

# マルチエージェント継続協調巡回問題における 分業によるロバスト性の向上

Effect of Divisional Cooperation on Robustness in Multi-Agent Continuous Cooperative Patrolling Problem

杉山 歩未  
Ayumi Sugiyama

菅原 俊治  
Toshiharu Sugawara

早稲田大学 基幹理工学研究科 情報理工・情報通信専攻  
Department of Computer Science and Communications Engineering, Waseda University

In this paper, we address the continuous cooperative patrolling problem by autonomous multi-agent who have limited activity time. We showed that agent with our method could construct a divisional cooperation in bottom-up manner by autonomous learning and simple one-to-one negotiation. In this paper, we focus on the effect of divisional cooperation created autonomously on the robustness of environmental changes, and evaluate our method. The results showed that the divisional cooperation improved the robustness of environmental changes since some agents respond to the changes and others continued their specific task. And more, we analyzed the reason of the improvement in detail.

## 1. はじめに

複数のエージェントによる自律的な協調は多くの分野で期待されている。これまでのトップダウンかつ中央集中的な処理・意思決定では処理能力の限界や故障時の影響が大きかった課題に対処するためである。そのため、複数の自律的なエージェントによる自律分散システムにより、ボトムアップにシステムを構築し、状況の変化に柔軟に対応することが望まれている。このような高度な自律性が求められる問題の1つとして、複数ロボットによる巡回問題がある。[David 11]はこれまでに行われてきた巡回問題のアプローチを体系的に整理し、複数ロボットによる巡回問題が自律性・分散性・通信の制約などの特徴を有し、多くのマルチロボットシステムに対して応用可能であることを示唆している。我々は巡回問題において特に継続性と協調に着目し、Continuous Cooperative Patrolling Problem (CCPP)を定義し、これを扱う。CCPPは巡回セールスマンのような最適経路探索ではなく、複数の自律エージェントが環境内を継続的に巡回し、各ノードの平均訪問間隔やイベントの検出時間を短縮させることを目的とする。

CCPPにおいてはエージェントが訪問すべき場所と頻度をどのように割り当てるかが重要であり、大きく分けて2つのアプローチがある。1つめのアプローチは環境をサブエリアに分割し、各サブエリアを1あるいは少数のエージェントが担当する手法である[Ahmadi 06, Elor 09]。2つめのアプローチは、環境を分割せずに適切な協調行動を獲得できるようなアルゴリズムやエージェントの行動戦略を提案するものである[Sampaio 10, Cheng 07]。我々も後者のアプローチに着目し、エージェント間の交渉を用いない自律的な戦略学習や、各ノードの重要度を学習することで間接的に他者のふるまいを考慮する手法を提案してきた[Yoneda 15], [Sugiyama 15]。

これらの研究をふまえて、全体の作業効率を向上させるためには、適切な分業体制を自律的に構築できるかが1つの重要な点となると考えられる。[Jones 03, Suseki 05]では、採餌問題や社会性昆虫をモデルに全体の統率者がいない環境でいかにして分業構造が生み出されるかという研究がなされている。このような統率者がいない環境において、エージェントの自律的行動

により効率的なシステムを構築できれば、交渉時のボトルネックを大きく低減できる。この観点から、我々は[Sugiyama 16]において責任をもつ範囲という概念を導入し、単純な1対1間交渉によってその範囲を自律的に調整することで役割分担・組織化が発生し、効率的な分業体制が構築されることを示した。

本稿では分業の自律的創発に着目した[Sugiyama 16]をベースに、自律的分業がシステムの柔軟性・環境の変化に対するロバスト性を高めることを示し、どのような要素がそのロバスト性に影響を与えるかを考察した。[David 11]は複数ロボットの巡回問題においてこれまで議論が不十分であった点として scalability, flexibility, resource utilization, interference, communication load or workload を挙げた。前述した多くの研究では、拡張性や通信の負荷は考慮されていたが、システムの柔軟性についての考察は十分になされていなかった。そこで本稿では、[Sugiyama 16]に対し追加的な実験を行い、自律的な分業体制の構築がシステムのロバスト性に与える影響を調査した。その結果、自律的に創発された分業体制はエージェントの停止という変化に対するロバスト性を向上させることを示し、その要因を分析した。

## 2. モデル

CCPPでは、環境内の各ノードでイベントが発生し、複数のエージェントがそれを検出するために協調的な巡回を行う。

### 2.1 環境

我々は離散時間  $tick$  を導入し、各  $tick$  ごとにイベントの発生やエージェントの移動を行う。エージェントが巡回を行う環境を  $G = (V, E) \subset \mathbb{R}^2$  と定義する。 $V = \{v_1, \dots, v_m\}$  はノードの集合であり、ノード  $v$  は座標  $x_v, y_v$  をもつ。 $E$  はエッジの集合であり、エージェントはエッジで接続された隣接ノードへ移動が可能である。ただし、ノード  $v$  上に障害物  $R_v \subset \mathbb{R}^2$  が存在する場合、そのノードへは移動できない。また、すべてのノードはイベントの発生確率  $p(v)$  ( $0 \leq p(v) \leq 1$ ) をもつ。時刻  $t$  にノード  $v$  において、検出されずに残存しているイベントの数を  $L_t(v)$  と表す。 $L_t(v)$  は  $p(v)$  に基づいて毎  $tick$  ごとに以下の式で更新する

$$L_t(v) \leftarrow \begin{cases} L_{t-1}(v) + 1 & (\text{イベント発生時}) \\ L_{t-1}(v) & (\text{その他}). \end{cases} \quad (1)$$

エージェントがノード  $v$  を訪問したとき、残存していたイベントは処理され  $L_t(v) = 0$  となる。

CCPP における巡回の評価指標について説明する。時刻  $t_s$  から  $t_e$  間の評価指標  $D_{t_s, t_e}$  を次式で定義する。

$$D_{t_s, t_e}(s) = \sum_{v \in V} \sum_{t=t_s+1}^{t_e} L_t(v), \quad (2)$$

ここで、 $t_s < t_e$  であり、 $s$  は 2.2.1 で述べるエージェントが取る行動戦略である。 $D_{t_s, t_e}(s)$  は環境全体における、検出されず放置されたイベントの時間積算を表しており、この値が小さいほど作業効率が良いとする。

## 2.2 エージェント

エージェントの集合を  $A = \{1, \dots, n\}$  と定義する。時刻  $t$  におけるエージェント  $i$  の位置を  $v_t^i \in V$  と表現する。エージェントは有限容量のバッテリーをもち、継続的な巡回のために定期的に充電基地  $v_{base}^i$  に戻り充電する必要がある。充電に関する制御は本稿の主目的と外れるため、詳しい制御方法は [Sugiyama 15] を参照されたい。エージェントは各ノード  $v$  の重要度  $p^i(v)$  を独立に学習・記録し、その集合  $P_v^i = \{(v, p(v)) | v \in V\}$  をもつ。重要度は数値で表現でき、環境において設定されているイベントの発生頻度  $p(v)$  と異なり、各エージェントが独立にもつ信念を表す。学習の詳細については後述する。

本モデルでは一般的な通信の制約を考慮した通信制御を想定した。エージェントは通信範囲  $d_{com} (> 0)$  を有し、エージェント  $i$  がエージェント  $j$  と通信ができるのはエージェント間のユークリッド距離  $m(v^i, v^j)$  を用いて、時刻  $t$  において  $m(v_t^i, v_t^j) < d_{com}$  の時とする。また、過度な通信によるコストの増加を防ぐため、 $i$  は  $j$  と最後に通信をおこなった時刻  $T_{last}^{i,j}$  を保持し、最低通信間隔  $T_{limit}^{i,j} (> 0)$  を定義し、 $T_{limit}^{i,j} \geq t - T_{last}^{i,j}$  のときは通信を行わないと仮定した。

### 2.2.1 目標決定戦略

エージェントの巡回は (1) 戦略  $s$  に従い目標ノード  $v_{tar}$  を決定。(2)  $v_{tar}$  までの経路を生成。(3) 生成した経路を通り  $v_{tar}$  に到着。という流れを繰り返すことで行う。以下でエージェントがとりうる戦略のコンセプト及び経路生成手法を簡単に述べるが、詳細なアルゴリズムは [Sugiyama 16] も参照にされたい。

**ランダム法**  $i$  は環境全体からランダムに目標を選択する。

**貪欲法**  $i$  は推定優先度  $EL_t^i(v)$  の上位  $N_g$  ノードから目標を選択する。

**訪問間隔優先法**  $i$  は環境全体から最後に訪問されてからの間隔  $I_t^i(v)$  の上位  $N_i$  ノードから目標を決定する。

**戦略型目標決定法**  $i$  は近隣の巡回を優先するかの閾値  $E_B$  を学習し、近隣の推定イベント蓄積数が  $E_B$  以下になるまで近隣を優先して巡回する。

また、我々はこれらの戦略から適切な戦略を学習する手法として Adaptive meta target decision strategy with learning of dirt accumulation probability (AMTDS/LD) を提案した [Sugiyama 15]。AMTDS/LD においてエージェント  $i$  は任意の戦略の集合  $S = \{s_1, \dots, s_n\}$  から最適と思われる戦略  $s^*$  を学習しながら巡回を行う。各エージェントは  $p(v)$  を未知とし  $p^i(v)$  の学習を行う。本稿で提案する手法と異なり、エージェント間の交渉は計算コストを考慮し行っていない。

経路決定には、最短経路周辺のイベント蓄積が大きいと推定される場所を経由するサブゴール型目標決定法を利用した。

## 3. 提案手法

我々は AMTDS/LD を拡張し、自律的な学習と 1 対 1 間の交渉というマイクロな行動戦略からシステム全体のマクロな分業創発を促進させる AMTDS with learning of event probabilities and enhancing divisional cooperation (AMTDS/EDC) 手法を提案する。ここでは、責任を持つ範囲という概念をエージェントに導入することで自身が重要だと考える場所への巡回の選択と集中を促し、1 対 1 の交渉によってその選択の改善を試みる。

### 3.1 重要度の (再) 学習と責任をもつ範囲

エージェントはノード  $v$  が最後に清掃された時刻  $t_{visit}^v$  を用いて、現在の時刻  $t$  までの経過時刻  $I_t^i(v)$  を以下のように計算する。

$$I_t^i(v) = t - t_{visit}^v. \quad (3)$$

そして、エージェント  $i$  はノード  $v$  に訪れたとき、 $I_t^i(v)$  を用いて  $p^i(v)$  を次式で更新する。

$$p^i(v) \leftarrow \begin{cases} (1 - \beta)p^i(v) + \beta \frac{1}{I_t^i(v)} (L_t(v) > 0 \text{ のとき}), \\ (1 - \beta)p^i(v) (L_t(v) = 0 \text{ のとき}), \end{cases} \quad (4)$$

ここで、 $\alpha$  ( $0 < \beta \leq 1$ ) は学習率である。

我々はエージェント  $i$  が責任をもつ範囲  $V_{self}^i (C V)$  を導入する。責任をもつとは、戦略  $s$  による目標ノード  $v_{tar}^i$  の候補を環境全体の  $V$  でなく、 $V_{self}^i$  に限定することで巡回対象の選択と集中を行うことである。ただし、訪問間隔優先法とランダム法は環境内で見逃されているノードの探索が目的であるため、これらの戦略では  $v_{tar}^i$  を  $V$  から選択する。エージェント  $i$  は充電基地に戻った際、 $V_{self}^i$  を更新する。 $V_{self}^i$  の初期値は  $V_{self}^i = V$  とする。 $V_{self}^i$  のサイズを  $N_{self}^i$  とし、この初期値も  $N_{self}^i = |V|$  となる。充電基地に戻った際、 $i$  は  $P^i$  の要素を  $p^i(v)$  の降順に並び替え、上位  $N_{self}^i$  のノードを  $V_{self}^i$  として定義する。 $p^i(v)$  が同値のノードが複数あった場合はランダムに決定される。

本研究では交渉にかかるコスト抑制のため、交渉には  $V_{self}^i$  から計算される 2 つの値のみを用いる。1 つは責任をもつノードの重要度の総和  $p_{sum}^i (\geq 0)$  であり、以下のように計算する

$$p_{sum}^i = \sum_{v \in V_{self}^i} p^i(v).$$

重要度  $p^i(v)$  が高いノードは頻繁な訪問を求められるので、 $p_{sum}^i$  は  $i$  が担当する作業量を表現している。2 つめは作業の重心  $C^i = (x_c^i, y_c^i)$  であり、 $p^i(v)$  の重みを考慮して以下のように設定する

$$x_c^i = \sum_{v \in V_{self}^i} \frac{p^i(v)}{p_{sum}^i} x_v, \text{ and } y_c^i = \sum_{v \in V_{self}^i} \frac{p^i(v)}{p_{sum}^i} y_v. \quad (5)$$

ここで、 $v = (x_v, y_v)$  である。ノード  $v_p$  からノード  $v_q$  への最短経路長を  $d(v_p, v_q)$  とすると、 $d(v_c^i, v) < d(v_c^i, v)$  のとき、ノード  $v$  はエージェント  $i$  の作業の中心にエージェント  $j$  よりも近く、訪問するために払うコストが小さいことをあらわす。

### 3.2 エージェント間交渉

エージェントは他者と交渉を行い、各々の  $V_{self}^i$  の要素の改善を試みる。 $V_{self}^i$  の改善には大きく分けて 2 つの段階がある。1 つは責任の差が大きい場合の公平性のための交渉である。こ

の交渉では、自分より少ない責任をもつ他者に、自分にとって重要でないタスクを任せることで、自分は重要な作業に集中し、相手は広く探索を行えるようになる。もう1つは、責任が均衡したときに責任をもつ作業を交換することで互いの効率を改善する交渉である。この交渉では、相手の方がより少ないコストで遂行できる作業を交換することで、大きな変動を与えずに効率を改善する。どちらの交渉においても、エージェントは  $p_{sum}^i$  と  $C^i$  のみを比較することで受け渡す情報を決定できる。

### 3.2.1 公平性のための交渉

式 (6) の条件を満たす場合、担っている責任に差があると判断し公平性のための交渉を行う。

$$1 + T_c < p_{sum}^i / p_{sum}^j. \quad (6)$$

$T_c$  ( $0 < T_c \ll 1$ ) は責任の差を判定する閾値である。この時、 $i$  は  $p^i(v)$  の降順に集合  $V_{self}^{i,j} = \{v \in V_{self}^i \mid d(C^i, v) > d(C^j, v)\}$  を計算する。 $i$  は  $V_{self}^{i,j}$  の  $p^i(v)$  の小さい値から、 $e_g$  ( $0 \leq e_g$ ) 個のノードを選択し、その  $p^i(v)$  の値を  $j$  に次式によって渡す。

$$p^j(v) \leftarrow p^j(v) + p^i(v) \times \delta \quad (7)$$

$$p^i(v) \leftarrow p^i(v) \times \delta. \quad (8)$$

$\delta$  ( $0 < \delta < 1$ ) は情報を受け渡す割合を表す。式 (7), (8) に示す  $p^i(v)$  値の受け渡しは、 $i$  が部分的に自分の責任を  $j$  に委託することを表す。 $e_g$  は交渉時に情報を受け渡されるノードの数であり、 $p_{sum}^i$  と  $p_{sum}^j$  の比率から以下のように決定する。

$$e_g = \min \left( N_{self}^i - 1, N_{gmax}^i, \left\lfloor \frac{p_{sum}^i}{p_{sum}^j} \times \gamma \right\rfloor \right), \quad (9)$$

ここで、 $N_{gmax}^i$  ( $0 < N_{gmax}^i \leq N_{self}^i$ ) は急激な変動を防ぐための上限であり、 $\gamma$  は受け渡すノード数を調整するパラメータである。情報の受け渡しが終わった後、エージェント  $i$  と  $j$  は  $V_{self}^i$  のサイズ  $N_{self}^i$  を次式で更新する

$$N_{self}^i \leftarrow N_{self}^i - e_g \quad (10)$$

$$N_{self}^j \leftarrow \min(|V|, N_{self}^j + e_g). \quad (11)$$

### 3.2.2 改善のための交渉

式 (12) の条件を満たす場合は改善のための交渉を行う。

$$1 - T_c < p_{sum}^i / p_{sum}^j < 1 + T_c. \quad (12)$$

$i$  は  $V_{self}^{i,j}$  の大きい  $p^i(v)$  の大きい値から  $e_g$  個のノードの  $p^i(v)$  を式 (7), (8) に従い  $j$  に渡す。 $e_g$  は次式で決定する

$$e_g = \min \left( N_{self}^i - 1, N_{cmax}^i \right), \quad (13)$$

ここで、 $N_{cmax}^i$  ( $> 0$ ) は上限である。情報の受け渡しが終わった後、エージェントは  $N_{self}^i$  を式 (10) に従い更新する。交換する  $p^i(v)$  の値を考慮し、 $N_{cmax}^i$  は  $N_{gmax}^i$  と比較して小さな値とする。式 (12) の条件を満たすとき、上記の過程は  $i$  と  $j$  で双方向に発生する。

## 4. 評価実験と考察

提案手法 (AMTDS/EDC) による責任を持つ範囲の導入とそれを調整する交渉手法を評価するために、実験を行った。実験では、全てのエージェントが環境全体に責任をもち、学習のみを行う AMTDS/LD との比較を行った。

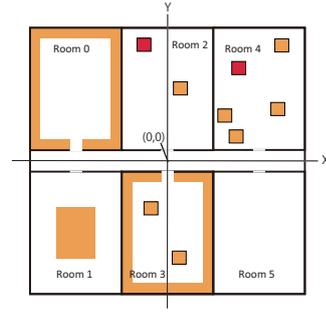


図 1: 実験環境

目標選択戦略	パラメータ	値
貪欲法	$N_g$	5
	$N_i$	5
	AMTDS/LD	
AMTDS/EDC	$\alpha$	0.1
	$\beta$	0.05
	$\varepsilon$	0.05
	$N_{gmax}$	100
通信の制約	$N_{cmax}$	10
	$T_c$	0.05
	$\gamma$	10
	$\delta$	0.5
	$T_{limit}$	10800

### 4.1 実験設定

本研究でエージェントが巡回を行う環境は、各ノードのイベント発生確率  $p^i(v)$  に偏りがあり、障害物によっていくつかの部屋  $Room N$  ( $N = 0, \dots, 5$ ) に分かれている  $101 \times 101$  の大きさのグリッド構造とした。その環境を図 (1) に示す。各ノードの  $p(v)$  の値は図 (1) に示す色によって赤色部:  $10^{-3}$ 、オレンジ色部:  $10^{-4}$ 、その他:  $10^{-6}$  とした。エージェント数は 20 で、全エージェントの充電基地は  $v_{base} = (0, 0)$  とした。全てのエージェントは充電基地からスタートし、充電のために定期的に戻る必要がある。エージェントのバッテリーの設定は容量を 900、容量 0 からの満充電に 2700tick を要し、隣接ノードへの移動は 1 消費すると設定した。よって、エージェントの巡回サイクルは最大 3600tick となり、評価指標の  $D_{t_g, t_e}(s)$  も 3600tick ごとに集計した。全てのエージェントは戦略として AMTDS/LD か提案手法の AMTDS/EDC のいずれかを用いる。また、モデルおよび提案手法で述べたパラメータは表 1 のように設定した。実験では、提案手法の環境の変化に対するロバスト性を評価するために一部のエージェントを途中で停止させた。100 万ステップ時にランダムに 10 のエージェントを停止させ、200 万ステップ時に停止したエージェントを復帰させた。エージェントは停止した瞬間から他者と通信を行えず、他のエージェントもその停止を知ることはできないとした。

### 4.2 エージェント停止時のロバスト性評価

評価指標  $D(s)$  の推移を図 (2) に示す。提案手法のエージェントは既存手法に比べて明らかに変化後の効率悪化を防いでいて、ピーク時では  $D(s)$  を 36.6% 減少させている。ロバスト性について考察する前に、[Sugiyama 16] で得られた自律的な分業の創発について簡単に述べる。図 (2) をみると、環境の変化がない時も提案手法の効率は良くなっている。その理由を説明

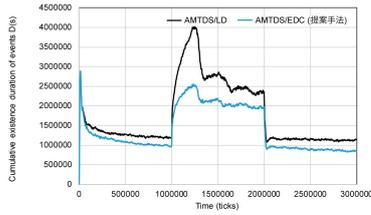


図 2: 巡回効率  $D(s)$  の推移

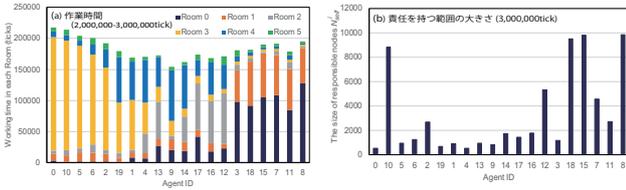


図 3: 各 Room における作業時間および  $N_{self}^i$

するために 200-300 万 tick における各エージェントの各 Room における作業時間と、試行終了時の  $N_{self}^i$  の大きさを図 (3) に示す。提案手法では各エージェントが主に作業する部屋に偏りが生じており、ある種のチームのような体制が創発されている。さらに、責任をもつ範囲  $N_{self}^i$  にも偏りが生じており、一部のエージェントは特定の狭い範囲に集中するスペシャリスト的な役割を、別のエージェントは比較的広範囲に責任をもつゼネラリスト的な役割を果たしている。さらに、同一のパターンで責任をもつエージェント郡のなかでも、重要度に大きな偏りがあるといった階層的な分業も行われていた。このような役割分担は巡回効率を高め、かつ、その分担は統率者のいない 1 対 1 交渉と自律的な再学習によってボトムアップに実現された。

以上をふまえ、自律的に創発された分業体制が環境の変化に追従できた理由について考察する。停止しなかったエージェントの責任の総和  $p_{sum}^i$  の推移を図 4(a) に、責任ノード数  $N_{self}^i$  の推移を図 4(b) に示す。図 4(a) から、停止以降に  $p_{sum}^i$  が大きく変動している。これは停止したエージェントが主に担当していた作業が残されていることを他のエージェントが発見・学習したためである。エージェントは提案手法に従い、図 4(b) に示すように責任の範囲を調整することで変化に対応している。特に、一部のゼネラリスト (エージェント 1,3,12 など) は環境の変化後に急激に  $N_{self}^i$  を減少させている。これは、変化時にはゼネラリストが新たに増加した作業を発見し責任範囲を狭めることで、情報の少ない新たな作業に集中し、スペシャリストは専任していた作業を続けることで、変化がシステム全体に影響することを防いでいると考えられる。また、本手法ではどこに責任をもつかを交渉で決定せず、各エージェントが独立に決定しており、交渉時の計算量を大きく抑えている。環境変化時には情報の不確実性が増加し予測が困難になるが、本結果は、

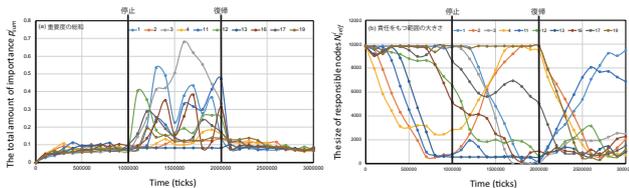


図 4:  $p_{sum}^i$  および  $N_{self}^i$  の推移

交渉時ではなく各エージェントに判断させることで低コストに効率的な対応を実現できる可能性を示唆している。

## 5. 結論

本研究では、高度な自律性と継続性、協調が求められる問題の 1 つとして CCPP を定義した。そして、ボトムアップに創発された自律的分業が環境の変化に対するロバスト性に与える影響について調査を行った。その結果、提案手法によって役割分担が創発され、それぞれが異なったパターンでタスクに対し責任をもつことで、エージェントの変化に対し柔軟に対応できることを示した。評価実験の結果、広範囲に責任をもつエージェントが変化に大きく反応し、狭い範囲に専任しているエージェントはその作業を続けることで、全員が全体に対し責任をもっている場合よりも、混乱を防げることを示唆した。今後は、突発的な変化に対するロバスト性だけでなく、エージェントの追加と停止がスケジューリングされるような変化が予測できる環境において、性能を維持するための要素も調査する。

## 謝辞

本研究は JSPS 特別研究員奨励費 JP16J11980 の助成を受けたものである。

## 参考文献

- [Ahmadi 06] Ahmadi, M., Stone, P.: A multi-robot system for continuous area sweeping tasks. In: Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation. pp. 1724–1729 (2006)
- [Cheng 07] Cheng, K., Dasgupta, P.: Dynamic area coverage using faulty multi-agent swarms. In: IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology. pp. 17–23 (2007)
- [David 11] David, P., Rui, R.: A survey on multi-robot patrolling algorithms. In: Proc. of Second IFIP WG 5.5/SOCOLNET Doctoral Conf. on Computing, Electrical and Industrial Systems, DoCEIS 2011, Portugal. pp. 139–146. Springer (2011)
- [Elor 09] Elor, Y., Bruckstein, A.M.: Multi-a(ge)nt graph patrolling and partitioning. In: Proc. of the 2009 IEEE/WIC/ACM Int. Joint Conf. on Web Intelligence and Intelligent Agent Technologies. pp. 52–57 (2009)
- [Jones 03] Jones, C., Mataric, M.J.: Adaptive division of labor in large-scale minimalist multi-robot systems. In: Proc. of the 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2003). vol. 2, pp. 1969–1974 (2003)
- [Sampaio 10] Sampaio, P., Ramalho, G., Tedesco, P.: The gravitational strategy for the timed patrolling. In: 22nd IEEE Int. Conf. on Tools with Artificial Intelligence. pp. 113–120 (2010)
- [Sugiyama 15] Sugiyama, A., Sugawara, T.: Meta-strategy for cooperative tasks with learning of environments in multi-agent continuous tasks. In: Proc. of the 30th Annual ACM Symposium on Applied Computing. pp. 494–500 (2015)
- [Sugiyama 16] Sugiyama, A., Sea, V., Sugawara, T.: Effective task allocation by enhancing divisional cooperation in multi-agent continuous patrolling tasks. In: IEEE 28th Int. Conf. on Tools with Artificial Intelligence. pp. 33–40 (2016)
- [Suseki 05] Suseki, K., Mizuguchi, T., Sugawara, K., Kosuge, K.: Proportion Regulation for Division of Labor in Multi-Robot System. In: Proc. of the 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. pp. 2339–2344 (2005)
- [Yoneda 15] Yoneda, K., Sugiyama, A., Kato, C., Sugawara, T.: Learning and relearning of target decision strategies in continuous coordinated cleaning tasks with shallow coordination. Web Intelligence 13(4), pp. 279–294 (2015)