{i \in I} b_i(s_i)$ と定義される．b_u は状態が s_u である確率を表し，b_i はエージェント i の局所状態が s_i である確率を表す．各エージェントはそれぞれ独立した信念状態を持つ．

ND-POMDP における目的は，初期信念状態 b から，最大ステップ数 T に対して系全体の報酬を最大化する結合ポリシ $\pi = \langle \pi_1, \ldots, \pi_n \rangle$ を探索することである．

ND-POMDP を適用可能な環境の例として，4 センサ (エージェント) の分散センサ網を図 1 に示す．各センサ I_i は行動 A_i として 4 方向 (東西南北) の内の 1 方向を観測することができる．その際，センサは観測した領域にターゲットが存在するか否かの観測情報 ω_i を得る．この観測には観測関数 O_i により *false positive* や *false negative* といった誤った観測情報を手に入れてしまう可能性もある．2 つのセンサが同時にターゲットを観測した時，発見したとみなし，報酬 R_l が得られる．例えば，図 1 の分散センサ網で，ターゲットが Loc1 に存在している場合，センサ 1 が東，センサ 2 が西を同時に観測することにより報酬が与えられる．しかし、片方のセンサのみで観測したり，ターゲットの存在しない領域を観測するとコストを負ってしまう．また，それぞれのセンサが得る観測情報や，ターゲットの行動は他のセンサの行動 (観測の方向) に依存しないものとする．

2.2. 基本的な学習手法

系全体の報酬を最大化する結合ポリシを求めるために，強化学習である Q 学習 [9] を用いる．結合観測履歴 $\vec{h}$ において結合行動 a をとることの価値を表すために，関数 $Q(\vec{h}, a)$ を定義する．結合観測履歴 $\vec{h}$ は結合行動 a と結合観測情報 ω の系列を表す．この場合の結合ポリシは以下のような式で表される．

$$\pi(\vec{h}) = \mathrm{argmax}_{a \in A} Q(\vec{h}, a) \tag{1}$$

間数 $Q(\vec{h}, a)$ を学習するために標準的な Q 学習を用いることができる．$Q(\vec{h}, a)$ の更新式は以下のような式で表される．

$$\begin{aligned} Q(\vec{h}^t, a^t) &= (1-\alpha) Q(\vec{h}^t, a^t) \\ &\quad + \alpha [r^t + \gamma \max_a Q(\vec{h}^{t+1}, a)] \end{aligned} \tag{2}$$

α は学習率と呼ばれ, $0 < \alpha \leq 1$ の実数値をとる．値が大きいと一般に学習の収束は早くなるが，価値観数の変動が大きくなるため，最適値に収束しない可能性がある．r^t は結合観測履歴 $\vec{h}^t$ で結合行動 a^t をとった時の報酬を表す．γ は割引率と呼ばれ，$0 \leq \gamma \leq 1$ の実数値をとり，値が大きいほど将来に期待される報酬を現在の報酬と同等に評価する．

この手法は最適な結合ポリシを導くことが可能であるが，現実的ではない．その原因として，２つの問題点が挙げられる．一つ目は，ポリシの組み合わせ数がエージェントの数に対して指数的となる点である．二つ目は，エージェントは他の全てのエージェントの行動 · 観測情報 · 報酬の情報を必要とする点である．これらの問題点を解決する方法として独立強化学習 (IL) と協調強化学習 (CL) が提案されている [10].

2.3. 独立強化学習

独立強化学習 (IL) [10] はエージェント間の協調を省き各エージェントが独立して強化学習を行う．観測履歴 $\vec{h}_i$ で行動 a_i をとることの価値を表すために，関数 $Q(\vec{h}_i, a_i)$ を定義する．

$$\begin{aligned} Q_i(\vec{h}_i^t, a_i^t) &= (1-\alpha) Q_i(\vec{h}_i^t, a_i^t) \\ &\quad + \alpha [r_i^t + \gamma \max_{a_i^*} Q_i(\vec{h}_i^{t+1}, a_i^*)] \end{aligned} \tag{3}$$

IL は他のエージェントの観測や報酬を無視し，自身の観測情報や報酬のみをもとに行動価値関数を学習する．r_i は局所的な報酬関数である R_l をリンク l に含まれるエージェント内で均等に分割したものである．

学習後，各センサは自身の関数 Q_i に基づいて行動 a_i を選択する．このときの結合ポリシは次式で表される．

$$\pi^*(\vec{h}) = \mathrm{argmax}_a \sum_{i \in I} Q_i^*(\vec{h}_i, a_i) \tag{4}$$

2.4. 協調強化学習

協調強化学習 (CL) [10] は強化学習をリンクごとにグループ化されたエージェントにより行う．本研究では以降このグループをリンクグループと呼ぶ．関数 $Q(\vec{h}, a)$ の近似関数として $\hat{Q}(\vec{h}, a)$ を定義する．$\hat{Q}(\vec{h}, a)$ は ND-POMDP のグラフをリンク単位で分解した局所関数 $Q_l(\vec{h}_l, a_l)$ の和であり，以下の式で表される．

$$\hat{Q}(\vec{h}, a) = \sum_{l \in E} Q_l(\vec{h}_l, a_l) \tag{5}$$

$Q_l(\vec{h}_l, a_l)$ はリンク l における結合観測履歴 $\vec{h}_l$ について結合行動 a_l をとることの価値を表す．$\hat{Q}(\vec{h}, a)$ を学習するために，式 (5) は式 (2) を用いて以下のように表される．

$$\begin{aligned} \sum_{l \in E} Q_l(\vec{h}_l^t, a_l^t) &= (1-\alpha) \sum_{l \in E} Q_l(\vec{h}_l^t, a_l^t) \\ &\quad + \alpha [\sum_{l \in E} r_l^t + \gamma \max_a \hat{Q}(\vec{h}^{t+1}, a)] \end{aligned} \tag{6}$$

しかし，式 (6) では各リンク l のエージェントは直接 $\max_a \hat{Q}(\vec{h}^{t+1}, a)$ を計算することができない．なぜならば，この計算では結合観測履歴 $\vec{h}^{t+1}$ と結合行動 a は全てのエージェントの行動 a_i および観測 ω_i 情報を必要とするためである．そこで，結合観測履歴 $\vec{h}^{t+1}$ におけ

る最適な結合行動 $a^* = \mathrm{argmax}_a \hat{Q}(\vec{h}^{t+1}, a)$ を用いることにより $\max_a \hat{Q}(\vec{h}^{t+1}, a)$ は以下のように書き換えられる．

$$\max_a \hat{Q}(\vec{h}^{t+1}, a) = \hat{Q}(\vec{h}^{t+1}, a^*) = \sum_{l \in E} Q_l(\vec{h}_l^{t+1}, a_l^*) \quad (7)$$

式 (6), (7) より最終的に各リンク l 上における $Q_l(\vec{h}_l, a_l)$ の更新式は以下の式で表される．

$$\begin{aligned} Q_l(\vec{h}_l^t, a_l^t) &= (1-\alpha) Q_l(\vec{h}_l^t, a_l^t) \\ &\quad + \alpha[r_l^t + \gamma Q_l(\vec{h}_l^{t+1}, a_l^*)] \end{aligned} \quad (8)$$

式 (8) の a_l^* を計算するために，CL では分散制約最適化問題 [7] の解法である Max-Sum アルゴリズム [1] を用いる．

2.4.1.CL の学習時の手順

CL では各リンクグループごとに任意の方法で代表センサを決める．代表センサは学習によってグループの関数 Q_l を更新する．CL の学習の手順を以下に示す．

1. 各ステップ t において，各リンクグループにおける結合行動 a_l^t の実行後，リンクグループ内のセンサは観測情報を代表センサに送る．
2. 代表センサはリンクグループへの報酬 r_l^t を受け取る．
3. 代表センサは結合観測履歴 $\vec{h}_l^t$ に結合行動 a_l^t と結合観測情報 ω_l^t を加え，新しい結合観測履歴 $\vec{h}_l^{t+1}$ を生成する．
4. 代表センサは結合観測履歴 $\vec{h}_l^{t+1}$ における最適な結合行動 a_l^* を Max-Sum アルゴリズムを用いて計算する．
5. 代表センサは式 (8) に従って関数 Q_l を更新する．
6. 代表センサはリンクグループ内のセンサに次にとるべき行動を通知する．このとき，行動は a_l^* または，適当な探索戦略 (ルーレット選択や ϵ グリーディ) に基づいて選択される．

上記の手順を繰り返すことにより，センサが各結合観測履歴でとるべき行動を学習させる．

しかし，式 (8) の関数 $Q_l(\vec{h}_l, a_l)$ では結合観測履歴 $\vec{h}_l$ が定義されているために，関数 Q_l の規模が最大ステップ数 T に関して指数的に増加することが問題となる．CL では結合観測履歴 $\vec{h}_l$ に含まれる結合行動 a_l と結合観測情報 ω_l のペアからなる系列の長さに制限を加える fixed-size window [10] を用いることによりこの問題を抑制する．

2.4.2.CL の実行時の手順

各センサは学習した関数 Q_l に基づいて行動を選択する．このときの結合ポリシは次式で表される．

$$\pi^*(\vec{h}) = \mathrm{argmax}_a \sum_{l \in E} Q_l^*(\vec{h}_l, a_l) \quad (9)$$

式 (9) に従って各リンクにおける $Q_l(\vec{h}_l, a_l)$ の総和が最大となる結合ポリシを選択し，行動を決定する．CL の実行の手順を以下に示す．

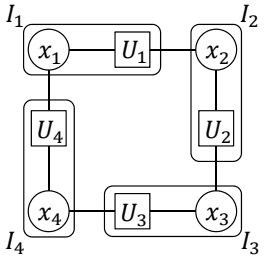

図 2: Max-Sum における問題のグラフ表現

1. 各ステップ t において，各リンクグループにおける結合行動 a_l^t の実行後，各センサは観測情報を受け取る．
2. 各センサは結合観測履歴 $\vec{h}_l^t$ に結合行動 a_l^t と結合観測情報 ω_l^t を加え，新しい結合観測履歴 $\vec{h}_l^{t+1}$ を生成する．
3. 各センサは関数 $Q_l(\vec{h}_l, a_l)$ から結合観測履歴 $\vec{h}_l^{t+1}$ における最適な結合行動 a_l^* を Max-Sum アルゴリズムにより計算する．

2.5.Max-Sum アルゴリズムによる行動の選択

各リンクグループの結合観測履歴 $\vec{h}_l$ における最適な行動 a^* とは以下の式で表される．

$$a^* = \mathrm{argmax}_a \sum_{l \in E} Q_l(\vec{h}_l, a_l) \quad (10)$$

CL では学習時および実行時にこの最適な結合行動を計算するために分散制約最適化問題 [7] の解法の 1 つである Max-Sum アルゴリズム [1] を用いる．

2.5.1. 分散制約最適化問題 (DCOP)

分散制約最適化問題 (DCOP) は問題の構成要素が複数のエージェントに分散して配置された離散組み合わせ最適化問題である．DCOP はエージェントの集合 I，変数の集合 X，各変数の値域 D，制約の集合 C，各制約に対応する評価関数の集合 F からなる．各エージェント I_i は変数 $x_i \in X$ と値域 $D_i \in D$ を持ち，変数は各エージェントの状態を表す．各制約 $c_{i,j} \in C$ について，評価関数 $f_{i,j}(x_i, x_j) \in F$ により変数値の割り当てに対する評価値が定義される．DCOP の目的は全ての制約についての評価関数の値の合計を最大化するような変数値の割り当てを求めることである．

2.5.2.Max-Sum アルゴリズム

Max-Sum アルゴリズム [1] は，DCOP の解法の一つである．Max-Sum アルゴリズムは関数ノード U と変数ノード x からなる factor グラフ上でメッセージを伝搬する．そのため，DCOP におけるグラフの表現を関数ノード U と変数ノード x からなる factor グラフとして表す．Max-Sum アルゴリズムでは図 2 のように各エージェントは factor グラフ上で自身の変数ノード x と評価関数を表す関数ノード U を持つように表現される．Max-Sum アルゴリズムでは変数ノード x と関数ノード U の間のメッセージ交換により大域的に (準) 最適な評価値となる解を得る．

Max-Sum アルゴリズムの処理を大きく分類すると変数ノード x_n から関数ノード U_m へのメッセージ $Q_{n\to m}(x_n)$ の計算と伝達，関数ノード U_m から変数ノード x_n へのメッセージ $R_{m\to n}(x_n)$ の計算と伝達，変数ノードによる周辺関数 Z_n の計算と最適な変数値の選択の三つに分けられる．これらの処理を繰り返すことにより各変数の (準) 最適な値を計算する．

2.5.3.CL における Max-Sum アルゴリズムの手順

CL の場合，各エージェントはセンサを表し図 2 の I_n に対応する．変数ノードは $a=\{a_1,\ldots,a_n\}$ となり図 2 の x_n に対応する．$a_i \in A_i$ はエージェント i がとりうる行動を表す．関数ノードは $Q=\{Q_l \mid l \in E\}$ となり図 2 の U_m に対応する，Q_l は各リンク l における関数 Q_l を表す．

(1) 変数ノードから関数ノードへのメッセージ　変数ノード i から関数ノード l へのメッセージは以下の式のような関数 $q_{i\to l}(a_i)$ をあらわす表を伝達する．

$$q_{i\to l}(a_i)=\sum_{g\in F_i\setminus\{l\}} r_{g\to i}(a_i)+c_{il} \tag{11}$$

$F_i\setminus\{l\}$ はメッセージのあて先である関数ノード l を除いた，変数ノード i に近接する他の全ての関数ノードを表す．c_{il} はグラフにサイクルを含む場合に，メッセージの値が無限に増加することを抑制する正規化のための係数である．

(2) 関数ノードから変数ノードへのメッセージ　関数ノード l から変数ノード i へのメッセージは以下の式のような関数 $r_{l\to i}(a_i)$ をあらわす表を伝達する．

$$r_{l\to i}(a_i)=\max_{a^l\setminus\{a_i\}}[Q_l(\vec{h}_l,a_l)+\sum_{g\in V_l\setminus\{i\}} q_{g\to l}(a_g)] \tag{12}$$

$V_l\setminus\{i\}$ はメッセージのあて先である変数ノード i を除いた，関数ノード l に近接する他の全ての変数ノードを表す．a^l は関数に関係する変数の集合を表す．$max_{a^l\setminus\{a_i\}}$ は $a^l\setminus\{a_i\}$ に含まれる変数の割り当てについて最大化することを意味する．

(3) 最適な変数値の計算　各変数ノードは周辺関数 $Z_n(a_i)$ を計算し，自変数の (準) 最適な変数値を決定する．この計算は次のように表される．

$$a_i^*=\mathrm{argmax}_{a_i} z_i(a_i) \tag{13}$$

$$z_i(a_i)=\sum_{g\in F_i} r_{g\to i}(a_i) \tag{14}$$

factor グラフがサイクルを含まなければ，各変数ノード i の変数値 a_i^* は最適解となる．また，factor グラフがサイクルを含む場合は，a_i^* は最適値ではない可能性があるが，その解品質は ND-POMDP に適用する場合には許容される程度である [10]．

2.6. 従来解法 IL と CL の利点と欠点

IL はポリシの組み合わせ数を小さく抑えたまま学習を行うことが可能である．本研究で扱う分散センサ網のモデルでは，エージェントがとりうる行動の種類は最大で 5 種類であり，観測情報の種類は最大で 2 種類である．従って，行動と観測情報の系列数が n である観測履歴のパターンは最大で $(5\times 2)^n$ 通りとなる．また，IL は他のエージェントの観測や報酬を無視し，自身の観測情報や報酬のみをもとに行動価値関数を学習するため，エージェント間の通信を必要としない．これらの利点を持つ反面，IL は他のセンサと協調しないために局所解に収束しやすい欠点がある．25 個のセンサをグリッド状に配置した分散センサ網に対して CL と IL を比較した場合，100 ステップ後における IL の解品質は CL のほぼ半分になることが実験により示されている [10]．

一方，CL は最適な行動を学習できるが二つの欠点がある．一つ目は fixed-sized window の上限値を増やすことにより結合観測履歴の系列長が指数的に増え，ポリシの組み合わせ数が膨大になる点である．IL と同様に，エージェントがとりうる行動の種類を最大で 5 種類，観測情報の種類を最大で 2 種類とすると，各リンクグループにおける結合行動と結合観測情報の系列長が n である結合観測履歴の組み合わせは最大で $(5^2\times 2^2)^n$ 通りある．そのため，fixed-sized window が比較的小さくても，組み合わせ数の増加は無視できない．また協調学習に Max-Sum アルゴリズムを用いるため，各センサは近傍センサ数の Q_l 値表を持つことになり，結果としてメモリ使用量が増加する．二つ目は Max-Sum アルゴリズムによる通信回数の増加である．CL では 1 ステップ毎に必要なラウンド数分のメッセージ交換を行わなければならない．これは帯域を制限される実際のアプリケーションでは無視できないコストとなりうる．特に，協調学習の段階ではある程度の通信は止むを得ないとしても，学習後の実行の段階には通信を省略できると望ましい．以上の問題点よりポリシの組み合わせ数と通信回数を削減する方法が望まれる．

3. 提案手法

本研究では，既存解法である CL に通信を介さない学習手法である IL を部分的に導入することにより協調が必要なセンサの数を削減する手法を提案する．この提案手法によりポリシの組み合わせ数の削減と系全体における Max-Sum アルゴリズムの通信回数の削減が期待される．

3.1. 提案手法の概要

既存解法の CL では, すべてのセンサが協調強化学習をする．この協調学習が 2.6 節で述べたポリシの組み合わせ数と通信回数の増加の原因となる．そこで，本研究では，系に含まれるセンサを，協調強化学習 CL を行うセンサと独立強化学習 IL を行うセンサに分類し，CL と IL を併用する手法を提案する．以降ではそれぞれの種類のセンサを CL センサおよび IL センサと呼ぶ．2.6 節で述べたように，IL を導入することによりポリ

シの組み合わせ数とエージェント間の通信を削減することが可能となる．しかし，IL は他のセンサと協調しないために局所解に収束しやすい欠点がある．そのため，無作為に CL センサを IL センサに置き換えるだけでは解品質が大幅に低下する可能性がある．解品質の低下を抑えつつ IL センサを組み込むためには IL が局所解に収束する原因に対処することが必要である．本研究では IL が局所解に収束する原因を大きく二つに分け，それらに対応して適切に IL センサを導入する方法を提案する．

3.1.1.IL が局所解に収束する原因

IL が局所解に収束しやすい原因の一つ目にエージェント間での協調が無いために，系全体での期待報酬を考慮しない点が挙げられる．例えば，図 1 で三つの領域全てにターゲットが存在し，報酬がそれぞれ Loc1 が 50，Loc2 が 80，Loc3 が 50 だと想定する．この場合，センサ 2 とセンサ 3 が Loc2 を観測するように収束すると系全体における期待報酬は 80 となる．また，センサ 1 とセンサ 2 が Loc1，センサ 3 とセンサ 4 が Loc3 を観測するように収束すると系全体における期待報酬は 100 となる．CL センサの場合，Max-Sum アルゴリズムを用いることにより各リンクグループは系全体での期待報酬が最大となるように結合行動 a_l^* を決定することができる．従って，後者の期待報酬が 100 の方に収束することができる．しかし，IL センサはセンサ間で協調することができない．従って，IL センサは 2 ホップ先のセンサの期待報酬を知ることができないことにより，近傍センサが複数ある IL センサはどの近傍センサと協調すれば系全体の期待報酬が高くなるか分からない．結果として，自身の期待報酬が最大となるように行動 a^* を決定するしかなく，前者の期待報酬が 80 の方に収束してしまう．

IL センサが局所解に収束しやすい原因の二つ目に協調的な行動選択が考慮されないために，協調的な行動についての学習が行われない点が挙げられる．ND-POMDP の報酬はリンク l に関係するエージェントの行動を引数として局所報酬関数 R_l により決定される．このとき，局所報酬関数 R_l により，片方のセンサだけがターゲットを観測しても報酬が与えられず，逆にコストが与えられる．CL センサの場合はリンクグループにおける結合観測履歴 h_l から結合行動 a_l に基づいて結合行動価値関数 $Q(h_l, a_l)$ を更新してゆく．従って，リンク l に関係するセンサの行動の組み合わせは最適な行動に収束する．しかし，IL センサの場合は自身の観測履歴 h_i から行動 a_i に基づいて行動価値関数 $Q(h_i, a_i)$ を更新してゆく．従って，図 1 で Loc1 にターゲットがいる場合，片方のセンサ 1 だけが Loc1 を観測してもコストが与えられることにより，その行動の期待報酬は小さい値に収束し，オフの行動を多く取るようになってしまう．

これら二つの原因より，CL センサを適切に IL センサに変更するには系全体の期待報酬に影響を与えず，且つ IL センサに協調的な行動選択をさせ関数 Q_i を学習させる必要がある．

3.1.2.CL センサと IL センサの併用

3.1.1 節で述べた 2 つの原因を回避するために，系全体で CL センサになる必要があるセンサを決定する．この条件は近傍センサの数が 2 つ以上で，且つ 2 ホップ先にセンサがいる場合であると言える．なぜなら，そのような条件のセンサは 2 ホップ先のセンサの期待報酬を考慮し，系全体での期待報酬を高くするように適切な近傍センサと協調行動を行う必要がある．この場合，IL センサは一つ目の原因より適当ではないと言える．逆に，系全体で CL センサになる必要がないセンサは IL センサの候補となる．この条件は CL センサになる条件の否定より，近傍センサの数が 1 つ以下，または 2 ホップ先にセンサがない場合となる．この条件の場合，IL センサは自身の期待報酬が最大となるように行動 a^* を決定しても，系全体の期待報酬には影響を与えない．例えば，図 1 で CL センサと IL センサの条件を考慮してセンサを配置すると，センサ 2 とセンサ 3 が CL センサとなり，センサ 1 とセンサ 4 が IL センサの候補となる．

また，IL センサの更新式 (3) の a_i^* を近傍センサとの協調的な行動にする必要がある．なぜなら，二つ目の原因より近傍センサとの協調行動によりターゲットを観測しないと正の報酬が与えられないからである．本提案では，学習時に IL センサの行動 a^* の選択を CL センサに制御させることにより期待報酬の低下を抑える．例えば，CL センサの条件に当てはまるセンサが一つ以上ある分散センサ網を考える．この場合，IL センサの条件により，IL センサの近傍センサ数は一つだけになっている．また，その近傍センサは必然的に CL センサになることにより，CL センサは IL センサの行動を制御することができる．例えば，図 1 で CL センサ 2 が Loc1 を観測する場合，IL センサ 1 に Loc1 を観測するように行動 a^* を通知する．逆に，CL センサ 2 が Loc2 を観測する場合，IL センサ 1 にオフの行動 a^* を通知する．これにより，二つ目の原因で述べた IL センサだけが領域を観測する事態を避けることが可能となる．CL センサは協調学習により高い解品質を得る学習をすることができる．従って，IL センサの行動選択を CL センサに制御させても比較的に解品質の高い結果を得ることが期待できる．この場合は，関数 Q_l の式 (8) は IL センサ間の行動も収束させる必要がある．そこで関数 Q_l の式 (8) の報酬 r_l に IL センサ間との報酬を追加することで以下のように変更する．

$$\begin{aligned} Q_{l_{CL}}(\vec{h}^t_{l_{CL}}, a^t_{l_{CL}}) &= (1-\alpha)Q_{l_{CL}}(\vec{h}^t_{l_{CL}}, a^t_{l_{CL}}) \\ &\quad +\alpha[(r^t_{l_{CL}} + \sum_{l_{IL}\in E_{l_{IL}}} r^t_{l_{IL}}) \\ &\quad +\gamma Q_{l_{CL}}(\vec{h}^{t+1}_{l_{CL}}, a^*_{l_{CL}})] \quad (15) \end{aligned}$$

式 (15) の $r_{l_{IL}}$ は IL センサ間のリンク l_{IL} における局所的な報酬関数 R_l を表し，$E_{l_{IL}}$ はリンクグループ l_{CL} に隣接するリンク l_{IL} の集合を表す．報酬 $r_{l_{IL}}$ を組み込むことにより CL センサは各結合観測履歴でリンク l_{CL} とリンク l_{IL} のうち，どのリンクを観測すれば良いか判断ができるようになる．このように学習側を変更

することにより CL センサは IL センサ間との行動についての学習も収束できるようになる．

IL が局所解に収束する原因を考慮し，適切に CL センサと IL センサを併用することにより，ポリシの組み合わせ数の削減と Max-Sum アルゴリズムを適用する範囲を縮小させることによる通信回数の削減が期待できる．また，IL センサを増やす場合，一つの CL センサの周りに集中しないように配置しなければならない．これは，CL センサは周囲に IL センサが増えると，学習の収束に要する試行回数が増えるためである．

3.2. 提案手法の学習時の手順

CL センサと IL センサの複合型の学習では，各センサの学習側の違いに応じて，CL の関数 Q_l と IL の関数 Q_i をそれぞれ並行して更新する．学習の手順を以下に示す．

1. 各ステップ t において，各リンクグループにおける結合行動 a_l^t の実行後，リンクグループ内のセンサは観測情報を代表センサに送る．
2. IL センサは行動 a_i^t の実行後，観測情報を受け取る．
3. 代表センサはリンクグループへの報酬 r_l^t と IL センサとの報酬 $r_{l_{IL}}^t$ を受け取る．
4. 代表センサは結合観測履歴 $\vec{h}_l^t$ に結合行動 a_l^t と結合観測情報 ω_l^t を加え，新しい結合観測履歴 $\vec{h}_l^{t+1}$ を生成する．
5. IL センサは観測履歴 $\vec{h}_i^t$ に行動 a_i^t と観測情報 ω_i^t を加え，新しい観測履歴 $\vec{h}_i^{t+1}$ を生成する．
6. 代表センサは結合観測履歴 $\vec{h}_l^{t+1}$ における最適な結合行動 a_l^* を Max-Sum アルゴリズムを用いて計算する．
7. 代表センサは IL センサに次にとるべき行動 a_i^* と報酬 $r_{l_{IL}}^t$ を均等に分割した r_i^t を通知する．
8. 代表センサは式 (15) を用いて関数 Q_l を更新する．
9. IL センサは式 (3) を用いて関数 Q_i を更新する．
10. 代表センサはリンクグループ内のセンサに次にとるべき行動を通知する．このとき，行動は a_l^* または，適当な探索戦略 (ルーレット選択や ϵ グリーディ) に基づいて選択される．

しかし，IL の関数 $Q_i(\vec{h}_i, a_i)$ においても，CL と同様に観測履歴 $\vec{h}_i$ が定義されているために，増加の程度は比較的小さいものの，Q_i 関数の規模が最大ステップ数 T に関して指数的に増加することは避けられない．そこで，IL の場合も観測履歴 $\vec{h}_i$ に含まれる行動 a_i と観測情報 ω_i のペアからなる系列の長さに制限を加える fixed-size window を用いる．

3.3. 提案手法の実行時の手順

学習結果に基づく行動決定では，各センサは学習した関数 Q_l または関数 Q_i に基づいて行動を選択する．このときの結合ポリシは次式で表される．

$$\pi^*(\vec{h}) = \mathrm{argmax}_a\{ \quad \sum_{l\in E_{CL}} Q_l^*(\vec{h}_l, a_l) + \sum_{i\in I_{IL}} Q_i^*(\vec{h}_i, a_i)\} \quad (16)$$

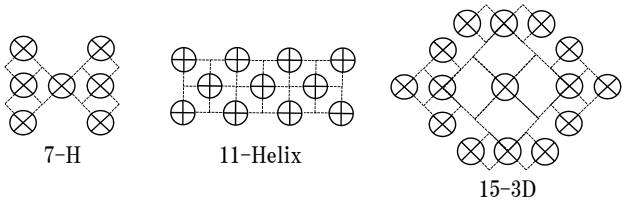

図 3: 実験で用いた分散センサ網

式 (16) の E_{CL} は CL センサのリンクグループのリンクの集合を表し，I_{IL} は IL センサの集合を表す．実行の手順を以下に示す．

1. 各ステップ t において，各リンクグループの結合行動 a_l^t と IL センサにおける行動 a_i^t の実行後，各センサは観測情報を受け取る．
2. CL センサは結合観測履歴 $\vec{h}_l^t$ に結合行動 a_l^t と結合観測情報 ω_l^t を加え，新しい結合観測履歴 $\vec{h}_l^{t+1}$ を生成する．
3. IL センサは観測履歴 $\vec{h}_i^t$ に行動 a_i^t と観測情報 ω_i^t を加え，新しい観測履歴 $\vec{h}_i^{t+1}$ を生成する．
4. CL センサは関数 $Q_l(\vec{h}_l, a_l)$ から結合観測履歴 $\vec{h}_l^{t+1}$ における最適な結合行動 a_l^* を Max-Sum アルゴリズムにより計算する．
5. IL センサは関数 $Q_i(\vec{h}_i, a_i)$ から観測履歴 $\vec{h}_i^{t+1}$ における最適な行動 a_i^* を選択する．

4. 実験と評価

提案手法の有効性を確認するために，実験により評価を行った．

4.1. 実験方法

従来解法 CL と提案手法を比較するために，図 3 の 3 種類の分散センサ網の例題を用いて実験した．これらの例題は ND-POMDP の実験 [6] [2] [10] で使用された代表的なものである．実験の設定を以下に示す．

- ターゲット数は 2 つとした．ターゲット 1，2 それぞれが観測されたときの報酬を+90，+70 とし，センサの行動をオフ以外としたが観測されなかったときのコストを-1 とした．
- 観測において $false\ \ positive$ もしくは $false\ negative$ が起きる確率は 10% とした．
- Q 関数の学習率 α は 0.001 とし，割引率 γ は 1.0 とする．
- 強化学習の探査戦略は ϵ グリーディとした．
- fixed-size window を用いた CL の結合観測履歴の結合行動と結合観測情報のペアの系列長は 3 以下に制限した．また，IL の観測履歴の行動と観測情報のペアの系列長は 8 以下に制限した．
- 結合行動を計算するための Max-Sum アルゴリズムのメッセージ交換回数は 4 ラウンドに制限した [10].

実験の手順を以下に示す．

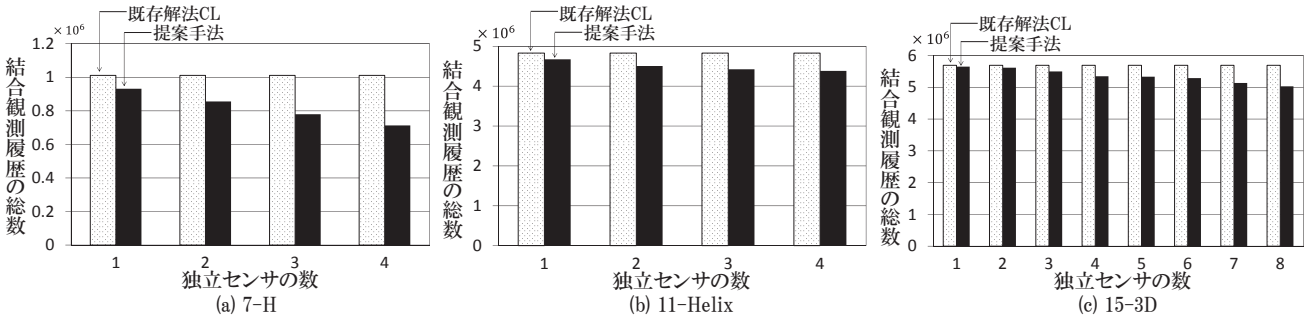

図 4: 全ての Q_l 値表に含まれる結合観測履歴の総数

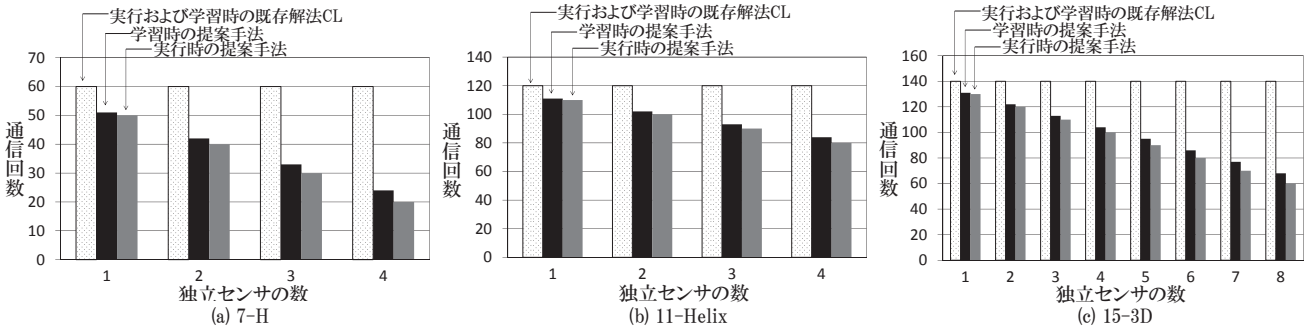

図 5: 1 ステップあたりの系全体の通信回数

1. 1 エピソードを 100 ステップ (T=100) とし，10000 エピソード学習させる．

2. 学習した結合ポリシで 10000 回実行し，解品質の平均を計算する．

また，提案手法の効果を示すために，さらに 2 つの比較手法を用いた．一つ目は IL センサに完全にランダムな行動を取らせる手法 (random) である．この手法と比較することにより IL センサが適切に学習できているか否かの判断ができる．二つ目は学習時の IL センサの行動 a^* の選択を CL センサに委任するのではなく，自身の関数 $Q_i(\vec{h}_i, a_i)$ をもとに決定する手法 (self-interest action) である．この手法と比較することにより学習時に IL センサが CL センサに行動 a^* の選択を委任する価値があるか否かの判断ができる．

4.2. 結果と評価

図 4 は従来解法 CL と提案手法について全ての Q_l 値表が保有する結合観測履歴の総数を比較したグラフである．結果は，従来解法 CL より提案手法の方が結合観測履歴の総数が少なくなっていることが分かる．これは部分的に IL センサを組み込むことにより，結合ポリシの組み合わせ数を削減したためである．また，Q_i 値表が保有する観測履歴の総数は IL センサ 1 つあたり 4000 程度となった．これは Q_l 値表より十分少ないといえる．

図 5 は各分散センサ網において従来解法 CL と提案手法について，1 ステップごとに要した系全体の通信回数で比較したグラフを表す．結果は，従来解法 CL より提案手法の方が系全体の通信回数が少なくなっていることが分かる．これは IL センサ数の増加に伴って，CL で用いる Max-Sum アルゴリズムなどに必要であった通信が，factor graph が縮小することにより減少し，通信回数が減っているためである．

図 6 は，各手法について解品質を比較したグラフを表す．結果は，IL センサが行動をランダムに選択する手法 (random) が最も解品質が低く，次に学習時に IL センサが行動選択を行う手法 (self-interest action) が低く，その次が提案手法の解品質となっていることが分かる．IL センサがランダムに行動を選択する手法より提案手法の解品質の方が高いことから，IL センサが適切に学習できていることと，その学習に効果があると言える．また，学習時に IL センサが自身で行動 a^* を決定する手法よりも提案手法の解品質の方が高いことから，CL センサに行動 a^* の選択を委任することにより局所解に陥ることを回避できていると言える．IL センサの増加に伴う解品質の低下を考察する．図 6 により (a) のグラフは 2 つ，(b) のグラフは 4 つ，(c) のグラフは 4 つまで IL センサを増加しても解品質の低下は比較的小さい．また, IL センサが複数隣接する CL センサ数は 7-H が 2 つ，11-Helix が 2 つ，15-3D が 4 つとなっている．従って，CL センサから IL センサに変更できる候補が複数 CL センサに隣接する場合は，一つだけを IL センサにしても解品質への影響は比較的小さいと言える．さらに IL センサを増やすと，IL センサ数と解品質はトレードオフの関係となった．このように，対象とする分散センサ網の環境において IL センサを増やしていくと，ポシリの組み合わせ数の削減によるポリシ空間の縮小と系全体の通信回数の削減が期待できる一方で，ターゲットの追跡において資源割り当ての精度が減少する影響がある．しかし，その影響は CL センサに複数隣接する IL センサの候補において，一つだけを IL センサに変更した場合，3 つの分散センサ網において解品質への影響は 5%より小さく，用途に

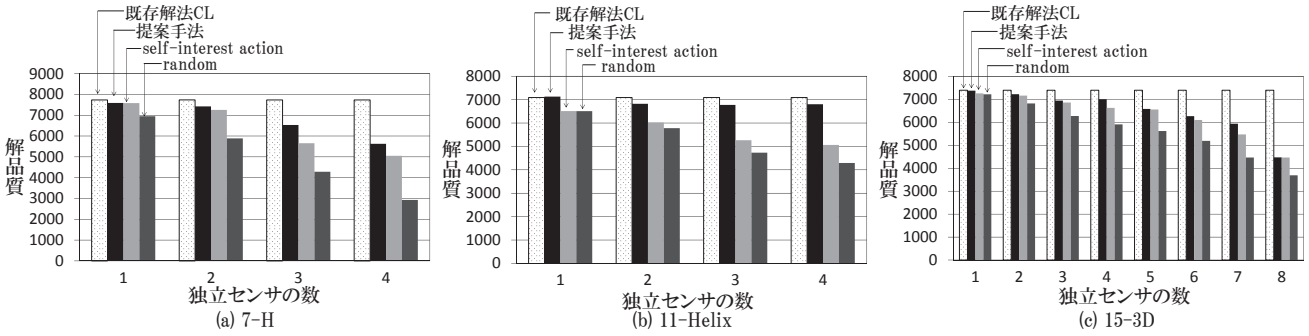

図 6: 1 エピソードあたりの解品質

よっては許容できる可能性がある範囲と考えられる．

5. まとめ

本研究では，不確かな状況下においてエージェント間の協調作用をモデル化した意思決定過程の一つであるネットワーク分散型部分観測可能マルコフ決定過程 (ND-POMDP) の従来解法である CL を改善し，ポリシの組み合わせ数と，系全体の通信回数を削減した．このために，協調学習 CL に部分的に独立学習 IL を導入する手法を提案した．提案手法を実験により評価し，解品質の低下を比較的小さい程度に抑制しつつ，ポリシの組み合わせ数を削減することが示された．また，提案手法は系全体の通信回数を削減できることが示された．

今後の研究課題として，適用可能な分散センサ網の拡張が挙げられる．現在の提案では近傍センサ数が二つ以上のセンサしかない分散センサ網へ適用することができない．例えばグリッド状の分散センサ網の場合において適切に IL を組み込む手法の検討などが挙げられる．また，解品質の低下をさらに抑制するための高度な学習方法を検討することにより CL における協調学習の範囲を削減することも挙げられる．

謝辞

本研究の一部は，科研費基盤研究 (C)25330257 および平成 23 年度人工知能研究振興財団研究助成による．

参考文献

[1] Farinelli, Alessandro, Rogers, Alex, and Petcu et al. Decentralised coordination of low-power embedded devices using the Max-Sum algorithm. In *7th International Conference on Autonomous Agents and Multi-Agent Systems.*, 2008.

[2] A kumar and S Zilberstein. Constraint-based dynamic programming for decentralized pomdps with structured interactions. In *7th international Conference on Autonomous Agents and Multi-Agent Systems.*, 2009.

[3] V Lesser, C Ortiz, and M Tambe. Distributed sensor networks: A multiagent perspective. In *Kluwer Academic Publishers.*, p. volume9, 2003.

[4] W Lovejoy. Computationally feasible bounds for partially observed markov decision processes. In *Operations Research*, Vol. 39, pp. 162–175, 1991.

[5] R.T Maheswaran, M Tambe, E Bowring, J.P Pearce, and P Varakantham. Taking dcop to the real world: Efficient complete solutions for distributed multi-event scheduling. In *3rd International Joint Conference on Autonomous Agents and Multiagent Systems.*, pp. 310–317, 2004.

[6] J Marecki, T Gupta, P Varakantham, M Tambe, and M Yokoo. Not all agents are equal: Scaling up distributed pomdps for agent networks. In *7th international Conference on Autonomous Agents and Multi-Agent Systems.*, pp. 485–492, 2008.

[7] A Petcu and B Faltings. DPOP: A scalable method for multiagent constraint optimization. In *IJCAI 05*, pp. 266–271, 2005.

[8] P Varakantham, M Tambe, and M Yokoo. Networked distributed pomdps: A synthesis of distributed constraint optimization and pomdps. In *19th international joint conference on Artificial intelligence.*, pp. 133–139, 2005.

[9] C.J.H Watkins and P Dayan. Technical note: Q-learning. In *Machine Learning*, Vol. 8, pp. 55–68, 1992.

[10] C Zhang and V Lesser. Coordinated multi-agent learning for decentralized pomdps. In *7th Annual Workshop on Multiagent Sequential Decision-Making Under Uncertainty, held in conjunction with AAMAS.*, 2012.