2Q-05

時間拡張グラフ上のナンバーリンクパズルとしてのマルチエージェント経路発見 *

宮田 直人 [1†]　平山 勝敏 [2‡]　沖本 天太 [2§]

神戸大学海事科学部 [1]　神戸大学大学院海事科学研究科 [2]

1 はじめに

マルチエージェント経路発見（Multi-Agent Path Finding(MAPF)）とは，グラフ上に存在する複数のエージェントを，それぞれの初期ノードから目標ノードへ互いに「衝突」しないよう移動させる経路を発見する問題である [2]．MAPF の問題例は，形式的にはグラフ $G = (V, E)$，エージェント集合 $A = \{1, \ldots, n\}$，および，時刻 t におけるエージェントのグラフ G 上での位置を決める写像を $\alpha^{(t)} : A \to V$ として，エージェントの初期位置 $\alpha^{(0)}$ とエージェントの目標位置 α^{+} を明記して四つ組 $\Sigma = (G = (V, E), A, \alpha^{(0)}, \alpha^{+})$ で表される．MAPF の問題例 Σ に対する解とは，写像の系列 $S(\Sigma) = \{\alpha^{(0)}, \alpha^{(1)}, \ldots, \alpha^{(\mu)}\}$ で，以下の条件を満たすものである．

- 最後の要素 $\alpha^{(\mu)}$ が目標位置 α^{+} に一致している．
- 任意の要素 $\alpha^{(t)}$ および任意の 2 エージェント a と b に対して $\alpha^{(t)}(a) \neq \alpha^{(t)}(b)$ である．すなわち，2 つ以上のエージェントが同時に同じノードに位置することはない．
- 連続する要素 $\alpha^{(t)}$ と $\alpha^{(t+1)}$ の差は，各エージェントの「妥当な移動」を反映する．ここで「妥当な移動」とは，グラフ上で隣接する何れかのノードへ移動する，あるいは，現在のノードに留まるのどちらかである．

なお，MAPF の最適化問題では，エージェントの移動にコストを定義し，その何らかの集計値（指標）を最小化することを目指す．コストの集計方法としては，全エージェントの移動コストの総和（sum-of-costs）を最小化する，あるいは，全エージェントにおける移動コストの最大値 (makespan) を最小化するという 2 通りの考え方があるが，いずれの指標でも MAPF の最適化問題は NP 困難であることが示されている [1][3]．

本論文では，MAPF を時間拡張グラフ上のナンバーリンクパズルと見なし，制約最適化問題に基づく MAPF の定式化を試みる．また，既存の制約最適化アルゴリズムを用いた予備実験の結果を報告する．

2 準備

2.1 ナンバーリンクパズル

ナンバーリンクパズルとは，盤面のマス上に任意に配置された数字に対し，上下左右に隣接するマスの間に線を引き，同じ数字同士を交差しない線で繋ぐパズルである．ナンバーリンクパズルは，充足可能性判定問題 (SAT) に定式化して解くことが可能であり，その際の SAT 符号化法が文献 [4] に詳しく記述されている．本研究では，文献 [4] の符号化を前提とした制約モデルを用いて，次に示す時間拡張グラフ上でのナンバーリンクパズルとして MAPF を記述する．

2.2 時間拡張グラフ

時間拡張グラフとは，時刻 0 から任意の上限時刻 μ までの各時刻 t について MAPF におけるグラフ $G = (V, E)$ をコピーした系列 $\{G^{(0)}, \ldots, G^{(t)}, G^{(t+1)}, \ldots, G^{(\mu)}\}$ を作り，連続する $G^{(t)}$ と $G^{(t+1)}$ のノード間にはエージェントにとって「妥当な移動」と考えられるすべての箇所に有向エッジを配置した非巡回グラフである．すなわち，時刻 t であるエージェントがあるノード i に位置すると仮定して，そのエージェントが次の時刻 $t+1$ において移動可能なノードを j とした場合，i から j へ有向エッジ $e_{ij}^{(t)} \in M^{(t)}$ を配置する．以下，時間拡張グラフの各時刻のグラフ $G^{(t)}$ を「層」とよび，すべてのエージェントは同じ移動能力（1 単位時間に隣接する何れかのノードに移動するか現在のノードに留まる）をもつと仮定して，層間有向エッジ $M^{(0)}, M^{(1)}, \ldots, M^{(\mu-1)}$ は基本的に均一に配置されていると仮定する．

3 定式化

時間拡張グラフ上のナンバーリンクパズルとして MAPF を解くにあたり，まず，以下の変数およびその値域を導入する．

$x_i^{(t)} \in \{0, 1, 2, \ldots, n\}, \forall t \in \{0, 1, \ldots, \mu\}, \forall i \in V$: 時刻 t においてノード i に位置するエージェントの識別子を値としてとる．値が 0 の場合は，時刻 t においてノード i にエージェントが存在しないことを意味する．

$e_{ij}^{(t)} \in \{0, 1\}, \forall t \in \{0, 1, \ldots, \mu - 1\}, \forall i, j \in V$: 時刻 t のノード i において，あるエージェントが層間有向エッジ $e_{ij}^{(t)}$ に対応した移動を選択すれば 1，選択しなければ 0 を値とし

---

* Multi-Agent Path Finding as Numberlink Puzzles on Time Expansion Graphs

† Naoto Miyata, Faculty of Maritime Sciences, Kobe University

‡ Katsutoshi Hirayama, Graduate School of Maritime Sciences, Kobe University

§ Tenda Okimoto, Graduate School of Maritime Sciences, Kobe University

てとる．なお，$e_{ii}^{(t)} = 1$ のとき，時刻 t のノード i において，あるエージェントがノード i に留まることを意味する．この移動を特に「ステイ」とよぶ．

次に，制約条件として以下を導入する．

境界制約　時刻 0 の変数 $x_i^{(0)}$ の値をエージェントの初期位置 $\alpha^{(0)}$ をもとに固定する．また，時刻 μ の変数 $x_i^{(\mu)}$ の値をエージェントの目標位置 α^+ をもとに固定する．

障害物制約　エージェントが侵入できないノード i については，すべての時刻 t において変数 $x_i^{(t)}$ の値を 0 に固定する．

移動制約　時刻 $t(\neq 0)$ においてノード i にエージェントがいるならば（$x_i^{(t)} > 0$），そのノードへ入る層間有向エッジとそのノードから出る層間有向エッジがそれぞれちょうど 1 本ずつ選択される．

不在制約 1　時刻 $t(\neq 0, \mu)$ のノード i に入る層間有向エッジが 1 本も選択されていなければ，時刻 t においてノード i にはエージェントはいない（$x_i^{(t)} = 0$）．

不在制約 2　時刻 $t(\neq \mu)$ においてノード i にエージェントがいなければ（$x_i^{(t)} = 0$），そのノードから出る層間有向エッジは 1 本も選択されない．

同一性制約　層間有向エッジ $e_{ij}^{(t)}$ が選択されているならば（$e_{ij}^{(t)} = 1$），時刻 t でノード i にいるエージェントと時刻 $t+1$ でノード j にいるエージェントは同じである（$x_i^{(t)} = x_j^{(t+1)}$）．

交換移動禁止制約　任意の 2 エージェント間において 1 単位時間で互いの位置を交換するような移動を許さない．すなわち，任意の層間有向エッジ $e_{ij}^{(t)}$ について $e_{ij}^{(t)} + e_{ji}^{(t)} \leq 1$ とする．

また，最適化のための条件として，各エージェントの無駄な移動をできるだけ抑える目的でステイの数を最大化するよう設定した．

4　予備実験

提案アルゴリズムの性能を予備実験により評価する．実験で用いた MAPF の問題例とその結果を図 1 に示す．この問題例では，5×5 の盤面にエージェント 1 から 4 が存在し，それぞれの初期位置と目標位置が上部に示されている．盤面の黒で塗りつぶしたマスは障害物を示している．

実験環境は，Intel Core i7-3960X (3.30GHz, 12 CPUs)，メモリ 32GB，Windows 7 Professional 64bit であり，制約最適化ソルバーとしては Scala 上で動作する制約プログラミング言語 Copris [*1] を使用した．Copris では制約を SAT に変換し，SAT ソルバーとしてデフォルトで Sat4j が使用される．今回の実験ではデフォルトの設定で使用した．ソルバーにより計算された各エージェントの移動の様子を図 1 の下部に示す．この問題例に対して Copris により生成された SAT の問題例のサイズは，変数の数が 12065，節の数が 221441 であり，求解に 163 分を要した．現状では，この程度のサイズの問題例でも多大な時間を要しているため，今後はより高速なソルバーを利用する，よりコンパクトな制約記述方法を検討する等，何らかの工夫が必要である．

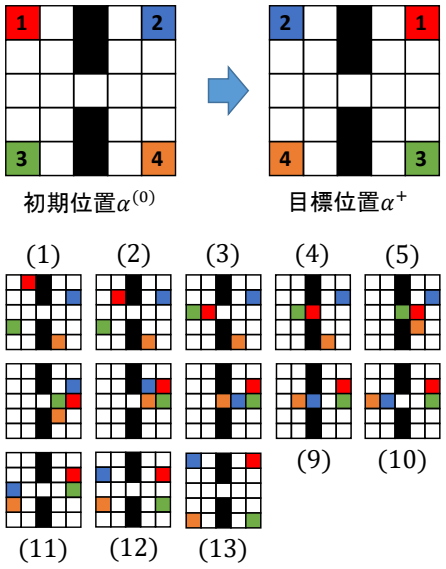

図 1　MAPF の問題例と実行結果

5　おわりに

本論文では，マルチエージェント経路発見（MAPF）問題を時間拡張グラフ上のナンバーリンクパズルと見なした上で，制約最適化問題に基づく MAPF の定式化を提案し，既存の制約最適化アルゴリズムで解くことを試みた．予備実験の結果によれば，エージェントによる妥当な振舞いが得られたと考えられる．今後の課題は，Sat4j 以外の高速 SAT ソルバーを求解に利用すること，01 整数計画問題など他の制約最適化問題による定式化を検討すること等が挙げられる．また，[2] 等にある MAPF に対する既存アルゴリズムとの詳細な比較も今後の研究課題である．

参考文献

[1] Surynek, P.: An Optimization Variant of Multi-Robot Path Planning is Intractable. *AAAI-2010*, pp. 1261–1263 (2010)

[2] Surynek, P., Felner, A., Stern, R., Boyarski, E.: Efficient SAT Approach to Multi-Agent Path Finding under the Sum of Costs. *ECAI-2016*, pp. 810–818 (2016)

[3] Yu, J., Lavalle, S. M.: Structure and Intractability of Optimal Multi-Robot Path Planning on Graphs. *AAAI-2013*, pp. 1443–1449 (2013)

[4] 田村直之，宋剛秀，番原睦則：　SAT とパズル. 情報処理, Vol. 57, No. 8, pp. 710–715 (2016)

[*1] http://bach.istc.kobe-u.ac.jp/copris/