特 別 研 究 報 告 書

題 目

ドローン配送における安全性と効率性を両立するマルチエージェント強化学習

指導教員

林　冬 惠

報 告 者

鄭　志 晟

岡山大学工学部工学科
情報・電気・数理データサイエンス系
情報工学コース

令和 7 年 2 月 7 日 提出

要約

物流業界では，新たな配送手段としてドローン配送が注目を集めている．ドローン配送は，無人航空機を用いて荷物を効率的に運搬する仕組みで，短距離への小型荷物配送や，アクセスが困難な遠隔地での運用に特に有用である．

しかし，ドローン配送では，複数のドローンが同時に空域を共有するため，衝突を回避しながら効率的に配送を行う経路計画が不可欠である．この課題はドローン配送問題（Drone Routing Problems, DRP）として定式化されている．オンデマンド配送の動的な経路計画に対応するためには，迅速に実用的な解を求めることが必要である．従来の探索型アルゴリズムは問題の規模が大きくなると計算時間が指数関数的に増大し，動的な経路計画への対応が困難である．そこで，近年ではマルチエージェント強化学習（Multi-Agent Reinforcement Learning, MARL）を用いた手法が注目されている．本研究では，MARL を用いた DRP の解決を検討する．

MARL はエージェントが試行錯誤を通じて最適な方策を学習する．DRP に適用する場合，学習過程でエージェント同士の衝突が頻発し，衝突がない効率的な経路計画を学習することが困難である．先行研究では，安全なマルチエージェント強化学習を用いることで安全性を確保する手法が提案されたが，安全性を向上させる一方で，エージェントの移動効率が低下するという課題が指摘されている．また，従来の報酬設計ではマップが大規模化するにつれて正の報酬が疎になるため，効率的な学習が困難であるという課題も存在した．

本研究では，学習効率とエージェントの移動効率を総合的に考慮した効率性と安全性の両立を実現するため，新たな手法を提案する．従来の報酬設計に加え，エージェントとゴール間の距離変化に応じた報酬を導入した．この報酬は，エージェントがゴールに近づくなら正，離れるなら負となり，その度合による連続値をとる．また，エージェントの安全性を確保しつつ，ノード占有率を抑える新たな安全制御を提案した．これにより，ノード上の過剰な停止を抑え，システム全体の移動効率を向上させた．さらに，分散的に安全制御を行う設計にすることで，エージェント数やマップサイズの増加に対するスケーラビリティを実現した．

エージェント数やマップ種類を変えて様々な実験を行った結果，提案手法は検証したすべての状況で衝突を回避させることができた．また，従来手法と比較して，学習の収束が大幅に早くなり，モデル性能が向上することを示した．特に，非グリッドなマップではより大きな向上が見られた．また，分散型安全制御は，従来の中央集権型安全制御と同等のゴール率およびコストに達成可能であり，中央集権方式が抱える課題を克服する可能性を示唆している．これらの結果から，提案手法は安全性と効率性を両立し，従来手法より優れた性能を発揮することが示された．

目次

図 目 次

表 目 次

第 1 章

はじめに

近年，物流業界は人手不足や環境負荷，サプライチェーンの複雑化といった課題に直面しており，技術革新やデジタル化，自動化を活用して効率的なシステムの構築が求められている．そこで，新たな物流手段としてドローン配送が注目されている．ドローン配送とは，無人航空機（ドローン）を使用して荷物を指定された場所へ配送するシステムである．特に，短距離の小型荷物の配送や，アクセスが困難な遠隔地への配送において有用性がある．また従来の配送手段と比べて環境負荷を軽減することも可能である．そのため，トラックやバイクなどの地上輸送手段を補完，代替する存在として期待されている．ドローン配送を想定するとき，同じエリア内を複数のドローンが飛行する状況において，各ドローンは目的地まで衝突なく移動することが求められる．このような課題を解決するため，ドローンが衝突なく効率的な配送計画をするための経路最適化問題として定式化されたドローン配送問題（Drone Routing Problems, DRP）[2] がある．

DRP は，衝突回避を主とした Multi-Agent Path Finding（MAPF）問題 [3] として考えることができる．経路計画の手法には，大きく分けて，A*アルゴリズムのような探索による手法と，行動を逐次的に選択することで経路を求める手法が存在する．前者は，エージェントが複数の場合，行動空間が指数関数的に増大するため，DRP のような問題には適さない．DRP で想定されるオンデマンド配送による動的な経路計画を行うには，最適でなくとも実用的な解を迅速に求める必要がある．後者に関して，近年ではマルチエージェント強化学習（Multi-Agent Reinforcement Learning, MARL）を用いた手法が提案されている [5] [6].

MARL では，エージェントの試行錯誤によって最適な方策を学習する．MARL を用いて DRP を学習させる場合，学習過程でエージェントの衝突が頻発する [4]．また，エージェント A が適切な行動を取った場合であっても，エージェント B が危険な行動（例えば，エージェント A に衝突するような行動）をした場合，結果的には低い報酬が与えられてしまい，行動を適切に評価させることが難しい [2]．以上の理由から，従来の MARL を単に DRP に適用するだけでは，衝突のない効率的な経路計画の方策を学習することが困難である．先行研究では，学習段階でエージェントの安全を確保することで．衝突を防ぎ，従来の MARL による学習と比較してモデル性能を向上さえることができた．しかし，学習効率の低さや，マルチエージェントシステム全体の移動効率の低下といった問題が指摘されている [4].

岡山大学工学部工学科
情報・電気・数理データサイエンス系
情報工学コース

そこで本研究では，学習効率の改善と，エージェントの安全性および移動効率の向上を両立するための手法を提案する．現状の DRP では，ゴール到達時のみ正の報酬を与えるため，マップが大きくなると学習初期での報酬が疎になり，効率的に学習を進めることが困難であった．そこで Potential-based Reward Shaping [12] をもとに，エージェントとゴール間の距離変化に応じた報酬を毎ステップ与える．これにより，エージェントがゴールへ向かうための動機づけを促進することを目指す．次に，エージェントの安全性と移動効率を両立させるため，衝突回避によるノード上での待機行動を抑えた安全制御を設計した．これにより，エージェントのノード占有率を低減させ，安全制御によるシステム全体の停滞を防ぐことを可能にした．また先行研究 [4] では，マップ内のエージェントは安全制御により中央主権的に管理されていたため，エージェント数の増加に伴うスケーラビリティの低さが課題であった．そこで，安全制御を分散的に運用可能な形に拡張を行った．

本稿は次のように構成されている．第 2 章では，研究背景として DRP の定義や MARL の概要を説明し，さらに，MARL を DRP に適用する際の課題を明確にする．第 3 章では，関連研究として安全なマルチエージェント強化学習や，先行研究における取り組みと課題について述べる．第 4 章では，本研究の提案手法を重要な要素ごとに詳細に説明する．第 5 章では，提案手法の有効性を検証するための評価実験の結果とその考察を示す．第 6 章では，本研究の総括と今後の展望を述べる．

第 2 章

背景

近年，EC 市場の急速な拡大により配達量が増加し配達員の負担が大きくなっている．配送拠点から最終顧客までの配送区間はラストワンマイルと呼ばれ，効率化を図ることが難しいとされている．その要因として，再配達や細かな配達時間帯の指定，交通渋滞などがある．また，物流業界では人手不足も深刻な問題となっている．そこで，新たな配送手段としてドローンを活用して配達を行うドローン配送が注目されている．ドローンを用いることで人手不足の助けになるだけでなく，空中を利用することで配送時間の短縮，また離島などの物流困難者に対しても有効であると考えられる．

本章では，まずドローン配送問題（Drone Routing Problems, DRP）について説明する．その後，マルチエージェント強化学習（Multi-Agent Reinforcement Learning, MARL）について説明し，従来の MARL を DRP に適用する際の課題について述べる．

2.1 ドローン配送問題

ドローン配送問題 (DRP) とは，ドローン配送における経路最適化問題を定式化したものである [2]．DRP では，配送エリアを図 2.1，図 2.2 のような無向グラフ $G =< V, E >$ で表す．ここで，$V = \{v_1, ..., v_{|V|}\}$ は位置情報 $l_k = (l_k^x, l_k^y)$ を持つノード v_k の集合である．また E は，$E = \{(v_j, v_k)\}$ として v_j と v_k をつなぐエッジ集合を表す．ノード v_a，v_b を連結するエッジ e_{ab} は連続値を持ちノード間の距離を表す．ドローンを模したエージェントの集合は，$N = \{1, 2, ..., i, ..., |N|\}$ と表され，各エージェント i にはエピソードごとにスタート地点 $s^i \in V$ とゴール地点 $g^i \in V$ が与えられる．エージェントは，特定のノードに向かって移動するか，もしくはその場に停止するという行動を選択する．ゴールに到達したエージェントは，そのエピソード終了までそのゴール地点で停止する．エージェントが 1 ステップで移動できる距離は $speed$ で定義される．また，DRP では以下の制約がある．

- 2 体のエージェントは同じノードに存在できない
- 2 体のエージェントは同じエッジを逆行できない

エピソードの終了条件は，すべてのエージェントが各ゴールに到達するか，任意の 2 体のエージェントが衝突するか，一定時間経過による強制終了の 3 つである．エージェント i の T ステップまでの行動の軌跡を $path^i = (l^i[0], l^i[1], ..., l^i[T])$ と表すとき，$l^i[0] = s^i$ であり，t ステップでゴール地点に到達したなら $l^i[t] = g^i$ となる．エージェント i がゴール地点 g^i に到達したとき，それ以降のステップは g^i で待機するため，$l^i[t']_{t'>t} = g^i$ となる．エージェント i の行動の軌跡を $path^i$ とするとき，コスト関数 $cost$ は次のように定義される．

$$cost(path^i) = \sum_{t=0}^{T-1} \|l^i[t+1] - l^i[t]\|_2 \tag{2.1}$$

DRP の目的は，上記の制約を満たしながら総移動コストを最小にする各エージェント i の経路を計画することである．すなわち,

$$min \sum_{epi} \sum_{i} cost(path^i_{epi}) \tag{2.2}$$

$$st. \forall i \in N, l^i[T] = g^i_{epi} \tag{2.3}$$

と表される．ここで，$l^i[T] = g^i_{epi}$ は，エージェント i が最大ステップ T でゴール地点 g^i_{epi} に存在していることを意味する．g^i_{epi} の表記は，エピソードごとにゴール地点が変化することを表している．

MARL を用いた DRP のためのシミュレーション環境として，MARL4DRP [7] がある．MARL4DRP では，エージェント間の距離が $speed$ 未満の場合，衝突が発生したと判定される．また，2024 年から International Conference on Autonomous Agents and Multiagent Systems（AAMAS）において，DRP Challenge というコンペティションが開催されている [15]．DRP Challenge は，複数のドローン配送シナリオに基づき，現実世界の地図上で複数のドローンに衝突のない最適な経路のセットを特定することを目的としている．実世界の配送シナリオを模倣した仮想プラットフォームを使用し，参加者は安全で効率的かつ費用対効果の高いドローン配送を促進するアルゴリズムを開発することが求められる．図 2.2 の非グリッドなマップは，AAMAS 2024 DRP Challenge にて使用されたマップである．

2.2 マルチエージェント強化学習

本研究では，マルチエージェント強化学習（MARL）のマルコフゲームによる定式化を基盤としている [10]．マルコフゲームとは，複数の主体が存在する場合のマルコフ決定過程のことである．マルコフゲームはタプル $(N, S, \{A^i\}_{i\in N}, P, \{R^i\}_{i\in N}, \gamma)$ で表現される．ここで，$N = \{1, ..., n\}$ はエージェントの集合，S はすべてのエージェントによって観測される状態空間，$A = A^1 \times \ldots \times A^n$ はすべてのエージェントの共同行動空間である．A^i はエージェント i の行動空間を表す．P は状態遷移確率であり，$P(s'|s, a) : S \times A \to [0, 1]$ で定義される．$R^i : S \times A \times S \to \mathbb{R}$ は，エージェント i の即時報酬関数である．$\gamma \in [0, 1]$ は，将来得られる報酬の割引率である．時刻 t での状態 s_t と共同行動 $a_t = \{a^1_t, ..., a^n_t\}$ から，確率 $P(s_{t+1}|s_t, a_t)$

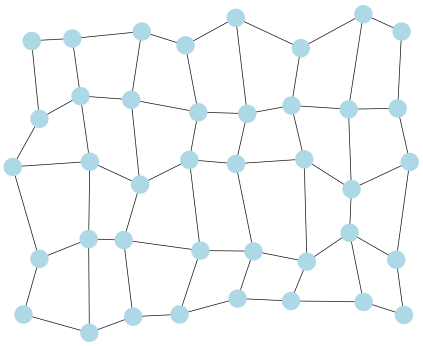


図　2.1: 8×5 のグリッドマップ

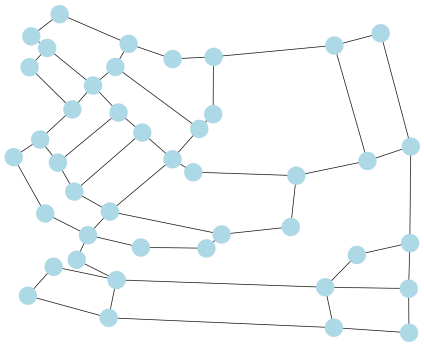


図　2.2: 実際の街をもとにしたマップ (Aoba00)

によって状態 s_{t+1} に遷移する．このとき，エージェント i は報酬 $R^i(s_t, a_t, s_{t+1})$ を受け取る．エージェント i の目的は，将来の累積報酬の期待値 $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R^i(s_t, a_t, s_{t+1})]$ を最適化する方策 $\pi^i : (a^i|s)$ を学習することである．エージェント i の行動は自身の方策だけでなく，他のエージェントの選択にも影響する．

近年の MARL では，中央集権型学習分散型実行（Centralized Training Distributed Execution, CTDE）が重要とされている．学習時はすべてのエージェントの情報を含む環境のすべての情報を利用し，実行時は各エージェント自身の部分観測のみを入力として方策を決定する．CTDE に基づくアルゴリズムには，MADDPG(Multi-Agent Deep Deterministic Policy Gradient) [11] や，QMIX [5] などがある．MADDPG は，協調と競争を含むような混合タスクに適し，連続行動空間に対応している．Actor-Critic 方式で学習により，各エージェントは独立した方策を学習しつつ，集中型 Critic により他のエージェントの情報を考慮した価値評価を行う．一方，QMIX は協調タスクに適し，離散行動空間に対応する．各エージェントが独立に Q 値を学習し，中央集権型の学習モジュールによってグローバルな Q 値を統合して最適化を行う．QMIX は，完全な協調型タスクで，すべてのエージェントが共同で目標を達成する場合に適しており，本研究ではベースとなる MARL アルゴリズムとして QMIX を採用した．

2.3　MARL を DRP に適用する際の課題

前述したとおり，DRP の目的は，すべてのエージェントが衝突なく各目的地へ到達することである．MARL はエージェントの試行錯誤によって最適な方策を学習するため，学習段階でエージェントの衝突が頻発する．そのため従来の MARL 手法を単に DRP に適用するだけでは，エージェントが安全かつ効率的な経路計画を学習することは困難である．

本研究の目的は，DRP において安全性を考慮しつつ効率的な経路計画を学習するための手法を提案することである．

第 3 章

関連研究

DRP の特性上，衝突が起きないという安全を確保することは必須である．本章では，安全を考慮した強化学習に関する研究について述べる．また，2.3 節で説明した課題について，先行研究における成果と問題点を述べる．

3.1　安全なマルチエージェント強化学習

強化学習において，エージェントは環境の状態を観測し，長期的な報酬を最大化することを目的として行動を選択する．しかし，高価なロボットプラットフォームのようにエージェントの安全性が特に重要とされるシナリオにおいては，長期的な報酬の最大化に加えて，物理的損傷やその他のリスクの回避を考慮する必要がある．このような背景のもと，近年では強化学習を実世界の問題に適用する際に安全性を保証するための安全な強化学習に関する研究が活発に行われている [8] [9]．一方，安全なマルチエージェント強化学習 (MARL) に関してはまだ十分に研究が行われていない．Elsayed-Aly らによる研究 [1] では，シールドを用いた安全なマルチエージェント強化学習のフレームワークが提案された．シールドとは，エージェントを監視し安全な行動になるように管理する役割をもつ．同研究では，一つのシールドが全エージェントを管理する Cetralized shielding と，複数のシールドで分割統治的にエージェントを管理する Factored shielding が提案された．エージェントと環境との間にシールドを挟み，各エージェントの行動と状態をシールドに入力する．シールドは各エージェントの行動が安全であるか判断し，もし安全でないなら安全な行動に置き換える．シールドは安全な行動を学習環境とエージェントに与え，学習環境は安全な行動から得られる状態と報酬をエージェントに渡す．このときシールドは，危険な行動をしたエージェントにペナルティとして負の報酬を与える．Factored shielding では，対象の問題をサブ問題に分解し，サブ問題ごとに独立にシールドを適用することで，分割統治的な処理が可能であることを示した．

同研究の提案手法は，既存の MARL にシールドを導入することで，学習時のエージェントの安全性を保証することを示した．この検証は，グリッドワールド上に配置された 2 体のエージェントが衝突を避けながらゴール地点を目指すタスクで実施された．そのため，エー

ジェントの行動は上下左右または停止の 5 種類であり，1 ステップで隣接のグリッドへ移動する．一方で，DRP で扱いたい実際の街をもとにしたマップ（例，図 2.2）では，非グリッドなマップ構造であり，ノード間の移動時間は連続量であるため，各エージェントが次の行動を決定・開始する時刻は同時ではない．このため，DRP の環境は，従来のマルチエージェント経路計画問題と比べ，より複雑である．

3.2　DRP への応用

Kaji らによる研究 [4] では，安全なドローン配送を目的として，視野の追加と，前述した Centralized shielding のフレームワークをもとに，方策から得られたすべてのエージェントの行動に安全制御を実施することで，エージェントの行動を管理する安全制御を提案した．視野とは，エージェントの周囲に他のエージェントが存在する場合，その情報を状態表現に組み込むことで実現される．通常，エージェントの状態表現には，現在の位置およびゴールに関する情報のみ含まれる．しかし，視野の概念を導入することで，エージェントは周囲に他のエージェントが存在するかどうかを認識することが可能になる．この視野は，MARL4DRP [7] で実装されている．

Kaji らの提案した安全制御では，特定の行動を危険な行動と定義し，危険な行動を安全な行動に変更することで衝突を回避させた．危険な行動とは，他のエージェントと同じエッジを通る行動や，次に目指すノードが他エージェントと同じである行動である．また，安全制御の対象は，次に進行するノードが未定のエージェント，すなわちノード上にいるエージェントに限定している．安全制御により危険と判断されたエージェントは，ノード上で待機する．この手法では，エージェントの衝突をほとんど無くすことができ，従来の MARL 手法に比べて性能が向上することが示された．

一方で，以下の課題が存在する．第一に，学習効率の低さである．これは，現状の DRP における報酬設計に改善の余地があることを示唆しており，特に図 2.2 で示したマップで学習を行った場合，学習の収束が著しく遅くなることが確認された．第二に，エージェントの安全性を確保できる一方で，移動効率が低下する点である．この原因として，安全制御によりエージェントがノード上で待機する頻度が増加し，結果として他のエージェントの行動も停滞することが挙げられる．第三に，中央主権的な安全制御による，エージェント数に対するスケーラビリティの低さである．本研究では，これらの課題を解決し，エージェントの安全性と効率性を両立した手法を提案する．

第 4 章

提案手法

3.2 節で，MARL を DRP に適用したときの現状の課題について説明した．これらを解決するため，本研究では以下の 2 つの要素からなる新たな手法を提案する．第一に，エージェントとゴール間の距離変化に基づく報酬を導入し，効率的な学習を促す．第二に，ノード占有率を抑えた安全制御の設計により，エージェントの移動効率を保つことを目指す．また，エージェント数が増加した場合にも，安全制御をスケーラブルに適用するための拡張を試みる．

4.1 ゴールまでの距離変化に応じた報酬の導入

従来の DRP における報酬設計 [4] [7] は，エージェントがゴールしたとき正の報酬を与え，停止，衝突，移動の場合は負の報酬を与える．マップが大きくなると学習の初期段階でエージェントがゴールに到達することが難しくなるため，正の報酬が疎になり効率的な学習が困難である．

Ng らによる Potential-Based Reward Shaping (PBRS) [12] では，タスクの環境や特性に基づいて設計されたポテンシャル関数を元の報酬関数に加えることで，方策の最適性を保持しつつ，探索効率を向上させることが示された．そこで本研究では PBRS を DRP に応用することで，学習効率の向上を目指す．各ステップで，エージェントとゴール間のユークリッド距離と，その次のステップにおけるユークリッド距離の差に応じた報酬 R^i_{step} を追加する．

$$R^i_{step} = \omega * \Delta d^i_t, \quad \omega \in \mathbb{R}_{>0} \tag{4.1}$$

$$\Delta d^i_t = d^i_t - d^i_{t+1} \tag{4.2}$$

$$d^i_t = \sqrt{(g^i_x - l^i_x)^2 + (g^i_y - l^i_y)^2} \tag{4.3}$$

ω は重み係数である．エージェント i の座標を $l^i = (l^i_x, l^i_y)$ とするとき，d^i_t が時刻 t におけるエージェント i のゴール $g^i = (g^i_x, g^i_y)$ までの距離，Δd^i_t は次の時刻での距離との差分である．エージェントが 1 ステップで移動する距離は $speed \in \mathbb{R}_{\geqslant 0}$ で定義されるため，$\max \Delta d_i = speed$

である．よって，$R^i_{step} \in [-\omega{*}\mathrm{speed}, \omega{*}\mathrm{speed}]$ となる．この報酬を追加することで，エージェントがゴールに向かう方向へ誘導されることが考えられ，大規模なマップにおいても効率的に学習が進むことが期待される．

4.2　ノード占有率を抑えた安全制御

3 章で紹介した Kaji らの手法における安全制御 [4] では，制御対象のエージェントはそのステップにおいてノード上にいるエージェントのみである．方策による行動が危険な行動だった場合，エージェントを停止させるため，そのエージェントは安全な行動と判断されるまでノード上で待機することになる．ノード上で待機する間，他のエージェントはそのノードへ移動できなくなる．無向グラフ内のエージェント密度が小さい場合には，この手法でも問題なく機能するが，エージェント密度が大きくなると，安全制御によりノード占有率が増え，システム全体の動作が停滞する．また，エージェントの接近度を考慮しないため，本来衝突しない場面でも停止させられるため，過剰な停止が発生する可能性がある．その結果，衝突を減少させることは可能であるが，エージェントの移動効率が低下する．

そこで本研究では，制御対象のエージェントを移動中のエージェントを含むすべてのエージェントとし，エージェントの接近度を考慮した設計を導入することで，ノード上での過剰な停止を防ぎ，移動効率を維持しながら衝突を防ぐ安全制御手法を提案する．提案する安全制御の概要を説明する．まず，エージェントの接近度を考慮するために図 4.1 のような危険領域 M を定義する．危険領域とは，衝突が予測される場所を中心とした円状の領域であり，半径 $r = speed + \alpha$ で定める．$\alpha \in \mathbb{R}_{>0}$ は，どの程度接近を許容するか調整するためのパラメータである．次のステップで，危険領域 M に 2 体のエージェントが侵入しているなら危険な行動と判断し，安全な行動に変更する．図 4.1 は，2 体のエージェント（緑と青）が同じノードに向かって移動している場合 (図 4.2 のケース 1) の危険な行動（右の例）となる状況を表している．

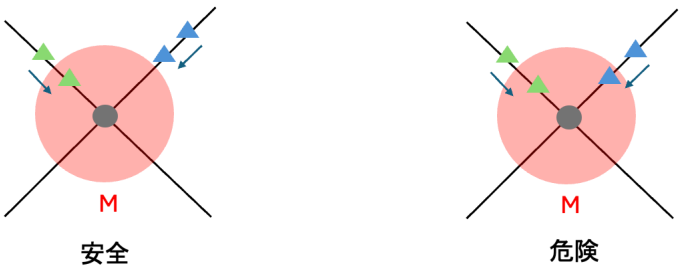


図　4.1: ノードを中心とした危険領域 M の例

2 体のエージェント (赤と青) が図 4.2 のような配置のときを考える．丸はノード，直線はエッジを表し，丸の中にエージェントがいるときはノード上にエージェントがいることを表す．危険領域 M は青エージェント視点のものとする．このときの，処理を次に示す．

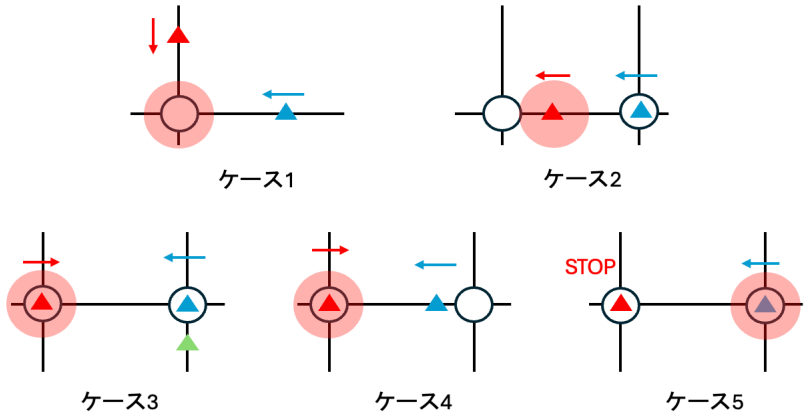


図　4.2: 2 体のエージェントの配置例

ケース 1) 行先ノードが赤エージェントのゴール地点ならば，エッジ上で停止後，青エージェントを優先して通過させる．このとき，赤エージェントが危険領域 M 内に存在するならば，青エージェントは迂回する（別のノードへ進む）．どちらのゴール地点でもないなら，交点に近い方を優先する．

ケース 2) 行先ノードが赤エージェントのゴール地点ならば，青エージェントは迂回する．そうでないなら，危険領域 M を赤エージェント中心としてエッジ上を移動.

ケース 3) 行動選択肢 (進行先に他のエージェントがいないエッジ) が多い方が迂回をする．図 4.2 の例だと赤エージェントが迂回する．

ケース 4) 赤エージェントが迂回する.

ケース 5) 青エージェントがすでにゴール済みならば赤エージェントは迂回，そうでないなら危険領域 M を青エージェント中心としてエッジ上を移動する．

前提として，エッジを移動中のエージェントは，移動途中で引き返すことはないものとしている．そのため，ケース 1 の場合では，行先ノードが片方のゴール地点でも引き返す行動が発生しないように，行先ノードへの接近度に応じて，相手を先に通過させるか迂回させるか判断している．

提案する安全制御では，安全な行動は，元の行動維持，エッジ上の停止，迂回，ノード上の停止の優先度に従って決定される．これにより，ノード上の停止を減らしエージェントの移動効率を保つことが期待される．

4.3　分散型安全制御への拡張

Kaji らによる中央集権型安全制御（図 4.3）では，エージェント数に対するスケーラビリティの低さが問題であった．これを解決するために，3.1 節で紹介した Factored shielding [1] を基に，DRP のための分散型安全制御を設計する．図 4.4 に分散型安全制御のアーキテクチャを示す．マップ割当の段階では，エージェントの状態と行動に基づき，適切なサブマップに割り当てる．各サブマップは互いに独立しており，サブマップごとにエージェントに対する安全制御を行う．統合の段階では，各サブマップで決定されたエージェントの行動を統合した際に衝突が発生しないか監視する．例えば，あるエージェントが別のサブマップに移動する場合，現在のサブマップ内で安全と判断された行動が，移動先のサブマップでは危険とみなされる可能性がある．この場合，統合の段階で新たな安全な行動に修正する．

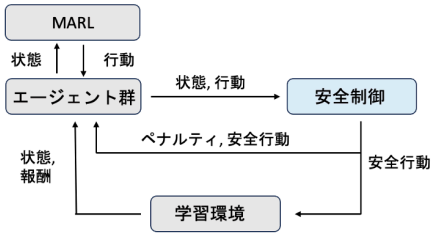


図　4.3: 中央集権型安全制御

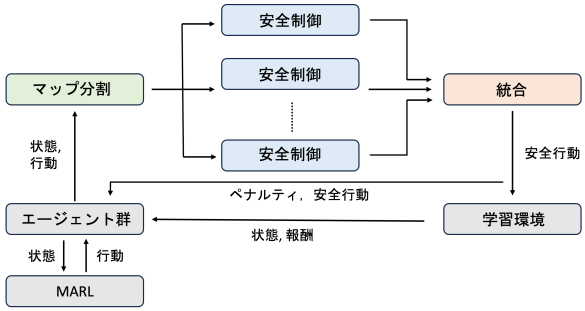


図　4.4: 分散型安全制御

各サブマップ内では，属するエージェントに対して中央集権型と同様に提案手法を適用する．DRP は無向グラフであるため，マップをサブマップに分割するとき，ノード上を分割境界とする．本研究では，分割境界上に位置するノードを共有ノードとよぶ．共有ノードは複数のサブマップに属するため，共有ノード上にいるエージェントは，複数のサブマップにおける安全制御の結果を受け取ることになる．統合段階では，共有ノード上にいるエージェントに対して，複数のサブマップからの安全制御による結果の整合性を確認する．整合性が取れていないとき，安全な行動に修正する．このように，統合段階では，共有ノードを監視することで，整合性の取れた最終的な安全制御の結果を出力する．

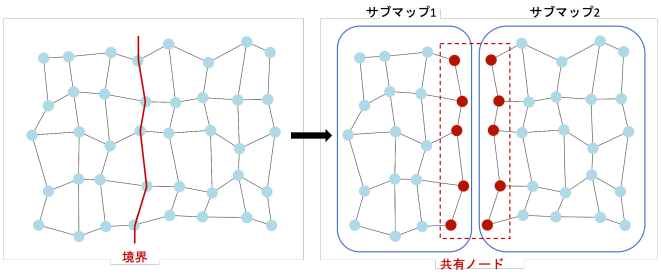


図　4.5: 2 つのサブマップに分割した例

第 5 章

評価

本章では，まず実験の設定について説明する．提案手法の有効性を検証するために，エピソードにおける衝突率，ゴール率，タイムアップ率，コストを比較する．エピソード終了状態は，衝突，ゴール（すべてのエージェントが目的地に到達したとき），タイムアップの 3 つ存在しており，衝突率，ゴール率，タイムアップ率とはそれぞれの状態で終了するエピソードの割合である．2 章で，DRP におけるコストを各エージェントの目的地に到達するまでの軌跡の距離の総和で定義したが，本研究の MARL では，ステップ数が距離に比例するため，ステップ数によってコストを再定義する．コスト C は，各エージェントが目的地に到達するまでのステップ数の総和と定義する．つまり，$C = \sum_{i=1}^{N} c_i$ と表せる．ここで，N はエージェント数，c_i はエージェント i が目的地に到達するまでに要したステップ数である．衝突または上限ステップ数以内に，すべてのエージェントが目的地に到達できなかった場合，コストをエージェント数と上限ステップ数との積として計算する．すなわち，1 エピソードにおける上限ステップ数を T とすると，コストは $C = N \times T$ である．このコストによって，効率的に移動できているか評価する．

5.1 実験の設定

DRP のためのシミュレーション環境は，Ding らが提供している MARL4DRP [7] を用いる．MARL4DRP の環境は，OpenAI が提供している Gym [13] の環境に準拠しているため，Gym 環境で使用できる強化学習フレームワークである EPyMARL [14] を用いる．学習のための MARL アルゴリズムは，QMIX [5] を用いる．また，先行研究との公平な比較を行うため，本研究におけるすべての手法に対してエージェントの視野（3.2 節参照）を追加するものとする．

エピソードの終了条件は，エージェントの衝突，すべてのエージェントが目的地へ到達する，または上限ステップ数の経過してタイムアップである．本研究では，すべてのエージェントが $speed = 5$ である．基本の報酬設定は，表 5.1 となっている．

表　5.1: 基本の報酬設計

エージェントの行動	報酬値
停止	$-10 \times speed$
移動	$-1 \times speed$
衝突	$-10 \times speed$
目的地到達	100

表　5.2: 各手法の説明

アルゴリズム名	説明
QMIX	中央集権学習分散実行型の価値分解手法
QMIX-SC（先行手法 [4]）	QMIX と Kaji らによる安全制御 (Safety Control，SC) を組み合わせた手法
QMIX-DbRS	QMIX にゴールまでの距離変化に応じた報酬 (Distance-based Reward Shaping，DbRS) を導入した手法
QMIX-ESC	QMIX とノード占有率を抑えた安全制御 (Enhanced Safety Control，ESC) を組み合わせた手法
QMIX-ESC&DbRS（提案手法）	QMIX とすべての提案要素を組み合わせた手法

5.2　提案手法の有効性の検証

ゴールまで距離変化に応じた報酬（4.1 節）における重み係数 ω は，パラメータ調整を目的とした事前検証実験の結果に基づき，$\omega = 5$ と設定した．また，ノード占有率を抑えた安全制御における危険領域 M（4.2 節）の α を 1 とした．提案手法および比較する手法を表 5.2 に示す．また，表 5.2 に記載されたアルゴリズム名は，以降の議論において参照するものとする．

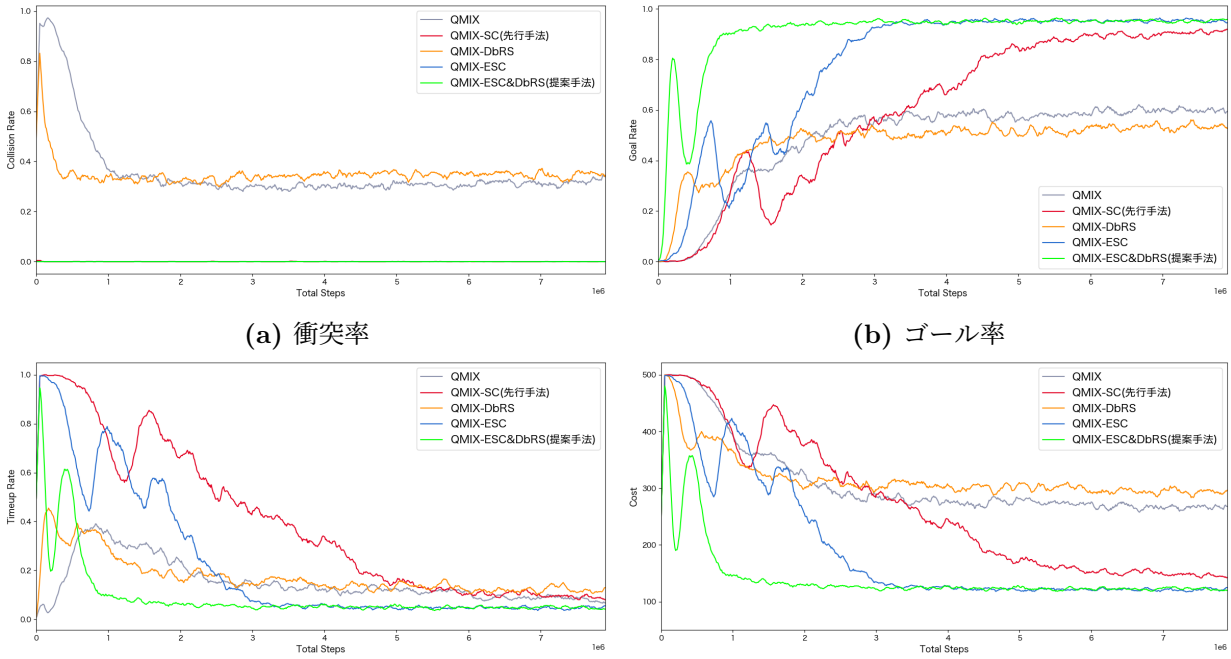
5.2.1　各手法間の性能評価

本節では，先行手法および提案手法の各構成要素を個別に適用した手法との比較結果を示す．これにより，提案手法の性能を評価するとともに，提案手法を構成する各要素の単独における効果についても検証を行う．マップは，8×5 のグリッドマップ（図 2.1）と実際の街をもとにした非グリッドマップ（図 2.2）を用いた．それぞれのマップ規模および学習の収束度合いに基づいて，実験設定を表 5.3 とした．上限ステップ数とは，1 エピソード内の最大ステップ数である．最大ステップ数とは，学習全体における累積ステップ数の上限を指し，最大ステップ数に達すると学習を終了する．

表　5.3: 実験 A,B の設定

	マップ	エージェント数	上限ステップ数	最大ステップ数
実験 A	8×5map	5	100	8M
実験 B	Aoba00	5	200	15M

まず，実験 A の結果を図 5.1 に示す．衝突率に関しては，QMIX および QMIX-DbRS 以外の手法において，衝突率はほぼ 0%となった．ゴール率に関しては，先行手法が約 90%であったのに対し，ノード占有率を抑えた安全制御（ESC）を導入した QMIX-ESC および提案手法では約 95%に向上した．コストのグラフ（図 5.1d）から各手法の収束過程を観察すると，ゴールまでの距離変化に応じた報酬（DbRS）を導入した提案手法が最も迅速に収束し，約 2M ステップ付近で収束していることがわかる．先行手法と比較すると，提案手法は早い段階で収束し，ゴール率も約 5%程度高いことがわかる．また，QMIX-ESC と提案手法を比較すると，最終的なコストは同等であるにもかかわらず，提案手法の方がより早く収束している．この結果から，DbRS を導入することで，性能を維持したまま効率的な学習が可能であることが示唆される．さらに，提案手法を含む ESC を導入した手法は，より低コストであることから，ノード占有率を抑制し，エージェントの移動効率を維持できていると考えられる．



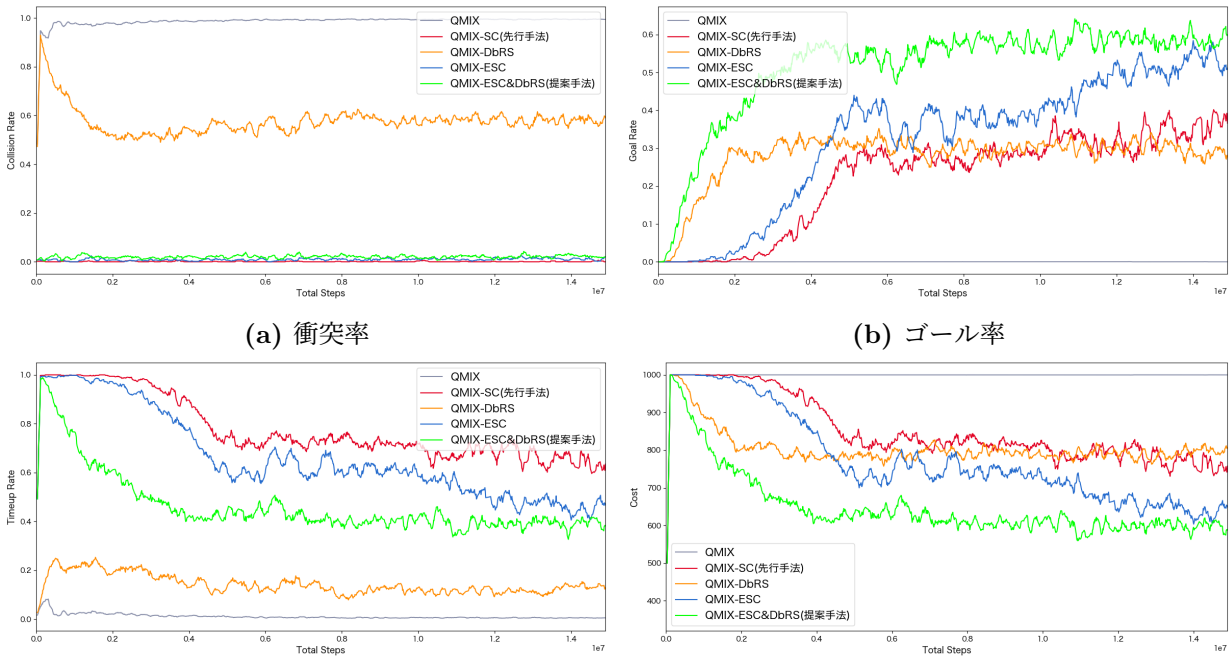
(a) 衝突率　(b) ゴール率

(c) タイムアップ率　(d) コスト

図　5.1: 実験 A の結果

次に，実験 B の結果を図 5.2 に示す．まず衝突率に関して，グラフ（図 5.2a）によると，先行手法はほぼ 0%であるが，ESC を用いた QMIX-ESC および提案手法では，わずかに衝突が発生していることが確認された．この原因については，5.4 節で後述する．ゴール率の

グラフ（図 5.2b）から，実験 A と同様に DbRS を導入した QMIX-DbRS および提案手法は，いずれも収束が早くなっている．先行手法による最終的なゴール率は約 38%であったのに対し，提案手法では約 59%となり，大幅な向上が見られた．グリッドマップ（図 2.1）を用いた実験 A では，その差が約 5%程であったこと踏まえると，非グリッドマップ（図 2.2）を用いた実験 B では，提案手法がより効果的に機能したと考えられる．しかし，実験 A と比べるとゴール率は約 60%と低下した．



(a) 衝突率　(b) ゴール率

(c) タイムアップ率　(d) コスト

図　5.2: 実験 B の結果

5.2.2　高エージェント密度環境における評価

次に，エージェント数を増やし，マップ内のエージェント密度を高めた場合の，提案手法の有効性について検証した．マップは前節の実験 A および B と同様のマップ（図 2.1，図 2.2）を用い，エージェント数の増加を考慮して最大ステップ数を 30M とした．実験の設定を表 5.4 に示す．また，本節では提案手法と先行手法による移動効率の変化を調べるため，新たに「探索成功時のエピソード長」を比較する．ここで，探索成功時とはすべてのエージェントが各ゴールに到達した状態を指す．前節，実験 A および B の 5 エージェントにおける探索成功時のエピソード長を図 5.3 に示す．

8×5 のグリッドマップでは，5 エージェントの場合，先行手法と提案手法の間で探索成功時のエピソード長の差は約 2 ステップであった（図 5.3a）．一方，エージェント数が増加した場合，10 エージェントでは約 10 ステップ，13 エージェントでは約 12 ステップの差が生じ

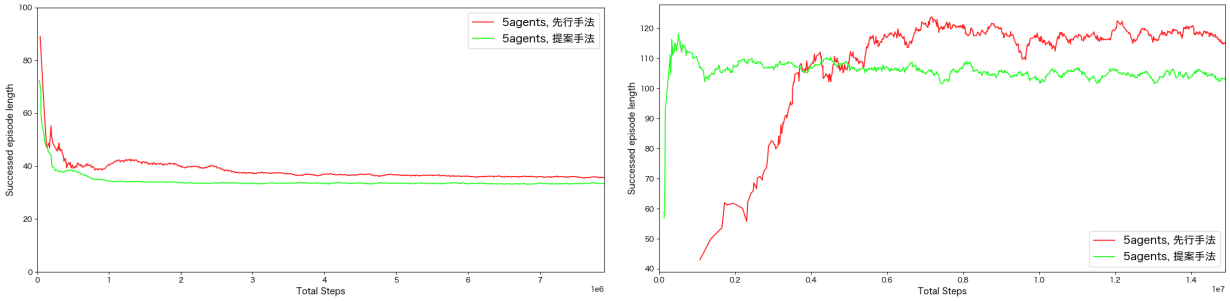
表　5.4: 実験 C,D の設定

	マップ	エージェント数	上限ステップ数	最大ステップ数
実験 C	8×5map	10, 13	100	30M
実験 D	Aoba00	8, 10	200	30M

た（図 5.4b）．さらに，両手法におけるゴール率の差は，5 エージェントでは約 5%であったが，10 エージェントおよび 13 エージェントでは約 12%の差に拡大した．また，学習の収束に関して，実験 A では提案手法が約 2M ステップ，先行手法が約 8M ステップで収束した．一方，エージェント数を 10 および 13 エージェントに増やした実験 C では，提案手法は約 5M ステップ，先行手法は約 30M ステップで収束した（図 5.4b）．

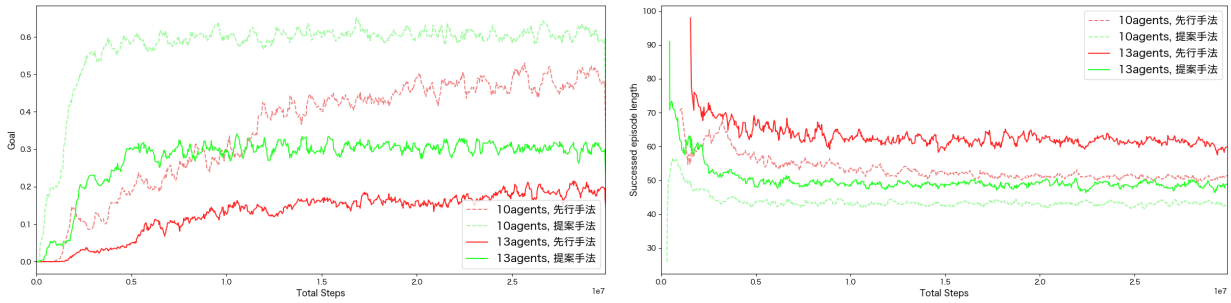
Aoba00 の非グリッドマップにおいて，両手法における探索成功時のエピソード長の差は，5 エージェントでは約 11 ステップであった（図 5.3b）．一方，8 エージェントでは約 21 ステップ,10 エージェントでは約 23 ステップの差が生じた．ゴール率に関しては，5 エージェントではその差が約 20%であったが，8 エージェントでは約 12%，10 エージェントでは約 7%となった．

両マップにおいて，提案手法は先行手法と比較して探索成功時のエピソード長が短縮され，特にエージェント密度が高まるにつれてその差が拡大することが確認された．この結果から，提案手法は先行手法と比べてエージェントの移動効率が向上していると考えられる．さらに，先行手法ではエージェント数の増加に伴い学習速度が急激に低下したが，提案手法では穏やかな低下を示した．



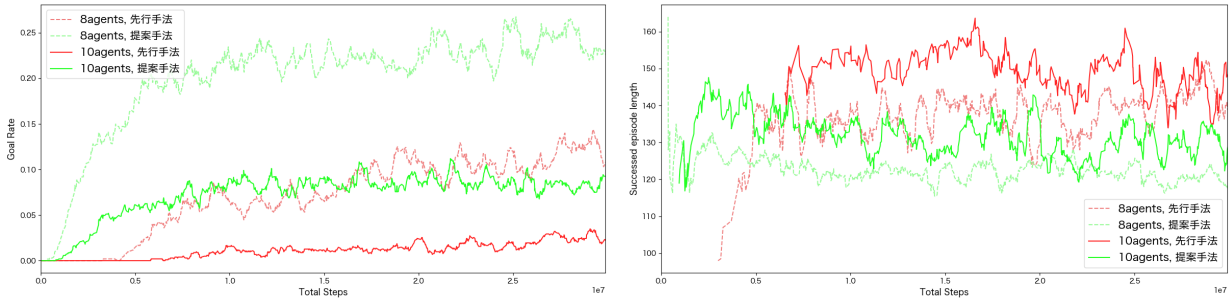
(a) 8×5map, 探索成功時のエピソード長　　**(b)** Aoba00, 探索成功時のエピソード長

図　5.3: 実験 A および B における 5 エージェントの探索成功時のエピソード長



(a) ゴール率　(b) 探索成功時のエピソード長

図　5.4: 実験 C の結果



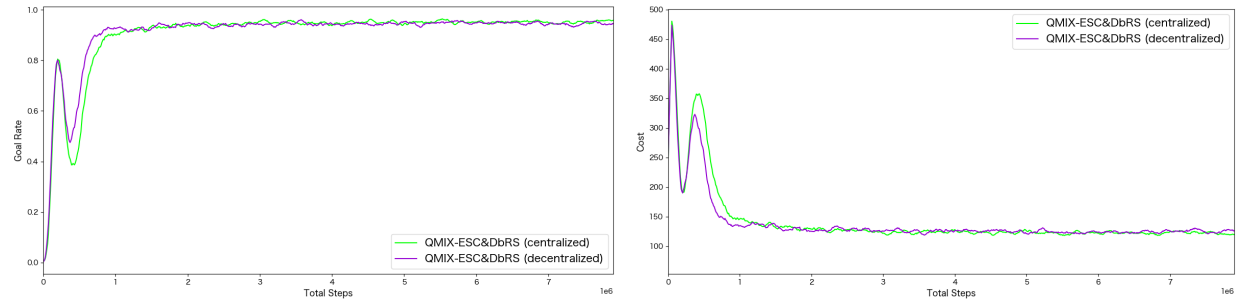
(a) ゴール率　(b) 探索成功時のエピソード長

図　5.5: 実験 D の結果

5.3　分散型安全制御の評価

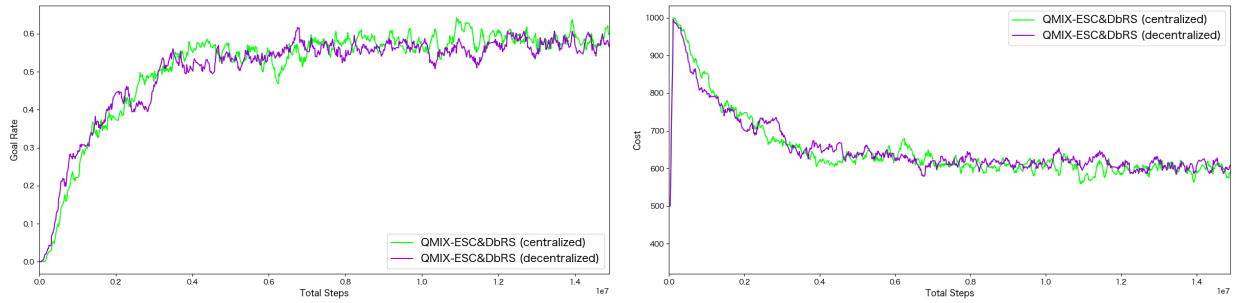
本節では，安全制御を分散実行した結果を示す．実験の設定は表 5.3 と同様である．マップの分割数は 4 とした．

提案手法を中央集権方式で実行した場合と，分散方式で実行した場合の比較を行った．その結果を図 5.6，図 5.7 に示す．どちらのマップでも，ゴール率およびコストに関して中央集権方式で実行した場合と同様の結果が得られた．このことから，分散型安全制御を行った場合でも中央集権的行う場合と同様の性能を発揮することができることがわかった．



(a) ゴール率　(b) コスト

図　5.6: 中央集権方式との比較 5agents, 8×5map



(a) ゴール率　　(b) コスト

図　5.7: 中央集権方式との比較 5agents, Aoba00

次に，エージェント数に対するスケーラビリティを検証するため，エージェント数を変えて実行時間を計測した．8×5 マップでは，1M ステップまでの実行時間，10×10 マップでは 0.1M ステップまでの実行時間を計測し，その結果を表 5.5 に示す．実行時間が短い方を太字で表す．表 5.5 の結果から，マップサイズやエージェント数を増加させた場合においても，分散型安全制御による時間的な有効性は確認されなかった．この原因として，プログラム全体に対する安全制御の計算負荷が軽微であることが考えられる．しかし，安全制御を分散的に実行可能であることが示されたため，将来的に安全制御による計算負荷が増大した場合において，本手法の有効性が期待される．

表　5.5: 中央集権方式および分散方式における実行時間（分: 秒）

設定	5 agents 8×5map	10 agents 8×5map	15 agents 10×10map	20 agents 8×5map	30 agents 10×10map
中央集権方式	**17:43**	**24:23**	9:28	57:35	**18:50**
分散方式	18:56	24:37	**8:47**	**54:03**	20:34

5.4　考察

提案手法として，エージェントとゴール間の距離変化に応じた報酬（DbRS）とノード占有率を抑えた安全制御（ESC）を組み合わせた手法を提案した．5.2.1 節で，提案手法の各構成要素単体の手法を含めた比較検証を行った結果，DbRS を導入することで先行手法より学習の収束が大幅に早まることが示された．特に，エージェント数が多い環境において，先行手法では学習速度が急激に低下していたが，DbRS を導入することで効率的に学習が進行することが確認された．また，ESC を導入することで，衝突を防止しつつゴール率やコストの改善が見られ，先行手法よりエージェントの移動効率が向上することが示唆された．さらに，これら 2 つの要素を組み合わせた提案手法は，先行手法と比較してより優れたゴール率およびコストを実現しつつ，学習効率の向上も可能であることが示された．

次に，5.2.1 節の実験 B において，ESC を導入した QMIX-ESC および提案手法によって，わずかに衝突が発生した原因について説明する．これは，マップの形状およびプログラムの

衝突判定方法（2.1 節参照）に起因している．実際に衝突が発生した箇所を観察したところ，すべて同一の領域で発生していることが確認された．この領域は，エッジ同士が鋭角（約 17 度）に交差しており，交点に近づくにつれてエッジ間の距離が急激に狭くなる．その結果，エージェントが異なるエッジ上に位置しながらも，衝突判定の基準内に同時に存在してしまうことが判明した．これを誤判定と扱うかは，現実世界の環境において同様の衝突が発生するかどうかに依存する．一方で，もしこのような衝突が許容されない場合，提案した ESC では，エージェントの安全を十分に確保できたとはいえない．エージェントの接近度に基づいて危険な行動かどうか判断していたが，エージェント同士の交差角度も考慮した設計とすることで，マップの形状に依存せず衝突回避が可能となると考えられる．

続いて，Aoba00 のような非グリッドマップにおけるゴール率の低下とその改善方法について検討する．先行手法でも同様の課題が議論されたが，その要因として，ノードの次数が小さいことによりエージェントの行動が限定されることがあげられる．さらに，ノードの接続関係によっては，エージェントが迂回する場合，大幅な移動距離の増加を生じる可能性があり，その結果，すべてのエージェントが一定時間内にゴールへ到達することが困難となる．また，このようなマップでは，エージェントに対して最適な迂回経路を学習させることも容易ではない．この課題を解決するための方法の一つとして，マップ情報（例えば，ノードの接続関係）に基づいてエージェントの行動優先度を決定する方法が考えられる．具体的には，ゴール優先度の高いエージェントを定め，それらを優先的に行動させることで，エージェント間の協調性を向上させ，不要な迂回の発生を抑制することが可能になると考えられる．

DRP の設定上，ゴールに到達したエージェントはエピソード終了までゴールノード上で待機するため，ノード占有率の増加を抑えることによりエージェントの移動効率を向上させた場合でも，ゴール済みのエージェントが増えるにつれてノード占有率は必然的に上昇する．このため，現状の DRP の設定では，ESC の真の有効性を十分に検証できたとは言い難い．この課題を解決するためには，ゴール到達後もエージェントが移動を続けるような問題設定を考えることで，移動効率の向上による影響をより適切に評価できたと考えられる．例えば，エージェントに対して次々に配送タスクが割り当てられる設定や，配送完了後に集荷地点へ戻る設定などが考えられる．このような問題設定の場合，すべてのエージェントが常にマップ上を移動しているため，より高い移動効率が求められる．このような状況では，提案手法の有効性がより顕著に発揮される可能性が高いと考えられる．

第 6 章

おわりに

ドローン配送問題の主たる目的は，エージェント同士の衝突が起きない効率的な配送経路を計画することであった．マルチエージェント強化学習（Multi-Agent Reinforcement Learning, MARL）の特性上，単に学習を進めるだけではこの目的を達成することは困難であった．

本研究の目的は，学習段階から衝突が起きないように安全制御することで，MARL 単体で学習する場合より，安全性とモデル性能を向上させることであった．そこで，エージェントとゴール間の距離変化に応じた報酬の導入と，ノード占有率の増加を抑えた安全制御を提案した．また，安全制御を分散的に行うための分散型安全制御の設計を試みた．

グリッドマップと非グリッドマップを用いて様々なエージェント数で実験を行った結果，提案手法は先行手法に対し，検証したすべての状況において，衝突なくより少ないコストでゴールに到達できることが示された．特に，非グリッドマップを用いた状況では，提案手法がより有効に働くことが確認された．また，先行手法と比べ学習効率が大幅に向上することが示された．この結果から，提案手法は学習効率の改善と，エージェントの移動効率およびそれに伴う性能の向上に貢献した．さらに，新たに設計された分散型安全制御は，従来の中央集権型安全制御と同等のゴール率およびコストを達成可能であり，中央集権方式が抱える課題を克服する可能性を示唆する結果が得られた．

今後の課題として，エージェント密度が高い環境においても対応可能な手法の考案が求められる．そのためには，エージェントの行動優先度を考慮した制御が必要であると考えられる．また，エージェントに対して連続的に配送タスクが割り当てられる設定での提案手法の有効性も検証することも重要な課題である．本研究の成果は，より安全かつ効率的なドローン配送の実現に寄与するものと期待される．

謝辞

本研究を行うにあたり，熱心な指導，助言をしてくださいました林冬惠准教授に厚く御礼申し上げます．そして，本研究を行うにあたり実験環境を提供してくださった京都大学の丁世堯助教に厚く御礼申し上げます．また，多くの有益な助言をくださいました林研究室の皆様方に心より感謝いたします．

参考文献

[1] Ingy Elsayed-Aly, Suda Bharadwaj, Christopher Amato, Rüdiger Ehlers, Ufuk Topcu, Lu Feng. Safe Multi-Agent Reinforcement Learning via Shielding. International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2021.

[2] 青山秀紀, 丁世堯, 林冬惠. ドローン配送計画最適化問題のための最短経路情報を利用したマルチエージェント強化学習. 人工知能学会全国大会論文集, Vol.JSAI2022, pp.3O3GS505–3O3GS505, 2022.

[3] Roni Stern, Nathan Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Eli Boyarski, Roman Bartak. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. arXiv preprint arXiv:1906.08291, 2019.

[4] Masahiro Kaji, Donghui Lin, Fumito Uwano. Safe Multi-agent Reinforcement Learning for Drone Routing Problems. The 25th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA), 2024.

[5] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, Shimon Whiteson. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent. Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research (PMLR),Vol.80, pp.4295-4304, 2018.

[6] Guillaume Sartoretti, Justin Kerr, Yunfei Shi, Glenn Wagner, T. K. Satish Kumar, Sven Koenig, Howie Choset. PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning. IEEE Robotics and Automation Letters, Vol.4, No.3, pp.2378–2385, 2019.

[7] Shiyao Ding, Hideki Aoyama, Donghui Lin. MARL4DRP: Benchmarking Cooperative Multi-agent Reinforcement Learning Algorithms for Drone Routing Problems. In Pacific Rim International Conference on Artificial Intelligence, pp. 459–465. Springer, 2023.

[8] Javier García, Fernando Fernández. A Comprehensive Survey on Safe Reinforcement Learning. Journal of Machine Learning Research, Vol.16, No.42, pp.1437-1480, 2015.

[9] Joshua Achiam, David Held, Aviv Tamar, Pieter Abbeel. Constrained Policy Optimization. arXiv preprint:1705.10528, 2017.

[10] Kaiqing Zhang, Zhuoran Yang, Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. arXiv preprint arXiv:1911.10635, 2019.

[11] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, Igor Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. arXiv preprint arXiv:1706.02275, 2020.

[12] Andrew Y. Ng, Daishi Harada, Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning, Vol.99, pp.278-287. Citeseer, 1999.

[13] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, Wojciech Zaremba. OpenAI Gym. arXiv preprint arXiv:1606.01540, 2016.

[14] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, Stefano V. Albrecht. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS), 2021.

[15] AAMAS-2024 Drone Routing Problems Challenge. https://drp-challenge.com/ (Accessed: Jan. 25, 2025).