

# エージェント間の通信遅延を考慮した 効率的なチーム編成手法の提案

舟戸峻也<sup>†</sup> 菅原俊治<sup>†</sup>

<sup>†</sup>早稲田大学基幹理工学研究科情報理工情報通信専攻

## 1 はじめに

近年、グリッドコンピューティングやIoTなど、複数の計算機や性質が異なったリソースを持つ機器、またその上で動作するソフトウェアの組み合わせで提供されるサービスが注目されている。こうしたサービスは複数の異なる部分的な要求で構成され、すべての要素を達成して初めて1つのサービスとして提供される。そのため、サービスの構成要素全てに対して適切な機器を割り当て、いわばチームを構成して一つのサービスを提供する必要がある。

このようなチーム編成の問題においては、小さな環境ならば集中制御でも現実的な時間で最適な割り当てを実現できる。しかし、現在、また将来のネットワークで想定されるような大きな環境 [1] になると、集中制御では最適解の導出にかかる時間が指数関数的に大きくなり現実的ではない。そこで分散制御により個々の機器が自律的にチーム編成を行いタスクを処理する必要がある。この課題に対し、[2] ではタスク負荷が動的に変わる環境において、各機器が自律的に適切な行動を学習してチーム編成を効率化する手法を提案している。しかし、ここでは通信遅延は考慮されていない。現実のアプリケーションには通信遅延が存在し、可能であれば早く通信できる範囲でチームを組む方が有利だと考えられる。そこで本研究では [2] のモデルを拡張し、通信遅延を含めたモデルを提案し、その環境下で既存のアルゴリズムがチーム編成に与える影響を調べるとともに、より効率的なチーム編成手法を提案する。

## 2 提案モデル

### 2.1 エージェントとタスク

エージェントの集合を  $A = \{1, \dots, n\}$  と置く。エージェント  $i \in A$  は自身の処理能力に相当する  $p$  種類のリソース  $H_i = \{h_i^1, \dots, h_i^p\}$  を持つ。ただし、 $\forall i \in A$  に対して  $\sum_{k=1}^p h_i^k > 0$  とする。また、 $i$  は自身の役割  $r_i = \{x \in \{\text{リーダー}, \text{メンバ}\}\}$  を自律的に決定する。ここで、リーダーはチーム編成を、メンバはチームに参加しタスクを処理する役割とする。

システムに一定時間ごとに追加するタスクを  $T$  と表す。  $T$  はそれを構成するサブタスクの集合  $S_T = \{s_1$

$, \dots, s_m\}$  で構成される。ここで、 $m$  はタスク  $T$  に含まれるサブタスクの数を表す。各サブタスク  $s_i$  は処理するためにリソースを要求し、そのリソースを  $R_{s_i} = \{r_{s_i}^1, \dots, r_{s_i}^p\}$  とかく。タスク  $T$  はすべてのサブタスク  $s \in S_T$  を処理したとき完了となる。そのためタスク  $T$  の処理には、各サブタスクを担当する複数のエージェントからなるチームを組む必要がある。

タスク  $T$  は環境で生成され、システムに用意されたキュー  $Q$  に追加される。キューのサイズ (最大長) を  $q_{max}$  と表し、その値を超えたときタスクはドロップされる。本モデルに離散時間を導入し、その単位を  $tick$  と呼ぶ。

エージェント  $i$  は、サブタスクが要求するリソースを持つとき、それを処理できる。サブタスクが要求するリソース番号を  $j$  とし、エージェント  $i$  がそれを処理できるとすると、その処理時間は  $\lceil R_{s_i}^j / h_i^j \rceil$  [ticks] で表される。

### 2.2 チーム編成

タスク  $T$  を実行するチームを  $(G, \sigma, T)$  と表す。ここで、 $G$  はタスク  $T$  を処理するエージェントの集合を表す。  $\sigma$  は割り当て関数であり、 $s \in S_T$  を  $i \in G$  に割り当てることを  $\sigma(s) = i$  と表す。すべてのサブタスク  $s$  についてエージェントの割り当てが完了したとき、チーム編成に成功したとしてタスクを処理できる。

チーム編成は各エージェントが役割を選択することから始まる。リーダーを選択したエージェントはキュー  $Q$  の先頭からタスクを取り出し、その中から自分のリソースで可能なサブタスクがあればそのうち一つを選ぶ。残ったサブタスクひとつにつきある戦略に基づき  $R \in \mathbb{Z}$  体のエージェントを選び、チーム参加要請を送信し、返信を待つ。その後、参加要請の返信を確認し、受理を返したメンバエージェントと自身の集合を仮チーム  $G^P$  とする。そして  $G^P$  の中で再びある戦略に基づき割り当て  $\sigma$  を決定し、割り当てるエージェントを  $G$  に入れる。すべてのサブタスクについて割り当てが決定したらチームの成立とみなし、 $i \in G$  にチームの成立通知とともにサブタスクを割り当てその処理を依頼し、自分もサブタスクを処理する。一つでも割り当て不可能なサブタスクがあった場合はチーム不成立とみなし、 $i \in G^P$  に不成立通知を送信する。

役割としてメンバを選んだエージェントは、受信したチーム参加要請を参照し、提案手法で述べる戦略により参加するチームを決め、その参加するチームのリー

Forming effective teams in a decentralized environment with communication delay

Ryoya FUNATO<sup>†</sup>, Toshiharu SUGAWARA<sup>†</sup>

<sup>†</sup>Department of Computer Science and Engineering, Waseda University

ダーに参加表明を行い、それ以外のリーダーには拒否を伝える。なお、各エージェントが同時に所属可能なチーム数は1つと制限する。このためタスク処理中のエージェントは、参加要請メッセージに対し、拒否のメッセージを出す。

### 2.3 通信遅延

チーム編成の一連の流れにおいて、 $\forall i \in A$  とのメッセージのやり取りに通信遅延  $D_i = \{d_1^i, \dots, d_n^i\}$  を設ける。ただし、 $d_i^i = 0$ ,  $1 \leq d_j^i \leq L$  ( $i \neq j$ ) とする。エージェント間のメッセージのやり取りには以下の項目が該当する。

1. リーダーから他エージェントへの参加要請
2. リーダーからの参加要請に対する受理・拒否返信
3. リーダーから  $i \in G$  へのチーム成立通知
- 3'. リーダーから  $j \in G^P$  へのチーム不成立通知

$d_i^i$  の値は、 $i$  と  $j$  の距離に依存するが、ここでは距離の定義は割愛する。

## 3 提案手法

[2] では、以下の項目をエージェントが学習し、その結果チーム編成を効率化できることを示した。

1. リーダーかメンバの役割選択
2.  $i$  の行動に対する  $j \in A \setminus \{i\}$  の協調可能性 (信頼度)
3. メンバが参加チームを選択する戦略

本研究では、モデルに通信遅延を導入したことに伴い、上記の項目2を拡張して通信遅延が無視できない分散環境での効率的なチーム編成の手法を提案する。

### 3.1 最短通信所要時間

エージェントが持つパラメータである最短通信所要時間  $Mt_i$  を定義する。 $i$  が  $j$  に返信を期待するメッセージを送信してからその返信を受信するまでの観測時間を  $Ct$  とする。すなわち、自身がリーダーなら、 $j$  に参加要請を送ってから  $j$  の受理・拒否返信を受信するまでの時間であり、メンバなら、 $j$  に受理返信を送ってから  $j$  のチーム成立・不成立通知を受信するまでの時間である。 $Mt_i$  を以下で更新する。

$$Mt_i = \min(Mt_i, Ct)$$

従って、 $Mt_i$  は暫定で最も近い相手との通信時間となる。

### 3.2 信頼度更新方法

エージェント  $i$  はエージェント  $\forall j (j \in A \setminus \{i\})$  に対する信頼度  $c_{ij}$  を持つ。信頼度は、自分の行動に対して  $j$  が協調してくれる可能性を表す。 $i$  がリーダーである場合、チーム参加要請を送るエージェントを決定する際に、信頼度の高いエージェントを優先する。また、 $i$  がメンバである場合、信頼度の高いリーダーを優先して受理返信を返す。 $c_{ij}$  を次のように更新する。

$$c_{ij} = (1 - \alpha) \times c_{ij} + \alpha \times \delta_c$$

ここで、 $\alpha$  は学習率であり、 $0 < \alpha (<< 1)$  の定数である。なお、エージェントの役割によって、その更新タ

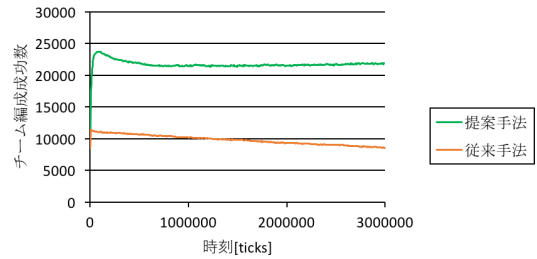


図1: タスク処理数の比較

イミングと  $\delta_c$  が異なる。自身がリーダーであるとき、更新タイミングは  $j$  からのチーム参加要請受理・拒否返信受信時で、

$$\delta_c = Mt_i / Ct \text{ (受理返信)}, 0 \text{ (拒否返信)}$$

で更新する。また、自身がメンバであるとき、更新タイミングは  $j$  からのチーム成立・不成立通知受信時で、

$$\delta_c = 1 \text{ (成立通知)}, c_{ij} \times (1 - Ct / Mt_i) \text{ (不成立通知)}$$

で更新する。これにより、リーダーとしては近いエージェントを高く評価し、メンバとしては自分を長く待たせたエージェントの評価を下げる。

## 4 評価実験と考察

本研究の有効性を示すため、[2] の手法 (以下、既存手法) と比較し、実験を行った。

図1より、提案手法では既存手法と比較してタスク処理効率が上がったことが確認できる。これは、信頼度の更新に距離を加味したことでより近いエージェントとの協調関係が結ばれ、メッセージのやり取りが相対的に高速化することで、短時間でチーム編成ができるようになったためである。

## 5 結論

本研究では、他のエージェントとの通信所要時間を未知とした上で効率的なチーム編成を実現する手法を提案した。これは [2] で提案したエージェントの自律的な学習を、通信遅延が無視できない環境に適用できるように拡張したものである。実験によって、既存手法と比べ提案手法の方がチーム編成を効率化できたことを示した。今後は、タスクの処理時間が変動する環境を想定してチーム編成の効率化を図るとともに、エージェントの数を増やした大規模な環境にも対応させる。

## 参考文献

- [1] 総務省. 平成 29 年版情報通信白書.
- [2] Masashi Hayano, Yuki Miyashita and Toshiharu Sugawara. "Adaptive Switching Behavioral Strategies for Effective Team Formation in Changing Environments," *Agents and Artificial Intelligence*. LNAI, pp.37-55, Springer, 2017.