J-062

# タスクベース効用関数を用いた経路交渉手法の実行時間の改善
# Improvement of Execution Time of Path Negotiation Method Using Task-Based Utility Function

平野 敦也† 打矢 隆弘† 内匠 逸†
Atsuya Hirano Takahiro Uchiya Ichi Takumi

## 1. はじめに

近年，自律移動ロボットの開発・普及が進んでおり，これらをエージェントとしてモデル化し制御する，マルチエージェントシステムが注目されている．具体例としては倉庫内自動搬送ロボットなどが挙げられる．しかしこのような例において膨大な搬送タスクを処理するためにエージェント数を増やすと，エージェント間で移動経路などの競合が発生する場合がある．競合が発生すると，かえって処理が非効率になるためエージェント間で協調が必要となる．

マルチエージェントシステムの研究では上記の問題は MAPD(Multiagent Pickup and Delivery)問題[1]として定式化されている．MAPD 問題では指定されたスタート地点からゴール地点まで移動する「タスク」が継続的に発生する．各エージェントは他との競合を回避しながら全タスクが終了するまでタスクを繰り返し処理する．各エージェントの経路を一度決定する問題は MAPF(Multiagent Path Finding)問題として定式化されているが，最適解の導出は一般に NP 困難であることが知られている．MAPD 問題はその繰り返しであるためより困難であると考えられる．

従来の MAPD 問題の研究では各エージェントは倉庫内自動搬送のように共通の目標を持つことが仮定されていた．しかし今後ロボットの利用が拡大すると，同じ空間に異なるオペレータによるロボットが動作することが考えられる．例えばあるオペレータによる案内ロボットと別のオペレータによる掃除ロボットが共通の空間で動作する可能性がある．このとき，単に競合を解消するだけでなく各オペレータの目標に応じた競合解消が必要となる．本研究では，先行研究で提案された経路交渉手法（図 1）について，交渉相手とするエージェントの削減により，実行時間の改善及び先行研究と同等のタスク成功率を目指す．

## 2. 関連研究

MAPD 問題の研究には特定のエージェントが全体を管理し全経路を決定する集中制御と，各エージェントが周囲の状況から自律的に行動を決定する分散制御に分けられる．集中制御の研究としては文献[2][3]がある．分散制御の研究としては文献[4][5]がある．例えば文献[4]では PIBT(Priority Inheritance Backtracking)という手法を提案している．これは各エージェントが毎ステップ優先度を計算し，近くのエージェントと通信をしながら優先度の高いエージェントから順に次の移動位置を決める．その際，優先度の低いエージェントを動かすことが可能である．次の位置に

†名古屋工業大学 大学院 工学研究科 工学専攻
Nagoya Institute of Technology，Graduate School of Engineering

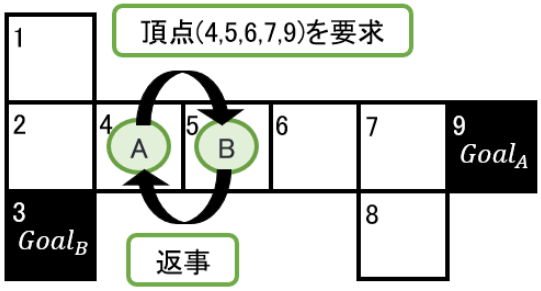


図 1　経路交渉イメージ

優先度の高いエージェントがいれば留まるか，優先順位を下げた次の位置に移動する．各エージェントが自律的に移動位置を決めるため，エージェント数の増加に対応が容易な手法である．しかしこの手法では，行き止まりを含むマップにおいてデッドロックが発生するため，実験環境に対し二重連結グラフでなければならないという制約がある．

## 3. 準備

### 3.1 問題設定

オペレータエージェントの集合を$K = \{o_1, o_2, \cdots, o_k\}$とする．各オペレータエージェントはそれぞれ一台のロボットエージェント$a_k$を持つ．各ロボット$a_k$はスタート位置$s_k$から動作を開始する．ロボット$a_k$はタスク$\tau_k$を遂行する．タスク$\tau_k$はゴール位置$g_k$と〆切$t_k$，報酬$r_k$から構成され，任意の時間にオペレータエージェントが自身のロボットに与える．オペレータは自身のロボットが〆切$t_k$までにゴール位置$g_k$に移動できれば報酬$r_k$を獲得できる．できなければ報酬は 0 となる．

マップは二次元空間に加え，垂直方向に時間の経過軸を持つグラフ$G(V)$として表される．時間は離散タイムステップ$t \in T$とし，${v^t}_k$はロボット$a_k$がタイムステップ$t$の時に占有する頂点$v \in V$であることを示す．1 タイムステップで，各ロボット$a_k$は現在位置${v^t}_k$から隣接頂点${v^{t+1}}_k$に辺$({v^t}_k, {v^{t+1}}_k)$を利用して移動するか，現在位置${v^t}_k$にとどまる$({v^t}_k = {v^{t+1}}_k)$．ロボット$a_k$の経路は各タイムステップで占有する頂点列とし，$P_k = ({v^0}_k, {v^1}_k, \ldots, {v^T}_k)$のように表される．なお，経路が衝突するとは頂点の衝突$({v^t}_k = {v^t}_{k'})$又は辺の衝突$(({v^t}_k = {v^{t+1}}_{k'}) \wedge ({v^{t+1}}_k = {v^t}_{k'}))$を指す．

### 3.2 タスク指向効用関数を用いた経路交渉手法

先行研究[6]のシステム及び経路交渉手法について説明する．この研究では予約ベースの経路割り当てによって経路の衝突を防止している．エリアマネージャがマップの全頂点の状態（”free”,”reserved”,”prohibited”）を記録するテーブ

ルを保持しており，”free”はどのロボットもその頂点を使用可能であること，”reserved”はあるロボットが予約しており他ロボットは使用不可能であること，”prohibited”はどのロボットも使用できない頂点を示す．各ロボットは探索した経路をエリアマネージャに送信し，予約する．この研究では FCFS メカニズムを採用しており，先に送信された有効な経路が承認され，該当頂点が予約される．先に予約された経路と衝突のある後から送信された経路は予約されず，送信したエージェントは別の経路を作成する必要がある．

次に先行研究での経路交渉手法について説明する．タスク付与されたロボットはエリアマネージャとの通信により経路の衝突を検知すると，全ロボットを停止させ順に以下の交渉を行う．まず，”free”頂点のみを用いて衝突のない経路$x^*_{org}$を探索する．次に交渉相手ロボットによって予約された頂点も使用して経路$x^*_{new}$を探索し，効用$(x^*_{org}, x^*_{new}, payment)$を計算する．計算した効用が正の値であるとき，交渉リクエストを交渉相手に送信し，負の値であるとき，他の交渉相手ロボットとの交渉に移る．交渉リクエストを受けたロボットはリクエストをしたロボットが要求する頂点を使用しない経路$x^*_{new}$を探索し，予約していた経路$x^*_{org}$も用いて効用を計算する．計算した効用が正の値になるように$payment$を設定する．つまり$payment$は要求された頂点をリクエストしたロボットに譲った時の効用減少分(コスト増加分)に設定される．次に，交渉をリクエストしたロボットは設定された$payment$を用いて再度効用を計算し，その値が正の値であれば交渉成立となりリクエストしたロボットが要求した頂点を予約可能となる．上述の交渉を全ロボットと順に行い，最も効用の高い結果となったロボットとの交渉結果を採用し，経路をエリアマネージャに送信・予約することで経路が確定する．

効用関数$U_{nego}$は以下の式で表される．

$$U_{nego}(V_{nego}) = u_{task}(x^*_{new}) - u_{task}(x^*_{org}) - p(V_{nego})$$
$$u_{task}(x) = r(x) - c(p)$$

$u_{task}$は各タスク遂行計画$x$を評価する関数である．$V_{nego}$は交渉対象となっている頂点列，$x^*_{new}$と$x^*_{org}$はそれぞれ交渉が成立した場合と不成立の場合のタスク遂行計画である．$p(V_{nego})$は交渉対象の頂点列$V_{nego}$に対する$payment$である．$r(x)$はタスクの報酬であり，〆切に間に合えば 100，間に合わなければ 0 である．$c(P)$は経路コストを表す．

## 4. 提案手法

先行研究のシステムでは，タスクが与えられたロボットが他の全ロボットを停止し順に交渉していた．しかしこれにより，マップサイズとエージェント数の拡大に伴い総交渉時間が増加した(図 2)．丸でプロットされたグラフが 20×20，四角形でプロットされたグラフが 10×10，三角形でプロットされたグラフが 5×5 の格子状マップでの結果である．横軸がエージェント数，縦軸が時間である．

本研究では上述の先行研究における問題点に対して，タスクが与えられたロボットが全ロボットと交渉するのではなく，自身の最短経路と衝突のある経路を予約しているロボットのみと交渉を行うことを提案する．

ここで，最短経路とはタスクが与えられたロボットがエリアマネージャと通信を行い，”free”，”reserved”状態にある頂点を用いて探索した経路のことである．衝突とは 3.1 節で述べたように，頂点の衝突$(v^t{}_k = v^t{}_{k'})$又は辺の衝突$((v^t{}_k = v^{t+1}{}_{k'}) \wedge (v^{t+1}{}_k = v^t{}_{k'}))$のことを指す．

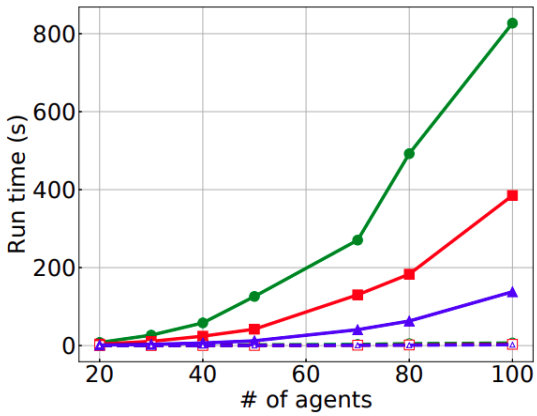


図 2　先行研究の実験結果：総交渉時間

つまり，探索した最短経路のうち，”reserved”頂点を予約しているロボットと交渉を行う．これにより交渉のために全ロボットが停止する時間が短縮されるため，実行時間が改善されると考えられる．効用関数を含めたロボット間で行う経路交渉のアルゴリズムは先行研究で提案されたものを用いる．

先行研究のシステムではタスクを付与されたロボットは他の全ロボットと順に交渉を行う．よって，例えば全ロボットが一度タスクを行うまでの交渉回数はロボット数を n とすると，$O(n^2)$となる．しかし提案手法では最短経路と衝突のあるロボットのみと交渉を行うため，全ロボットが一度タスクを行うまでの交渉回数は，$O(nl)$となる($l$は各ロボットの最短経路長)．つまり，$n > l$の時，提案手法が有効になると考えられる．

## 5. まとめ

本研究では，先行研究で提案された経路交渉手法において交渉を行うエージェント数を削減することを提案した．今後はシミュレーション実験を行い，総交渉時間の改善及び先行研究と同等のタスク成功率が達成されることを示す．

### 参考文献

[1] Ma, H., Li, J., Kumar, T. and Koenig, S.: “Lifelong multiagent path finding for online pickup and delivery tasks”, AAMAS, pp. 837–845, 2017.
[2] Sharon, G., Stern, R., Felner, A. and Sturtevant, N. R.: “Conflict-based search for optimal multi-agent pathfinding”, Artif.Intel, Vol. 219, pp. 40–66 , 2015.
[3] Luna, R. and Bekris, K. E.: “Push and swap: Fast cooperative pathfinding with completeness guarantees”, IJCAI, pp. 294–300 , 2011.
[4] Okumura, K., Machida, M., Defago, X. and Tamura, Y.: “Priority inheritance with backtracking for iterative multi-agent path finding”, IJCAI , 2019.
[5] Ma, H., Li, J., Kumar, T. and Koenig, S.: “ Lifelong multiagent path finding for online pickup and delivery tasks”, AAMAS, pp. 837–845 , 2017.
[6] Hiroaki Inotsume , Aayush Aggarwal, Ryota Higa, and Shinji Nakadai, “Path Negotiation for Self-interested Multirobot Vehicles in Shared Space”, IEEE/RSJ IROS, 2020.