

博士（工学）論文

エージェントの協調・非協調動作の  
干渉に基づく  
渋滞とその解消の解析

望月優加理

電気通信大学大学院 情報理工学研究科

機械知能システム学専攻

2022年9月

博士論文審査委員会

主任指導教員 主査 澤田 賢治 准教授

指導教員 委員 金子 修 教授

指導教員 委員 小木曾 公尚 准教授

委員 田中 基康 教授

委員 新 誠一 名誉教授

Copyright ©2022  
Yukari Mochizuki  
All Rights Reserved

# An analysis of traffic congestion and its resolution due to interference of cooperative and uncooperative behavior of agents

Yukari Mochizuki

## Abstract

This doctoral thesis evaluates traffic congestion due to agent control and its resolution in a multi-agent system with a mix of cooperative and non-cooperative agents, based on interference among the agents' behaviors. In existing studies, traffic congestion is defined in terms of agent density and flow rate. And cooperative functions have been proposed to predict, avoid, and mitigate the expansion of the traffic congestion. In the evaluation of cooperative functions, the effectiveness of traffic congestion mitigation has been evaluated on a simulation basis. In the transition process of introducing technologies for multi-agent systems, such as automation of existing facilities or renewal of facilities, a situation in which cooperative agents and non-cooperative agents are mixed together occurs.

Therefore, this doctoral thesis focuses on the resolution of the traffic congestion after it occurs, and proposes Demand-for graph that represents the behavior demands generated by the interference between cooperative and uncooperative behavior. We apply the proposed method to the problems of maze exploration in a two-dimensional plane, transportation in an automated warehouse, and lane change of an automated vehicle, and derive the size of the traffic congestion and its resolution conditions. The Demand-for-graph traffic congestion analysis for the three applications provides a theoretical evaluation of the impact on traffic congestion of the increase in the reduction speed of existing congestion and new interference caused by the introduction of a new cooperative function for the introduction of a new cooperative function.

By discussing the above, this doctoral thesis contributes to the diffusion and development of cooperative control technology for multi-agent systems. In terms of the diffusion of the technology, it realizes the evaluation of the impact of the cooperative behavior of agents on interference in an environment with a mixture of cooperative and non-cooperative agents, and enables the performance evaluation during the transitional period of the diffusion of the technology.

For the development of the technology, we provide design guidelines for the communication function and cooperative control of multi-agent systems from the viewpoint of resolving interference caused by uncoordinated behavior of agents.

## 概要

本博士論文は協調型・非協調型エージェントの混在するマルチエージェントシステムにおける、エージェントの制御による渋滞とその解消をエージェント間の動作干渉から評価するものである。マルチエージェントシステムはその適用範囲の広さから様々な分野で研究されている。中でも、エージェントの協調機能による渋滞緩和は、複数の研究分野で扱われる横断的研究課題である。既存研究ではエージェントの密度と流量に対して渋滞を定義し、渋滞の予見、回避、拡大抑止を目的とした協調機能が提案されている。既存研究では、シミュレーションベースで渋滞緩和の効果が評価されている。しかしながら、既存設備の自動化や設備更新などマルチエージェントシステムの技術導入とシステム更新は、その移行過程において協調型エージェントと非協調型エージェントが混在する状態が発生する。したがって、新規技術の導入においては、エージェントの協調動作と非協調動作の干渉が渋滞に与える影響の理論的評価が必要となる。このとき、協調・非協調動作の干渉に対する渋滞のモデル化が課題となる。

そこで、干渉の表現方法として新たに協調・非協調動作で生じる行動要求を表す Demand-for graph を提案する。Demand-for graph による干渉表現は、行動要求の達成によるグラフ構造の変化を通じてエージェントの動作による干渉関係変化の観測を可能にする。本博士論文では、マルチエージェントシステムの二次元平面における迷路探索、自動倉庫での搬送、自動運転車両の車線変更の問題で生じる渋滞に Demand-for graph による干渉解析を適用し、渋滞の規模とその解消条件を導出する。

迷路探索ではエージェント間の疑似移動による探索引き継ぎによるデッドロックの検知と回復に Demand-for graph による解析を適用する。これにより、協調型エージェント間でのデッドロックを起点とする渋滞に関して、渋滞の解消条件と通信範囲に対する渋滞の解消速度を導出した。渋滞解消速度と疑似移動の所要時間に基づき、デッドロック解消を目的とした疑似移動による渋滞緩和の条件を導出した。

自動倉庫での搬送問題では、エージェントの荷積みと荷降ろしで発生する渋滞とその自然解消の解析に Demand-for graph を適用する。Demand-for graph を用いた渋滞の発生・拡大・解消のモデル化を通じて、発生した渋滞の拡大速度と解消速度を導出した。これにより、非協調型エージェント間の渋滞に関して、渋滞の解消条件を搬送路の構造とエージェント密度の関係として導出した。また、渋滞幅の拡大による渋滞の再拡大とデッドロックに着目し、エージェントの停止時間に基づく渋滞幅の拡大予想に取り組んだ。

自動運転車両の車線変更問題では、車群全体の速度確保を目的とした自動運転車両の車線変更で発生する渋滞の解析に Demand-for graph を適用する。自動運転車両の車線変更で生じるエージェント間の車線変更、減速、加速の要求を Demand-for graph で表し、発生した渋滞の解消速度を導出した。これにより、協調型エージェントと非協調型エージェントが混在するシステムで生じる渋滞に関して、エージェントの協調動作による渋滞の発生とその解消条件を導出した。また、渋滞の解消条件に基づき非協調型エージェントを考慮した車線変更アルゴリズムとして、要求棄却機能を有する車線変更を提案した。

以上を論ずることで本博士論文はマルチエージェントシステムの協調制御技術普及と発展に貢献をなす。技術の普及においては協調型・非協調型エージェントの混在する環境下でのエージェントの協調動作が干渉に与える影響の評価を実現し、技術普及の過渡期における性能評価を可能にする。技術の発展においてはエージェントの非協調動作による干渉の解決可否の観点から、マルチエージェントシステムの通信機能と協調制御の設計指針を示す。

# 目次

1	序論	1
1.1	背景	1
1.2	概要	3
1.3	構成	8
2	既存の渋滞解析手法	9
2.1	渋滞	9
2.2	デッドロック	10
3	Demand-for graph による干渉解析	12
3.1	はじめに	12
3.2	Wait-for graph による渋滞規模評価	12
3.3	Demand-for graph による渋滞解消速度の評価	13
3.4	Demand-for graph による渋滞表現の具体例	15
3.4.1	2台のエージェントのデッドロック	15
3.4.2	3台のエージェントのデッドロックと渋滞	16
4	グラフ探索	19
4.1	ASEP	19
4.2	ASEP の直列ネットワーク	19
4.3	複数レーン ASEP	20
4.4	二次元平面	21
4.5	エージェントの移動の数理モデル	21
4.6	Demand-for graph のダイナミクス	23
5	迷路探索	25
5.1	探索行動の数理モデル	26
5.2	疑似移動によるデッドロック解決	27
5.3	Demand-for graph による渋滞解析	30
5.4	渋滞解消速度	32
5.5	数値実験	33
6	搬送問題	36
6.1	荷積み・荷降ろしと移動要求	36
6.2	渋滞の Demand-for graph 表現	38
6.3	渋滞の解消条件	40
6.3.1	ASEP ネットワーク	41
6.3.2	二次元平面	43
6.4	協調機能による渋滞緩和への応用	44
6.4.1	ASEP ネットワーク	45

6.4.2	二次元平面	46
6.5	数値実験	46
6.5.1	Beth Network 探索	46
6.5.2	ASEP ネットワーク探索	49
6.5.3	二次元平面探索	51
7	<b>車線変更と車群形成</b>	53
7.1	自動運転車両の車線変更と車群形成	53
7.2	渋滞の Demand-for graph 表現	54
7.3	渋滞の解消速度と解消条件	59
7.4	非協調エージェントを考慮した協調動作	60
7.5	数値実験	61
8	<b>まとめ</b>	64
8.1	結論	64
8.2	今後の発展性	64
	<b>参考文献</b>	66

# 1 序論

## 1.1 背景

マルチエージェントシステムは個々のエージェントが周辺環境と他のエージェントの状態に応じて意思決定し、エージェントとシステム全体の目標を達成する。マルチエージェントシステムはその適用範囲の広さから様々な分野で研究されている。分散コンピューティング、ロボティクスの分野では、ロボットの編隊走行などのロボット間での協調動作が研究されている。ソフトウェア科学の分野では複数のソフトウェアエージェントのネットワーク探索による、大規模システムの管理、情報収集、探索の効率化が課題となる。また、人工生命の分野では強化学習の手法として複数のエージェントが、環境との相互作用に基づき独立に学習するマルチエージェント強化学習が研究されている。各領域における主要研究課題は、エージェントの活動空間を分割した区画と区画間の接続関係のグラフ表現によってグラフ探索問題に帰着させることができる。

ロボットなどの複数の移動物体で構成されるマルチエージェントシステムの代表的な適用先として、迷路探索などの空間探索 [1-19]、編隊制御などのベンチマーク問題 [21-24]、自動倉庫における荷物の搬送 [25-45]、自動運転車の車群形成 [56, 58-71] などの複数の移動物体による協調動作が挙げられる。これらの協調動作は非対称単純排他過程 (Asymmetric Simple Exclusion Process, 以降 ASEP) とそのネットワーク [46-51]、およびセルオートマトン [52-54] に基づくエージェントの動作空間のグラフ化によって、マルチエージェントシステムのグラフ探索問題として表現できる。

マルチエージェントシステムのグラフ探索の代表的な適用先の応用研究の、研究課題とその解決手法に基づく分類分けを表 1 に示す。

表 1: マルチエージェントシステムにおける研究課題

	研究課題		
	最適化	合意形成	安全確保
経路計画	A-1 [1-8, 10-13, 16-19] B [26, 32, 35, 36, 38, 45] C [56, 61, 66]		A-1 [9, 14, 15]
スケジューリング	B [28, 33, 34, 37, 40-44]		
軌道制御	C [58, 62]		A-1 [9, 14, 15] A-2 [21-24] B [45] C [64, 65, 67, 69]
システム構築	B [27, 29, 30]		
ネットワーク構築		A-2 [21-24] C [59, 60, 62]	

表 1 の A-1, A-2 はそれぞれ、迷路探索と編隊制御を扱う研究である。B, C はそれぞれ、自動倉庫における荷物の搬送と自動運転を扱う研究である。マルチエージェントシステムの応用研究における研究課題は、大きく最適化、協調動作のための合意形成、エージェントや障害物との衝突回避による安全確保の三つに大別される。また、課題解決に用いられるマルチエージェントシステムの要素技術として、



経路計画, スケジューリング, 軌道計画と制御, システム設計, 通信ネットワーク構築が挙げられる。

迷路探索の分野ではセンサの被覆範囲の最大化 [5, 6, 10, 12, 13] や, エージェントの移動経路長の最小化 [1-4, 7, 8] のためのエージェントの協調動作が研究されている。文献 [5, 6, 10, 12, 13] はエージェントの密集によるセンシング範囲の干渉を考慮した, 個々のエージェントの経路計画手法を提案している。このとき, エージェント間の衝突とデッドロックの回避 [9, 14-19] が課題となる。衝突回避はエージェントの安全確保, デッドロックの回避は渋滞回避を目的とした最適化に該当する。文献 [9, 14-17] はエージェント間の衝突回避と衝突回避の結果生じる膠着状態であるデッドロックの回避を考慮した経路計画手法を提案している。これに対して, 先行研究 [17-19] では, エージェント間での協調動作を用いたデッドロック解決手法を提案している。

自動倉庫内の搬送問題では適用先の例として, 直方体の上面をエージェントが移動する多層式 [25-29] や, コンベアや棚の間などの決められた通路をエージェントが走行する構造 [31-44] が提案されている。最適化の観点ではエージェントの流量の最大化 [46-48] とエージェントの移動経路長の最小化の最適化 [42] が研究されている。文献 [46-48] はエージェントの非協調動作に基づき, 搬送路のレイアウトに対するエージェント密度と流量の関係を解析している。これにより, 流量の最大化を目的とした搬送路とエージェント数の設計を実現している。搬送問題の分野における渋滞回避の課題として, エージェントの協調動作によるデッドロックの解決 [40-42, 44] が挙げられる。デッドロックの解決手法にはデッドロックの予見と回避で構成されるデッドロック回避 [42, 44] と, 発生したデッドロックの検知と解消で構成されるデッドロック回復 [41] が存在する。文献 [42, 44] はデッドロック回避のためのスケジューリング手法を提案している。また, 文献 [40] はデッドロック回避のための経路計画手法を提案している。デッドロック回復に関しては, 文献 [41] が人力でのエージェントの退避を用いたデッドロック回復の計算複雑さを解析している。

自動運転の分野ではエージェント間の協調動作による渋滞緩和 [56, 58-67, 69] が課題となる。文献 [63] では自動運転による交通円滑化の施策として, 協調型自動運転による車線変更を用いた単路部の交通円滑化を挙げている。文献 [61, 68] は協調型自動運転車両による車線変更の具体的な手法を提案している。文献 [56, 61, 66] はどちらも, 渋滞緩和のための経路計画の研究である。文献 [56, 57, 59, 60] は渋滞緩和を目的とした, 自動運転車両の車速, 車間距離, 軌道の制御手法を提案している。自動運転の分野では歩行者や手動運転車両との衝突回避など, 安全性の保証 [64, 65, 67, 69] が課題となる。

表 1 に示す通り, システムの全体最適化, 部分最適化は複数の領域にまたがる横断的研究課題であると言える。最適化の対象の一つに, 渋滞緩和によるシステムの稼働率 [11, 18, 19, 26, 32, 35, 36, 38, 40-45, 56, 58, 61, 62, 66] が挙げられる。渋滞緩和における課題として, 稼働率低下の原因となる渋滞 [56, 58, 61, 62, 66] とデッドロック [18, 19, 40-44] の解決が挙げられる。既存研究における渋滞緩和の取り組みは渋滞の予見と協調動作による渋滞の回避や緩和 [39-44, 55] を主体としている。このとき, 渋滞の予見には文献 [46-55] などの非協調動作に基づく, 渋滞モデルが利用される。一方で, 非協調型のエージェントと協調型のエージェントが混在する環境では, 協調動作による渋滞の発生を考慮する必要がある。文献 [56, 64, 65] では自動運転車両と手動運転車両の混在流を対象に渋滞緩和を研究しているが, その評価はシミュレーションに基づく評価であり理論的解析には至っていない。協調動作による渋滞緩和の理論的評価においては, エージェントの協調動作に対する渋滞の発生・拡大・解消のモデル化が必要となる。

マルチエージェントシステムの応用研究における共通課題として, 協調動作を実現するための局所的な通信によるエージェント間での情報共有手法の提案が挙げられる。

文献 [21-24] ではマルチエージェントシステムの通信ネットワーク構築のベンチマーク問題として,

ロボットのフォーメーション形成のための合意問題が研究されている。

文献 [56,59] は自動運転における通信の研究として、V2V 通信の制約下で実現可能な渋滞緩和のための自動運転車の車線変更と車線変更のための車群形成アルゴリズムを提案している。また、実環境でのマルチエージェントシステムの運用では、ヘテロジニアスなエージェント群による協調や人間との協働が想定される。文献 [7] では、トラックと UAV で構成されるヘテロジニアスな探索システムの荷物の搬送問題におけるタスク割り当て手法を提案している。文献 [56,59] では人間が運転する手動運転車両が混在する車群における協調的走行速度制御手法を提案している。日本においては 2020 年 3 月から第 5 世代移動通信システム (5G) の商用展開が開始された。文献 [77] によると 5G は超高速通信, 超低遅延かつ高信頼性を有する通信, 多数同時接続通信の三つのユースケースに適合できるように設計されている。また, 5G では企業や自治体自らが 5G の通信技術を用いて自営通信エリアを構築するローカル 5G が制度化されている。このように, 協調動作による課題解決においては, 協調動作を実現するためのエージェント間の通信ネットワークの構築とそれを実現する通信手段の選択が課題となる。

## 1.2 概要

本博士論文はマルチエージェントシステムの横断的研究課題の一つである協調機能による渋滞緩和に対して, 協調機能を構成するエージェントの動作が渋滞の発生とその解決に与える影響を評価するものである。本博士論文における協調機能の導入による渋滞緩和効果の評価は

- a. 協調機能に基づく自然渋滞の解消の評価
- b. 自然発生した渋滞の協調機能を介さない自然解消との比較
- c. 協調機能によって新たに生じる渋滞とその解消の評価

の三段階で構成される。評価の各段階に対して渋滞の解消条件の導出と, 渋滞緩和のための協調機能実装におけるエージェント間の通信機能の制約導出に取り組む。

図 1 に本博士論文で扱うマルチエージェントシステムの概念図を示す。エージェントは移動・通信の機能を持つ, 自律移動ロボットである。エージェントは通信により, 情報共有を行う。また, 周辺環境と共有情報に基づき, システム全体, または自身の目的達成のために協調機能を行使する。

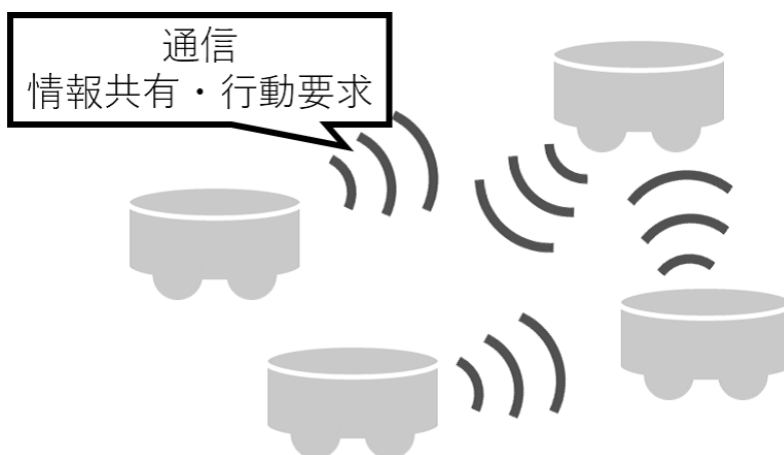


図 1: マルチエージェントシステム

本博士論文では, マルチエージェントシステムの協調機能による渋滞の発生と解消への影響評価のた

めに、新たに協調動作、非協調動作の概念を導入する。協調動作はエージェントがシステム全体、または他のエージェントの目的達成のために行う動作を意味する。このとき、エージェントの協調動作はシステム全体の協調機能を構成する要素技術の一つであり、動作の実行に他のエージェントから通信で取得した情報の利用を要するものとする。同様に、エージェントが単体で行使する、通信を要さない動作を非協調動作と表す。また、エージェント間の協調・非協調動作の干渉の表現手法として、行動要求を導入する。エージェントは自身の目的達成に他のエージェントの協調・非協調動作を要する場合に、該当の動作に対する行動要求を送信する。行動要求は渋滞とそれによる稼働率低下の原因となるエージェントの動作の干渉を表すとともに、渋滞の解消に必要となる他のエージェントの動作を表す。

既存研究における渋滞解析手法の多くは協調動作に基づき渋滞の発生を未然に防ぐ [7, 40, 42–44, 78], あるいは渋滞の拡大を防ぐこと [56, 57, 59–61, 63] を主体としている。一方で、既存設備の自動化や設備更新などマルチエージェントシステムの技術導入とシステム更新では、その移行過程において文献 [56, 59, 60, 64, 64, 65] のように協調型エージェントと非協調型エージェントが混在する状態が発生する。協調型エージェントと非協調型エージェントが混在する状態では、エージェントの協調動作と非協調動作の干渉により避けられない渋滞の発生が想定される。また、協調型エージェントと非協調型エージェントが混在する状況下での渋滞緩和に関する先行研究の多くは、シミュレーションによる評価が中心であり理論的解析を行ったものは少ない。そこで、本博士論文ではエージェントの協調・非協調動作の干渉の観点から、渋滞の発生と渋滞の拡大・解消速度の評価に取り組む。これにより、協調型エージェントと非協調型エージェントの混在環境における、エージェントの協調行動による渋滞緩和の性能評価を実現する。更に、協調動作による渋滞緩和手法の実用化に向けた課題として、協調動作実現に対するエージェント間の通信ネットワークの要件を議論する。

エージェントの渋滞規模の解析においては、渋滞の原因となるエージェントの協調・非協調動作の特定が課題となる。この課題の解決に、エージェントによる資源占有と資源開放の待ち関係を表す Wait-for graph [72–76, 80, 81] を用いる。Wait-for graph は文献 [42, 43] のデッドロック回避手法における、デッドロックの予見に用いられる。更に、本博士論文ではエージェントの協調・非協調動作による渋滞の発生・拡大・解消のモデル化のために、新たな渋滞表現手法として Wait-for graph に協調・非協調動作に対するエージェント間の行動要求の概念を導入した Demand-for graph を提案する。行動要求の導入により、従来研究 [72–76, 80, 81] のデッドロック解決手法であるデッドロックの予見と回避に加えて、発生したデッドロックの検知と膠着状態からの回復に基づくデッドロック解決を実現する。行動要求によってエージェントの協調・非協調動作の干渉を表す。本博士論文では Demand-for graph を用いた、エージェントの協調・非協調動作の干渉に対する渋滞規模の定義を示す。また、エージェントの協調・非協調動作に対する Demand-for graph の形状変化に基づき、エージェントの動作による渋滞の発生・拡大・解消の速度を導出する。

本博士論文では、マルチエージェントシステムの迷路探索、自動倉庫内での搬送問題、車線変更と車群形成の問題に対して、Demand-for graph による渋滞、デッドロックの解析を適用する。全適用先での共通課題は渋滞の拡大速度と解消速度の関係に基づく、渋滞の解消条件導出である。各探索問題の渋滞解析の適用を通じて、非協調動作に起因する渋滞の協調動作による解消、非協調動作に起因する渋滞の非協調動作による自然解消、協調動作に起因する渋滞の発生とその解消を論じる。

はじめに、マルチエージェントシステムのグラフ探索のベンチマーク問題である迷路の協調探索のデッドロック解決を通じて、エージェントの非協調動作に起因する渋滞の協調動作による解消について論じる。これは、協調機能による渋滞緩和評価における a) 協調機能に基づく自然渋滞の解消の評価に該当する。迷路探索におけるデッドロックは、エージェントの非協調的な移動経路の干渉によって生

じる。本博士論文ではエージェントの協調動作として Wait-for graph に基づくデッドロックの検知と、検知したデッドロックからの回復のための探索引き継ぎと疑似移動を提案する。疑似移動は通信によりエージェントの役割と区画移動後の予想状態を共有し、共有した通信情報に基づき役割の交換と予想される状態の再現による探索引き継ぎを行う区画移動を伴わない協調動作である。探索引き継ぎに対して発生する要求の Demand-for graph 表現と、要求達成による Demand-for graph の形状変化の解析から探索引き継ぎにに基づくデッドロックからの回復条件を導出する。また、デッドロックの検知と探索引き継ぎの所要時間により、エージェントの通信保証範囲と通信手段に対するデッドロックの解消速度を評価する。これにより、既存のデッドロック回避手法で予見・回避が不可能なデッドロックの解消に対して、探索引き継ぎによるデッドロックからの回復が有効であることを示す。また、迷路探索問題における固有課題として、探索引き継ぎ動作を考慮した通信範囲に対する渋滞解消速度の導出に取り組む。数値実験により、探索引き継ぎ動作を考慮した渋滞解消速度を検証する。固有課題への取り組みを通じて、エージェント間の通信可能距離、1 時刻あたりの通信可能情報量に基づき協調機能実現のための通信手段の選択指針を示す。

次に、自動倉庫内での搬送問題では、ホモニアスなエージェント群による ASEP の直列ネットワーク探索での荷積み・荷降ろしを起点とする渋滞の解析を通じて、エージェントの非協調動作による渋滞の自然解消について論じる。これは、協調機能による渋滞緩和評価における b) 自然発生した渋滞の協調機能を介さない自然解消との比較に該当する。迷路探索問題で提案する疑似移動を用いたデッドロックの解決は、搬送問題の荷積み・荷降ろしで生じたデッドロックの解消にも有効である。図 2 と図 3 に ASEP の直列ネットワークと二次元平面での搬送問題の概念図を示す。図 2 の例では、エージェントは商品棚の間の通路を走行する。エージェントの荷積み・荷降ろしは通路上で行われる。図 3 の例では、エージェントは商品棚の上面を走行する。

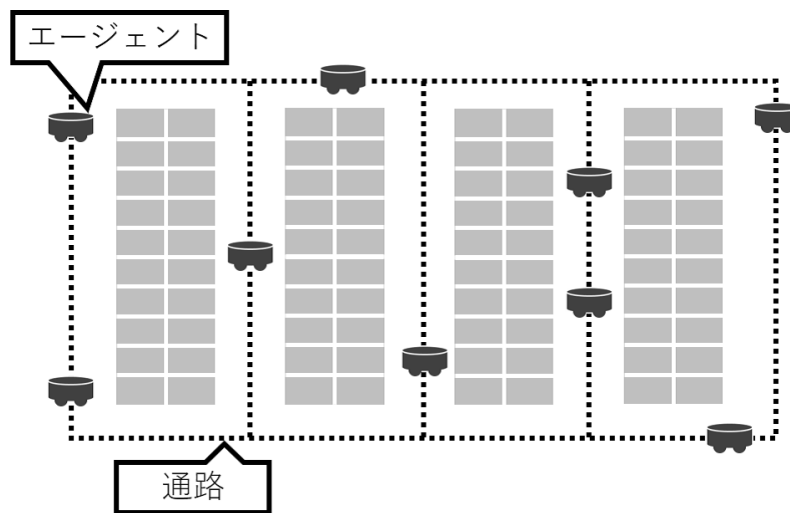


図 2: ASEP の直列ネットワークの搬送問題

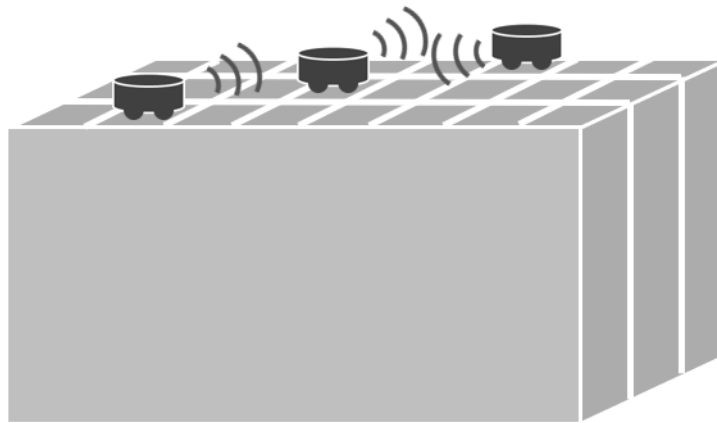


図 3: 二次元平面の搬送問題

搬送問題では非協調動作である荷積み・荷降ろしによる一時停止で、渋滞が発生する。本博士論文ではエージェントの荷積み・荷降ろしを識別する情報として、新たにインシデント ID を導入する。インシデント ID は渋滞の原因となる非協調動作と一対一で対応する渋滞の識別子である。本博士論文ではインシデント ID の設定により、流量と密度の関係に基づく従来の渋滞解析手法では評価が困難であった起点となるエージェントの動作別の渋滞の拡大・解消の評価を実現する。インシデント ID とエージェントの移動経路干渉の Demand-for graph 表現により、荷積み・荷降ろしによる渋滞の発生とエージェントの移動による渋滞の拡大・解消を評価する。また、渋滞の拡大速度と解消速度の関係から、エージェントの移動速度に対する渋滞の解消速度を導出する。渋滞の解消速度を導出したことにより、エージェントの密度と搬送路の構造的な特徴に対する渋滞の解消条件を明らかにする。これにより、エージェントの非協調動作による渋滞の解消、緩和を定量的に評価することが可能になる。また、搬送問題における固有課題として、渋滞解消条件の協調的経路計画への応用とそれを実現するための通信手段を提案する。固有課題への取り組みでは、エージェントの密度と予想流量に基づく協調的な経路計画手法を提案する。更に、提案する経路計画手法実現に向けた通信ネットワークの設計として、情報共有における通信距離制約と共有情報の情報量に基づく通信負荷の評価を示す。

車線変更と車群形成ではヘテロジニアスな自動運転車両と手動運転車両の混在する車群による複数レーン ASEP 探索の渋滞の解析を通じて、協調動作を起点とする渋滞の発生とその解消について論じる。これは、協調機能による渋滞緩和評価における c) 協調機能によって新たに生じる渋滞とその解消の評価に該当する。協調動作を起点とする渋滞の拡大・解消の観測には、搬送問題搬送問題で提案したインシデント ID を用いる。図 4 に自動運転車両の車線変更、車群形成の概念図を示す。

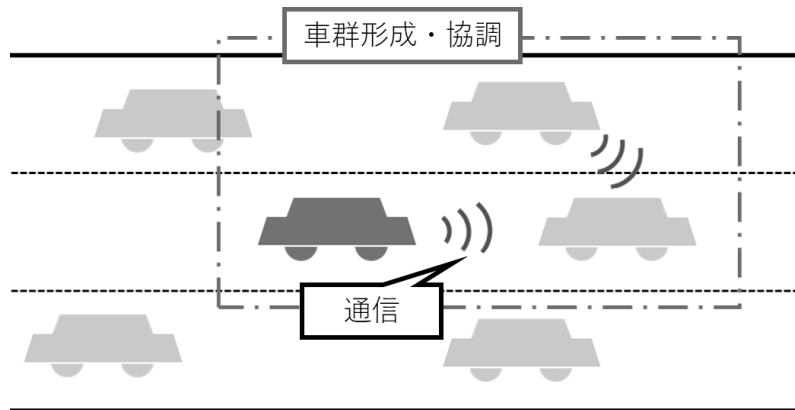


図 4: 自動運転車両の車線変更と車群形成

エージェントは、協調動作として文献 [61] の提案する後続車両を加速させるための車線変更を行う。協調動作としての車線変更は動作の主体となるエージェントの車速上昇ではなく、後続車両の車速確保のための車線変更を要求する通信情報に基づく動作であるという点で利他的と表現される。文献 [61] では利他的な車線変更の導入による、交通流全体の車速の向上の効果をシミュレーションにより検証している。車線変更に際して、エージェントは図 4 に示す通りに車群を形成する。車群内のエージェントでの通信により、車線変更とそのため車間距離の確保を達成する。本博士論文では車線変更の際に車群後方で発生する一時的な減速に着目し、協調動作による渋滞の発生とその解消の評価に取り組む。そこで、Demand-for graph による渋滞解析に、新たに車群後方の減速状態に対する加速要求を導入する。加速要求は利他的な車線変更のために減速中のエージェントとの区画移動の干渉解消に対する行動要求である。加速要求の達成に要する前方エージェントの区画移動は、実施に通信情報を要さない非協調動作である。加速要求と搬送問題で提案したインシデント ID の利用により、協調動作である車線変更を起点とする渋滞の発生とその非協調動作による解消の評価を実現する。本博士論文では協調動作としての車線変更で生じる車線変更・減速・加速要求を Demand-for graph で表す。後続車両の減速とそれによって生じる加速要求の関係から、車線変更に起因する渋滞の規模を評価する。要求達成による Demand-for graph の形状変化の解析から、車線変更による渋滞の発生と発生する渋滞の解消条件を導出する。これにより、エージェントの協調動作による非協調型エージェントの流量低下と、協調動作による交通円滑化の効果の比較を実現する。また、車線変更と車群形成における固有課題として、非協調型エージェントの動作を考慮した車線変更のアルゴリズムを提案する。数値実験により協調型・非協調型エージェントの混在流における、非協調型エージェントの割合と交通流量の関係を示す。更に、提案する車群形成のアルゴリズムに対して、既存の利他的車線変更のアルゴリズムとの比較で用いられる通信手段の通信距離と同時通信台数を対象に実現可能性を評価する。

共通課題への取り組みを通じて、エージェントの協調動作による既存渋滞解消効率上昇と新規渋滞の発生とその解消を総合した、渋滞緩和効果の評価を実現する。また、固有課題への取り組みを通じて、各適用先におけるマルチエージェントシステムの実用化に向けた通信手段通信可能距離、同時通信台数、通信相手の選択の選択指針を示す。これにより、搬送、交通分野への自動化技術の導入による効率改善の効果を示すとともに、自動化技術の実用化に向けた通信技術の課題を明らかにする。

### 1.3 構成

2章では渋滞解析の既存研究としてエージェントの密度と交通流量に基づく渋滞規模の評価と、エージェントの動作干渉に基づくデッドロック検知・予見手法を示す。3章ではデッドロック検知・予見に用いられる Wait-for graph の、エージェントの動作干渉に基づく渋滞規模評価への応用を示す。また、エージェントの協調・非協調動作に基づく渋滞規模の変化を評価可能な渋滞表現手法として、Wait-for graph にエージェントの協調・非協調動作に対する行動要求の概念を導入した Demand-for graph を提案する。4章では ASEP, ASEP の直列ネットワーク, 複数レーン ASEP, 二次元平面を対象に, マルチエージェントシステムのグラフ探索の数理モデルを示す。5章では迷路探索問題での疑似移動によるデッドロック回復に対する, Demand-for graph 表現の適用を示す。6章では ASEP の直列ネットワークと二次元平面の搬送問題での荷積み・荷降ろしに起因する渋滞の, 非協調動作による自然解消に対する Demand-for graph 表現の適用を示す。7章では自動運転車両の利他的車線変更に起因する渋滞とその解消に対する Demand-for graph 表現の適用を示す。最後に8章に本博士論文の結論と今後の発展性を示す。

---

表記1: ノードの有限集合  $V$  とエッジの有限集合  $E$  で構成される有向グラフを  $G = (V, E)$  と表す。有向グラフ  $G$  についてノード  $v_i \in V$  を始点,  $v_j \in V$  を終点に持つエッジを  $(v_i, v_j)$  と表す。

表記2: 有向グラフ  $G = (V, E)$  について, ノード  $v \in V$  の出次数を  $\text{outdeg}(G, v)$  と表す。同様に, ノード  $v$  の入次数を  $\text{indeg}(G, v)$  と表す。また,  $\text{outdeg}(G, v) = 0$  と  $\text{indeg}(G, v) = 0$  を満たすノード  $v$  を, それぞれ出端点と入端点と表す。

表記3: 有向グラフ  $G = (V, E)$  について2つのノード間の距離を, ノード間を結ぶ最短経路のエッジ数とする。  $v_i \in V, v_j \in V$  間の距離を  $\Delta(G, v_i, v_j)$  と表す。  $v_i \in V$  と  $v_j \in V$  を結ぶ経路が存在しない場合,  $\Delta(G, v_i, v_j) = \infty$  とする。

表記4: 命題  $a$  が真であることが, 命題  $b$  が真であることの十分条件であることを

$$a \rightarrow b \iff \neg a \vee b$$

を表す。

表記5:  $\{0, 1\}^n$  は要素が0または1の  $n$  次の列ベクトルとする。  $\{0, 1\}^{n \times m}$  は要素が0または1の  $n \times m$  行列とする。  $n$  字の単位行列を  $I_n$  と表記する。全ての要素が0である  $n$  次の列ベクトルと全ての要素が1である  $n$  次の列ベクトルを, それぞれ  $\mathbf{0}_n, \mathbf{1}_n$  と表す。また, ある行列  $M$  の転置行列を  $M^T$  と表す。

## 2 既存の渋滞解析手法

本章では、本博士論文で取り扱う渋滞とデッドロックに関して、既存の解析手法を示す。

### 2.1 渋滞

本節では、既存の渋滞解析手法として、マルチエージェントシステムの渋滞解析の手法について述べる。既存研究の渋滞解析手法の多くは、エージェントの密度と流量に関係から渋滞の規模を評価している [46–54, 56, 58–62, 70, 71]。

交通工学の分野において、渋滞は密度と交通流量の関係を示す基本図で表現される。交通工学において、密度は道路の単位長さあたりに存在する車の台数を、交通流量は道路のある一点を単位時間あたりに通過する車の台数を意味する。文献 [54, 55] では、渋滞を基本図において密度の上昇に対して交通流量が減少する状態と表現している。渋滞はある道路区間に関してその道路区間を通過しようとする単位時間当たりの車の台数（交通需要）が、対象区間の交通容量を超過する場合に発生する。このとき、渋滞の発生要因は交通集中による交通需要の増加と、事故や交通規制、路面状況による交通容量の減少の二つに大別される。

文献 [55] では単路の渋滞解析手法として、OR 分野で使用される待ち行列と累積図による渋滞解析を紹介している。文献 [55] で紹介される手法では、待ち行列の長さがそのまま渋滞の長さとしてされる。累積図はある道路断面を通過する車の累積台数の時間推移である累積曲線で構成される。累積図では渋滞解析区間の上流端における累積曲線を到着曲線、下流端における累積曲線を出発曲線と表す。累積図における到着曲線と出発曲線の垂直方向の差が待ち行列の長さに該当する。

ネットワーク交通流においては交通需要が車の移動経路選択によって変化するため、この場合、累積図作成のための観測点（渋滞の上流端と下流端）の設定ができないため、待ち行列長さによる渋滞解析は困難である。そのため、文献 [55] ではネットワーク交通流における渋滞解析手法として、ネットワーク内での単位時間あたりに移動を完了する車の数を指標とする渋滞解析を挙げている。

文献 [46–51] では渋滞解析のための数理モデルとして、非対称単純排他過程（Asymmetric Simple Exclusion Process, 以降 ASEP）を用いている。ASEP 上の移動において、エージェントは自身の前方の区画が空いている場合に移動確率  $p$  ( $0 \leq p \leq 1$ ) で移動する。このとき、ASEP の交通容量は移動確率  $p$  によって求まる。

文献 [46, 54] は、単路の ASEP の移動確率  $p$  に対する交通容量を導出している。ASEP 上のエージェント密度  $\rho$  ( $0 \leq \rho \leq 1$ ) が交通容量を上回る場合、ASEP 上で渋滞が発生する。文献 [48] では ASEP の直列ネットワークに関して、ネットワーク構造と移動確率  $p$  に対する交通容量を導出している。

エージェントの協調動作による渋滞緩和 [40–42, 44, 46–48, 56, 57, 59–61, 63] は、マルチエージェントシステムの横断的研究課題の一つである。しかしながら、文献 [46, 48, 54, 54, 55] に示す既存の渋滞解析手法は観測区間における交通需要と交通容量の関係による解析を主としており、エージェントの協調・非協調動作が渋滞の原因である交通需要増加と交通容量減少に与える影響の理論的解析には至っていない。そこで、本博士論文ではエージェントの協調・非協調動作による渋滞の発生、拡大、解消の評価に取り組む。このとき、エージェントの協調・非協調動作を起点とした、渋滞規模の時間変化の観測が課題となる。



## 2.2 デッドロック

デッドロックは複数のプロセスがお互いに待ち状態に陥ったことで発生するシステムの膠着状態である。本博士論文ではプロセスの待ち状態を渋滞としてとらえ、デッドロックを解消不能な渋滞の一形態として扱う。プロセス間の待ち状態は、プロセスが占有している共有資源とプロセスの資源割り当て要求によって発生する。

以降では、 $I$  個のプロセスの集合  $P$  と共有資源の集合  $S$  を例に、デッドロックを説明する。ここで、 $P$  における  $i$  番目 ( $i \in \{1, 2, \dots, I\}$ ) のプロセスを  $P_i$  と表記する。また、 $S$  における  $j$  番目 ( $j \in \{1, 2, \dots, |S|\}$ ) のプロセスを  $S_j$  と表記する。

プロセスによる共有資源の占有と、資源割り当て要求、プロセス間の待ち状態について述べる。プロセス  $P_i \in P$  が占有する共有資源の集合を  $O(P_i) \subseteq S$  で表す。  $S_j \in O(P_i)$  は  $P_i$  に共有資源  $S_j$  が割り当てられていることを意味する。プロセス  $P_i \in P$  による資源割り当て要求を  $R(P_i) \in S$  で表す。  $S_j = R(P_i)$  はプロセス  $P_i$  による、タスク開始のための共有資源  $S_j$  の割り当て要求を意味する。このとき、プロセス  $P_i$  の資源割り当て要求  $R(P_i)$  がプロセス  $P_j$  の占有資源  $O(P_j)$  に対して

$$R(P_i) \in O(P_j) \quad (1)$$

を満たすとき、 $P_i$  は  $P_j$  が共有資源  $R(P_i)$  の占有を開放するまでの間はタスクを開始できない状況となる。この状態を、 $P_i$  の  $P_j$  に対する待ち状態と表す。

オペレーティングシステムにおけるデッドロックは、下記の4つの状態が同時に満たされる場合に発生する。

- a. 複数のプロセスが使用可能な共有資源について、同時に複数のプロセスが資源を使用することが禁止される状態。
- b. 1つのプロセスが1つ以上の共有資源を使用している状態で、且つほかのプロセスが使用している共有資源の開放を待っている状態。
- c. 共有資源の開放は、資源を占有しているプロセスのタスク完了の後に自発的に行うことでのみ達成可能である状態
- d. 次の条件を満たすプロセスの集合  $\hat{P} := \{P_0, P_1, P_2, \dots, P_n\}$  が存在する状態。  $k \in \{0, 1, 2, \dots, n-1\}$  が、  $R(P_k) \in O(P_{k+1})$  を満たし、且つプロセス  $P_0$  と  $P_n$  が  $R(P_n) \in O(P_0)$  を満たす。

OS の分野では制約 d) の表現手法として、Wait-for graph [72, 73, 80, 81] を用いる。以降では、デッドロックの条件の一つである制約 d) の解消に着目した、既存のデッドロック解決手法について述べる。デッドロック解決手法はデッドロック回避とデッドロック回復の2種類に大別される。

デッドロック回避ではデッドロックを予見し、予見したデッドロックを回避するためのプロセスの資源割り当て計画手法 [14, 16, 62, 78] やスケジューリング手法 [33, 37, 40–44, 76] が研究されている。デッドロック回避では、デッドロックの予見において Wait-for graph に基づきエージェントの動作干渉を評価する。文献 [14] では、限定的な状況でエージェント間のすれ違い移動を許容し、すれ違い移動が許可されない区間に交通ルールを設定することでデッドロックフリーな経路計画を実現している。文献 [16, 78] では、ホモジニアスなエージェント群の予見したデッドロックを回避のするための経路計画手法が提案されている。文献 [42] では、事前に与えられたエージェントの移動経路に基づいた、デッドロックの予見と予見結果に基づくエージェントの移動順序制御によるデッドロック回避を提案している。一方で、文献 [41, 78] では、予期しないエージェントの一時停止や、エージェント間の速度のばら

つきから、経路計画手法によるデッドロックの完全回避は困難であるとされている。ヘテロジニアスなエージェント群によるグラフ探索や、大量のエージェントによる同時探索においては回避困難なデッドロックの発生頻度が上昇する。

デッドロック回復は発生したデッドロックの検知と、デッドロック要因の資源占有を解消することによるデッドロックからの回復で構成される。デッドロック回復ではデッドロックの検知において Wait-for graph に基づきエージェントの動作干渉を評価する。一般に、デッドロックからの回復はデッドロックに陥ったプロセスの一つをロールバックすることで達成される。文献 [41] では AGV システムでのデッドロックからの回復手法として、システムの一時停止とデッドロックの要因となっている AGV の退避を提案している。文献 [41] ではデッドロックに陥った AGV の一時停止と退避によってプロセスのロールバックを表現し、デッドロック検知時に発生したすべてのデッドロックを解消できる AGV の退避経路導出を実現している。このとき、AGV の退避経路導出の計算複雑さは、同時に発生するデッドロックの数とデッドロックに陥った AGV の台数に影響される。また、デッドロック回復ではロールバックしたプロセスを再開した後、同箇所デッドロックが再発する可能性がある。文献 [41] においても、退避した AGV の移動再開により同箇所でのデッドロックの発生、AGV の退避が繰り返される将棋における千日手のような状況が発生する可能性が示されている。エージェントの移動を要さないデッドロック回復手法として文献 [79] では、エージェント同士のすれ違い移動ができない場合に有効な荷物の受け渡し機能を実現するためのロボットマニピュレータの基礎研究がされている。文献 [78] では、デッドロックからの回復手法として荷物の受け渡し機能を利用してバケツリレー形式で荷物の受け渡しが紹介されている。荷物の受け渡しによるデッドロックからの回復では、プロセスのロールバックが発生しないため同箇所でのデッドロックの再発を防止できる。ただし、文献 [78] が述べている通り、荷物の受け渡しに高精度の制御が要求される場合はデッドロック回避やロールバックによるデッドロックからの回復をデッドロック解決手法に用いることが望まれる。

文献 [14, 16, 41, 42, 62, 78] では、Wait-for graph によりエージェントの協調・非協調動作とデッドロックの関係が示されている。

本博士論文では、大量のヘテロジニアスなエージェントによるグラフの協調探索を対象とする。大量のヘテロジニアスなエージェントを運用する例として、大規模な EC 倉庫でのエージェントによる荷物の搬送問題が挙げられる。EC 倉庫のような多品種少量生産の環境では、新規の搬送タスクが動的に発生する。この場合、デッドロックの発生頻度の上昇が想定されるため、短時間で確実にデッドロックを解消することが望まれる。そこで、デッドロック解決の手法としてエージェント間での荷物の受け渡しによるデッドロック回復を採用し、検知したデッドロックからの回復における荷物の受け渡しの最小化による効率的な渋滞緩和を考える。

## 3 Demand-for graph による干渉解析

### 3.1 はじめに

本章では、本博士論文で扱う渋滞とデッドロックの例とその解析手法について述べる。本博士論文ではデッドロックの前提条件の制約 d) を表す Wait-for graph [80] により渋滞を表現する。渋滞の Wait-for graph 表現により、エージェントの動作干渉に基づく渋滞規模の評価とデッドロックの検知が可能となる。

更に、本博士論文ではエージェントの協調・非協調動作による、渋滞およびデッドロックの発生とその拡大・解消の評価に取り組む。渋滞の拡大・解消の評価のためには、エージェントの協調動作と非協調動作に基づく渋滞規模とその時間変化の定式化が課題となる。そこで、エージェントの協調・非協調動作による渋滞規模の変化を評価するための新たな干渉関係表現として、エージェント間の行動要求を表す Demand-for graph を提案する。本論文では、Demand-for graph が表す行動要求の干渉関係について、Wait-for graph が表すエージェントの動作干渉との関係を示す。これにより、エージェントの行動要求の干渉に基づく渋滞規模の評価を実現する。また、Demand-for graph による渋滞解析に、エージェントの協調・非協調動作による要求達成とそれによる Demand-for graph の形状変化の評価を導入する。これにより、エージェントの協調・非協調動作に基づく、渋滞の拡大・解消速度の評価を実現する。

本章では、本博士論文で扱うマルチエージェントシステムのグラフ探索における、Wait-for graph と Demand-for graph による渋滞表現を示す。

### 3.2 Wait-for graph による渋滞規模評価

Wait-for graph はプロセスによる共有資源の占有と、資源割り当て要求に基づき、プロセス間の待ち状態を表現する。

Wait-for graph はプロセスを表すノードの有限集合  $N_w := \{n_1, n_2, \dots, n_I\}$  と、プロセス間の待ち状態を表すアークの有限集合  $A_w$  で構成される。Wait-for graph において、ノード  $n_i$  はプロセス  $P_i$  に対応する。Wait-for graph のアーク  $A_w$  は

$$\begin{aligned} A_w &= \sum_{n_i \in N_w} \sum_{n_j \in N_w/n_i} f_1(t, i, j) \\ f_1(t, i, j) &= \begin{cases} (n_i, n_j) & \text{if } (R(P_i) \in O(P_j)) \\ \emptyset & \text{otherwise} \end{cases} \end{aligned} \quad (2)$$

と与えられる。式 (2) において  $(n_i, n_j) \in A_w$  は、プロセス  $P_i$  が  $P_j$  の待ち状態であることを意味する。式 (2) より、制約 d) を満たすプロセス群は Wait-for graph 上の閉路で表現される。したがって、デッドロックの発生は Wait-for graph 上の閉路の有無で判断可能である。

ここで、文献 [46, 48, 54] における ASEP 探索を例に、マルチエージェントシステムのグラフ探索での式 (2) の Wait-for graph に基づく渋滞規模の評価を説明する。ASEP の探索においては移動するエージェントがプロセス、ASEP を構成する区画が共有資源に該当する。ASEP 探索 [46, 48] において、エージェントは自身の前方の区画に他のエージェントが存在する場合に衝突回避のため一時停止を選択する。密度と流量に基づく渋滞解析において、衝突回避のための一時停止は高密度状態での流量低下、すなわち渋滞を意味する。エージェントの一時停止の条件は、プロセス  $P_i$  とその資源割り当て要

求, および他プロセスの占有資源  $O(P_j)$  を用いて

$$\exists P_j \in P, \text{s.t. } R(P_i) \in O(P_j) \quad (3)$$

と表される. そこで, 本博士論文においては式 (3) が成り立つ状況を渋滞として扱う. 式 (2), (3) により, Wait-for graph 上で  $n_i$  の出次数  $\text{outdeg}(G, n_i)$  が 1 以上の場合に  $P_i$  は式 (3) を満たす. このとき, 式 (3) に示す渋滞は Wait-for graph の連結成分に表れる. 以上から, 本博士論文では Wait-for graph 上の端点間の距離を渋滞の長さとして定義する. グラフ上のノード間の距離は, 二つのノードを結ぶ最短経路の長さを意味する. このとき, 渋滞の長さは渋滞最後尾のエージェントが渋滞から分離するために解消する必要がある待ち状態の数を表す.

次に, デッドロックに対する渋滞規模の評価について記す. デッドロックは Wait-for graph 上の閉路で表現される. しかしながら, 閉路とその連結成分には  $\text{outdeg}(n_i) = 0$  となる端点が存在しない. そこで, 本論文では  $\text{outdeg}(n_i) = 0$  となる端点が存在しない場合の渋滞の長さを, 閉路と端点との距離, および閉路の大きさの和で定義する. このとき, 閉路の大きさは, 閉路上に存在するノードの数を意味する. 以降では, デッドロックに対する渋滞規模をデッドロック規模と表記する.

### 3.3 Demand-for graph による渋滞解消速度の評価

Wait-for graph は個々のエージェントの行動に対する待ち状態の変化の情報に含まれない. 本博士論文では, 新たな渋滞表現の方法として Wait-for graph にエージェントの非協調動作と協調動作に対する行動要求の概念を導入した Demand-for graph を提案する. ここで, 協調動作はその実行に他のエージェントとの通信情報を利用する動作を意味する. 対して, 非協調動作は他のエージェントの通信情報を利用せず, 個々のエージェントが単独で実行する動作を意味する. 行動要求は協調・非協調動作の干渉によって生じる待ち状態の解消に要する, 他のエージェントの協調・非協調動作の実施要求を意味する. エージェントの協調・非協調動作に対する行動要求の発生を解析することで, Demand-for graph による動作干渉と渋滞規模の評価を実現する. また, エージェントの協調・非協調動作による行動要求の達成に基づき, 協調・非協調動作による渋滞解消の評価を実現する. 以降では, エージェント間の行動要求の設定と達成に基づく, Demand-for graph のダイナミクスを示す.

時刻  $t$  におけるプロセス  $P_i$  による行動要求の有限集合を  $R_p(t, P_i) \subset \{P_1, P_2, \dots, P_I\}$  で表す.  $P_j \in R_p(t, P_i)$  は  $P_i$  から  $P_j$  に対する行動要求を表す. プロセス間の行動要求は, 協調動作または非協調動作によって達成可能なものとする.  $P_i$  から  $P_j$  に対する行動要求が達成されたとき,  $R_p(t+1, P_i)$  から  $P_j$  を削除する.

Demand-for graph はプロセスの行動要求を表す. Demand-for graph は時刻  $t$  におけるプロセスを表すノードの有限集合  $N_d := \{n_1, n_2, \dots, n_I\}$  と, プロセス間の行動要求を表すアークの有限集合  $A_d(t)$  で構成される. このとき, 時刻  $t$  における Demand-for graph  $G_d(t)$  は  $R_p(t, P_i)$  に対して,

$$\begin{aligned} A_d(t) &= \sum_{n_i \in N_d} \sum_{n_j \in N_d/n_i} f_2(t, i, j) \\ f_2(t, i, j) &= \begin{cases} (n_i, n_j) & \text{if } (P_j \in R_p(t, P_i)) \\ \emptyset & \text{otherwise} \end{cases} \end{aligned} \quad (4)$$

と与えられる.

このとき, 行動要求  $R_p(t, P_i)$  が

$$P_j \in R_p(t, P_i) \iff R(t, P_i) \in O(P_j) \quad (5)$$

を満たすならば、式 (2), (4) より  $G_w(t) = G_d(t)$  が成り立つ。このとき、Demand-for graph 上の端点間の距離は、渋滞の長さを表す。また、Demand-for graph 上の閉路は、デッドロックの発生を意味する。更に、 $G_w(t) = G_d(t)$  は  $P_i$  から  $P_j$  に対する行動要求達成によりプロセス  $P_i$  の  $P_j$  に対する待ち状態が解消されることを意味する。したがって、要求達成、すなわちエージェントの非協調動作と協調動作による Demand-for graph の構造変化から、渋滞とデッドロックの解消の評価が可能になる。

次に、Demand-for graph のダイナミクス表現を示す。本博士論文では時刻  $t$  における Demand-for graph の隣接行列を  $\Phi(t) \in \{0, 1\}^{I \times I}$  と表す。 $\Phi(t)$  の  $i$  行  $j$  列要素  $\phi_j^i(t) \in \{0, 1\}$  は、時刻  $t$  の Demand-for graph  $G_d(t)$  におけるエッジ  $(n_i, n_j)$  の有無を表す。したがって、 $\phi_j^i(t) = 1$  は  $(n_i, n_j) \in A_d(t)$  と同義である。Demand-for graph の形状変化を

$$\Phi(t+1) = B\mathbf{u}_1(t) - D\mathbf{u}_2(t) \quad (6)$$

と与える。式 (6) において、 $bm\mathbf{u}_1(t)$  は動作干渉による行動要求の設定を表す。 $\mathbf{u}_2(t)$  はエージェントの動作による行動要求の達成を表す。行列  $B$  と  $D$  は  $\mathbf{u}_1(t)$  と  $\mathbf{u}_2(t)$  に対する Demand-for graph の変化を表す入力行列である。 $B$  と  $\mathbf{u}_1(t)$  の設定により、特定の動作に起因する渋滞の規模の観測を実現する。 $D$  と  $\mathbf{u}_2(t)$  の設定により、協調・非協調動作による行動要求の達成を定義する。

ここで、Demand-for graph に対して、式 (6) における各行列の設定例を示す。式 (4) より  $\mathbf{u}_1(t)$  を  $R_p(t, P_i)$  に対して

$$\mathbf{u}_1(t) = \begin{bmatrix} \mathbf{U}_1(t, 1) \\ \vdots \\ \mathbf{U}_1(t, I) \end{bmatrix}, \quad (7)$$

$$\mathbf{U}_1(t, i) = \begin{bmatrix} f_3(t, i, 1) & 0 & \cdots & 0 \\ 0 & f_3(t, i, 2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & f_3(t, i, I) \end{bmatrix},$$

$$f_3(t, i, j) = \begin{cases} 1 & (\text{if } P_j \in R_p(t, i)) \\ 0 & \text{otherwise} \end{cases}$$

と与える。また、 $\mathbf{u}_1(t)$  に対する入力行列  $B$  を

$$B = \begin{bmatrix} \mathbf{1}_I^T & \mathbf{0}_I^T & \cdots & \mathbf{0}_I^T \\ \mathbf{0}_I^T & \mathbf{1}_I^T & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_I^T \\ \mathbf{0}_I^T & \cdots & \mathbf{0}_I^T & \mathbf{1}_I^T \end{bmatrix} \quad (8)$$

と与える。式 (7) において  $\mathbf{U}_1(t, i)$  は  $j$  行  $j$  列要素に  $f_3(t, i, j) \in \{0, 1\}$  を持つ対角行列である。 $f_3(t, i, j)$  は  $P_i$  の  $P_j$  に対する行動要求の有無を表す。式 (8) より、 $B$  は要素数  $I$  の行ベクトル  $\mathbf{1}_I^T$  を対角要素に持つ、対角ブロック行列である。式 (7), (8) より、式 (6) における  $B\mathbf{u}_1(t)$  の  $i$  行  $j$  列要素は  $P_i$  の  $P_j$  に対する行動要求の有無に該当する。

ここで、時刻  $t$  におけるプロセス  $P_i$  によるタスク完了を  $T(t, P_i) \in \{0, 1\}$  と表す。時刻  $t$  において  $P_i$  がタスクを完了し共有資源を開放する場合、 $T(t, P_i) = 1$  となる。式 (6) について、 $\mathbf{u}_2(t)$  を  $T(t, P_i)$  に対して

$$\mathbf{u}_2(t) = \begin{bmatrix} \mathbf{U}_2(t, 1) \\ \vdots \\ \mathbf{U}_2(t, I) \end{bmatrix}, \quad (9)$$

$$\mathbf{U}_2(t, i) = \begin{bmatrix} f_3(t, i, 1)T(t, P1) & 0 & \cdots & 0 \\ 0 & f_3(t, i, 2)T(t, P2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & f_3(t, i, I)T(t, PI) \end{bmatrix}$$

と与える。また、 $\mathbf{u}_2(t)$  に対する入力行列  $\mathbf{D}$  を

$$\mathbf{D} = \begin{bmatrix} \mathbf{1}_I^T & \mathbf{0}_I^T & \cdots & \mathbf{0}_I^T \\ \mathbf{0}_I^T & \mathbf{1}_I^T & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_I^T \\ \mathbf{0}_I^T & \cdots & \mathbf{0}_I^T & \mathbf{1}_I^T \end{bmatrix} \quad (10)$$

と与える。式 (9) において  $\mathbf{U}_2(t, i)$  は  $j$  行  $j$  列要素に  $f_3(t, i, j)T(t, Pj)$  を持つ対角行列である。 $f_3(t, i, j)T(t, Pj)$  は  $Pj$  のタスク完了による  $Pi$  から  $Pj$  への行動要求の達成を意味する。式 (10) より、 $\mathbf{D}$  は  $\mathbf{B}$  と同じく、要素数  $I$  の行ベクトル  $\mathbf{1}_I^T$  を対角要素に持つ対角ブロック行列である。式 (9)、(10) より、式 (6) における  $\mathbf{D}\mathbf{u}_2(t)$  の  $i$  行  $j$  列要素は、 $Pj$  のタスク完了による  $Pi$  から  $Pj$  への行動要求の達成を意味する。式 (7)、(9) より、式 (6) について

$$\mathbf{u}_2(t) \leq \mathbf{u}_1(t) \quad (11)$$

が真となる。また、式 (8)、(10) より  $\mathbf{B} = \mathbf{D}$  であることから、式 (6) において  $\mathbf{u}_1(t) - \mathbf{u}_1(t-1)$  が渋滞の拡大を  $\mathbf{u}_2(t)$  が渋滞の解消を意味する。

### 3.4 Demand-for graph による渋滞表現の具体例

本節では、 $I$  台の移動物体を模したエージェント群  $\mathcal{I} := \{1, 2, \dots, I\}$  によるグラフ探索における渋滞とデッドロックの例を用いて、Demand-for graph による渋滞とデッドロックの解析の課題を示す。以降では探索システムにおける  $i$  番目 ( $i \in \mathcal{I}$ ) のエージェントを  $\mathcal{A}_i$  と表記する。 $\mathcal{A}_i$  は 2.2, 3.2, 3.3 節における  $Pi$  に対応する。

本節では、移動要求に対する Demand-for graph を示す。エージェントは自身の目標移動先に他のエージェントが存在している場合、対象のエージェントに移動要求を発する。このとき、行動要求に対して式 (5) が真となるため、Demand-for graph  $G_d(t)$  と Wait-for graph  $G_w(t)$  の形状は一致する。また、本節ではエージェントの非協調動作である区画移動によってのみ、移動要求が達成可能であると仮定する。したがって、 $\mathcal{A}_i$  による行動要求の達成とそれによる共有資源の開放である  $T(t, \mathcal{A}_i) = 1$  は  $\mathcal{A}_i$  の区画移動と同義である。

#### 3.4.1 2 台のエージェントのデッドロック

本博士論文で取り扱う最も簡単な問題は 2 台のエージェントの移動要求の干渉によるデッドロックである。2 台のエージェントによるデッドロックの例を図 5 に示す。

図 5 において丸はエージェントを表し、矢印はエージェントの進行方向を表す。図 5 の例では、 $\mathcal{A}_1$  と  $\mathcal{A}_2$  について

$$R(t, \mathcal{A}_1) = O(t, \mathcal{A}_2) \wedge R(t, \mathcal{A}_2) = O(t, \mathcal{A}_1) \quad (12)$$

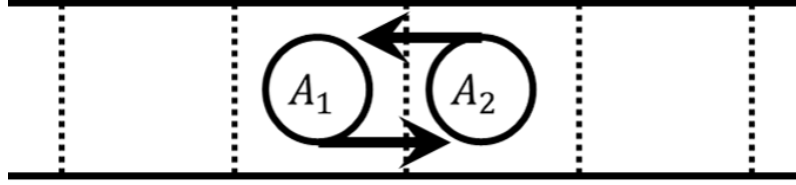


図 5: 2 台のエージェントのデッドロック

が真となる。すなわち、 $A_1$  と  $A_2$  は互いの現在位置に移動を試みている状態である。また、移動要求に対して

$$A_i = R_p(t, A_j) \wedge A_j = R_p(t, A_i) \quad (13)$$

が真となる。すなわち、 $A_1$  と  $A_2$  は互いに対して移動要求を発している状態である。このとき、式 (12) よりエージェントは区画移動を行えないため

$$T(t, A_1) = T(t, A_2) = 0 \quad (14)$$

が成り立つ。式 (6 - 10), (14) より導出した、図 5 に対する Demand-for graph を図 6 に示す。

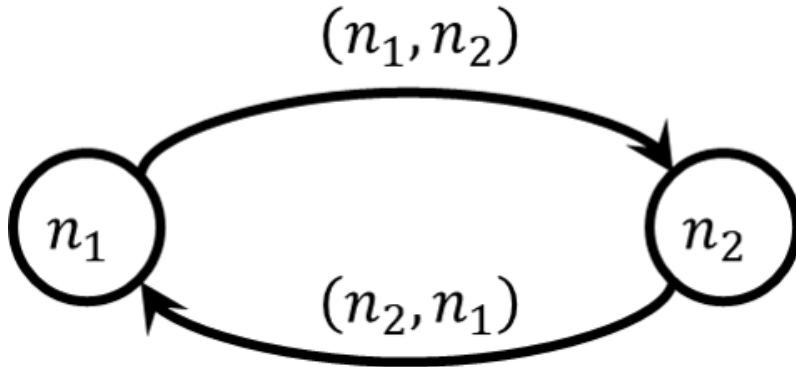


図 6: 2 台のエージェントのデッドロック

図 6 においてデッドロックの規模は閉路上のノードの数、すなわち 2 である。また、図 6 を Wait-for graph としてとらえると、2 台のエージェントは互いに待ち状態にあるため区画移動が行えない状況である。式 (6 - 10), (14) より、式 (6) の  $Du_2(t)$  は零行列となる。したがって、Demand-for graph からエッジの削除がされないことがわかる。このとき、単位時刻あたりの渋滞長の減少量を渋滞の解消速度とすると、渋滞の解消速度は 0 となる。

### 3.4.2 3 台のエージェントのデッドロックと渋滞

3 台のエージェントが関係する干渉関係では、渋滞とデッドロックが同時に発生する状況が発生する。はじめに、3 台のエージェントによる渋滞の例を図 7 に示す。

図 7 において丸はエージェントを表し、矢印はエージェントの進行方向を表す。図 7 において  $A_2$  は  $A_1$  に対する待ち状態にあり、 $A_3$  は  $A_2$  に対する待ち状態にある。図 7 におけるエージェント間の待ち関係は、

$$\begin{aligned} R(t, A_2) &= O(t, A_1) \\ R(t, A_3) &= O(t, A_2) \end{aligned} \quad (15)$$

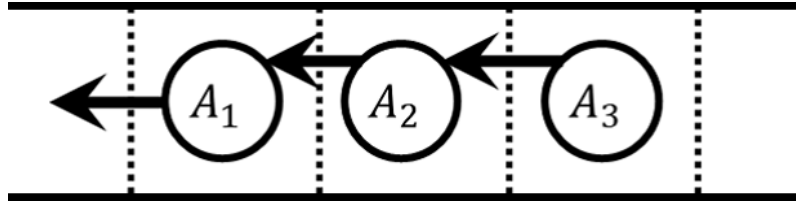


図 7: 3 台のエージェントのデッドロックと渋滞

となる。また、移動要求は

$$\begin{aligned} R_p(t, A_2) &= A_1 \\ R_p(t, A_3) &= A_2 \end{aligned} \quad (16)$$

である。図 7 において  $A_1$  のみが区画移動可能であるため  $T(t, A_i)$  は

$$\begin{aligned} T(t, A_1) &= 1 \\ T(t, A_2) &= T(t, A_3) = 0 \end{aligned} \quad (17)$$

となる。図 7 に対する Demand-for graph を図 8 に示す。

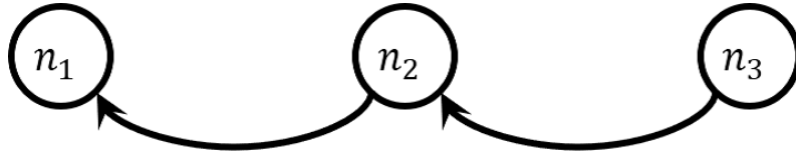


図 8: 2 台のエージェントのデッドロック

図 8 において渋滞規模は  $G_w$  上の  $n_3$  と  $n_1$  の距離、すなわち 2 である。また、図 8 を Wait-for graph としてとらえると、 $A_2$  と  $A_3$  は待ち状態にあるため区画移動が行えない状況である。したがって、達成が可能な移動要求は  $R_p(A_2)$  が発する、 $A_1$  への移動要求のみである。式 (6 - 10), (17) より、式 (6) の  $Du_2(t)$  の 2 行 1 列成分は 1 となる。したがって、単位時刻あたりの渋滞長の減少量を渋滞の解消速度とすると、渋滞の解消速度は  $A_1$  の移動確率と一致する。

次に、3 台のエージェントによるデッドロックと渋滞の同時発生例を図 9 に示す。

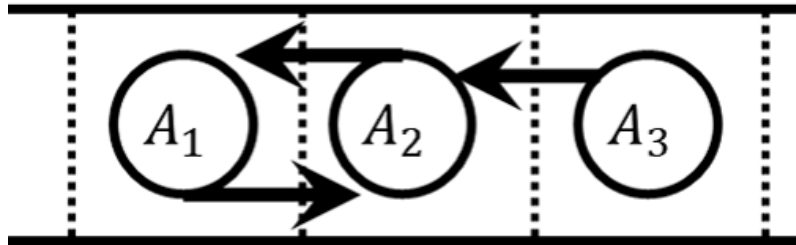


図 9: 3 台のエージェントのデッドロックと渋滞

図 9 の丸はエージェントを表し、矢印はエージェントの進行方向を表す。図 9 におけるエージェント間の待ち関係は、

$$R(A_1) = O(A_2)$$



$$\begin{aligned} R(A2) &= O(A1) \\ R(A3) &= O(A2) \end{aligned} \tag{18}$$

となる。また、移動要求は

$$\begin{aligned} R_p(A1) &= A2 \\ R_p(A2) &= A1 \\ R_p(A3) &= A2 \end{aligned} \tag{19}$$

である。移動要求の達成を表す  $T(t, Ai)$  は

$$T(t, A1) = T(t, A2) = T(t, A3) = 0 \tag{20}$$

となる。

したがって、デッドロック前提条件の制約 d), 式 (1) より図 9 において  $A1$  と  $A2$  の関係はデッドロック,  $A3$  の状態は渋滞に該当する。図 9 の状態に対する Wait-for graph と Demand-for graph を図 10 に示す。

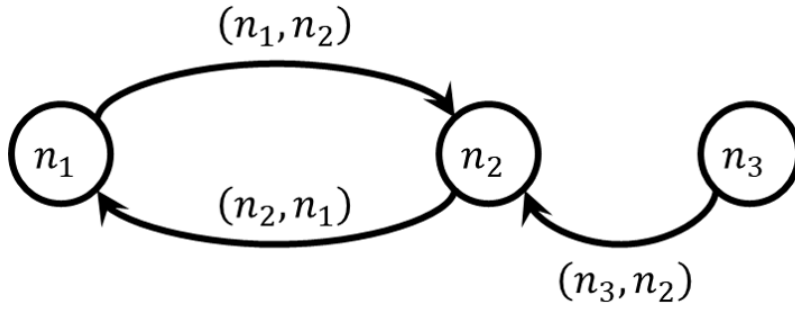


図 10: 3 台のエージェントに対する Demand-for graph

図 10 の例において、渋滞規模は  $A1$  と  $A2$  のデッドロック規模 2 と  $n_3$  と  $n_2$  の距離の和で 3 になる。式 (18), (19) より, 式 (6 - 10), (20) より, 式 (6) の  $Du_2(t)$  は零行列となる。したがって、渋滞の解消速度は 0 となる。

## 4 グラフ探索

本章では Demand-for graph による渋滞解析の適用対象となる，迷路探索，搬送，車線変更と車群形成のグラフ探索におけるエージェント移動の数理モデルを示す．本誌論文では探索対象を ASEP の直列ネットワーク，複数レーン ASEP，二次元平面とする．探索対象は区画と区画間の境界で構成される．

以降では，探索対象を有向グラフ  $N = (V, E)$  と表す． $V$  は探索対象を構成する区画の有限集合である．また， $(v_i, v_j) \in E$  は区画  $v_i$  から  $v_j$  に移動可能であることを表す．

### 4.1 ASEP

ASEP の直列ネットワーク，および複数レーン ASEP について説明するにあたり，ASEP の基本構造と ASEP 上のエージェントの動作を示す．ASEP は有限個の区画とその境界で構成される 1 次元格子である．ASEP はエージェントの流入口である始端と，流出口である終端をそれぞれ一つ持つ．エージェントは ASEP の始端から終端に向かって移動する．本博士論文では，ASEP を構成する区画の数を ASEP の長さとして扱う．

ASEP の例を図 11 に示す．図 11 は長さ 8 の ASEP である．図 11 の  $v_1$  は ASEP の始端であり，エージェントが存在しない場合は一定の確率で  $v_1$  にエージェントが生成される．図 11 の  $v_8$  は ASEP の終端であり，エージェントが存在する場合は一定の確率でそのエージェントが消失する．ASEP 上のエージェントは ASEP の始端から終端に向かって一定の確率で 1 区画ずつ移動する．このとき，既にほかのエージェントが存在する区画への移動が制限される．

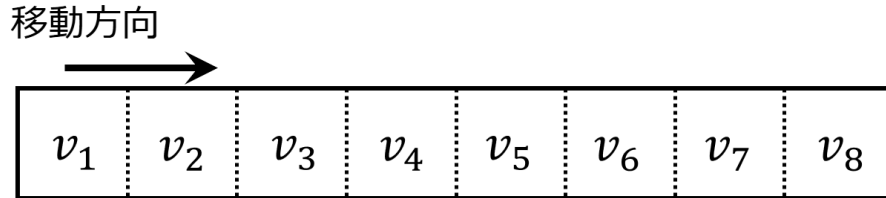


図 11: ASEP

### 4.2 ASEP の直列ネットワーク

本論文で扱う ASEP の直列ネットワークは，長さの同じ複数の ASEP と頂点サイトで構成される．頂点サイトは ASEP の終点と始点を接続する区画である．ASEP は頂点サイトによって直列に接続される．以降では， $X$  個の長さ  $Y$  の ASEP と  $Z$  個の頂点サイトで構成される ASEP の直列ネットワークについて説明する．ASEP の直列ネットワークにおいて  $\mathcal{L}x := \{v_1^x, v_2^x, \dots, v_Y^x\}$  ( $1 \leq x \leq X$ ) は  $N$  の  $x$  番目の ASEP を表す． $v_y^x \in \mathcal{L}x$  は ASEP  $\mathcal{L}x$  の  $y$  番目の区画を表す． $v_{cz} \in V$  は  $N$  の  $z$  番目の頂点サイトを表す．

ASEP の直列ネットワークの例を図 12 に示す．

ASEP の直列ネットワークにおいて，ASEP の終端  $v_Y^x$  に存在するエージェントは一定の確率で隣接する頂点サイト  $v_{cz}$  に移動する．また，頂点サイト  $v_{cz}$  に存在するエージェントは隣接する ASEP の始端  $v_1^x$  から移動先の区画を選択し，一定の確率で選択した ASEP の始端に移動する．頂点サイトへの移

動, および頂点サイトからの移動においても, 既にほかのエージェントが存在する区画への移動が制限される. また, 複数のエージェントが同時に頂点サイトへの移動を試みる場合, 衝突回避のために一台のエージェントのみが頂点サイトに移動する.

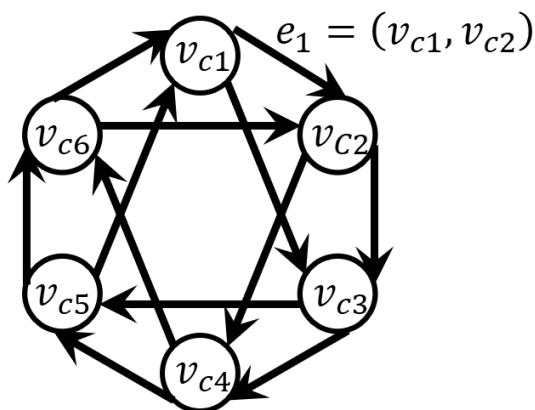


図 12: ASEP の直列ネットワーク

### 4.3 複数レーン ASEP

複数レーン ASEP は長さの等しい複数の ASEP が並列して接続する ASEP ネットワークである. 本論文では, レーン数  $X$ , 長さ  $Y$  の複数レーン ASEP を  $N := \mathcal{L}1 \cup \mathcal{L}2 \cup \dots \cup \mathcal{L}X$  で表す.  $\mathcal{L}x := \{v_1^x, v_2^x, \dots, v_Y^x\}$  ( $1 \leq x \leq X$ ) は  $N$  の  $x$  番目のレーンに位置する ASEP を表す.  $v_y^x \in \mathcal{L}x$  は ASEP  $\mathcal{L}x$  の  $y$  番目の区画を表す.

レーン数 3, 長さ 8 の複数レーン ASEP を図 13 に示す. 複数レーン ASEP 上のエージェントは前方の区画への移動のほか, 隣接する ASEP への車線変更が可能である. 図 13 において  $v_4^2$  に位置するエージェントは,  $v_4^1, v_4^3$  への車線変更, または  $v_5^2$  への移動が可能である.

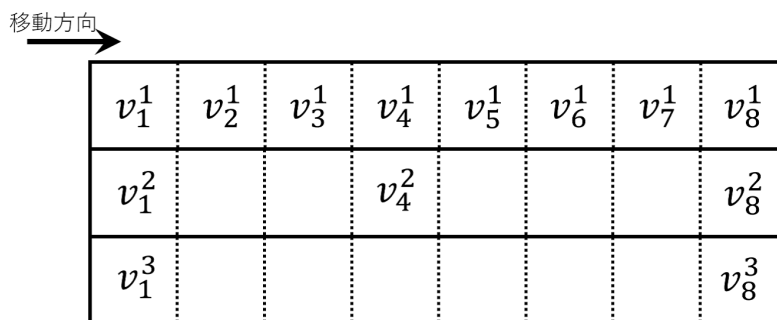


図 13: 複数レーン ASEP

複数レーン ASEP では, エージェント間の衝突防止のために直進優先のルールを適用する. これにより, 車線変更では移動先の区画だけではなく, 移動先の区画の後方に隣接する区画にエージェントが存在する場合も移動を制限する.

#### 4.4 二次元平面

$X \times Y$  平面は横  $X$  区画, 縦  $Y$  区画の二次元平面である. エージェントは, 隣接する区画から移動先の区画を選択し, 一定確率で選択した区画に移動する. また, 複数のエージェントが同時に同一の区画への移動を試みる場合, 衝突回避のために一台のエージェントのみが移動する. 二次元平面上での移動ルールは, ASEP の直列ネットワークでの頂点サイト上の移動ルールと一致する. 図 14 に二次元平面の例とその ASEP の直列ネットワーク近似表現を示す. 二次元平面において,  $v^i \in V$  は  $N$  の  $i$  番目の区画を表す.

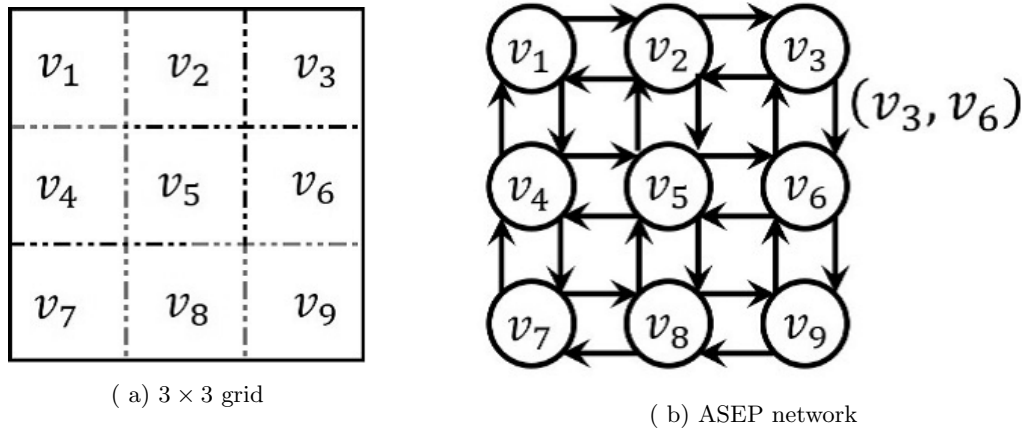


図 14: Two-dimensional grid and ASEP network

#### 4.5 エージェントの移動の数理モデル

本博士論文で扱う探索システムは, 移動, 速度変更, 通信の機能を持つエージェント群で構成される分散制御型探索システムである. 本節では, エージェントの移動の数理モデルを示す.

本博士論文ではエージェントの移動と通信による情報共有に以下の制約を設ける.

- a. 全エージェントは区画移動と通信を繰り返し行い, 移動と通信のタイミングは同期される.
- b. 1 回の区画移動における移動距離は最大で 1 区画である.
- c. 同一のノードに複数のエージェントが同時に移動することはできない. また, 他のエージェントが存在するノードに移動することはできない.
- d. エージェントは通信により移動先のノードに存在するエージェント, および同一のノードへの移動を試みる他のエージェントを検知可能である.

制約 c) より, 複数のエージェントが同一区画に同時移動することはエージェントの衝突を意味する. エージェントは制約 d) に基づく周辺のエージェント検知により, 他のエージェントとの衝突を回避する.

ここで,  $I$  台のエージェントによる探索対象  $N$  の探索について考える. 以降では探索システムにおける  $i$  番目 ( $i \in I := \{1, 2, \dots, I\}$ ) のエージェントを  $A_i$  と表し, このときの  $i$  の値をエージェントの ID として扱う.

エージェントは自己位置同定と周囲のエージェント検知の機能を持つ。自己位置同定と周囲のエージェント検知機能について、時刻  $t$  における区画  $v_x$  での  $A_i$  の有無を  $\varepsilon(t, i, v_x) \in \{0, 1\}$  で表す。時刻  $t$  に  $A_i$  が区画  $v_x \in N$  に存在する場合、 $\varepsilon(t, i, v_x)$  は 1 となる。そうでない場合、 $\varepsilon(t, i, v_x)$  は 0 となる。時刻  $t$  における  $A_i$  の位置を  $\varepsilon_A(t, i) \in N$  と表す。時刻  $t$  に  $A_i$  が存在しない場合、 $\varepsilon_A(t, i)$  は  $\emptyset$  となる。また、時刻  $t$  に区画  $v_x \in N$  に存在するエージェントの ID を  $\varepsilon_N(t, v_x) \in (\{0\} \cup \mathcal{I})$  と表す。区画  $v_x$  にエージェントが存在しない場合、 $\varepsilon_N(t, v_x)$  は 0 とする。このとき、 $\varepsilon_N(t, v_x)$  は  $\varepsilon(t, i, v_x)$  に対して

$$(\varepsilon_N(t, v_x) = i) \iff (\varepsilon(t, i, v_x) = 1), \quad (21)$$

$$(\varepsilon_N(t, v_x) = 0) \iff (\varepsilon(t, i, v_x) = 0, \forall i \in \mathcal{I}) \quad (22)$$

を満たす。

エージェントの移動に関して、時刻  $t$  における  $A_i$  の  $v^x$  への移動を  $\delta(t, i, v^x) \in \{0, 1\}$  と表す。時刻  $t$  に  $A_i$  が  $v^x$  に移動する場合、 $\delta(t, i, v^x)$  は 1 となる。そうでない場合、 $\delta(t, i, v^x)$  は 0 となる。時刻  $t$  における  $A_i$  の移動先の区画を  $\delta_A(t, i) \in N$  と表す。 $\delta_A(t, i)$  は  $\delta(t, i, v^x)$  に対して、

$$(\delta_A(t, i) = v^x) \iff (\delta(t, i, v^x) = 1) \quad (23)$$

を満たす。また、時刻  $t$  における  $A_i$  の目標移動先区画を  $\hat{\delta}_A(t, i) \in N$  と表す。制約 b) より  $\hat{\delta}_A(t, i)$  は

$$\neg \left( \hat{\delta}_A(t, i) \neq \varepsilon_A(t, i) \right) \vee \left( \left( \varepsilon_A(t, i), \hat{\delta}_A(t, i) \right) \in E \right) \quad (24)$$

を満たす。制約 d) より、エージェントは衝突回避を考慮して、目標移動先区画への移動と一時停止を選択する。このとき、 $\delta_A(t, i)$ 、 $\hat{\delta}_A(t, i) \in N$ 、 $\varepsilon_A(t, i)$  について

$$\neg (\delta_A(t, i) = v_x) \vee \left( \left( \hat{\delta}_A(t, i) = v_x \right) \vee (\varepsilon_A(t, i) = v_x) \right) \quad (25)$$

が真となる。式 (25) において  $\delta_A(t, i) = \hat{\delta}_A(t, i)$  は、エージェントが目標区画に移動することを意味する。 $\delta_A(t, i) = \hat{\delta}_A(t, i)$  は、エージェントが一時停止を選択したことを意味する。

制約 c) より、 $\delta(t, j, v_y^x)$  は  $\delta_A(t, i)$ 、 $\varepsilon(t, i, v_y^x)$  に対して

$$\varepsilon(t, j, \delta_A(t, i)) + \delta(t, j, \delta_A(t, i)) = 0, \quad \forall j \in (\{1, 2, \dots, I\} \setminus \{i\}) \quad (26)$$

を満たす。式 (26) において  $\varepsilon(t, j, \delta_A(t, i))$  は  $A_i$  の移動先の区画に存在するエージェントの有無を、 $\delta(t, j, \delta_A(t, i))$  は  $A_i$  の移動先の区画に移動するエージェントの有無を表す。したがって、式 (26) が真となることは、時刻  $t$  において  $A_i$  の移動先の区画に他のエージェントが存在しないこと、および移動先の区画に同時に区画移動するエージェントが存在しないことを意味する。時刻  $t$  における  $A_i$  の移動によるエージェントの位置更新を  $\delta_A(t, i)$  に対して

$$\varepsilon_A(t+1, i) = \delta_A(t, i) \quad (27)$$

と表す。このとき、式 (26), (27) より  $\varepsilon(t, i, v_x)$  は

$$\sum_{i \in \mathcal{I}} \varepsilon(t, i, v_x) \leq 1, \quad \forall v_x \in V \quad (28)$$

を満たす。

次に、エージェントの移動に対する待ち状態の設定と Wait-for graph による渋滞解析について示す。時刻  $t$  の Wait-for graph を  $G_w(t) = (N_w, A_w(t))$  と表す。本博士論文では、 $G_w(t)$  を  $\hat{\delta}_A(t, i)$ ,  $\varepsilon_A(t, i, v_x)$ ,  $\delta_A(t, i, v_x)$  に対して、

$$A_w(t) = \sum_{n_i \in N_w} \sum_{n_j \in N_w \setminus \{n_i\}} \begin{cases} \{(n_i, n_j)\} & \text{if } \left( \varepsilon(t, j, \hat{\delta}_A(t, i)) + \delta(t, j, \hat{\delta}_A(t, i)) > 0 \right) \\ \emptyset & \text{otherwise} \end{cases} \quad (29)$$

と与える。式 (25), (26), (29) より、

$$\neg(\exists n_j \in N_w, \text{ s. t. } (n_i, n_j) \in A_w(t)) \vee (\delta_A(t, i) = \varepsilon_A(t, i)) \quad (30)$$

が成り立つ。Wait-for graph  $G_w(t)$  について、ノード  $n_i \in N_w$  の出次数を  $\text{outdeg}(n_i)$  と表す。また、 $G_w(t)$  における  $n_i \in N_w$  と  $n_j \in N_w$  の距離を  $\Delta(G_w(t), n_i, n_j) \in \{1, 2, \dots, \infty\}$  と表す。式 (26), (28), (29) より、 $G_w(t)$  は

$$\text{outdeg}(n_i) \leq 1, \quad \forall n_i \in N_w \quad (31)$$

を満たす。

#### 4.6 Demand-for graph のダイナミクス

本項ではマルチエージェントシステムのグラフ探索における、Demand-for graph のダイナミクスの基本形を示す。ここで、時刻  $t$  における Demand-for graph  $G_d(t)$  の基本形を Wait-for graph に対して

$$G_d(t) = G_w(t) \quad (32)$$

と与える。式 (29) より、式 (32) について

$$(n_i, n_j) \in A_d(t) \iff \varepsilon(t, j, \hat{\delta}_A(t, i)) + \delta(t, j, \hat{\delta}_A(t, i)) = 1 \quad (33)$$

が真となる。したがって、 $A_i$  の区画移動と  $A_j$  の現在位置および区画移動が干渉する場合に、 $A_i$  から  $A_j$  に対する移動要求を設定する。

式 (29), (32) より、式 (6) における  $\mathbf{u}_1(t)$  を

$$\mathbf{u}_1(t) = \begin{bmatrix} \mathbf{U}_1(t, 1) \\ \vdots \\ \mathbf{U}_1(t, I) \end{bmatrix}, \quad (34)$$

$$\mathbf{U}_1(t, i) = \begin{bmatrix} f_3(t, i, 1) & 0 & \dots & 0 \\ 0 & f_3(t, i, 2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & f_3(t, i, I) \end{bmatrix},$$

$$f_3(t, i, j) = \varepsilon(t, j, \hat{\delta}_A(t, i)) + \delta(t, j, \hat{\delta}_A(t, i))$$

と与える。また、 $\mathbf{u}_1(t)$  に対する入力行列  $\mathbf{B}$  を式 (8) で与える。式 (34) において  $\mathbf{U}_1(t, i)$  は  $j$  行  $j$  列要素に  $f_3(t, i, j) \in \{0, 1\}$  を持つ対角行列である。 $f_3(t, i, j)$  は  $A_i$  の  $A_j$  に対する待ち状態と行動要求の有無を表す。式 (34), (8) より、式 (6) における  $\mathbf{B}\mathbf{u}_1(t)$  の  $i$  行  $j$  列要素は  $\varepsilon(t, j, \hat{\delta}_A(t, i)) + \delta(t, j, \hat{\delta}_A(t, i))$  と一致する。したがって、 $\mathbf{B}\mathbf{u}_1(t)$  はエージェントの一時停止による流量の低下、すなわちエージェント間の非協調動作の干渉による渋滞の拡大を表す。

式 (32) で表す Demand-for graph は、エージェントの非協調動作である区画移動によってのみ移動要求を達成可能としている。そのため、式 (6) について、 $\mathbf{u}_2(t)$  を

$$\begin{aligned} \mathbf{u}_2(t) &= \begin{bmatrix} \mathbf{U}_2(t, 1) \\ \vdots \\ \mathbf{U}_2(t, I) \end{bmatrix}, \\ \mathbf{U}_2(t, i) &= \begin{bmatrix} f_3(t, i, 1)f_4(t, 1) & 0 & \cdots & 0 \\ 0 & f_3(t, i, 2)f_4(t, 2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & f_3(t, i, I)f_4(t, I) \end{bmatrix}, \\ f_4(t, i) &= \begin{cases} 1 & (\text{if } \delta_{\mathcal{A}}(t, i) = \hat{\delta}_{\mathcal{A}}(t, i)) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (35)$$

と与える。また、 $\mathbf{u}_2(t)$  に対する入力行列  $\mathbf{D}$  を式 (10) と与える。式 (9) において  $\mathbf{U}_2(t, i)$  は  $j$  行  $j$  列要素に  $f_3(t, i, j)f_4(t, j)$  を持つ対角行列である。このとき、 $f_4(t, j) = 1$  は  $\mathcal{A}_i$  の区画移動を意味することから、 $f_3(t, i, j)f_4(t, j) = 1$  は  $\mathcal{A}_i$  の区画移動による  $\mathcal{A}_i$  から  $\mathcal{A}_j$  への行動要求の達成を意味する。式 (10) より、 $\mathbf{D}$  は  $\mathbf{B}$  と同じく、要素数  $I$  の行ベクトル  $\mathbf{1}_I^T$  を対角要素に持つ対角ブロック行列である。式 (35), (10) より、式 (6) における  $\mathbf{D}\mathbf{u}_2(t)$  の  $i$  行  $j$  列要素は、 $\mathcal{A}_i$  の区画移動による  $\mathcal{A}_i$  から  $\mathcal{A}_j$  への行動要求の達成を意味する。

このとき、式 (34), (35) より、グラフ探索においても式 (11) が成り立つ。したがって、 $\mathbf{u}_1(t) - \mathbf{u}_1(t-1)$  は渋滞の拡大に、 $\mathbf{u}_2(t)$  は渋滞の解消に対応する。

式 (25), (26), (29), (32) より  $\delta_{\mathcal{A}}(t, i)$  と  $G_d(t)$  について、

$$\neg (\delta_{\mathcal{A}}(t, i) = \hat{\delta}_{\mathcal{A}}(t, i)) \vee (\text{outdeg}(n_i) = 0) \quad (36)$$

が真となる。式 (36) は式 (35) により式 (6) に対する命題

$$\neg (f_4(t, i) = 1) \vee (\text{outdeg}(n_i) = 0) \quad (37)$$

と変換できる。式 (6), (10) より式 (37) が真となることは、非協調動作である区画移動による渋滞の自然解消について渋滞の解消が Demand-for graph の出端点でのみ発生することを意味する。これは、渋滞学における、渋滞の分離が渋滞先頭のエージェントから発生する性質と一致する。式 (34) - (37) に示す Demand-for graph の状態変化は、エージェントの非協調動作である区画移動とその干渉に対して渋滞規模の時間変化評価を実現する。

## 5 迷路探索

迷路探索問題では  $I$  台のエージェントとエージェントと同数の探索ユニットで構成される分散協調型探索システムによる二次元平面の協調探索を通じて、協調動作による渋滞緩和の有効性を検証する。

探索システムは目的地への到達や迷路形状の把握などの複数のミッションがあらかじめ与えられる。与えられる全てのミッションはエージェント単体により達成可能である。探索システムの目的は分散型探索による有限時間内の全ミッションの達成である。エージェントは移動、通信による周囲の他のエージェントとの情報共有、センサによる周辺状況把握の機能を持つ。

エージェントはそれぞれ異なる役割を持ち、役割に応じて探索行動を決定する。エージェントの役割の表現には [18, 19] で用いた探索ユニットを用いる。探索ユニットはミッション、エージェントの経路計画手法、探索記録で構成される情報である。探索ユニットはミッションと経路計画手法にしたがって、エージェントの移動経路を決定する。

探索システムは与えられたミッションと経路計画手法の組み合わせから探索ユニットを生成し、生成した探索ユニットを各エージェントに1つずつ割り当てる。探索ユニットはエージェント間での交換による再割り当てが可能である。本博士論文ではデッドロック解決のための協調機能として、エージェント間での探索ユニット交換と疑似移動を提案する。探索ユニットを交換すると、エージェントの移動経路が変化する。これにより、デッドロックの原因である相互進路妨害を解消する。また、エージェントは探索ユニットの交換に際して、探索引き継ぎのために先行研究 [18] の疑似移動を行う。疑似移動は探索ユニット交換による、ミッションと探索アルゴリズム組み合わせに対するエージェントの移動経路への影響を抑えるための区画移動を伴わない動作である。エージェントはデッドロック解消を目的とした協調動作として、探索ユニットの交換と疑似移動を行う。

迷路探索では  $I$  台のエージェントで構成され、 $I$  個のミッションを持つ分散協調探索システムにおけるデッドロック解決を考える。以降では、探索システムにおける  $i$  番目の探索ユニットを  $U_i$  と表記し、この時の  $i$  の値を探索ユニットの ID として扱う。

迷路探索問題における探索対象  $N = (V, E)$  は区画と隣接する区画間の境界で構成される二次元平面である。 $v_x \in V$  は  $x$  番目 ( $x = 1, 2, \dots, |V|$ ) のノードを表し、 $(v_x, v_y) \in E$  は  $v_x$  から  $v_y$  へつながるアークを表す。探索対象の例として二次元迷路とその有向グラフ表現を図 15 に示す。

図 15 a は隣接する区画間の境界を点線で、境界上の障害物を実線で示す。



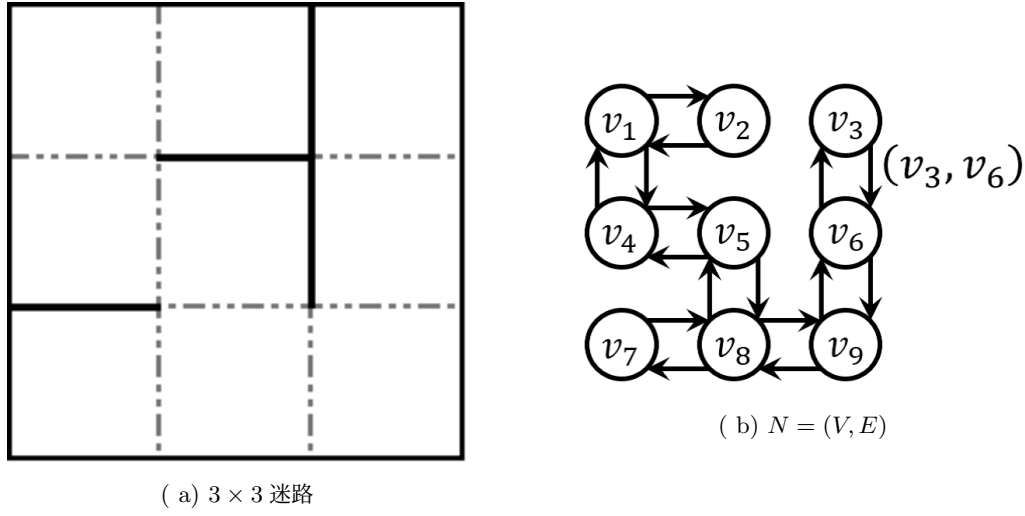


図 15: 探索対象 (迷路探索)

## 5.1 探索行動の数理モデル

迷路探索問題ではエージェントはデッドロック解消を目的とした協調動作として、探索ユニットの交換と疑似移動を行う。

はじめに、迷路探索における協調動作である探索ユニットの交換と疑似移動に関わる、探索ユニットの割り当てと ID について述べる。時刻  $t$  における  $\mathcal{A}_i$  に割り当てられた探索ユニットの ID と  $U_i$  が割り当てられたエージェントの ID をそれぞれ  $\mu_{\mathcal{A}}(t, i) \in \{0\} \cup \mathcal{I}$ ,  $\mu_U(t, i) \in \{0\} \cup \mathcal{I}$  と表す。ただし、 $\mu_{\mathcal{A}}(t, i) = 0$  は  $\mathcal{A}_i$  に探索ユニットが割り当てられていない状態を表す。同様に、 $\mu_U(t, i) = 0$  は  $U_i$  がいずれのエージェントにも割り当てられていないことを表す。 $\mathcal{A}_i$  の移動経路  $\hat{\delta}_{\mathcal{A}}(t, i)$  はエージェントに割り当てられた探索ユニットが決定する。時刻  $t$  において探索ユニットが割り当てられていないエージェントは、時刻  $t$  における経路計画を現在位置での一時停止とする。このとき、 $\varepsilon_{\mathcal{A}}(t, i)$  と  $\hat{\delta}_{\mathcal{A}}(t, i)$  について

$$\neg(\mu_{\mathcal{A}}(t, i) = 0) \vee \left( \hat{\delta}_{\mathcal{A}}(t, i) = \varepsilon_{\mathcal{A}}(t, i) \right) \quad (38)$$

が真となる。

$\mathcal{A}_i$  は探索ユニット  $U_i$  が  $\hat{\delta}_{\mathcal{A}}(t, i)$  を決定後、移動を開始する。このとき、 $\mathcal{A}_i$  は時刻  $t$  の移動先区画を  $\mathcal{A}_i$  は式 (25), (26) を満たすように  $\delta_{\mathcal{A}}(t, i)$  を決定し、衝突を回避する。衝突回避に際して、エージェントは式 (26) に基づき、移動要求を設定する。移動要求は対象のエージェントの移動、または疑似移動によって達成される。時刻  $t$  における  $\mathcal{A}_i$  の移動要求を  $p(t, i) \in \{0\} \cup \mathcal{I}$  と表す。 $p(t, i) = j$  ( $j \neq i$ ) は時刻  $t$  における  $\mathcal{A}_i$  から  $\mathcal{A}_j$  への移動要求を表す。これは、時刻  $t$  において  $\mathcal{A}_i$  が  $\mathcal{A}_j$  に対する待ち状態であることと同義である。また、 $p(t, i) = 0$  は時刻  $t$  において  $\mathcal{A}_i$  が他のエージェントの待ち状態がなく、移動要求を発していないことを表す。式 (26) より  $p(t, i)$  は

$$p(t, i) = j \iff \left( \varepsilon_{\mathcal{A}} \left( \left( t, j, \hat{\delta}_{\mathcal{A}}(t, i) \right) + \delta_{\mathcal{A}} \left( t, j, \hat{\delta}_{\mathcal{A}}(t, i) \right) = 1 \right) \wedge (j \neq i) \right) \quad (39)$$

と与えられる。式 (39), (29) より、 $p(t, i)$  は  $G_w(t)$  に対して

$$p(t, i) = j \iff (n_i, n_j) \in A_w(t) \quad (40)$$

を満たす。迷路探索問題において、エージェントの区画移動による位置更新は式 (27) に従う。

ここで、情報共有のための機能としてエージェント間の通信の数理モデルを示す。エージェントの通信状態は他のエージェントとの距離や障害物の有無によって時間変化する。迷路探索問題では、エージェントの通信機能の性能を通信保証範囲で表す。本博士論文では、迷路探索における通信保証範囲を探索対象のグラフ上のエージェント間のマンハッタン距離で与える。このとき、 $A_i$  について  $\varepsilon_{\mathcal{A}}(t, i)$  の示す現在地を中心とする、通信保証範囲内に存在する他のエージェントとの情報共有が保証される。

通信保証範囲を  $d \in \mathbb{N}$  と表す。また、時刻  $t$  における  $A_i$  と  $A_j$  の距離を  $\Delta(N, \varepsilon_{\mathcal{A}}(t, i), \varepsilon_{\mathcal{A}}(t, j)) \in \mathbb{N}$  と表す。  $\Delta(N, \varepsilon_{\mathcal{A}}(t, i), \varepsilon_{\mathcal{A}}(t, j))$  は探索対象  $N$  における、 $A_i$  と  $A_j$  が存在するノード間の距離である。有向グラフにおける二つのノード間の距離は二つのノードを結ぶ最短経路の辺数である。時刻  $t$  における  $A_i$  から  $A_j$  への通信状態を  $c_j^i(t) \in \{0, 1\}$  で表す。  $c_j^i(t)$  は

$$\Delta(N, \varepsilon_{\mathcal{A}}(t, i), \varepsilon_{\mathcal{A}}(t, j)) \leq d \implies c_j^i(t) = c_i^j(t) = 1 \quad (41)$$

を満たす。式 (41) より、 $A_i$  と  $A_j$  が互いに通信保証範囲内にいる場合、通信の双方向性が保証される。通信の双方向性は探索ユニット交換の合意条件に影響する。

時刻  $t$  において  $A_i$  が  $A_j$  の送信情報  $\omega(t, j)$  を受信したとき、その受信情報を元情報  $\omega(t, j)$  と区別するために  $\bar{\omega}(t, i, j)$  と表す。  $\bar{\omega}(t, i, j)$  は  $c_j^i(t) \in \{0, 1\}$  によって

$$\bar{\omega}(t, i, j) = c_j^i(t)\omega(t, j) \quad (42)$$

となる。  $A_i$  が  $A_j$  の送信情報を受信できなかった場合、 $\bar{\omega}(t, i, j)$  は 0 になる。

## 5.2 疑似移動によるデッドロック解決

本節では、迷路探索問題における協調動作として、デッドロック解決のための疑似移動を示す。

迷路探索問題においてエージェントは移動要求に基づきデッドロックを検知し、探索ユニットの交換と疑似移動によって検知したデッドロックからの回復を実現する。制約 d) より、式 (29) の  $G_w(t)$  上の閉路がデッドロックを意味する。ここで、 $G_w(t)$  上の二つのノード  $n_i$  と  $n_j$  の距離を  $\Delta(G_w(t), n_i, n_j) \in \{0, 1, \dots, \infty\}$  と表す。  $n_i$  が  $G_w(t)$  の閉路上に存在する場合、 $\Delta(G_w(t), n_i, n_i) \neq \infty$  となる。したがって、 $\Delta(G_w(t), n_i, n_i) \neq \infty$  は  $A_i$  がデッドロック状態にあることと同義である。

デッドロックの解決は  $G_w(t)$  上の閉路の検知によるデッドロックの検知と、探索ユニットの交換を用いたデッドロックからの回復動作で構成される。探索ユニットの交換により、エージェント間でミッションと探索記録を交換することができる。以降では、エージェント間での探索ユニットの交換を役割交換と表す。本博士論文では、デッドロックからの回復のための役割交換の目標を

$$\mu_{\mathcal{A}}^{\mu_{\mathcal{A}}(t, i)}(t+1) = \begin{cases} i & \text{if } \Delta(G_w(t), n_i, n_i) = \infty \\ p(t, i) & \text{otherwise} \end{cases} \quad (43)$$

と表す。式 (43) は  $G_w(t)$  上の閉路の形状に沿った役割交換を意味する。デッドロックの原因となるエージェント間の役割交換と疑似移動により、エージェントの経路計画結果を変化させることでデッドロックを解消する。

エージェント間での情報共有による、式 (43) の役割交換実現のための通信手段について述べる。エージェント間での探索ユニットの再割り当ての表現には探索引き継ぎ指令  $r(t, i) \in \mathcal{I}$  を用いる。  $r(t, i) = j$  は時刻  $t$  において  $\mu_{\mathcal{A}}(t+1, j) = \mu_{\mathcal{A}}(t, i)$  を満たす探索ユニットの再割り当ての要求を意味する。式 (43) を  $\mu_{\mathcal{A}}(t+1, i)$  と  $r(t, i)$  から

$$(\mu_{\mathcal{A}}(t+1, i) = \mu_{\mathcal{A}}(t, j)) \iff (r(t, j) = i), \quad (44)$$

$$(r(t, i) = p(t, j)) \iff (\Delta(G_w(t), n_i, n_i) \neq \infty) \quad (45)$$

と表す。このとき、式 (31) より式 (45) が成り立つとき、

$$\neg(r(t, j) = i) \vee (r(t, k) \neq i), \forall k \in \mathcal{I} \setminus \{j\} \quad (46)$$

は真となる。

本博士論文ではエージェントの経路計画に制限を設けることで、過去の共有情報を利用した継続的なデッドロック検知を実現する。以降では、エージェントの経路計画について、

$$\neg \left( \varepsilon_{\mathcal{A}}(t+1, i) \neq \hat{\delta}_{\mathcal{A}}(t, i) \wedge \mu_{\mathcal{A}}(t+1) = \mu_{\mathcal{A}}(t+1) \right) \vee \left( \hat{\delta}_{\mathcal{A}}(t+1, i) = \hat{\delta}_{\mathcal{A}}(t, i) \right), \quad (47)$$

$$\forall i \in \{1, 2, \dots, I\}$$

の制約を設ける。

式 (39) よりエージェントはデッドロック検知のために、 $p(t, i)$  を共有し  $\Delta(G_w(t), n_i, n_j) \neq \infty$  を満たすノード  $n_j$  の集合  $\hat{N}_w^i(t) \subseteq N_w$  を導出する。デッドロック検知のための情報共有では、エージェントを中継した情報共有を行う。 $\mathcal{A}_i$  は  $\hat{N}_w^i(t)$  にしたがって連結範囲の情報の共有と更新を繰り返すことにより、 $\Delta(G_w(t), n_i, n_j) \neq \infty$  を満たすノード  $n_j$  の集合を推定する。このとき、情報の共有と更新の繰り返し回数を中継数  $h \in \mathbb{N}$  と表す。時刻  $t$  において  $\mathcal{A}_i$  が  $h'$  回目の情報共有で送信する連結範囲の情報を  $P_{h'}^i(t) \subseteq N_w$  と表す。 $n_j \in P_{h'}^i(t)$  は  $\Delta(G_w(t), j, i) \neq \infty$  を意味する。情報共有において、 $\mathcal{A}_i$  が  $\mathcal{A}_j$  から受信する  $p(t, j)$  は式 (42) から  $\bar{p}(t, i, j)$  となる。また、式 (42) より、 $\mathcal{A}_i$  が  $\mathcal{A}_j$  から受信する  $P_{h'}^i(t)$  を  $\bar{P}_{h'}^i(t, i, j)$  として

$$\bar{P}_{h'}^i(t, i, j) = \begin{cases} P_{h'}^j(t) & \text{if } c_i^j(t) = 1 \\ \emptyset & \text{otherwise} \end{cases} \quad (48)$$

と与える。

デッドロック検知において、 $\mathcal{A}_i$  は Algorithm 1 にしたがって  $\hat{N}_w^i(t)$  を求める。

---

**Algorithm 1** Derive  $\hat{N}_w^i(t)$

---

```

N ← {n_i}
flag ← true
while flag do
  flag ← false
  for j ← 1; j < I + 1; j ++ do
    k ←  $\bar{p}(t, i, j)$ 
    if n_j ∉ N ∧ n_k ∈ N then
      N ← N ∪ {n_j}
      flag ← true
    end if
  end for
end while
return  $\hat{N}_w^i(t) \leftarrow N/n_i$ 

```

---

次に、 $\mathcal{A}_i$  は過去のデッドロック検知結果である  $\hat{P}^i(t) \subseteq N_w$  と  $\hat{N}_w^i(t)$  から  $P_1^i(t)$  を

$$P_1^i(t) = \hat{P}^i(t) \cup \hat{N}_w^i(t) \quad (49)$$

と求める。  $P_1^i(t)$  を共有し,  $P_{h'+1}^i(t)$  を

$$P_{h'+1}^i(t) = \sum_{j=1}^I f_5(t, i, j)$$

$$f_5(t, i, j) = \begin{cases} \bar{P}(t, i, j) & (\text{if } n_j \in \hat{N}_w^i(t)) \\ \emptyset & \text{otherwise} \end{cases} \quad (50)$$

と求める。

$P_h^i(t)$  導出後,  $\mathcal{A}_i$  は  $P_{h+1}^i(t)$  を共有する。  $\mathcal{A}_i$  は  $P_{h+1}^i(t)$  の共有結果から探索引き継ぎ指令  $r(t, i)$  を決定し, 共有する。 継続検知型では  $r(t, i)$  を

$$r(t, i) = \begin{cases} p^i(t) & (\text{if } n_i \in P_{h+1}^i(t)) \\ i & (\text{otherwise}) \end{cases} \quad (51)$$

と求める。 探索ユニットの再割り当てではエージェント間でのデッドロック検知のタイミングのずれを考慮し, ユニットのバッファ  $\mu_B(t, i) \in \{0\} \cup \mathcal{I}$  を設ける。 エージェントは Algorithm 2 にしたがって  $r(t, i)$  を共有し,  $\mu_{\mathcal{A}}(t+1, i)$  と  $\mu_B(t, i)$  を決定する。 また, エージェントは  $P_{h+1}^i(t)$  を共有し,  $\hat{P}^i(t+1)$  を

$$\hat{P}^i(t+1) = \begin{cases} \emptyset & \text{if } ((\delta_{\mathcal{A}}(t, i) \neq \varepsilon_{\mathcal{A}}(t, i)) \vee r(t, i) \neq i) \\ P_{h+2}^i(t) & \text{otherwise} \end{cases} \quad (52)$$

と求める。

---

**Algorithm 2** Derive  $\mu_{\mathcal{A}}(t+1, i)$  and  $\mu_{\mathcal{A}}(t, i)$

---

```

if  $r^i(t) \neq i$  then
   $\mu_{\mathcal{A}}(t+1, i) \leftarrow 0$ 
else
   $\mu_{\mathcal{A}}(t+1, i) \leftarrow \mu_{\mathcal{A}}(t, i)$ 
end if
 $\mu_B(t+1, i) \leftarrow 0$ 
for  $j \leftarrow 1; j < I+1; j++$  do
  if  $\bar{r}(t, i, j) == i \& \& j \neq i$  then
    if  $r^i(t) \neq 0$  then
       $\mu_{\mathcal{A}}(t+1, i) \leftarrow \bar{\mu}_{\mathcal{A}}(t, i, j);$ 
    else
       $\mu_B(t+1, i) \leftarrow \bar{\mu}_{\mathcal{A}}(t, i, j)$ 
    end if
  break;
  end if
end for

```

---

### 5.3 Demand-for graph による渋滞解析

迷路探索問題では、移動要求  $p^i(t)$  に対して Demand-for graph による渋滞解析を適用する。時刻  $t$  における Demand-for graph  $G_d(t)$  を移動要求  $p(t, i)$  に対して

$$\begin{aligned} A_d(t) &= \sum_{n_i \in N_d} \sum_{n_j \in N_d/n_i} f_6(t, i, j) \\ f_6(t, i, j) &= \begin{cases} (n_i, n_j) & \text{if } (p(t, i) = j) \\ \emptyset & \text{otherwise} \end{cases} \end{aligned} \quad (53)$$

と与える。式 (53) より、 $G_d(t)$  のエッジ  $(n_i, n_j) \in A_d(t)$  は、 $\mathcal{A}_i$  の  $\mathcal{A}_j$  に対する移動要求を表す。すなわち、 $G_d(t)$  は  $p^i(t)$  に対して

$$(n_i, n_j) \in A_d(t) \iff p(t, i) = j \quad (54)$$

を満たす。このとき、式 (40), (54) より、

$$G_w(t) = G_d(t) \quad (55)$$

が成り立つ。すなわち、迷路探索問題における Wait-for graph と Demand-for graph の形状は一致する。式 (55) よりデッドロック未発生時の渋滞の長さは、 $G_d(t)$  上の端点間の距離と一致する。また、発生したデッドロックの規模は  $G_d(t)$  上の閉路の大きさと一致する。 $\Delta(G_w(t), n_i, n_i) \neq \infty$  のとき、 $\Delta(G_w(t), n_i, n_i)$  がデッドロックの規模となる。

迷路探索では Demand-for graph の隣接行列  $\Phi(t)$  の時間変化式 (6) について  $\mathbf{u}_1(t)$  を

$$\begin{aligned} \mathbf{u}_1(t) &= \begin{bmatrix} \mathbf{U}_1(t, 1) \\ \vdots \\ \mathbf{U}_1(t, I) \end{bmatrix}, \\ \mathbf{U}_1(t, i) &= \begin{bmatrix} f_7(t, i, 1) & 0 & \cdots & 0 \\ 0 & f_7(t, i, 2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & f_7(t, i, I) \end{bmatrix}, \\ f_7(t, i, j) &= \begin{cases} 1 & \text{(if } (p(t, i) = j)) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (56)$$

と、 $\mathbf{u}_2(t)$  を

$$\begin{aligned} \mathbf{u}_2(t) &= \begin{bmatrix} \mathbf{U}_2(t, 1) \\ \vdots \\ \mathbf{U}_2(t, I) \end{bmatrix}, \\ \mathbf{U}_2(t, i) &= \begin{bmatrix} f_3(t, i, 1)f_8(t, 1) & 0 & \cdots & 0 \\ 0 & f_3(t, i, 2)f_8(t, 2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & f_3(t, i, I)f_8(t, I) \end{bmatrix}, \\ f_8(t, i) &= \begin{cases} 1 & \text{(if } (\delta_{\mathcal{A}}(t, i) = \hat{\delta}_{\mathcal{A}}(t, i)) \vee \mu_{\mathcal{A}}(t+1, i) \neq \mu_{\mathcal{A}}(t, i)) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (57)$$

と与える.  $B, D$  はそれぞれ式 (8) と式 (10) で与える. このとき, 式 (39) より式 (56) と式 (34) について

$$f_7(t, i, j) = f_3(t, i, j) \quad (58)$$

が常に真となる. したがって, 式 (56) と式 (34) は同義である. また, 式 (57) の  $f_8(t, i)$  における  $\mu_A(t+1, i) \neq \mu_A(t, i)$  は  $A_i$  の疑似移動を意味する. したがって, 式 (35), (57), (58) より, 式 (6) に示す Demand-for graph は基本形の式 (35) に協調動作である疑似移動による移動要求の達成を導入した式である.

このとき, 式 (56), (57) より, 迷路探索においても式 (11) が成り立つ. したがって,  $\mathbf{u}_1(t) - \mathbf{u}_1(t-1)$  は渋滞の拡大に,  $\mathbf{u}_2(t)$  は渋滞の解消に対応する.

以降では, デッドロックからの回復条件とデッドロックからの回復速度について述べる. 移動要求  $p(t, i)$  は  $\mathcal{A}p(t, i)$  の移動, または疑似移動によって達成される. このとき, デッドロックに陥ったエージェントは移動が不可能なため,  $G_d(t)$  上の閉路の解消には  $\mathcal{A}p(t, i)$  による探索引き継ぎと疑似移動が必要である. したがって,  $r(t, i) = p(t, i)$  となり式 (45) の探索引き継ぎが実行されることで,  $G_d(t+1)$  で  $(n_i, n_j)$  が削除されてデッドロックが解消する.

このとき, 探索引き継ぎと疑似移動によるデッドロック解決について, 定理 1 が成り立つ.

**定理 1.** 継続検知型での規模  $S \in \{1, 2, 3, \dots, \mathcal{I}\}$  のデッドロックからの回復のための役割交換と疑似移動は以下を満たす.

- a. 解決前のデッドロックに対してのみ役割交換と疑似移動を行う.
- b. デッドロックの規模  $S$  に対して, デッドロック発生後  $\lfloor \frac{S}{dh} \rfloor$  ステップ以内に役割交換と疑似移動が開始される.

**証明.** 定理 1 (a) は  $r^i(t)$  と  $(0 \leq s \leq \frac{1}{dh})$  に対して,

$$\neg (r^i(t+s) = j \wedge j \neq i) \vee \left( \exists s, \text{ s.t. } (\Delta(G_w(t), n_j, n_i) \neq \infty) \wedge \left( r^i(t+s') = i, \forall 0 \leq s' \leq s \right) \right) \quad (59)$$

が真となることと同義である.

Algorithm 1 より,  $\hat{N}_w^i(t)$  は  $G_w(t)$  について

$$\neg (n_j \in \hat{N}_w^i(t)) \vee (\Delta(G_w(t), n_j, n_i) \neq \infty) \quad (60)$$

を満たす. 式 (41), (29) より  $G_w(t)$  について

$$\neg (\Delta(G_w(t), n_j, n_i) \leq d \wedge p^j(t) \neq 0) \vee (c_i^j(t) = 1) \quad (61)$$

が真となる. Algorithm 1 より式 (61) 式が成り立つとき,  $\hat{N}_w^i(t)$  は

$$\neg (\Delta(G_w(t), j, i) \leq d) \vee (n_j \in \hat{N}_w^i(t)) \quad (62)$$

を満たす.

式 (47), (52) および式 (62) 式より式 (50) が成り立つとき,  $P_{h+1}^i(t)$  は

$$\neg \left( \left( \mu_B^j(t) \neq 0, \forall n_j \in P_{h+1}^i(t) \right) \wedge n_i \in P_{h+1}^i(t) \right) \vee (\Delta(G_w(t), n_i, n_i) \neq \infty) \quad (63)$$

を満たす. 式 (38), (47), (52), 式 (62) と式 (63) 式より式 (59) 式が成り立つ.

定理 1 (b) は  $r^i(t)$  と  $s$  に対して

$$\neg(\Delta(G_w(t), n_i, n_i) \neq \infty) \vee (\exists s, s.t. n_i \in P_{h+1}^i(t+s)) \quad (64)$$

が真となることと同義である。式 (41) と式 (61) 式より式 (50) が成り立つとき、 $P_{h'+1}^i(t)$  は  $\hat{G}_w(t)$  について

$$\neg(\Delta(G_w(t), n_j, n_i) \leq d) \vee (P_{h'}^j(t) \subseteq P_{h'+1}^i(t)) \quad (65)$$

を満たす。式 (65) 式より式 (52) が成り立つならば、 $\hat{P}^i(t+1)$  は

$$\neg(\delta_{\mathcal{A}}(t, i) = \varepsilon_{\mathcal{A}}(t, i) \wedge r(t, i) = i) \wedge (\Delta(G_w(t), n_i, n_j) \leq d(h+1)) \vee (P_1^j(t) \subseteq \hat{P}^i(t+1)) \quad (66)$$

を満たす。式 (66) 式を  $s(0 \leq s \leq \lfloor \frac{S}{dh} \rfloor)$  を用いて  $\hat{P}^i(t+s+1)$  の式に拡張すると

$$\neg(\delta_{\mathcal{A}}(t+s', i) = \varepsilon_{\mathcal{A}}(t, i) \wedge r(t+s', i) = i, \forall 0 \leq s' \leq s) \wedge (\Delta(G_w(t), n_i, n_j) \leq ds(h+1)) \vee (P_1^j(t) \subseteq \hat{P}^i(t+s+1)) \quad (67)$$

となる。

また、式 (47), (49), (62), (67) 式より式 (64) 式が成り立つ。  $\square$

定理 1 より、発生したデッドロックのすべてが解決可能であることがわかる。

## 5.4 渋滞解消速度

迷路探索問題の固有課題として、デッドロック解決におけるデッドロック検知と疑似移動の所要時間に基づく渋滞解消速度評価に取り組む。デッドロックの発生から検知までの所要時間と疑似移動の所要時間を、それぞれ  $\theta_d \in \mathbb{N}$  と  $\theta_i \in \mathbb{N}$  で表す。渋滞解消速度の評価では、Demand-for graph  $G_d(t)$  上のノード間の距離の最大値を渋滞規模として扱う。時刻  $t$  における渋滞規模を  $l$  と表す。迷路探索では、単位時刻あたりの  $l$  の減少量の推測値を渋滞解消速度として扱う。渋滞解消速度を  $\sigma_r$  ( $0 \leq \sigma_r$ ) と表す。

定理 1 より、規模  $S$  のデッドロックの発生から検知までの最大所要時間は  $\lfloor \frac{S}{dh} \rfloor$  ステップである。したがって、 $\theta_d$  は  $S$  に対して、

$$\theta_d \leq \left\lfloor \frac{S}{dh} \right\rfloor \quad (68)$$

を満たす。ここで、疑似移動の所要時間  $\theta_i$  を固定値とすると、渋滞発生から解消までの所要時間  $\theta_d + \theta_i$  は

$$\theta_d + \theta_i \leq \left\lfloor \frac{S}{dh} \right\rfloor + \theta_i \quad (69)$$

を満たす。

式 (44), (45), (57) より、デッドロック解消による  $l$  の減少量はデッドロックの規模  $S$  と同値となる。このとき、式 (69) より渋滞解消速度  $\sigma_r$  は  $S$  に対して、

$$\sigma_r \geq \frac{S}{\theta_d + \theta_i} \geq \frac{dhS}{S + dh(\theta_i + 1)} \quad (70)$$

を満たす。式 (70) より、 $d, h, \theta_i$  が固定値であるため  $\sigma_r$  の値は  $S$  が大きくなるほど増加し  $dh$  に近づく。したがって、 $\sigma_r$  の値は最小規模のデッドロック、すなわち  $S = 2$  のときに最小となる。このとき、式 (68) より  $\theta_d = 1$  となるため、 $\sigma_r$  は

$$\sigma_r = \frac{2}{1 + \theta_i} \quad (71)$$

となる.

ここで、協調動作による渋滞緩和効果の比較対象として、エージェントの非協調動作である移動による渋滞の自然解消に着目する. デッドロックが発生していない場合、疑似移動と役割交換が実施されなため、式 (57) の渋滞長の減少は区画移動に対してのみ発生する. このとき、式 (57) よりエージェントの移動確率の制限により渋滞の解消速度はエージェントの区画移動の実施確率と一致するため

$$\sigma_r \leq 1 \quad (72)$$

を満たす. 式 (71), (72) より  $\theta_i$  が

$$\theta_i \leq 1 \quad (73)$$

を満たす場合、疑似移動による渋滞解消速度は非協調動作による渋滞の自然解消速度を上回る. したがって、式 (71), (72) は疑似移動の所要時間が通常の区画移動の所要時間を下回る場合、本来は流量低下の原因であるデッドロックの発生とその解消動作が流量増加に貢献することを示している.

## 5.5 数値実験

協調動作による既存渋滞の解消について、定理 1 に示す提案する疑似移動を用いたデッドロック解決の性質を数値実験により検証する.

迷路探索では先行研究 [19] と同様に流量の評価指標に区画からエージェントが移動する確率であるエージェント流量と、エージェントが区画移動または探索ユニットの交換を行う確率であるユニット流量を用いる. 探索終了までのエージェント流量とユニット流量の時間平均を、それぞれ  $Q_A$ ,  $Q_U$  と表す. 本実験では、探索期間  $T \in \mathbb{N}$  に対して  $Q_A$  と  $Q_U$  を

$$Q_A = \frac{1}{T|V|} \sum_{t=1}^T \sum_{i=1}^I f_9(t, i), \quad (74)$$

$$f_9(t, i) = \begin{cases} 1 & \text{if } (\delta_A(t, i) \neq \varepsilon_A(t, i)) \\ 0 & \text{otherwise} \end{cases} \quad (75)$$

$$Q_U = Q_A + \frac{1}{T|V|} \sum_{t=1}^T \sum_{i=1}^I f_{10}(t, i), \quad (76)$$

$$f_{10}(t, i) = \begin{cases} 1 & \text{if } (\mu_A^i(t+1) \neq \mu_A^i(t) \wedge \mu_A^i(t+1) \neq 0) \\ 0 & \text{otherwise} \end{cases} \quad (77)$$

と求める.

探索対象の迷路を図 16 に示す. 探索対象は  $10 \times 10$  の二次元迷路である. 図 16 の探索対象は、行き止まりが少ないため大規模なデッドロックが発生しやすいという特徴を持つ.

また、本実験では  $A_i$  と  $A_j$  の通信状態  $c_j^i(t)$  を通信保証範囲  $d$  に対して

$$c_j^i(t) = 1 \iff \Delta(N, \varepsilon_A(t, i), \varepsilon_A(t, j))(t, i, j) \leq d \quad (78)$$

と与える. 式 (78) は、エージェント間のマンハッタン距離が  $d$  以下であれば通信可能であることを意味する.



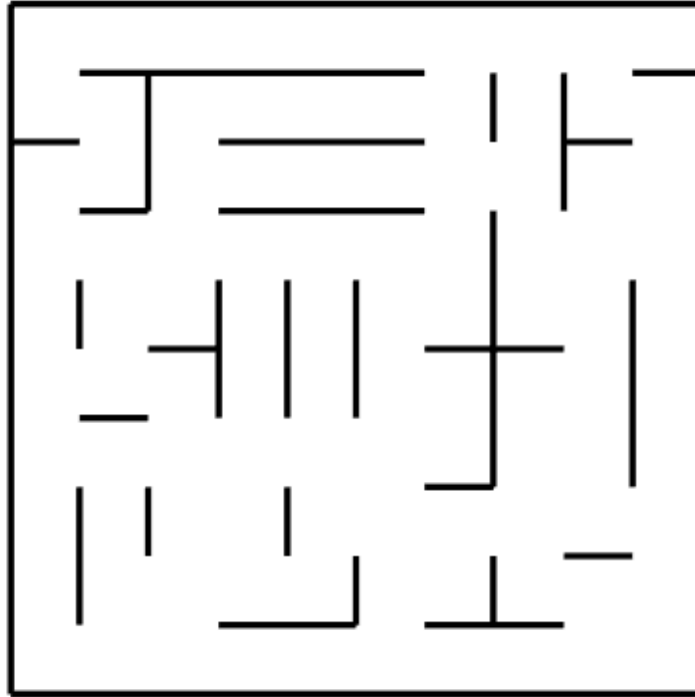


図 16: 探索対象 (迷路探索)

本実験では、探索期間を  $T = 2500$  に設定し、エージェント数を  $1, 10, 20, \dots, 80$  と変化させる。通信保証範囲を  $d = 18$ 、中継数を  $h = 5$  に設定する。図 16 よりエージェント間のマンハッタン距離は最大 18 であるため、本実験では全てのエージェントに対して常に通信による情報共有が保証される。また、 $\theta_i = 0$  とし、疑似移動にかかる時間は考慮しない。各エージェントはあらかじめ与えられた経路に沿って迷路を 1 巡することをミッションに持つ。移動経路はエージェントの初期位置と迷路上の障害物の位置を考慮したものであり、障害物回避のほかに移動方向の制約を持たない。また、エージェントの移動経路は式 (47) の条件を満たす。全エージェントがミッションを達成した場合、その時刻を探索完了時刻とし探索を終了する。

移動経路はエージェントの初期位置と迷路上の障害物の位置を考慮したものであり、障害物回避のほかに移動方向の制約を持たない。また、エージェントは式 (47) の条件を満たす。全エージェントがミッションを達成した場合、その時刻を探索完了時刻とし探索を終了する。

Demand-for graph に基づく渋滞解析の結果が定理 1 より発生した全てのデッドロックの解決が保証している。そのため、エージェント数に関わらず全エージェントの探索完了が予想される。

エージェント数に対する探索完了時刻を表 2 に示す。表 2 では、Demand-for graph に基づく渋滞解析の結果の通りエージェント数に関わらず 2500 ステップ以内に探索が完了していることが確認できる。

表 2: 探索完了時刻 ( $d = 18, h = 5$ )

Agent num				
1	20	40	60	80
261	351	658	1165	1811

次に、通信保障範囲を  $d = 2$ 、中継数を  $h = 5$  に設定し、デッドロック検知の所要時間  $\theta_d$  が流量に与える影響を検証する。このとき、式 (69) および  $\theta_i = 0$  から、デッドロック解決のための疑似移動による流量の増加が期待できる。

平均流量を図 17 に示す。

図 17 よりエージェント数 70 以上の探索においても流量の低下が抑えられており、過去情報の利用による継続的なデッドロック検知が流量確保に有効であることが確認できた。

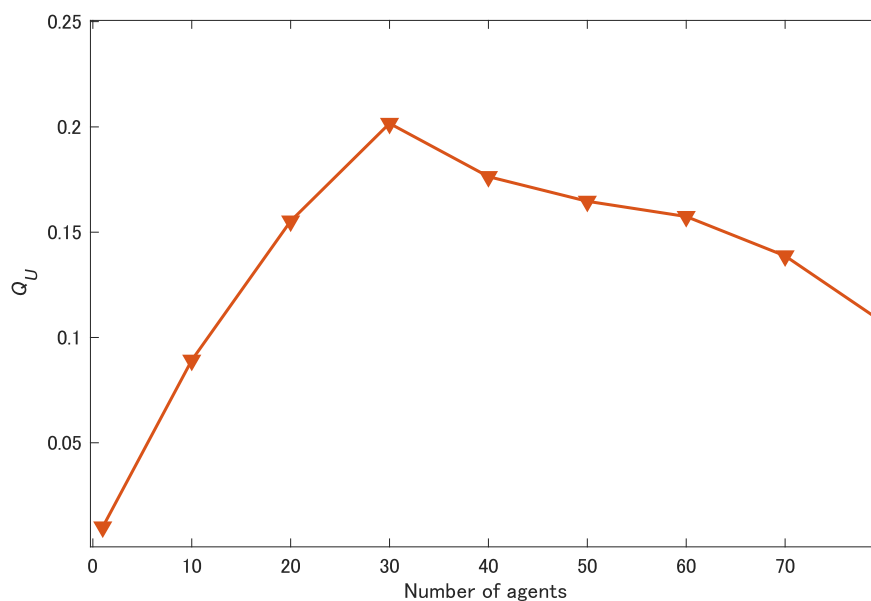


図 17: 実験結果 ( $Q_u$ )

## 6 搬送問題

搬送問題では  $I$  台のエージェントによる, ASEP の直列ネットワーク, および二次元平面の探索を通じて, 通信を要さない非協調機能による渋滞の自然解消を評価する. これにより, 迷路探索で提案した疑似移動を含む協調動作による既存渋滞の解消について, 非協調動作による自然渋滞との渋滞解消速度の比較を可能にする.

自動倉庫内の搬送問題の研究課題として, 搬送効率の最大化を目的とした搬送路の構成と導入するロボット数の設計, および経路計画とスケジューリングによる搬送計画が挙げられる. 既存研究の多くでは搬送人の定常状態における移動効率や, 搬送開始から終了までの平均移動効率により搬送効率を評価している. これに対し, 本博士論文ではエージェントの非協調動作である荷積みや荷降ろしのための一時停止に対して生じる渋滞と, エージェントの非協調動作による渋滞の自然解消を評価する.

以降では, エージェントの荷積みと荷降ろしを非協調動作として扱い, 非協調動作に起因する渋滞の発生とその解消を論ずる. 搬送問題では渋滞の発生と解消速度の評価に, インシデント ID と移動要求を用いる. インシデント ID は渋滞の原因となったエージェントの協調・非協調動作とそれによって発生した渋滞を関連付ける識別子である. インシデント ID により, 従来の密度と流量に基づく渋滞評価に加えて渋滞の原因となった動作毎に渋滞規模を評価することが可能になる. 移動要求はエージェントの区画移動に関する待ち状態解消に要する, 他のエージェントへの区画移動の要求である. インシデント ID に基づく移動要求の設定と, 移動要求の Demand-for graph 表現によって荷積みと荷降ろしによって発生する渋滞を識別する.

### 6.1 荷積み・荷降ろしと移動要求

本節では, エージェントの荷積み・荷降ろし動作と, 移動要求の数理モデルを示す. エージェントは移動のほかに, 荷積みと荷降ろしの機能を持つ. 荷積みと荷降ろしの際, エージェントは移動を停止する. 本博士論文ではエージェントの荷積み, 荷降ろしによる一時停止に対して, 固有の識別子としてインシデント ID を設定する. エージェントはインシデント ID に基づき, 一時停止中の前方のエージェントに対して移動要求を設定する.

時刻  $t$  における,  $A_i$  の荷積み, 荷降ろしのための一時停止を  $p(t, i) \in \{0, 1\}$  で表す. 時刻  $t$  に  $A_i$  の荷積み, 荷降ろしのために一時停止を選択する場合,  $p(t, i)$  は 1 となる.

$\xi_{\mathcal{N}}(t, v_y^x)$  は時刻  $t$  における区画  $v_y^x$  のインシデント ID を表す.  $\xi_{\mathcal{N}}(t, v_y^x) = 0$  は時刻  $t$  において区画  $v_y^x$  が荷積み・荷降ろしに起因する渋滞の範囲外であることを意味する.  $\xi_{\mathcal{A}}(t, i)$  は時刻  $t$  における  $A_i$  のインシデント ID を表す.  $\xi_{\mathcal{A}}(t, i) = 0$  は時刻  $t$  において  $A_i$  が荷積み・荷降ろしに起因する渋滞の範囲外に存在することを意味する.  $\xi_{\mathcal{N}}(t, v_y^x)$  を  $\xi_{\mathcal{A}}(t, i)$  に対して,

$$\xi_{\mathcal{N}}(t, v_y^x) = \xi_{\mathcal{A}}(t, \varepsilon_{\mathcal{N}}(t, v_y^x)) \quad (79)$$

と与える. このとき,  $\xi_{\mathcal{A}}(t, i)$  は

$$\xi_{\mathcal{A}}(t, 0) = 0 \quad (80)$$

を満たす. インシデント ID  $n$  の渋滞に含まれる区画群を  $V_{t,n}(t) \subseteq V$  と表す.  $V_t(t, n)$  は  $\xi_{\mathcal{N}}(t, v_y^x)$  に対して

$$v_y^x \in V_t(t, n) \iff \xi_{\mathcal{N}}(t, v_y^x) = n \quad (81)$$

を満たす.

時刻  $t$  において,  $A_i$  が移動を要求するエージェントの ID を  $r(t, i) \in \{0\} \cup \mathcal{I}$  で表す. 時刻  $t$  において,  $A_i$  が他のエージェントに移動を要求しない場合は  $r(t, i) = 0$  となる. 以降では  $r(t, i)$  を時刻  $t$  における  $A_i$  の移動要求と表す. 本博士論文では  $r(t, i)$  をエージェントのインシデント ID  $\xi_{\mathcal{A}}(t, i)$  に対して

$$r(t, i) = \begin{cases} \varepsilon_{\mathcal{A}}(t, \hat{\delta}_{\mathcal{A}}(t, i)) & \left( \text{if } \xi_{\mathcal{N}}(t, \hat{\delta}_{\mathcal{A}}(t, i)) \neq 0 \right) \\ 0 & \text{(otherwise)} \end{cases} \quad (82)$$

と与える. 式 (79)-(82) よりエージェントは自身の目標移動先にインシデント ID を持つエージェントが存在している場合に, 該当のエージェントに対して移動を要求する.

本博士論文ではエージェントの荷積み, 荷降ろし, 移動に基づき, 渋滞の発生, 拡大, 分離とそれに伴うインシデント ID の設定を以下のように与える.

$p(t, i) = 1$  は時刻  $t$  に  $A_i$  が荷積み, 荷降ろしを行い, 渋滞が発生したことを意味する. このとき, インシデント ID  $\xi_{\mathcal{A}}(t, i)$  と  $\xi_{\mathcal{N}}(t, \varepsilon_{\mathcal{A}}(t, i))$  に  $n \in \{1, 2, \dots, \mathcal{I}\}$  を設定する. 次に, 渋滞の拡大を説明する. 時刻  $t$  における  $A_i$  の移動  $\hat{\delta}_{\mathcal{A}}(t, i)$  が

$$\xi_{\mathcal{N}}(t, \hat{\delta}_{\mathcal{A}}(t, i)) \neq 0 \quad (83)$$

を満たすとき,  $A_i$  は渋滞に合流する. このとき, インシデント ID  $\xi_{\mathcal{A}}(t, i)$  と  $\xi_{\mathcal{N}}(t, \varepsilon_{\mathcal{A}}(t, i))$  に  $\xi_{\mathcal{N}}(t, \hat{\delta}_{\mathcal{A}}(t, i))$  を設定する. 最後に, エージェントの渋滞からの分離について説明する.  $\xi_{\mathcal{A}}(t, i) \neq 0$  を満たす  $A_i$  の移動が

$$\varepsilon_{\mathcal{A}}(t+1, i) \neq \varepsilon_{\mathcal{A}}(t, i) \quad (84)$$

を満たす場合,  $A_i$  が区画移動により渋滞から分離したものとして扱う. このとき, インシデント ID  $\xi_{\mathcal{A}}(t, i)$  と  $\xi_{\mathcal{N}}(t, \varepsilon_{\mathcal{A}}(t, i))$  を 0 に初期化する.

式 (83), (84) より,  $\xi_{\mathcal{A}}(t, i)$  と  $\xi_{\mathcal{N}}(t, v_y^x)$  の設定方法を Algorithm 3 と与える.

---

**Algorithm 3** Set  $\xi_{\mathcal{A}}(t+1, i)$  and  $\xi_{\mathcal{N}}(t+1, v_y^x)$

---

```

1: if ( $p(t, i) = 1$ ) then
2:    $\xi_{\mathcal{A}}(t+1, i) \leftarrow i$ 
3: else
4:   if ( $\xi_{\mathcal{A}}(t, i) \neq 0$ ) then
5:     if ( $\varepsilon_{\mathcal{A}}(t+1, i) \neq \varepsilon_{\mathcal{A}}(t, i)$ ) then
6:        $\xi_{\mathcal{A}}(t+1, i) \leftarrow 0$ 
7:     else
8:        $\xi_{\mathcal{A}}(t+1, i) \leftarrow \xi_{\mathcal{A}}(t, i)$ 
9:     end if
10:  else
11:    if ( $\xi_{\mathcal{N}}(t, \hat{\delta}_{\mathcal{A}}(t, i)) \neq 0$ ) then
12:       $\xi_{\mathcal{A}}(t+1, i) \leftarrow \xi_{\mathcal{N}}(t, \hat{\delta}_{\mathcal{A}}(t, i))$ 
13:    end if
14:  end if
15: end if
16:  $\xi_{\mathcal{N}}(t+1, \varepsilon_{\mathcal{A}}(t, i)) \leftarrow \xi_{\mathcal{A}}(t+1, i)$ 

```

---

Algorithm 3 の 2 行目, 6 行目, 12 行目はそれぞれ, 渋滞の発生, 分離, 合流に対するインシデント ID の設定を表す. このとき, Algorithm 3 より  $\xi_{\mathcal{N}}(t, v_y^x)$  について

$$\neg \xi_{\mathcal{N}}(t, v_y^x \neq 0) \vee \varepsilon_{\mathcal{N}}(t, v_y^x) \neq 0 \quad (85)$$

が真となる.

## 6.2 渋滞の Demand-for graph 表現

搬送問題ではエージェントの移動要求に基づき渋滞規模を評価する. 搬送問題における Demand-for graph  $G_d(t)$  を

$$\begin{aligned} A_d(t) &= \sum_{n_i \in N_d} \sum_{n_j \in N_d/n_i} f_{11}(t, i, j) \\ f_{11}(t, i, j) &= \begin{cases} (n_i, n_j) & \text{if } (r(t, i) = j) \\ \emptyset & \text{otherwise} \end{cases} \end{aligned} \quad (86)$$

と与える. このとき, 式 (82), Algorithm 3 より,  $r(t, i)$  について

$$\neg r(t, i) = j \neq 0 \vee \xi_{\mathcal{A}}(t, j) \quad (87)$$

が真となる. 式 (86), (87) より,  $(n_i, n_j) \in A_d(t)$  は時刻  $t$  においてインシデント  $\text{ID}_{\xi_{\mathcal{A}}}(t, j) \neq 0$  に対応する渋滞に含まれる  $\mathcal{A}_j$  によって  $\mathcal{A}_i$  の移動が阻害されることを意味する.

以上から, 搬送問題の荷積み・荷降ろしに起因する渋滞規模を移動要求に基づく Demand-for graph  $G_d(t)$  で評価する.

ここで, 荷積み・荷降ろしで生じる渋滞規模の評価項目として,  $G_d(t)$  の構造に基づき渋滞の長さと同幅を定義する.

本博士論文では渋滞規模を Demand-for graph  $G_d(t)$  の出端点と入端点に基づき定義する. 以降では,  $G_d(t)$  の出端点に対応するエージェントとその存在区画を渋滞の先頭と表す. また, 入端点に対応するエージェントとその存在区画を渋滞の最後尾と表す.  $V^h(t) \in V(t)$  と  $V^t(t) \in V(t)$  は, それぞれ渋滞の先頭と最後尾に位置する区画の集合である. このとき,  $V^h(t)$  と  $V^t(t)$  は

$$v_y^x \in V^h(t) \iff d_{out}(v_y^x) = 0 \quad (88)$$

と

$$v_y^x \in V^t(t) \iff d_{in}(v_y^x) = 0 \quad (89)$$

を満たす.

$G_d(t)$  における渋滞の先頭と最後尾の平均距離を  $l(t) \in \mathbb{N}$  と表す. このとき,  $l(t)$  は時刻  $t$  における渋滞の長さを表す. 式 (86) より, 渋滞の長さ  $l(t)$  は渋滞最後尾のエージェントが渋滞から分離するために達成を要する移動要求の平均値を意味する.  $l_n(t)$  を

$$l_n(t) = \frac{1}{|V^t(t)|} \sum_{v_y^x \in V^t(t)} f_3(t, v_y^x) \quad (90)$$

と与える. 式 (90) において  $f_3(t, v_y^x)$  は渋滞の最後尾の区画  $v_y^x$  から最も近い渋滞の先頭区画までの距離を表す. Algorithm 3 と 式 (88)-(90) より, 渋滞の最後尾の区画  $v_y^{x'}$  が

$$\exists v_y^x \in N \text{ s.t. } \delta(v_y^x, v_y^{x'}, t) = 1 \wedge \xi(v_y^x, t) = 0. \quad (91)$$

を満たすとき、 $l(t)$ が増加する。また、Algorithm 3と式(88)-(90)より、渋滞の先頭区画 $v_y^x$ が

$$\exists v_y^x \in N \text{ s.t. } \delta(t, v_y^x) = 1. \quad (92)$$

を満たすとき、 $l(t)$ が減少する。

$G_d(t)$ における出端点と入端点の数を、それぞれ $w^r(t) \in \mathbb{N}$ ,  $w^e(t) \in \mathbb{N}$ と表す。以降では、 $w^r(t)$ と $w^e(t)$ を、それぞれ時刻 $t$ における渋滞の解消幅、拡大幅と表す。渋滞の解消幅 $w^r(t)$ は渋滞から分離可能な、渋滞先頭のエージェントの数を意味する。渋滞の拡大幅 $w^e(t)$ は渋滞への合流が発生する、渋滞最後尾のエージェントの数を意味する。 $w^r(t)$ と $w^e(t)$ をそれぞれ、

$$w^r(t) = |V^h(t)|, \quad (93)$$

と

$$w^e(t) = |V^t(t)|. \quad (94)$$

で与える。

渋滞の長さは渋滞解消の所要時間に、渋滞の幅は長さに対する流量の低下量に影響する。

図18に二次元平面上の、荷積み・荷降ろしに起因する渋滞例を示す。図18の灰色の丸( $a_1 - a_7$ )は荷積み・荷降ろしに起因する渋滞内に位置するエージェントを表す。対して、白色の丸( $a_8 - a_{10}$ )は渋滞外に位置するエージェントを表す。図中の矢印は渋滞内に位置するエージェントの目標移動先区画を表す。図18の渋滞の先頭は $a_1$ と $a_2$ であり、渋滞の最後尾は $a_5 - a_7$ である。図18の例において、渋滞の長さ、拡大幅、解消幅は、それぞれ $l(t) = 3$ ,  $w^e(t) = 3$ ,  $w^r(t) = 2$ である。

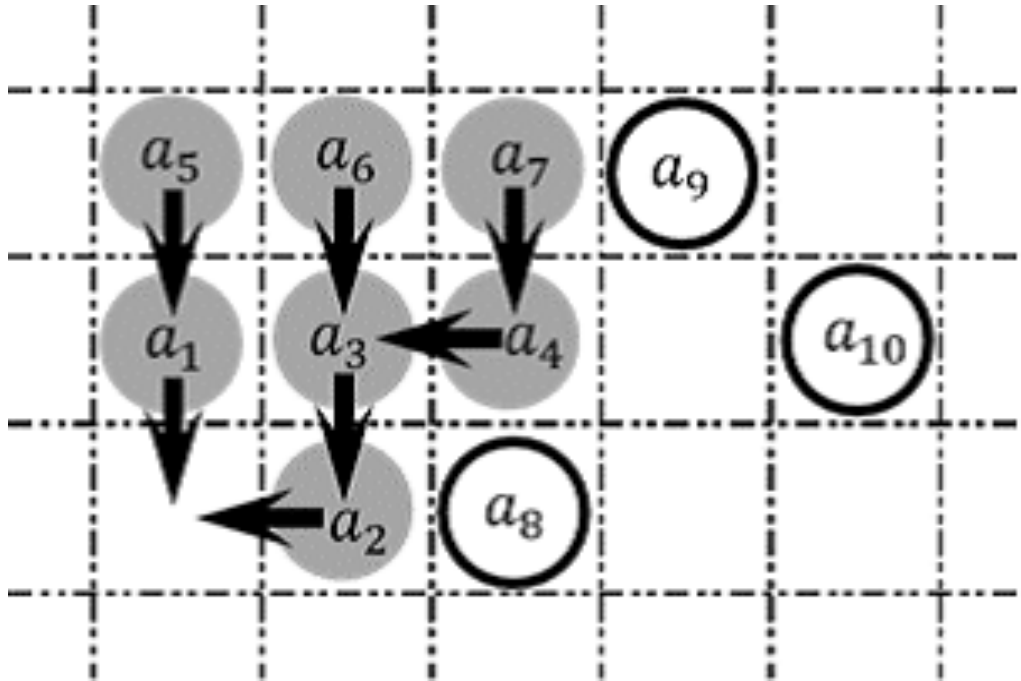


図18: 渋滞規模

搬送問題では Demand-for graph の隣接行列  $\Phi(t)$  の時間変化式 (6) について  $u_1(t)$  を

$$\begin{aligned}
\mathbf{u}_1(t) &= \begin{bmatrix} \mathbf{U}_1(t, 1) \\ \vdots \\ \mathbf{U}_1(t, I) \end{bmatrix}, & (95) \\
\mathbf{U}_1(t, i) &= \begin{bmatrix} f_{12}(t, i, 1) & 0 & \cdots & 0 \\ 0 & f_{12}(t, i, 2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & f_{12}(t, i, I) \end{bmatrix}, \\
f_{12}(t, i, j) &= \begin{cases} 1 & \left( \text{if } \left( \delta_{\mathcal{A}}(t, i) = \hat{\delta}_{\mathcal{A}}(t, i) \wedge \xi_{\mathcal{A}}(t, i) \neq 0 \right) \right) \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

と,  $\mathbf{u}_2(t)$  を

$$\begin{aligned}
\mathbf{u}_2(t) &= \begin{bmatrix} \mathbf{U}_2(t, 1) \\ \vdots \\ \mathbf{U}_2(t, I) \end{bmatrix}, & (96) \\
\mathbf{U}_2(t, i) &= \begin{bmatrix} f_{12}(t, i, 1)f_4(t, 1) & 0 & \cdots & 0 \\ 0 & f_{12}(t, i, 2)f_4(t, 2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & f_{12}(t, i, I)f_4(t, I) \end{bmatrix}
\end{aligned}$$

と与える.  $\mathbf{B}$ ,  $\mathbf{D}$  はそれぞれ式 (8) と式 (10) で与える. このとき, 式 (39) より式 (95) と式 (34) について

$$f_{12}(t, i, j) = \begin{cases} f_3(t, i, j) & \text{(if } \xi_{\mathcal{A}}(t, i) \neq 0) \\ 0 & \text{otherwise} \end{cases} \quad (97)$$

が成り立つ. したがって, 式 (95) と式 (96) から求める渋滞を表す Demand-for graph は, 基本形の Demand-for graph の式 (34) と式 (35) にインシデント ID による観測区間の制限を設けた部分グラフである.

このとき, 式 (95), (96) より, グラフ探索においても式 (11) が成り立つ. したがって,  $\mathbf{u}_1(t) - \mathbf{u}_1(t-1)$  は渋滞の拡大に,  $\mathbf{u}_2(t)$  は渋滞の解消に対応する.

### 6.3 渋滞の解消条件

本章では, 荷積み・荷降ろしに起因する渋滞の拡大・解消速度, および渋滞解消条件の導出を示す. 以降では, 移動要求の Demand-for graph 表現  $G_d(t)$  に対して, 渋滞の長さや幅の拡大速度と解消速度を導出する.

単位時刻あたりの渋滞の最後尾におけるエージェントが合流による渋滞長増加数の期待値を, 渋滞長の拡大速度として扱う.  $\sigma_{el}$  ( $0 \leq \sigma_{el}$ ) は時刻  $t$  における渋滞長の拡大速度を表す. 同様に, 単位時刻あたりの渋滞先頭に位置するエージェントの渋滞長減少数の期待値を, 渋滞長の解消速度として扱う.  $\sigma_{rl}$  ( $0 \leq \sigma_{rl} \leq 1$ ) は時刻  $t$  における渋滞長の解消速度を表す. 本博士論文では,  $\sigma_{el}$  と  $\sigma_{rl}$  を, エージェントの移動確率  $p$  と密度  $\rho$  に基づき導出する. 移動確率  $p$  と密度  $\rho$  はどちらも, ASEP およびそのネットワーク上での渋滞解析で用いられる指標である [46–48].

渋滞幅の評価では, 渋滞の拡大幅と解消幅の増加数の予測値から渋滞の拡大・解消速度を評価する. 単位時刻あたりの渋滞拡大幅の増加数の予測値を拡大速度として扱う.  $\sigma_{ew}$  ( $0 \leq \sigma_{ew}$ ) は時刻  $t$  にお

る渋滞幅の拡大速度を表す。本博士論文では、 $\sigma_{ew}$  を渋滞長の増加速度  $\sigma_{el}$  と渋滞発生からの経過時間に基づき導出する。同様に、単位時刻あたりの渋滞解消幅の増加数の予測値を、渋滞幅の解消速度として扱う。 $\sigma_{rw}$  ( $0 \leq \sigma_{rw}$ ) は時刻  $t$  における渋滞幅の解消速度を表す。本博士論文では、 $\sigma_{rw}$  を渋滞長の増加速度  $\sigma_{rl}$  とエージェントの渋滞からの分離開始からの経過時間に基づき導出する。 $\sigma_{ew}$  と  $\sigma_{rw}$  は、探索対象のグラフ構造に影響を受ける。

以降では、ASEP の直列ネットワークと二次元平面の搬送問題に関して、それぞれ渋滞の拡大・解消速度を導出する。

### 6.3.1 ASEP ネットワーク

単路の ASEP では、 $\sigma_{ew}$  と  $\sigma_{rw}$  はどちらも 1 に固定される。

本博士論文では Algorithm 3 に基づく渋滞長の拡大を評価するために、新たに潜在的渋滞区域の概念を導入する。潜在的渋滞区域は渋滞最後尾のエージェントの存在区画に連続する区画のうち、インシデント ID が 0 に設定されているエージェントが存在する区画群である。潜在的渋滞区域の具体例を図 19 に示す。図 19 において、灰色の丸は渋滞内に位置するエージェントを表す。同様に、白色の丸は渋滞外に位置するエージェントを表す。図 19 の例では、渋滞最後尾に連続して位置する、2 台の白色の丸で表すエージェントの存在区画が潜在的渋滞区域に該当する。以降では、潜在的渋滞区域外における渋滞長の拡大速度を導出する。

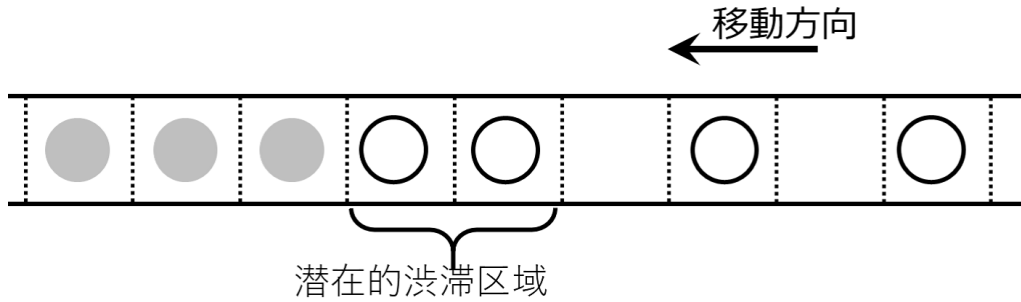


図 19: 潜在的渋滞区域

式 (91), (95) より、インシデント ID が  $\xi_A(t, i) \neq 0$  を満たすエージェントとの区画移動の干渉により渋滞長が拡大する。潜在的渋滞区域の外では、渋滞最後尾の隣接区画に存在するエージェントのみ式 (91) すなわち  $\xi_A(t, i) \neq 0$  を達成可能である。そのため、ASEP における渋滞長の拡大速度  $\sigma_{el}$  は

$$\sigma_{el} = p\rho \quad (98)$$

と導ける。式 (96) より、エージェントの区画移動によって渋滞長が減少する。このとき、Algorithm 3 の 8 行目より、渋滞の先頭区画  $v_y^x$  は常に  $\varepsilon_N(t, v_y^x) \neq 0$  を満たす。すなわち、渋滞の先頭区画には必ずエージェントが存在する。そのため、SEP における渋滞長の解消速度  $\sigma_{rl}$  は

$$\sigma_{rl} = p \quad (99)$$

と導ける。

ASEP の直列ネットワーク  $N = (V, E)$  において、渋滞長の拡大速度はエージェントの密度に依存する。以降では、 $N$  上で高密度相と低密度相の相分離が発生した場合を想定する。このとき、高密度相と



低密度相では渋滞の拡大速度が異なる．高密度相におけるエージェントの密度を  $\rho_h$  と表すと，高密度相における  $\sigma_{el}$  は

$$\sigma_{el} = p\rho_h \quad (100)$$

となる．低密度相におけるエージェントの密度を  $\rho_l$  と表すと，低密度相における  $\sigma_{el}$  は

$$\sigma_{el} = p\rho_l. \quad (101)$$

となる．このとき，ASEP 上の高密度相の割合を  $\gamma(0 < \gamma \leq 1)$  とするとネットワーク  $N$  全体における  $\sigma_{el}$  は  $\gamma p\rho_h + (1 - \gamma)p\rho_l$  となる．したがって，式 (100)，(101) より， $\sigma_{el}$  は式 (98) と同様に  $\rho$  となる．渋滞の解消速度は，単路の ASEP の結果式 (99) と変わらず  $\sigma_{rl}$  となる．

式 (88)，(89)，(93)，(94) より，渋滞の拡大幅と解消幅の増加は ASEP ネットワークのジャンクション区画で発生する．したがって， $\sigma_{ew}$  と  $\sigma_{rw}$  は  $N$  の平均次数に依存する．以上から， $N$  における  $\sigma_{ew}$  を，式 (98)，移動確率  $p$ ，エージェントの密度  $\rho$ ， $N$  の平均次数  $c$ ，ASEP 長  $L$  に対して

$$l\sigma_{ew} = f_8 \sigma_{el} c^{f_{13}\sigma_{el}(t_e-1)} \log c, \quad (102)$$

$$f_8 = \frac{1}{L+1}.$$

と求める．式 (103) において， $t_e \in \{0, 1, \dots, \infty\}$  は渋滞発生からの経過時間を表す．同様に， $\sigma_{ew}$  を

$$\sigma_{rw} = f_8 \sigma_{rl} c^{f_8 \sigma_{rl}(t_r-1)} \log c. \quad (103)$$

と求める．式 (103) において， $t_r \in \{0, 1, \dots, \infty\}$  は渋滞の解消開始からの経過時間を表す．

渋滞長の拡大速度と解消速度の関係から，渋滞の解消条件を導出する．渋滞長の解消速度  $\sigma_{rl}$  が拡大速度  $\sigma_{el}$  を上回る場合，渋滞長の減少により荷積み・荷降ろしの一時停止で発生した流量低下が解消される．式 (98)，(99)，(100)，(101)，および密度に対する上限  $\rho \leq 1$  より，ASEP ネットワーク上の  $\sigma_{el}$  と  $\sigma_{rl}$  について

$$\sigma_{el} \leq \sigma_{rl}. \quad (104)$$

が真となる．式 (104) より ASEP ネットワークの搬送問題において，荷積み・荷降ろしで生じる渋滞は非協調動作による自然解消が保証されると言える．

ただし，エージェントの長期間停止によって渋滞の長さが ASEP 長  $L$  を超過した場合，渋滞幅の拡大による大幅な流量低下やデッドロックが生じる可能性がある．より，エージェントの分離開始後の渋滞長の減少速度は

$$\sigma_{rl} - \sigma_{el} = (1 - \rho)p \quad (105)$$

となる．式 (105) より，密度  $\rho$  が大きい場合にエージェントの長時間停止による渋滞幅拡大のリスクが大きくなる．

以降では，荷積み・荷降ろしによる一時停止時間に対する，渋滞幅の拡大予想を記す．荷積み・荷降ろしで生じるエージェントの一時停止の期間を  $t_p \in \mathbb{N}$  と表す．このとき，渋滞の先頭エージェントが渋滞から分離する時点での渋滞長の予想値は

$$t_p \sigma_{el} = t_p p \rho \quad (106)$$

となる．式 (105)，(106) より，渋滞長が 0 になるまでの所要時間の予想値は

$$\frac{t_p p \rho}{(1 - \rho)p} \quad (107)$$

と求まる。渋滞最後尾が ASEP 始端に到達すると、渋滞幅が拡大する。したがって、式 (107) までの間に渋滞長の拡大値が ASEP 長を超える場合、渋滞の拡大幅が上昇する。以上から、ASEP 長  $L$  に対して  $t_p$  が

$$\left( \frac{t_p p \rho}{(1-\rho)p} + t_p \right) \sigma_{el} = \left( \frac{t_p p \rho}{(1-\rho)p} + t_p \right) p \rho \geq L \quad (108)$$

を満たす場合に渋滞幅の拡大が予想される。式 (108) を式変形すると、

$$t_p \geq \frac{L(1-\rho)}{p\rho^2} \quad (109)$$

となる。

### 6.3.2 二次元平面

二次元平面上の探索では渋滞最後尾  $V_t(t)$  に位置する区画の平均次数が渋滞長の解消速度に影響する。本博士論文では、渋滞はその先頭区画を中心に放射状に拡大すると仮定する。二次元平面における渋滞の拡大例を図 20 に示す。

図 20 において、灰色と黒色の丸は全て渋滞内に位置するエージェントを表す。特に、渋滞最後尾のエージェントを黒色の丸で表す。

二次元平面の探索で、エージェントは自身の移動方向を隣接する 4 区画から選択する。図 20 の灰色の区画において、エージェントは移動確率  $p$  に対して  $p/2$  の確率で渋滞に合流する。黒色の区画において、エージェントは  $p/4$  の確率で渋滞に合流する。したがって、十分に渋滞が拡大した場合、渋滞最後尾の隣接区画に位置するエージェントが渋滞に合流する確率は  $p\rho/2$  となる。式 (25), (91) より、二次元平面における  $\sigma_{el}$  は

$$\sigma_{el} = \sum_{n=1}^{\infty} \left( \frac{p\rho}{2} \right)^n = \frac{p\rho}{2-p\rho}. \quad (110)$$

導ける。

文献 [48] より、ASEP ネットワークではエージェントの流量の上限はジャンクション区画の次数  $c$  によって制限される。このとき、 $N$  上のエージェント密度が  $\rho = 0.5$  となるとき、流量が最大となる。また、二次元平面においては隣接する 4 区画に存在するエージェントが同一区画への移動を試みた場合、1 台のエージェントのみが移動可能となる。以上から、二次元平面における  $\sigma_{rl}$  の上限を、 $N$  の ASEP ネットワークの近似表現に基づき

$$\sigma_{rl} \leq \frac{p}{8}. \quad (111)$$

と求める。

次に、渋滞幅に対する拡大速度と解消速度を導出する。二次元平面の ASEP 直列ネットワークの近似表現において、ASEP 長は  $L = 0$  である。図 20 は渋滞最後尾のエージェント数が、渋滞発生時の渋滞の先頭区画から渋滞最後尾までの長さの 4 倍になることを示している。したがって、 $\sigma_{ew}$  と  $\sigma_{rw}$  の上限を  $N$  の平均次数と式 (110), (111) に対して

$$\sigma_{ew} \leq 4\sigma_{el} \quad (112)$$

$$\sigma_{rw} \leq 4\sigma_{rl}. \quad (113)$$

と求める。

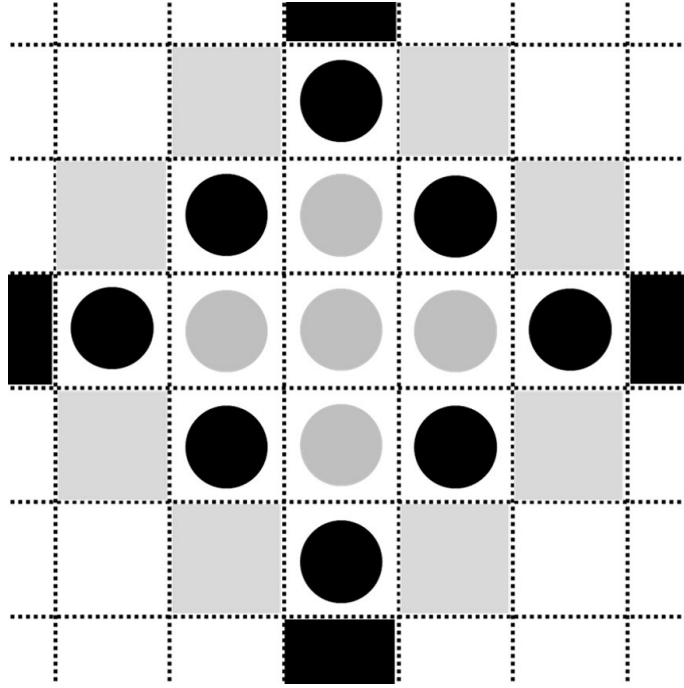


図 20: 渋滞の拡大例

式 (110), (111), (104) より, 二次元平面における渋滞の解消条件を

$$\rho < \frac{2}{p+8}. \quad (114)$$

と求める. 式 (114) より, エージェント密度  $\rho$  が移動確率  $p$  に対して大きくなると荷積み・荷降ろしで生じる渋滞の自然解消が困難になることがわかる.

ここで, 式 (109) の渋滞の拡大幅増加予想を二次元平面の探索に適用する. 二次元平面においては, ASEP 長  $L$  は 0 として扱われる. したがって, 二次元平面の探索では渋滞の拡大幅の増加を阻止することは困難である. また, 二次元平面では ASEP ネットワークと比較して, 渋滞の拡大や分離開始後のエージェントの渋滞への再合流によりデッドロックの発生確率が高くなる. デッドロックが発生した場合, 渋滞の解消速度  $\sigma_{rl}$  と  $\sigma_{rw}$  は共に 0 である. したがって, デッドロックが発生した場合は, 渋滞の自然解消は不可能である. 渋滞の拡大幅の増加予想とデッドロック発生リスクを考慮すると, 二次元平面での搬送ロボットの運用において, 協調的な経路計画および迷路探索問題で提案した疑似移動によるデッドロック解決などの協調機能による既存渋滞の解決が必要であるとわかる.

#### 6.4 協調機能による渋滞緩和への応用

本節では, 搬送問題の固有課題として渋滞緩和のための協調機能の検討と, 協調機能の実現に要する通信機能の要件を示す. 本博士論文では非協調動作である荷積み・荷降ろしで発生した渋滞の緩和の課題として, 式 (109) に示す渋滞の拡大幅増加抑止とデッドロックの解決に取り組む. 以降では, 協調機能実現のための通信について, エージェント間の距離に対する通信制限を考える. 搬送問題においてもエージェント間の距離が通信保証範囲  $d$  以下で, すなわち  $\Delta(N, \varepsilon_A(t, i), \varepsilon_A(t, j)) \leq d$  であれば通信可能であると仮定する.

#### 6.4.1 ASEP ネットワーク

デッドロックの発生条件である循環待ち条件より、デッドロックが発生する場合は渋滞の拡大幅が 2 以上である必要がある。したがって、ASEP ネットワークでは、式 (109) の渋滞の拡大幅の増加抑止が渋滞緩和の課題となる。そこで、本博士論文では式 (109) に基づく渋滞の拡大幅増加リスクの評価と、協調的な再経路計画による渋滞の拡大幅増加抑止を提案する。ここで、渋滞迂回のための再経路計画の機能はエージェント、頂点サイト、中央管理装置のいずれかに実装するものとする。再経路計画機能はエージェントの一時停止と移動経路の情報に基づき渋滞の発生とその拡大幅の増加リスクを評価し、拡大幅の増加リスクが高いと判断した場合に再経路計画を実施する。

式 (109) より渋滞の拡大幅の増加リスクは、エージェントの移動確率  $p$ 、ASEP 上の密度  $\rho$ 、ASEP 長  $L$ 、エージェントの一時停止時間  $t_p$  に影響される。本博士論文ではエージェントの移動確率  $p$  と ASEP 長  $L$  は固定値とする。式 (91), (92) より密度  $\rho$  は渋滞長の拡大速度  $\sigma_{el}$  にのみ影響する。そのため、渋滞後方、すなわち ASEP 始端側のエージェント流入量の監視により式 (109) における  $\rho$  に相当する値を入手可能である。以上から、エージェントの荷積み・荷降ろしによる一時停止と ASEP への一時停止  $p_A(t, i)$  と、ASEP 始端へのエージェントの流入量の監視により式 (109) による渋滞幅の増加リスクの評価が可能となる。

エージェント、中央管理装置で拡大幅の増加リスクを評価する場合、搬送路  $N$  上の全エージェントの  $p_A(t, i)$  と経路選択を再経路計画機能を持つエージェント、中央管理装置が把握する必要がある。拡大幅の増加リスク評価機能を頂点サイトに実装する場合、本論文では、通信距離の制約から搬送路  $N$  全域での通信が困難な状況を想定し、エージェントと頂点サイト間の通信による  $p_A(t, i)$  の監視と流入量  $p\rho$  の推定を提案する。

以降では、頂点サイトによる渋滞の拡大幅増加リスクの評価と、渋滞緩和のための再経路計画の要求の流れを示す。提案する協調的再経路計画は以下の機能で構成される。

- a. 頂点サイトによるエージェントの流入量計測
- b. エージェントによる頂点サイトへの一時停止・移動再開通知
- c. 頂点サイトによる再経路計画要求

機能 a) では、頂点サイトは接続する ASEP へのエージェントの流入量を計測する。機能 a) はエージェントの通過を検知する非接触型センサや、エージェントによる頂点サイトへの移動経路情報伝達によって実現する。機能 a) をエージェント、頂点サイト間の通信により実現する場合、通信における情報量は搬送路  $N$  を構成する ASEP 数  $X$  に対して  $\log_2 X$  bit 以下となる。機能 b) において  $A_i$  は荷積み・荷降ろしによる一時停止とその解消の情報として  $p(t, i)$  を伝達する。このとき、送信する情報量は  $p(t, i) \in \{0, 1\}$  より 1bit である。機能 c) では、頂点サイトが渋滞拡大幅の増加リスクをエージェント、または中央管理システムに通知する。隣接する ASEP のみの渋滞幅拡大リスクを考慮する場合、機能 c) の実現に要する通信の情報量は最低で再経路計画の要否を表す 1bit となる。また、再経路計画機能をエージェントに実装する場合、機能 a)~c) は全て、頂点サイトと隣接する ASEP 上のエージェントとの通信で完結する。したがって、情報共有に要する最大通信距離は ASEP 長と一致することから、協調機能の実現に対する通信範囲  $d$  の条件は

$$\Delta(N, v_1^x, v_Y^x) \leq d, \quad \forall x \in \{1, 2, \dots, X\} \quad (115)$$

$$d \geq Y \quad (116)$$

と求まる.

以上から, 渋滞緩和のための通信機能は, ASEP 長に応じた最大通信距離と同時接続数を有することが要件となる.

#### 6.4.2 二次元平面

二次元平面ではその ASEP ネットワーク近似における ASEP 長が  $L = 0$  であることから, エージェントの一時停止時間  $t_p$  の値に関わらず式 (109) の条件を満たすことは不可能である. そこで, 二次元平面の探索では渋滞拡大幅の増加抑止による渋滞緩和ではなく, エージェントの渋滞への再合流で生じるデッドロックの解決に着目する.

式 (53), (86) より, 搬送問題における Demand-for graph は迷路探索における Demand-for graph の部分グラフである. したがって, 搬送問題の荷積み・荷降ろしで発生するデッドロックは, 5 章に示す迷路探索におけるデッドロック解決機能によって解消可能である.

探索引き継ぎと疑似移動によるデッドロック解決機能は, エージェント間の通信によって達成される. デッドロック解決機能の通信における最大情報量はエージェント数  $I$  と一致する. 定理 1 に示す通り通信距離の制限はデッドロック解決の所要時間にのみ影響し, デッドロック解消は保証される. したがって, 協調機能の実装における通信手段の選択では, 通信距離と比較して通信速度と同時接続数が優先される.

疑似移動によるデッドロック解決を行う場合, 探索引き継ぎのための積み荷の交換が必要となる. このとき, 5 章の設定と同じく, 疑似移動には通常の区画移動以上の時間が要する場合がある. したがって, デッドロック解決のための疑似移動では, その協調動作のための一時停止によって新たな渋滞が発生する状況が想定される. 協調動作によって生じる渋滞とその解消条件の評価は, 7 章で自動運転車両の利他的な車線変更を題材に説明する.

### 6.5 数値実験

搬送問題の数値実験では ASEP ネットワーク探索での  $\sigma_{el}$  と  $\sigma_{rl}$  の検証と, 二次元平面における渋滞の再突入によるデッドロック発生を確認する.

Demand-for graph による渋滞解析の結果から, ASEP ネットワーク探索について式 (99), (100) より  $\sigma_{el} \leq \sigma_{rl}$  が成り立つ. したがって, 式 (109) を満たす場合は, 渋滞の解消が保証される.

また, 二次元平面においては式 (109) における  $L$  が 0 となるため, 渋滞へのエージェントの再突入やデッドロックによる渋滞解消失敗が予想される.

#### 6.5.1 Beth Network 探索

全頂点サイトの出次数と入次数が等しい, Beth Network 上での渋滞の拡大速度と解消速度を検証を示す. 探索対象図 21 に探索対象の ASEP の直列ネットワークを示す. 図 21 は ASEP 長  $L = 100$ , 頂点サイト数  $\|V\| = 10$ , 頂点サイトの平均次数 2 の Beth Network である.

本実験では, はじめに密度  $\rho$  の定常流を発生させる. その後,  $e_1$  の終端に位置するエージェントの荷積みにより, 渋滞を発生させる. 渋滞の拡大幅が 2 に増加するまでの間  $V_t(t)$  の拡大を観測し, 渋滞発生から拡大幅上昇までの経過時間に基づき  $\sigma_{el}$  を計測する. 最後に, 渋滞の先頭エージェントの荷積みによる一時停止を解除し,  $V_t(t)$  の観測結果に基づき  $\sigma_{rl}$  を計測する. また, 渋滞解消までの最大渋滞幅を計測する. 本実験では同一の密度と移動確率の設定に対して 5 回の実験を行い,  $\sigma_{el}$  と  $\sigma_{rl}$  の計測値

の平均値を示す.

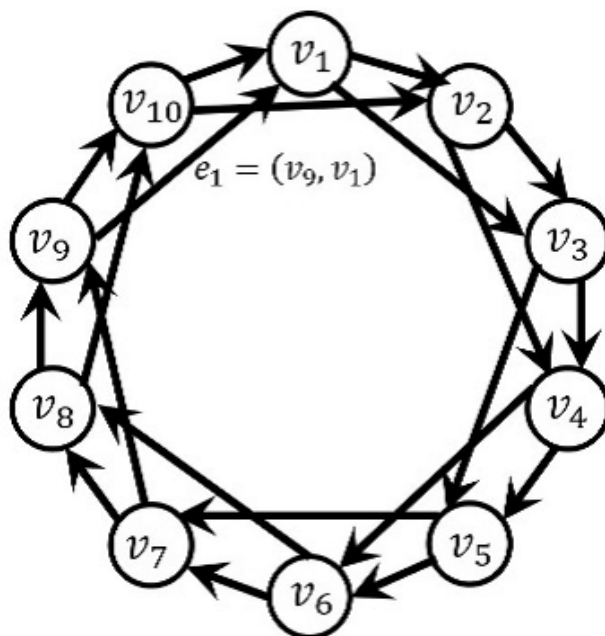


図 21: 探索対象 (Beth Network)

式 (99), (100) より渋滞長の拡大速度  $\sigma_{el}$  は移動確率  $p$  と密度  $\rho$  に比例して上昇し, 渋滞長の解消速度  $\sigma_{rl}$  は密度  $\rho$  に関わらず移動確率  $p$  に近似されると予想される.

$\rho$  と  $p$  に対する  $\sigma_{el}$  と  $\sigma_{rl}$  の計測結果を, それぞれ図 22 と図 23 に示す. 図 22 は  $\rho$  に対する  $\sigma_{el}$  の推移を示す. 図 23 に示す  $\sigma_{el}$  は,  $\rho \leq 0.7$  の範囲において式 (100) で導出した予測値と近い値となった.  $\rho \geq 0.8$  の範囲においては  $\sigma_{el}$  が, 予測値を上回る結果となった. これは, 潜在的渋滞区域では渋滞の拡大速度が上昇することを意味する.

図 23 は  $\rho$  に対する  $\sigma_{rl}$  の推移を示す. 図 23 に示す  $\sigma_{rl}$  の値は, 式 (101) で導出した予測値と近い数値となった. また, 図 23 において  $\sigma_{rl}$  の  $\rho$  に対する推移はほぼ横ばいとなっており, 渋滞の解消速度は密度の影響を受けにくいことがわかる. 以上の結果から Beth Network において導出した式 (100), 式 (101), 式 (104) に示す渋滞の解消条件が, 渋滞解消の必要条件として妥当であることが示せた.

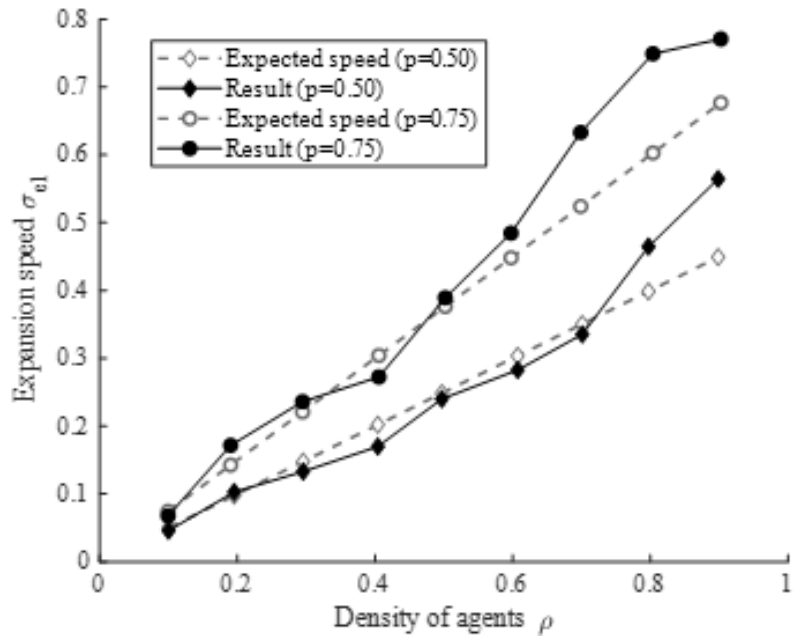


図 22: 渋滞の拡大速度 (Beth Network)

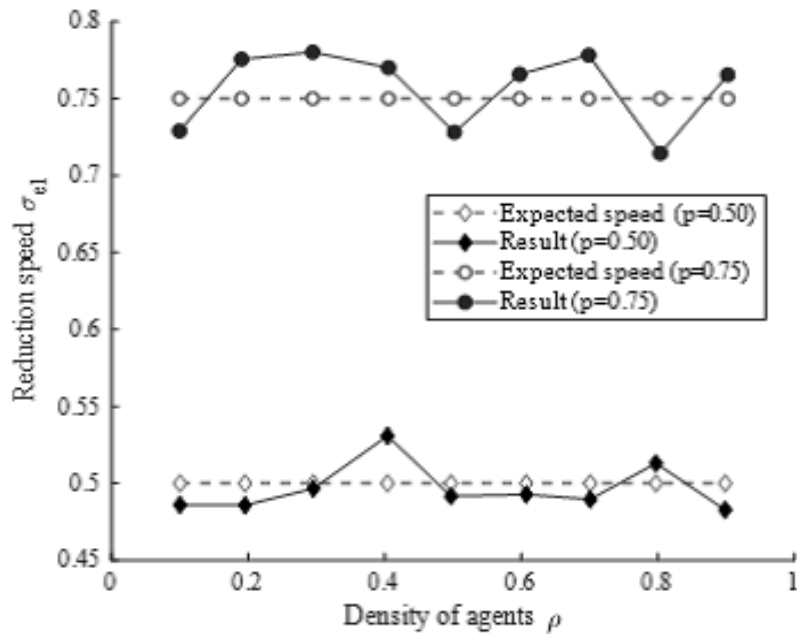


図 23: 渋滞の解消速度

### 6.5.2 ASEP ネットワーク探索

ASEP ネットワーク探索では潜在的渋滞区域外の  $\sigma_{el}$  と  $\sigma_{rl}$  の検証を示す．本数値実験の探索対象は ASEP 長  $L = 100$ ，頂点サイト数  $\|V\| = 15$  の ASEP の直列ネットワーク  $N$  である．探索対象に密度  $\rho = 0.1, 0.2, \dots, 0.9$ ，移動確率  $p = 0.5, 0.75$  の流れを発生させる．図 24 に探索対象の ASEP の直列ネットワークを示す．

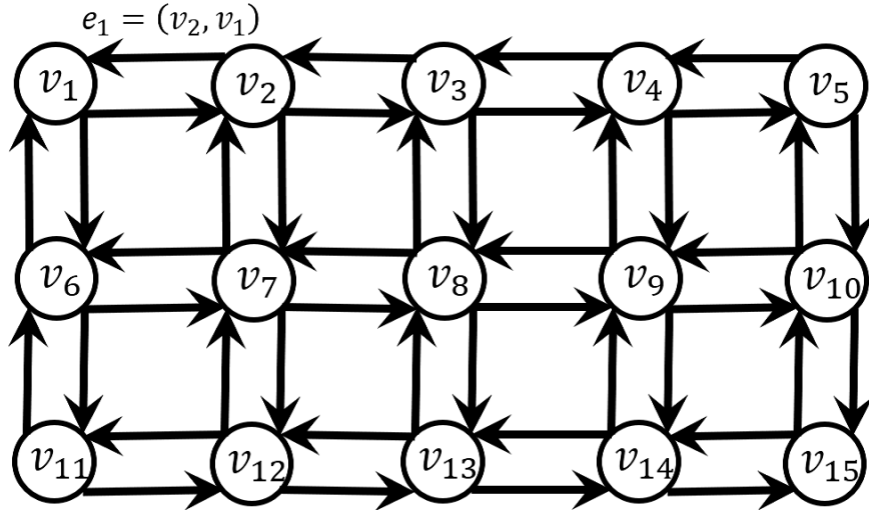


図 24: 探索対象 (ASEP network)

本実験では，はじめに密度  $\rho$  の定常流を発生させる．その後， $e_1$  の終端に位置するエージェントの荷積みにより，渋滞を発生させる．渋滞の拡大幅が 2 に増加するまでの間  $V_i(t)$  の拡大を観測し，渋滞発生から拡大幅上昇までの経過時間に基づき  $\sigma_{el}$  を計測する．最後に，渋滞の先頭エージェントの荷積みによる一時停止を解除し， $V_i(t)$  の観測結果に基づき  $\sigma_{rl}$  を計測する．また，渋滞解消までの最大渋滞幅を計測する．本実験では同一の密度と移動確率の設定に対して 5 回の実験を行い， $\sigma_{el}$  と  $\sigma_{rl}$  の計測値の平均値を示す．

Demand-for graph に基づく渋滞解析の結果において，式 (100) より渋滞長の解消速度  $\sigma_{rl}$  は移動確率のみに依存するため Beth-Network 上での  $\sigma_{rl}$  から変化しないことが予想される．対して，式 (99) より渋滞長の拡大速度  $\sigma_{el}$  は密度  $\rho$  が影響するため，図 24 上での  $\sigma_{el}$  は Beth-Network 上での値から変化すると予想できる．また，渋滞の拡大幅の増加に関して，式 (109) の渋滞の拡大幅の増加予想に  $p = 0.75$ ， $t_p \sigma_{el} = 100$ ， $L = 100$  を代入する．このとき，渋滞の拡大幅の増加が予想される密度は  $\rho \geq 0.82$  となる．したがって， $\rho = 0.9$  の結果に対して，渋滞の拡大幅増加による大幅な流量低下と渋滞への再突入による渋滞解消失敗が予想される．

$\rho$  と  $p$  に対する  $\sigma_{el}$  と  $\sigma_{rl}$  の計測結果を，それぞれ図 25 と図 26 に示す．図 25 は  $\rho$  に対する  $\sigma_{el}$  の推移を示す．図 25 に示す  $\sigma_{el}$  は，式 (100) で導出した予測値を下回る結果となった．これは，探索対象である  $N$  が Beth Network ではないため，頂点サイトを結ぶ ASEP によって密度が変化するためである．図 22 と比較すると，ネットワーク上での密度の偏りは渋滞長の拡大速度が低下していることが確認できた．

図 26 は  $\rho$  に対する  $\sigma_{rl}$  の推移を示す．図 26 に示す  $\sigma_{rl}$  の値は，式 (101) で導出した予測値と近い数値となった．また，図 26 において  $\sigma_{rl}$  の  $\rho$  に対する推移はほぼ横ばいとなっている．図 23 の結果と



併せて、渋滞の解消速度は探索対象のネットワーク構造や密度の影響を受けにくいことがわかる。以上の結果から導出した式 (100), 式 (101), 式 (104) に示す渋滞の解消条件が、渋滞解消の必要条件として妥当であることが示せた。

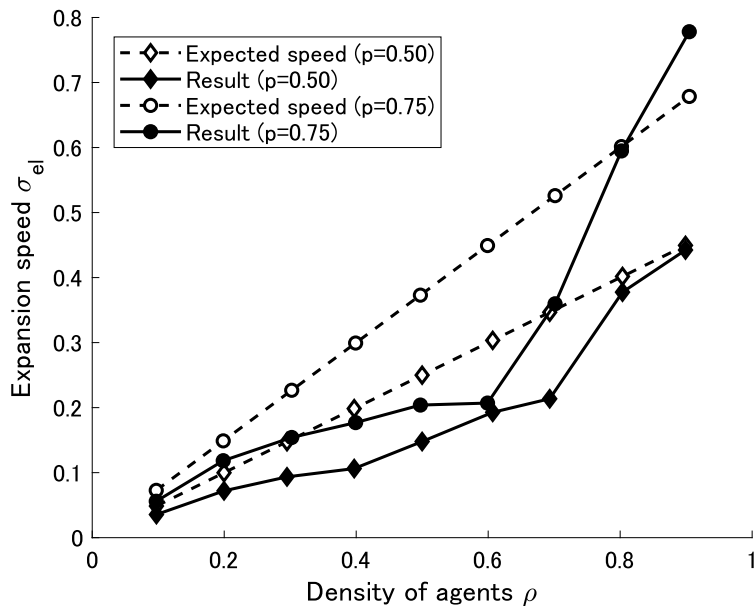


図 25: 渋滞の拡大速度 (ASEP network)

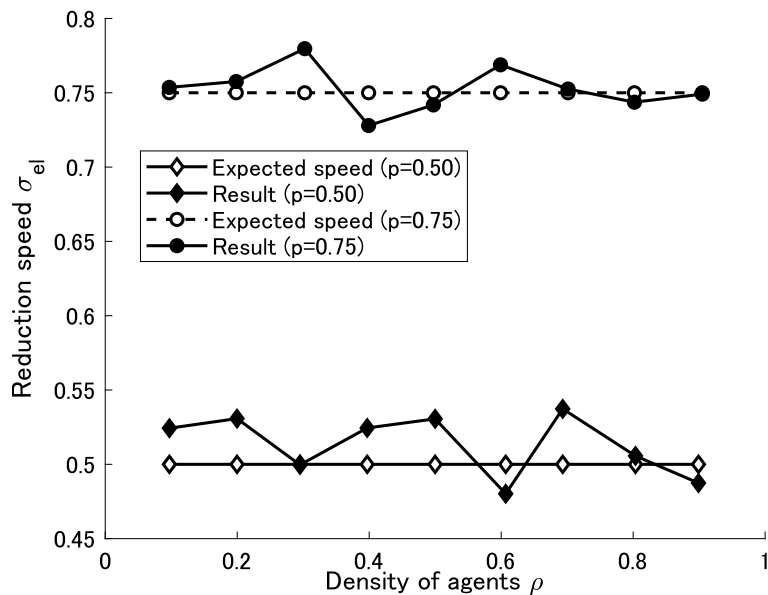


図 26: 渋滞の解消速度 (ASEP network)

次に、渋滞拡大幅の最大値の計測結果を示す。表 3 に  $p = 0.75$  に対する、渋滞拡大幅の最大値の計測結果を示す。表 3 について  $0.7 \leq \rho$  の結果を確認すると、渋滞からの分離開始したのちも渋滞幅が拡大していることがわかる。この場合、渋滞長の減少開始後も渋滞幅の拡大により、流量は低下し続ける。この結果は、式 (103) と同じく、探索対象  $N$  において ASEP 長  $L$  が大きく平均次数  $c$  が小さい場合に渋滞による流量低下を軽減できることを示す。  $0.8 \leq \rho$  の結果に着目すると、渋滞は拡大と解消を繰り返し完全に解消されることはなかった。図 25 と図 26 を確認すると、  $\rho = 0.8$  の結果において  $\sigma_{el}$  は  $\sigma_{rl}$  を下回っている。式 (109) より、渋滞の  $\rho = 0.8$  における渋滞の完全解消失敗の原因は、渋滞の拡大幅増加により渋滞への再合流が発生したためと言える。表 3 の結果と併せて渋滞の完全解消の評価には式 (103) の渋滞解消条件だけではなく、式 (109) による渋滞の拡大幅増加予想と渋滞への再突入を考慮する必要があることがわかる。

表 3: 最大渋滞拡大幅

$\rho$	0.1	0.3	0.5	0.7	0.9
Maximum width	3.0	3.0	3.0	4.0	20.8

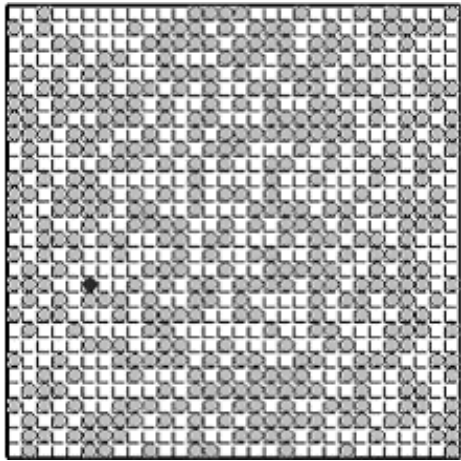
### 6.5.3 二次元平面探索

本実験では二次元平面における  $\sigma_{el}$  の予想値と、渋滞から分離したエージェントの再合流によるデッドロックの発生を検証する。探索対象  $N$  は  $30 \times 30$  の二次元平面である。  $N$  上のエージェントの密度は  $\rho = 0.5$ 、エージェントの移動確率は  $p = 0.75$  である。本実験では 20 回の探索シミュレーションを行い、平均値に基づき  $\sigma_{el}$  を計測する。

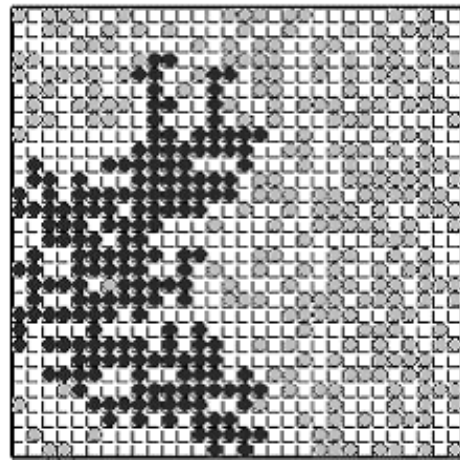
一回目の探索シミュレーションにおける渋滞の拡大と解消の様子を図 27 に示す。実験では、はじめに  $\rho = 0.5$  の定常流を発生させる。その後、任意のエージェント 1 台に荷積みを模した一時停止を発生させる。渋滞発生後は 100 ステップの間  $V_t(t)$  を観測し、渋滞拡大の結果から  $\sigma_{el}$  と渋滞の拡大幅を計測する。最後に、一時停止させたエージェントの移動を再開させ、  $V_t(t)$  によって渋滞の解消を観測した。

図 27 の黒色の丸は荷積み・荷降ろしで発生した渋滞内に位置するエージェントを表す。灰色の丸は渋滞外に位置するエージェントを表す。図 27 (a) に示す通り、  $t = 101$  において渋滞を発生させている。その後、  $t = 201$  までの間、渋滞の拡大を観測している。  $t = 201$  における渋滞長は 25 区画、渋滞の拡大幅は 62 区画となった。20 回のシミュレーションによる、  $t = 201$  での渋滞長と渋滞の拡大幅の平均値はそれぞれ 23.4 区画、47.6 区画となった。式 (110) による渋滞長の予想値は 23.08 であり、シミュレーションで計測した渋滞長と近い値となった。一方で、式 (112) に基づく渋滞の拡大幅の予想値は 92.32 である。シミュレーションによる計測値は、予想値の半分程度の値となった。式 (112) は渋滞拡大幅の上限を予想するために、渋滞が同心円状に拡大していく状況を想定している。実験結果からも、渋滞の拡大幅は式 (112) の予想値を下回ることが確認できた。

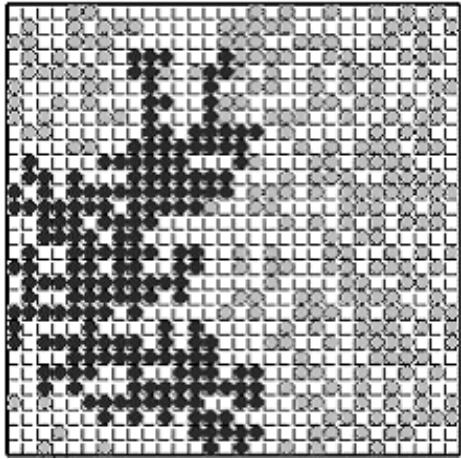
図 27 (b) は渋滞からのエージェントの分離開始直後の様子である。渋滞先頭のエージェントの移動は  $t = 204$  に開始された。(c) は  $t = 208$  においてエージェントの渋滞再合流によりデッドロックが生じた様子を表している。その後 (d) に示す通り、最終ステップまで渋滞は拡大を続けた。実験結果より渋滞の解消する協調的な経路計画には渋滞の解消条件だけではなく、デッドロック解決の考慮が必要であることを確認できた。



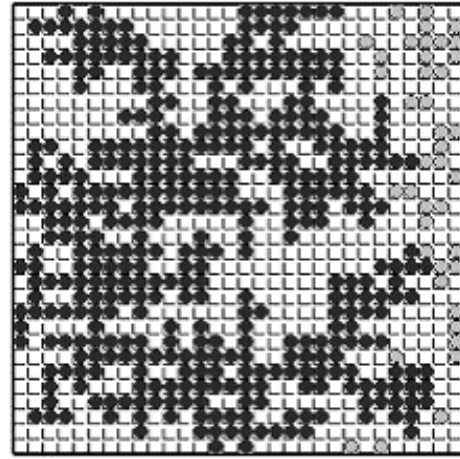
( a )  $t = 101$



( b )  $t = 204$



( c )  $t = 208$



( d )  $t = 300$

图 27: 探索結果 (二次元平面)

## 7 車線変更と車群形成

自動運転車両の車線変更と車群形成では  $I$  台のエージェントによる複数レーン ASEP 探索を通じて、協調機能によって新たに生じる渋滞の規模とその解消を評価する。迷路探索問題と搬送問題での渋滞解析で得られた協調機能による既存渋滞の緩和と総合して、協調機能の導入による渋滞発生と渋滞緩和効果の評価を実現する。エージェントは協調機能として、後続車両の車速確保を目的とした利他的車線変更 [61] を行う。

以降では、協調動作である利他的な車線変更による渋滞の発生と、車線変更後の渋滞の解消速度を評価する。利他的な車線変更で生じる渋滞の評価には、搬送問題で提案したインシデント ID を利用する。

### 7.1 自動運転車両の車線変更と車群形成

はじめに、問題設定として、交通円滑化のための自動運転車両の利他的な車線変更について述べる。

文献 [61] では速度の異なる自動運転車両群に関して、個々の車両の速度を確保することで車両群全体の流量を確保するための車線変更の手法を提案している。各車両は自身の最高速度、現在速度、目標車間距離、および周囲の車両の最高速度の情報に基づき、前進と車線変更を判断する。

文献 [61] では各車両は単体の車線変更では最高速度までの加速が不可能であると判断した場合、前方の車両に交通円滑化のための車線変更を要求する。以降では文献 [61] に倣い車線変更を要求した車両を Current vehicle、車線変更の要求を受け取った車両を Front vehicle と表す。Front vehicle は速度の近い車両の区画占有によって車線変更が阻害された場合、車線変更の妨げとなっている車両に車間確保のための減速を要求する。以降では Front vehicle から減速要求を受け取った車両を Assistant vehicle と表す。本論文では、Front vehicle の Current vehicle からの要求に基づく車線変更を利他的な車線変更として扱う。

利他的な車線変更に際して、Current vehicle、Front vehicle、Assistant vehicle は協調制御のために車群を形成する。[61] では、制御を簡単化するためにすでに車群に属しているエージェントは新たな車線変更や減速の要求を拒否する。Front vehicle は自身の車線変更完了後に、Current vehicle と Assistant vehicle に車線変更完了を通知する。Front vehicle の車線変更完了通知によって、利他的な車線変更のための車群は解除される。

利他的な車線変更と車線変更のための車群形成の例を図 28 に示す。

図 28 は  $A1$  が自身より最高速度の低い 3 台の車両  $A2$ ,  $A3$ ,  $A4$  に前方をふさがれる状況を想定している。また、 $A2$ ,  $A3$ ,  $A4$  の走行速度は同程度とする。このとき、 $A1$  は 3 車線のどの車線に移動した場合も他の車両に進路を妨げられ、自身の最高速度に到達することができない。そのため、 $A1$  は Current vehicle として同一車線前方に位置する  $A3$  に対して車線変更を要求する。これにより、 $A3$  が Front vehicle に設定される。しかしながら、 $A3$  の車線変更は速度の近い 2 台のエージェント  $A2$ ,  $A4$  の区画占有によって阻害される。そのため、 $A3$  は車線変更に必要な車間距離を確保するために、 $A2$  に減速を要求する。これにより、 $A2$  が Assistant vehicle に設定される。 $A3$  は車線変更が可能になるまでの間は現在速度での前進を継続する。 $A3$  は車線変更完了後、 $A1$  と  $A2$  に車群解除の通知を送信する。 $A2$  は車群解除の通知を受信後、自身の最高速度まで加速し車速を確保する。また、 $A3$  の車線変更により、 $A1$  は自身の最高速度までの加速が可能となる。

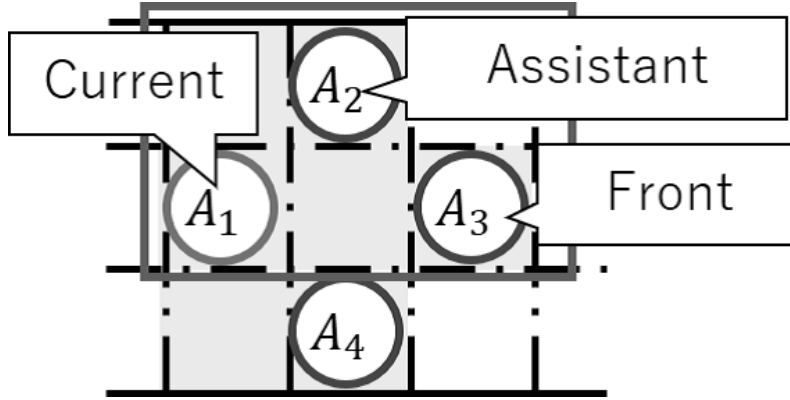


図 28: 利他的な車線変更の例

本博士論文では、利他的な車線変更に要する Assistant vehicle の減速と車群後方、特に Assistant vehicle の後続車両に生じる一時的な車速低下に着目し、利他的な車線変更に起因する渋滞を評価する。

本論文では、エージェントの速度を移動確率で表現する。エージェントは速度のパラメータとして、最高速度と現在速度を有する。  $A_i$  の最高速度を  $p_{\max}(i)$  ( $0 \leq p_{\max}(i) \leq 1$ ) と表す。時刻  $t$  における  $A_i$  の現在速度を  $p(t, i)$  ( $0 \leq p(t, i) \leq p_{\max}(i)$ ) と表す。

## 7.2 渋滞の Demand-for graph 表現

本論文では、利他的な車線変更と車線変更のための減速が引き起こす渋滞を評価するため、自動運転車両の制御に新たに加速要求の概念を導入する。エージェントは車群に属しているエージェントへの衝突回避のために一時停止を選択した場合、進路妨害元のエージェントに加速を要求し、車群に合流する。このとき、加速要求を発したエージェントを Follow vehicle と表す。これにより、利他的な車線変更が引き起こす渋滞を、Follow vehicle の速度低下と密度上昇として評価する。

以降では、自動運転車両の車群形成・車線変更アルゴリズムの複数レーン ASEP 探索への拡張と、エージェントの車線変更、減速、加速の要求設定の数理モデルを示す。

移動速度確保のための車線変更に関して、エージェントは車線変更、減速、加速を前方のエージェントに要求する。時刻  $t$  において、 $A_i$  が車線変更を要求するエージェントの ID を  $r_c(t, i) \in (\{0\} \cup \mathcal{I})$  と表す。時刻  $t$  において、 $A_i$  が他のエージェントに車線変更を要求しない場合は  $r_c(t, i) = 0$  となる。時刻  $t$  において、 $A_i$  が減速を要求するエージェントの ID を  $r_d(t, i) \in (\{0\} \cup \mathcal{I})$  と表す。時刻  $t$  において、 $A_i$  が他のエージェントに減速を要求しない場合は  $r_d(t, i) = 0$  となる。時刻  $t$  において、 $A_i$  が加速を要求するエージェントの ID を  $r_a(t, i) \in (\{0\} \cup \mathcal{I})$  と表す。時刻  $t$  において、 $A_i$  が他のエージェントに加速を要求しない場合は  $r_a(t, i) = 0$  となる。時刻  $t$  において、 $A_i$  が車線変更、減速、加速のいずれかを要求するエージェントの ID を  $r(t, i) \in (\{0\} \cup \mathcal{I})$  と表す。時刻  $t$  において、 $A_i$  が他のエージェントに車線変更、減速、加速のいずれも要求しない場合は  $r(t, i) = 0$  となる。このとき、 $r_c(t, i)$ ,  $r_d(t, i)$ ,  $r_a(t, i)$  は

$$\begin{aligned}
 &\neg(r_c(t, i) \neq 0) \vee (r_d(t, i) = r_a(t, i) = 0) \\
 &\neg(r_d(t, i) \neq 0) \vee (r_a(t, i) = r_c(t, i) = 0) \\
 &\neg(r_a(t, i) \neq 0) \vee (r_c(t, i) = r_d(t, i) = 0)
 \end{aligned} \tag{117}$$

を満たす。式 (117) は時刻  $t$  において、 $A_i$  が設定できる要求の最大数が 1 であることを意味する。本

論文では式 (117) より  $r(t, i) = 0$  を  $r_c(t, i)$ ,  $r_d(t, i)$ ,  $r_a(t, i)$  に対して

$$r(t, i) \leftarrow r_c(t, i) + r_d(t, i) + r_a(t, i) \quad (118)$$

と与える。

また、エージェントは車線変更、減速、加速の要求に際して、要求先のエージェントと車群を形成する。時刻  $t$  において、 $\mathcal{A}_i$  が車群に含まれるか否かを  $\xi(t, i) \in \{0, 1\}$  と表す。時刻  $t$  において、 $\mathcal{A}_i$  が車群に含まれる場合、 $\xi(t, i) = 1$  となる。車群の分離に関して、時刻  $t$  における  $\mathcal{A}_i$  の車群分離通知を  $g_c(t, i) \in \{0, 1\}$  と表す。時刻  $t$  における  $\mathcal{A}_i$  が車群から分離する場合、 $g_c(t, i) = 1$  となる。

車群形成と要求の設定に関して、エージェントは下記の順序で探索機能を使用する。

- a. 自力での加速可否判定
- b. 車線変更要求の設定
- c. 減速要求の設定
- d. 車線変更・直進選択
- e. 減速処理
- f. 移動、加速要求の設定
- g. 車群分離

はじめに、エージェントは車線変更要求の要否を判定する。 $\mathcal{A}_i$  は各レーンに対して、確保する目標車間距離  $d \in \{1, 2, \dots, \infty\}$  の範囲で  $p_{\max}(i)$  まで加速可能かを判定する。 $\mathcal{A}_i$  は  $\varepsilon_{\mathcal{A}}(t, i)$  と目標車間距離内のエージェントの速度から、レーン  $x'$  での加速可否を Algorithm 4 にしたがって判断する。

---

**Algorithm 4** Function1( $x'$ ) : Acceleration availability at  $\mathcal{L}x'$

---

```

1: Set  $x$  and  $y$  from  $v_y^x = \varepsilon_{\mathcal{A}}(t, i)$ ;
2: if ( $|x' - x| > d$ ) then
3:   return false;
4: else
5:   for  $l \leftarrow 1; l < d - |x' - x|; l++$  do
6:     if ( $y + l \leq Y$ ) then
7:       if  $\left( \begin{array}{l} p_{\mathcal{A}}(t, \varepsilon_{\mathcal{N}}(t, v_{y+l}^{x'})) \\ < p_{\max}(t, i) \end{array} \right)$  then
8:         return false;
9:       end if
10:    end if
11:  end for
12:  return true;
13: end if

```

---

$\mathcal{A}_i$  は各レーンでの加速可否の判定結果から、Algorithm 5 によって自身の車線変更による最高速度までの加速が可能であるかを判定する。 $\mathcal{A}_i$  は車線変更による加速が不可能であると判断した場合、同一レーンの前方最近傍に位置するエージェントに車線変更要求を送信する。このとき、要求先のエージェントの ID を  $r_c(t, i)$  に設定する。ただし、対象のエージェントがすでに車群に属している場合、車

線変更要求は拒否される。したがって、 $r_c(t, i)$  は  $\xi(t, i)$  に対して

$$\neg((r_c(t, i) \neq r_c(t-1, i)) \wedge (r_c(t, i) \neq 0)) \vee (\xi(t, r_c(t, i)) = 0) \quad (119)$$

を満たす。  $r_c(t, i) \neq 0$  の場合、  $\mathcal{A}_i$  は Current vehicle として、  $\mathcal{A}_{r_c(t, i)}$  は Front vehicle として車群に合流する。

車線変更要求を受信した場合、  $\mathcal{A}_i$  は式 (26) の左辺の衝突回避条件に基づき隣接するレーンへの車線変更の可否を判定する。車線変更が可能な場合は、隣接するレーンへの車線変更を  $\hat{\delta}_{\mathcal{A}}(t, i)$  に設定する。車線変更が不可能な場合は隣接するレーンの後方最近傍のエージェントの ID を  $r_d(t, i)$  に設定し、  $\mathcal{A}_{r_d(t, i)}$  に減速要求を送信する。ただし、対象のエージェントがすでに車群に属している場合、減速要求は拒否される。したがって、  $r_d(t, i)$  は  $\xi(t, i)$  に対して

$$\neg((r_d(t, i) \neq r_d(t-1, i)) \wedge (r_d(t, i) \neq 0)) \vee (\xi(t, r_d(t, i)) = 0) \quad (120)$$

を満たす。  $r_d(t, i) \neq 0$  となる場合、  $\mathcal{A}_{r_d(t, i)}$  が Assistant vehicle として車群に加わる。

---

**Algorithm 5** Function2(): Acceleration on its own or not

---

```

1: Set  $x$  from  $v_y^x = \varepsilon_{\mathcal{A}}(t, i)$ ;
2: if Function1( $x$ )=true then
3:   return true;
4: end if
5: for ( $m \leftarrow 1$ ;  $m < d$ ;  $m++$ ) do
6:   if Function1( $x+m$ )=true then
7:     return true;
8:   end if
9:   if Function1( $x-m$ )=true then
10:    return true;
11:  end if
12: end for
13: return false;

```

---

$\mathcal{A}_i$  は現在のレーンでの加速可否、および車線変更要求  $r_c(t, i)$  に基づき時刻  $t$  における目標移動区画  $\hat{\delta}_{\mathcal{A}}(t, i)$  を設定する。目標移動区画決定後、  $\mathcal{A}_i$  は周囲のエージェントの減速要求  $r_d(t, j, i)$  にしたがって移動確率  $p(t, i)$  を

$$p(t, i) \leftarrow \begin{cases} p_d & \left( \begin{array}{l} \text{if } \exists j \in \mathcal{I}, \\ \text{s. t. } (r_d(t, j) = i) \end{array} \right) \\ p_{\max}(i) & \text{otherwise} \end{cases} \quad (121)$$

と設定する。移動確率  $p(t, i)$  を設定後、  $\mathcal{A}_i$  は Algorithm 6 にしたがって移動する。このとき、  $\mathcal{A}_i$  は Algorithm 6 の 3 - 8 行目の通りに加速要求  $r_a(t, i)$  を設定する。Algorithm 6 の 3 行目の if 文の条件は式 (26) の左辺と一致する。  $r_a(t, i) \neq 0$  となる場合、  $\mathcal{A}_i$  は Follow vehicle として車群に合流する。  $\mathcal{A}_i$  は移動結果に基づいて  $\xi(t+1, i)$  を更新し、車群の分離を行う。エージェントは自身の車線変更、減速、加速要求が満たされた場合に車群から分離する。  $\xi(t+1, i)$  の更新アルゴリズムを Algorithm 7 に示す。Algorithm 7 の 3 行目は  $\mathcal{A}_i$  が車線変更要求に基づく車線変更を完了した場合に、要求元に車線変更完了を通知することを意味する。また、5 行目は  $\mathcal{A}_i$  が車線変更を完了した場合に、減速要求の要

求先の Assistant vehicle と車線変更の要求元である Current vehicle に車線変更完了と車群分離を通知することを意味する。

---

**Algorithm 6** Moving agents

---

```

1: Set  $x$  and  $y$  from  $v_y^x = \hat{\delta}_{\mathcal{A}}(t, i)$ ;
2: if  $\text{rand}() < p(t, i)$  then
3:   if  $\left( \sum_{j \in (\mathcal{I} \setminus \{i\})} \varepsilon(t, j, v_y^x) + \varepsilon(t, j, v_{y-1}^x) = 1 \right)$  then
4:      $\delta_{\mathcal{A}}(t, i) \leftarrow \varepsilon_{\mathcal{A}}(t, i)$ ;
5:     if  $\xi(t, \varepsilon_N(t, \hat{\delta}_{\mathcal{A}}(t, i))) = 1$  then
6:        $r_a(t, i) \leftarrow \varepsilon_N(t, \hat{\delta}_{\mathcal{A}}(t, i))$ ;
7:     end if
8:   else
9:      $\delta_{\mathcal{A}}(t, i) \leftarrow \hat{\delta}_{\mathcal{A}}(t, i)$ ;
10:  end if
11: else
12:    $\delta_{\mathcal{A}}(t, i) \leftarrow \varepsilon_{\mathcal{A}}(t, i)$ ;
13: end if

```

---



---

**Algorithm 7** Separation of vehicle groups

---

```

1: Set  $x$  and  $y$  from  $v_y^x = \varepsilon_{\mathcal{A}}(t, i)$ ;
2: if  $(\exists j \in \mathcal{I} \text{ s.t. } r_c(t, j) = i)$  then
3:   if  $\left( (\delta_{\mathcal{A}}(t, i) = v_y^{x+1}) \vee (\delta_{\mathcal{A}}(t, i) = v_y^{x-1}) \right)$  then
4:      $g_c(t, j) \leftarrow 1$ ;
5:     if  $(r_d(t, i) \neq 0)$  then
6:        $g_c(t+1, i) \leftarrow 1$ ;
7:        $g_c(t+1, r_d(t, i)) \leftarrow 1$ ;
8:     end if
9:   end if
10: else if  $(r_a(t, i) \neq 0)$  then
11:   if  $\left( (\delta_{\mathcal{A}}(t, i) = \hat{\delta}_{\mathcal{A}}(t, i)) \wedge (\xi(t, r_a(t, i)) = 0) \right)$  then
12:      $g_c(t, i) \leftarrow 1$ ;
13:   end if
14: end if

```

---

$\mathcal{A}i$  は  $r(t, i)$  と  $g_c(t, i)$  から  $\xi(t+1, i)$  を,

$$(\xi(t+1, i) = 0) \iff (r(t, i) = 0 \wedge (r(t, j) \neq i, \forall j \in I)) \vee (g_c(t, i) = 1) \quad (122)$$

と求める。式 (118), (122) より, Current vehicle, Front vehicle, Assistant vehicle, Follow vehicle によって車群が形成される。また,  $\mathcal{A}i$  は  $r(t, i) \neq 0$  となる場合に車群に合流し,  $g_c(t, i) = 1$  となる場合に車群から分離する。また,  $\mathcal{A}i$  は  $r(t, i) \neq 0$ , すなわち他のエージェントに車線変更, 減速, 加速を要求している場合, 要求が達成されるまで  $r(t, i)$  を維持する。したがって,  $\xi(t+1, i)$  と  $r(t+1, i)$  に



ついて

$$\neg(\xi(t+1, i) = 1) \vee ( (r_c(t+1) = r_c(t)) \wedge (r_d(t+1) = r_d(t)) \wedge (r_a(t+1) = r_a(t)) ) \quad (123)$$

が成り立つ。

以降では、本論文で提案する Demand-for graph による利他的な車線変更に対する渋滞表現を示す。

時刻  $t$  における Demand-for graph を  $G_d(t) = (N_d, A_d(t))$  と表す。  $N_d : \{n_1, n_2, \dots, n_I\}$  と  $A_d(t)$  はそれぞれエージェントを表すノードと、エージェント間の要求を表すエッジの有限集合である。  $n_i \in N_d$  は  $A_i$  に対応するノードである。  $(n_i, n_j) \in A_d(t)$  は時刻  $t$  における  $A_i$  から  $A_j$  への要求を表す。本論文では  $G_d(t)$  を、  $r(t, i)$  に対して

$$A_d(t) \leftarrow \sum_{i \in I} f_{14}(t, i), \quad (124)$$

$$f_{14}(t, i) = \begin{cases} \emptyset & (\text{if } r(t, i) = 0) \\ (n_i, n_{r(t, i)}) & (\text{otherwise}) \end{cases}$$

と与える。

このとき、Algorithm 6 より  $r_a(t, i) \neq 0$  となる場合、  $A_i$  は一時停止を選択する。式 (26), (118), (124) より、  $(n_i, n_j) \in A_d(t)$  は  $A_i$  の速度低下を意味する。そのため、利他的な車線変更のための車群を、車線変更で生じる渋滞として扱う。利他的な車線変更のための車群は、Demand-for graph では連結成分として現れる。

本論文では、渋滞の先頭を Demand-for graph 上で出次数が 0 となる端点  $n_i$  に対応するエージェントとして定義する。式 (124) より、  $A_i$  が渋滞の先頭に位置する場合、  $r(t, i) = 0$  が成り立つ。また、Demand-for graph での端点間の距離を渋滞の長さとして定義する。ここで、グラフ上のノード間の距離は、二つのノードを結ぶ最短経路の長さを意味する。式 (122), 式 (124) より、渋滞の長さは渋滞最後尾のエージェントが車群から分離するために達成する必要がある要求の数を表す。したがって、発生した渋滞の完全解消の所要時間は、渋滞の長さに比例する。

自動運転車両の車線変更では Demand-for graph の隣接行列  $\Phi(t)$  の時間変化式 (6) について  $\mathbf{u}_1(t)$  を

$$\mathbf{u}_1(t) = \begin{bmatrix} U_1(t, 1) \\ \vdots \\ U_1(t, I) \end{bmatrix}, \quad (125)$$

$$U_1(t, i) = \begin{bmatrix} f_{15}(t, i, 1) & 0 & \cdots & 0 \\ 0 & f_{15}(t, i, 2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & f_{15}(t, i, I) \end{bmatrix},$$

$$f_{15}(t, i, j) = \begin{cases} 1 & \left( \text{if } \left( (\delta_{\mathcal{A}}(t, i) = \hat{\delta}_{\mathcal{A}}(t, i) \wedge \xi_{\mathcal{A}}(t, i) \neq 0) \right) \vee (r_c(t, i) = j) \vee (r_d(t, i) = j) \right) \\ 0 & \text{otherwise} \end{cases}$$

と、  $\mathbf{u}_2(t)$  を

$$\begin{aligned}
\mathbf{u}_2(t) &= \begin{bmatrix} U_2(t, 1) \\ \vdots \\ U_2(t, I) \end{bmatrix}, & (126) \\
U_2(t, i) &= \begin{bmatrix} f_{15}(t, i, 1)f_{16}(t, 1) & 0 & \cdots & 0 \\ 0 & f_{15}(t, i, 2)f_{16}(t, 2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & f_{15}(t, i, I)f_{16}(t, I) \end{bmatrix} \\
f_{16}(t, i) &= \begin{cases} 1 & (\text{if } g_c(t, i) = 1) \\ f_3(t, i) & \text{otherwise} \end{cases}
\end{aligned}$$

と与える.  $B$ ,  $D$  はそれぞれ式 (8) と式 (10) で与える. このとき, 式 (125) と式 (95) について

$$(f_{15}(t, i, j) = 1) \iff (f_{12}(t, i, j) = 1 \vee r_c(t, i) = j \vee r_d(t, i) = j) \quad (127)$$

が常に真となる.  $r_c(t, i)$  が表す車線変更要求と  $r_d(t, i)$  が表す減速要求は, どちらも通信を要する協調動作に対する行動要求である. 式 (127) より, 式 (125) では自動運転車両の利他的な車線変更を構成する, 車線変更と減速の二つの協調動作を起点とする渋滞の拡大を観測可能である. また, 式 (126) と式 (96) について

$$(f_{16}(t, i, j) = 1) \iff f_4(t, i, j) = 1 \vee g_c(t, i) = 1 \quad (128)$$

が常に真となる. 式 (128) は, 式 (126) に示す渋滞の解消が式 (96) に示す渋滞の自然解消に, 協調動作の達成を表す通信情報  $g_c(t, i)$  を導入したものであることを表す. したがって, 式 (125) と式 (126) に基づく渋滞の Demand-for graph は, 協調動作に対する行動要求の設定と達成に基づく渋滞規模変化の評価を実現する.

このとき, 式 (125), (126) より, グラフ探索においても式 (11) が成り立つ. したがって,  $\mathbf{u}_1(t) - \mathbf{u}_1(t-1)$  は渋滞の拡大に,  $\mathbf{u}_2(t)$  は渋滞の解消に対応する.

### 7.3 渋滞の解消速度と解消条件

自動運転では, 時刻  $t$  における渋滞長の減少値の期待値を渋滞の解消速度  $\sigma_{rl}(t)$  として扱う.  $\sigma_{rl}(t)$  は**定理 2** を満たす.

**定理 2.** 時刻  $t$  において渋滞の先頭に位置するエージェントが *Follow vehicle* の場合,  $\sigma_{rl}(t)$  は渋滞の前から 2 番目に位置するエージェントの移動確率と一致する.

**証明.** 式 (126) より, 渋滞長の減少はエージェントの区画移動に対して発生する. 渋滞の前から 2 番目に位置するエージェントの ID を  $i' \in \mathcal{I}$  と置く.  $Ar_a(t, i')$  は渋滞の先頭に位置するため, 式 (118) より

$$r_a(t, r_a(t, i')) = 0 \quad (129)$$

が成り立つ. このとき,  $Ar_a(t, i')$  は *Follow vehicle* のため式 (123), (129) より

$$\xi(t, r_a(t, i')) = 0 \quad (130)$$

が成り立つ. Algorithm 6, 7 および式 (26), (122), (130) より,  $\sigma_{rl}(t)$  は

$$\sigma_{rl}(t) = p(t, i') \quad (131)$$

と求まる. □

**定理 2** より車群の先頭に位置するエージェントの加速で、 $\sigma_{rl}(t)$  の確保が可能であると言える。また、新たなエージェントの車群への合流間隔は、車群内外のエージェントの相対速度に反比例する。したがって、渋滞先頭に位置するエージェントが Follow vehicle で、且つ車群内外のエージェントの移動速度が等しい場合、利他的な車線変更によって生じた渋滞の解消速度は渋滞の拡大速度を上回る。

次に、渋滞の先頭に位置するエージェントが Front vehicle , Assistant vehicle の場合の渋滞解消速度について述べる。このとき、 $\sigma_{rl}(t)$  は**定理 3** を満たす。

**定理 3.** Front vehicle の車線変更が達成されない場合、 $\sigma_{rl}(t) = 0$  となる。

**証明.** 渋滞先頭に位置する Front vehicle の ID を  $i'$  と置く。式 (122), (126), Algorithm 7 より、 $\mathcal{A}i'$  の車線変更が達成されない場合、 $\mathcal{A}i$  が Front vehicle, Assistant vehicle, Current vehicle いずれかであれば

$$\neg(\xi(t, i) = 1) \vee (\xi(t + 1, i) = 1) \quad (132)$$

が真となる。このとき、Algorithm 6, 7, 式 (122), (132) より、 $\mathcal{A}i$  が Follow vehicle の場合も式 (132) は真となる。したがって、Front vehicle の車線変更が達成されない場合、 $\sigma_{rl}(t) = 0$  となる。□

**定理 3** より、Front vehicle の車線変更要求の達成は、車線変更に起因する渋滞解消の必要条件である。また、**定理 2** より、Front vehicle の車線変更完了後は渋滞の解消速度が拡大速度を上回るため、渋滞の解消が保証される。したがって、一般車両と自動運転車両が混在する状況下での自動運転車両の制御では、他のエージェントの車線変更・減速要求に対する協調機能を持たない車両が Front vehicle と Assistant vehicle に設定されないように考慮する必要がある。

#### 7.4 非協調エージェントを考慮した協調動作

本節では、協調動作に起因する渋滞の発生とその解消条件に基づき、自動運転車両と手動運転車両の混在流における車線変更のアルゴリズムを検討する。ここで、自動運転車両は利他的な車線変更と車線変更のための減速を協調機能を持つ協調型エージェントである。対して、手動運転車両は利他的な車線変更と減速を行わない非協調型エージェントである。本論文では、車速の計測は車載センサで行うものとし、エージェント間での通信は車線変更・減速要求の設定と達成の通知にのみ利用する。手動運転車両は車載センサによって周囲のエージェントの車速測定を利用した、自身の車速確保のための車線変更の機能を持つものとする。

自動運転車両の車群形成と車線変更においては、エージェント間のマンハッタン距離に加えて通信情報を自身の行動選択に加味しない非協調型エージェントによる通信制限を考慮する。以降では、協調型エージェントに対する通信保証範囲を目標車間距離  $d$  で与える。非協調型エージェントは通信保証範囲内に他のエージェントが存在する場合も、通信情報は利用しない。また、非協調型エージェントは行動要求の設定と達成状況を含めて、情報共有のための通信による情報発信は行わないものとする。

**定理 2, 3** より、Front vehicle の車線変更達成は利他的な車線変更で生じる渋滞の解消に対する必要条件である。式 (126) について、行動要求  $r_c(t, i)$  および  $r_d(t, i)$  に基づく車線変更と減速は協調動作に該当する。したがって、 $r_c(t, i)$  と  $r_d(t, i)$  に対応する渋滞長の減少には、 $\mathcal{A}r_c(t, i)$  と  $\mathcal{A}r_d(t, i)$  が通信可能である必要がある。言い換えると、車線変更要求は、エージェントの協調動作である利他的な車線変更によって達成が判断される。そのため、Front vehicle には協調型エージェントを設定する必要がある。同様に、車間確保のための減速は協調動作に該当する。そのため、Assistant vehicle に非協調型エージェントが設定された場合、車間確保に失敗するため Front vehicle による利他的な車線変更

が困難になる．対して，Follow vehicle による加速要求は，非協調動作である区画移動によって達成される．すなわち，利他的な車線変更で生じる渋滞を解消するためには，非協調型エージェントの Front vehicle, Assistant vehicle への設定を避ける必要がある．また，協調型エージェントと非協調型エージェントの区別のために，エージェントの通信状態を利用する． $c(i) \in \{0, 1\}$  は  $\mathcal{A}_i$  の通信状態を表す． $\mathcal{A}_i$  が協調型エージェントの場合， $c(i)$  は 1 となる．本論文では，Demand-for graph  $G_d(t)$  と通信状態  $c(i)$  に対して

$$\neg(((n_i, n_j) \in A_d(t)) \wedge (c(j) = 0)) \vee (r_a(t, i) = j) \quad (133)$$

を満たす車線変更要求  $r_c(t, i)$  と減速要求  $r_d(t, i)$  の設定を目指す．

本論文では利他的な車線変更の車群形成に対して，新たに要求棄却機能を導入する．Current vehicle は Front vehicle の設定時に，対象のエージェントが非協調型の場合は車線変更要求の設定を中止する．このとき，車線変更要求  $r_c(t, i)$  は  $c(i)$  に対して

$$r_c(t, i) (1 - c(r_c(t, i))) = 0, \quad \forall i \in \{1, 2, \dots, \mathcal{I}\} \quad (134)$$

を満たす．

また，Front vehicle は Assistant vehicle を設定する際に，対象のエージェントの通信状態を確認する．対象が非協調型エージェントの場合は減速要求の設定を中止し，Current vehicle に車線変更要求の棄却を通知する．このとき，減速要求  $r_d(t, i)$  は  $c(i)$  に対して，

$$r_d(t, i) (1 - c(r_d(t, i))) = 0, \quad \forall i \in \{1, 2, \dots, \mathcal{I}\} \quad (135)$$

を満たす．また， $r_c(t, i)$ ,  $r_d(t, i)$  に対して

$$r_d(t, r_c(t, i)) (1 - c(r_d(t, r_c(t, i)))) = 0, \quad \forall i \in \{1, 2, \dots, \mathcal{I}\} \quad (136)$$

が成り立つ．

式 (134), (135), (136) より，エージェントの要求は式 (133) を満たす．

## 7.5 数値実験

本章では利他的な車線変更で発生する渋滞による流量低下と車線変更達成後の Current vehicle の加速による交通円滑化による流量上昇の効果を評価する．自動運転車両の車線変更では，渋滞と交通円滑化の影響の評価指標として，エージェントの密度と流量を用いる．

時刻  $t$  におけるエージェントの密度を  $\rho(t)$  ( $0 \leq \rho(t) \leq 1$ )，探索範囲全体での平均流量を  $Q(t)$  ( $0 \leq Q(t) \leq 1$ ) と表す．密度に対して流量が高くなるほど，個々のエージェントの移動速度が高くなる．本論文では，エージェントの密度  $\rho(t)$  と  $Q(t)$  をそれぞれ

$$\rho(t) = \sum_{v_y^x \in N} \sum_{i \in \mathcal{I}} \varepsilon(t, i, v_y^x) / |N|, \quad (137)$$

$$\begin{aligned} Q(t) &= \sum_{i \in \mathcal{I}} f_{17}(t, i) / |N| \\ f_{17}(t, i) &= \begin{cases} 1 & (\text{if } \delta_{\mathcal{A}}(t, i) \neq \varepsilon_{\mathcal{A}}(t, i)) \\ 0 & (\text{otherwise}) \end{cases} \end{aligned} \quad (138)$$

と求める．

利他的な車線変更による渋滞と交通円滑化の影響を数値実験により評価する．図 29 に探索対象の複数レーン ASEP を示す．

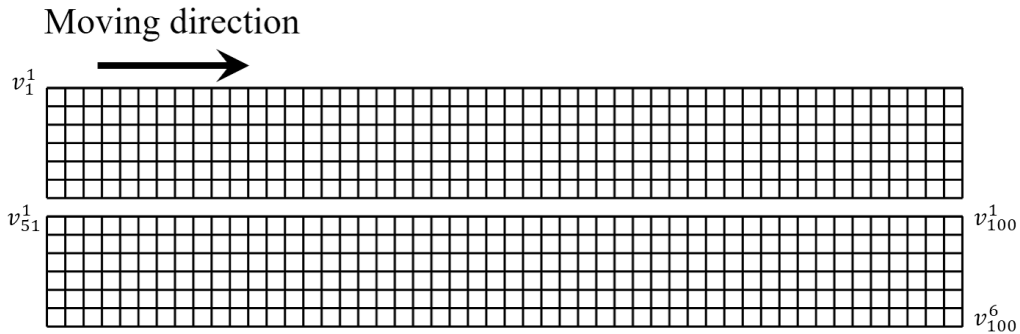


図 29: 探索対象 (自動運転)

図 29 はレーン数  $X = 6$ 、長さ  $Y = 100$  の複数レーン ASEP である．図 29 の左上が  $v_1^1$ 、右下が  $v_{100}^6$  である． $v_1^x$  は単路の流入口であり、エージェントが存在しない場合は確率 0.5 で新たなエージェントが生成される． $v_{100}^x$  は単路の流出口であり、時刻  $t$  に  $v_{100}^x$  に存在するエージェントは時刻  $t+1$  に消滅する．

本実験では利他的な車線変更を行わない場合と、目標車間距離  $d = 5$  で車線変更を行う場合の定常流について密度と流量を比較する．利他的な車線変更を行わない場合、交通流の密度と流量は全エージェントが非協調型のエージェントの交通流の結果と一致する．目標車間距離  $d = 5$  で車線変更を行う場合、全エージェントはネットワーク  $N$  上でマンハッタン距離  $d$  以下の範囲に位置するエージェントとの通信による情報共有を行う．

文献 [61] では利他的な車線変更によるエージェントの速度によるレーン分けと、それによる渋滞緩和の効果が報告されている．また、利他的な車線変更に起因する渋滞は交通流の上流側でより多く発生すると予想される．これに対して、本博士論文で示した定理 2 は、協調型エージェントの交通流および要求棄却を導入した場合において利他的な車線変更で生じる渋滞は時間経過による自然解消が保証されることを意味している．したがって、本実験の利他的な車線変更を導入した場合の交通流について、交通流の下流側では渋滞緩和による密度の低下と速度に基づくレーン分けが予想される．

はじめに、車線変更を行わない場合の定常流を示す．本実験では、3000 ステップの間エージェントの移動を繰り返して定常流を生成し、 $\rho(3000)$  と  $Q(3000)$  を定常流の密度と流量として扱う．車線変更を行わない場合の  $t = 3000$  の複数レーン ASEP の状態を図 30 に示す．図 30 において、赤色の丸は移動確率 0.75、黄色の丸は移動確率 0.50、水色の丸は移動確率 0.25 のエージェントを表す．このとき、最終時刻の密度は  $\rho(3000) = 0.59$ 、流量は  $Q(3000) = 0.1117$  となった．

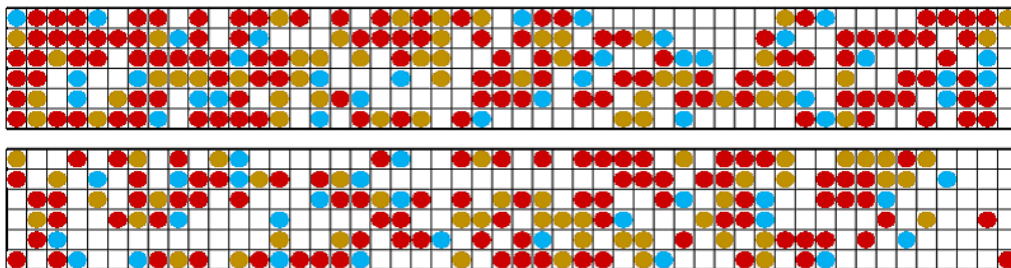


図 30: シミュレーション結果 (利他的な車線変更なし)

次に、目標車間距離  $d = 5$  で車線変更を行う場合の定常流を示す。車線変更を行わない場合と同様に、3000 ステップの間エージェントの移動を繰り返して定常流を生成し、 $\rho(3000)$  と  $Q(3000)$  を定常流の密度と流量として扱う。車線変更を行わない場合の  $t = 3000$  の複数レーン ASEP の状態を図 31 に示す。図 31 において、赤色の丸は移動確率 0.75、黄色の丸は移動確率 0.50、水色の丸は移動確率 0.25 のエージェントを表す。このとき、最終時刻の密度は  $\rho(3000) = 0.375$ 、流量は  $Q(3000) = 0.1317$  となった。

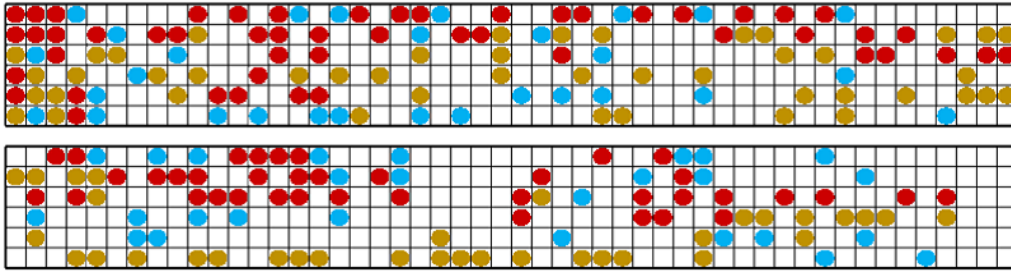


図 31: シミュレーション結果 ( $d = 5$ )

図 31 より、上流側比べて下流側では車間距離が確保され密度が下がっていることがわかる。また、下流側では移動確率 0.75 のエージェントが上側のレーンに集中している様子が見られた。これは、上流で利他的な車線変更が行われ、個々のエージェントの移動速度が確保された効果である。また、利他的な車線変更を行わない場合と利他的な車線変更を実施する場合には、利他的な車線変更を実施する場合に  $\rho(3000)$  は減少し  $Q(3000)$  は増加している。この結果より、利他的な車線変更による探索システム全体の移動速度向上による交通円滑化は、車線変更で生じる渋滞による移動速度低下の影響を上回ることが示せた。

## 8 まとめ

### 8.1 結論

本博士論文ではマルチエージェントシステムのグラフ探索における渋滞緩和の評価手法として、エージェントの協調・非協調動作で生じる行動要求に基づく渋滞解析手法を提案した。渋滞解析に際してエージェントの協調・非協調動作の干渉の表現手法として、新たに動作に対する行動要求の関係を表す有向グラフ Demand-for graph を提案した。Demand-for graph は行動要求に対するエージェント間の待ち状態の関係から、エージェントの協調・非協調動作に起因する渋滞の長さとの幅の評価を可能にする。また、エージェントの協調・非協調動作による行動要求の設定・達成に基づき、渋滞の発生と拡大・解消速度の評価を可能にする。

提案する Demand-for graph による渋滞解析を、マルチエージェントシステムの迷路探索、自動倉庫内の搬送問題、自動運転車両の車線変更に適用した。各適用先での Demand-for graph を用いた渋滞解析により、協調機能に基づく自然渋滞の解消の評価、自然発生した渋滞の協調機能を介さない自然解消との比較、協調機能によって新たに生じる渋滞とその解消の評価に取り組んだ。各適用先ではそれぞれ非協調動作に起因するデッドロックの協調動作による解消、非協調動作に起因する渋滞の非協調動作による自然解消、協調動作に起因する渋滞とその解消に対してデッドロックと渋滞の解消条件を導出した。迷路探索問題ではデッドロック解決のための協調動作である疑似移動の所要時間の概念を導入し、デッドロック解消の所要時間と疑似移動による流量増加の比較により渋滞緩和の条件を示した。搬送問題では渋滞幅の拡大予想とデッドロック解決の観点から、渋滞緩和のための再経路計画手法と疑似移動によるデッドロック回復を提案した。自動運転車両の車線変更では利他的な車線変更に起因する渋滞の解消条件に基づき、非協調型エージェント混在時の利他的な車線変更アルゴリズムを示した。Demand-for graph の各適用先での個別課題を通じて、エージェントの協調動作による渋滞緩和効果の定量評価を実現した。また、協調動作の実現における、エージェント間の通信機能の要件を明らかにした。

### 8.2 今後の発展性

本博士論文では、エージェントの協調・非協調動作の Demand-for graph 表現により、エージェントの協調動作による渋滞緩和効果の理論的解析を実現した。本成果の発展性として、既存の協調機能による渋滞緩和効果の Demand-for graph による評価の適用、および渋滞緩和のための新たな協調機能の提案とその実装が挙げられる。

既存の協調機能に対して Demand-for graph による渋滞解析を適用することで、提案技術の実運用における交通渋滞の発生とその解消の評価が可能となる。

新たな協調機能の提案に関して渋滞緩和のための協調機能の実現方法は、協調的経路計画による迂回路の選択や、スケジューリング、エージェントの軌道・速度制御が挙げられる。このとき、エージェントの非協調動作に対する Demand-for graph による渋滞解析結果を活用することで、渋滞の発生・拡大原因の特定と協調機能による効果的な渋滞緩和が期待できる。また、その実装における課題として、既存の通信手段の制約下での協調動作の実現と通信機能の改善が挙げられる。

## 謝辞

本研究を行うにあたり、学部4年から長く指導してくださった澤田賢治准教授に深く感謝申し上げます。博士前期課程の修了後、社会人博士として再び大学で研究に従事する機会をいただきました。学部生のときから変わらず発言における自らの主張の整理を苦手としており、論文投稿や学会発表の際にはご迷惑をおかけすることもありましたが都度丁寧にご指導いただきました。

出張や研究環境の整備に関しましては、長田様にお世話になりました。登学の際には、登学の頻度の低い私のことを気遣っていただきました。

また、澤田研究室の学生の皆様にも感謝申し上げます。ゼミや輪講を通じて様々な刺激をいただき、本博士論文の成果につながる発想を得ることができました。

最後に、進学のお機会を与えてくださったアズビル株式会社の村中敦様、水上正様に感謝申し上げます。



## 参考文献

- [1] V. J. Lumelsky: A Comparative Study on the Path Length Performance of Maze-Searching and Robot Motion Planning Algorithms; *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 1, pp. 57–66 (1991)
- [2] K. Knight: Are Many Reactive Agents Better Than a Few Deliberative Ones?; *Proceedings of the Thirteen International Joint Conference on Artificial Intelligence*, pp. 432–437 (1993)
- [3] 横尾真：淘汰を用いたマルチエージェント実時間探索の高速化：協調探索への競争の導入；コンピュータソフトウェア, Vol. 14, No. 4, pp. 47–55 (1997)
- [4] R. S. Bharathi, R. Priyadharshni, S. Ramesh: Artificial Bee Colony Algorithm to Find Optimum Path for Mobile Agents in Wireless Sensor Networks; *IEEE International Conference on Advanced Communication Control and Computing Technologies*, pp. 842–846 (2014)
- [5] K. Kobayashi, M. Kido and Y. Yamashita: Computationally Efficient Model Predictive Control for Multi-Agent Surveillance Systems; *IEICE Trans. Fundamentals*, VOL. E102-A, NO. 2 pp. 372–378 (2019)
- [6] S. Lee and Y. Kim: Cooperative Reactive Persistent Surveillance Algorithm Using Multiple UAVs Considering Incident Arrivals; *20th World Congress the International Federation of Automatic Control*, pp. 2383–2388 (2017)
- [7] X. Bai, M. Cao, W. Yan and S. S. Ge: Efficient Routing for Precedence-Constrained Package Delivery for Heterogeneous Vehicles; *IEEE Transactions on automation science and engineering*, Vol. 17, No. 1, pp. 248–260(2020)
- [8] B. H. Kazerouni, M. B. Moradi and P. H. Kazerouni: Variable Priorities in Maze-Solving Algorithms for Robot’s Movement; *IEEE International Conference on Industrial Informations* Vol. 6, pp. 181–185 (2003)
- [9] E. H. Kivelevich and K. Cohen: Multi-Agent Maze Exploration; *Journal of Aerospace Computing, information, and communication*, Vol. 7, pp. 391–405 (2010)
- [10] Guangming Xiong, Jianwei Gong, Huiyan Chen and Zhibao Su: Multi-robot Exploration Based on Market Approach and Immune Optimizing Strategy; Third International Conference on Autonomic and Autonomous Systems (ICAS’07), pp. 1-6 (2007)
- [11] Shiguang Wu, Xiaojie Liu, Xingwei Wang, Xiaolin Zhou and Mingyang Sun: Multi-robot Dynamic Task Allocation Based on Improved Auction Algorithm; 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE), pp. 1-5 (2021)
- [12] Farouq Zitouni, Saad Harous and Ramdane Maamri: A Distributed Approach to the Multi-Robot Task Allocation Problem Using the Consensus-Based Bundle Algorithm and Ant Colony System; *IEEE Access*, Vol. 8, pp. 27479–27494 (2020)
- [13] Faiza Gul, Imran Mir, Wan Rahiman and Tauqeer Ul Islam: Novel Implementation of Multi-Robot Space Exploration Utilizing Coordinated Multi-Robot Exploration and Frequency Modified Whale Optimization Algorithm; *IEEE Access*, Vol. 9, pp. 22774–22787 (2021)
- [14] 加藤 晋, 西山 栄枝, 武野 純一：交通ルールの適用による複数の移動ロボットの協調行動, 日本ロボット学会誌, Vol. 12, No. 2, pp. 291–298 (1994)

- [15] 岩朝 睦美, 戸田 雄一郎, 久保田 直行: 予測可能な移動障害物のある環境における時空間グラフを用いた大域的経路探索と行動計画; 日本機械学会論文集, Vol. 85, No. 876, pp. 1-14 (2019)  
<https://doi.org/10.1299/transjsme.18-00254>
- [16] 覺張陽則, 高瀬國克, 羽田芳郎: 複数台移動ロボットのためのデッドロックフリーナビゲーション法, 電気学会論文誌C (電子・情報・システム部門誌), Vol. 127, No. 3, pp. 380-388 (2007)
- [17] 望月優加理: 自律協調ロボット群による複数目的地探索の研究, 平成 27 年度電気通信大学知能機械工学科卒業論文 (2016)
- [18] 望月優加理, 澤田賢治, 新誠一: 通信可能範囲を考慮した分散協調制御システムに基づく迷路探索; システム制御情報学会論文誌, Vol. 31, No. 4, pp. 167-176 (2018)
- [19] 望月優加理, 澤田賢治, 新誠一: 分散協調制御システムに基づく迷路探索問題における通信手段と探索効率の関係; システム制御情報学会論文誌, Vol. 32, No. 3, pp. 101-112 (2019)
- [20] 望月優加理, 澤田賢治: ヘテロジニアスなエージェント群のグラフ探索における役割交換と探索効率の関係; システム制御情報学会論文誌, Vol.34, No.10 (2021)
- [21] 西野 智也, 石井 光治: CSMA/CA に基づく合意形成のためのバックオフ期間制御; 電気学会論文誌C (電子・情報・システム部門誌), Vol. 141, No. 9, pp. 1008-1015 (2021)
- [22] 吉岡 愛, 滑川 徹: マルチエージェントシステムの合意問題とそのフォーメーション制御への応用; 計測自動制御学会論文集, Vol. 44, No. 8, pp. 663-669 (2008)
- [23] Kentaro Kobayashi, Shunsuke Noro, Hiraku Okada and Masaaki Katayama: Adaptive transmission probability for CSMA/CA-based consensus control of multi-agent systems; Vol. 8, No. 9, pp. 393-398 (2019)
- [24] 桜間 一徳: マルチエージェントシステムの制御: III 合意制御 (1); システム/制御/情報, Vol. 57, No. 9, pp. 386-396 (2013)
- [25] Arianna Alfieri, Marco Cantamessa, Anna Monchiero and Francesca Montagna: Heuristics for puzzle-based storage systems driven by a limited set of automated guided vehicles; Journal of Intelligent Manufacturing, Vol. 23, No. 5, pp. 1-11 (2012)
- [26] K.J.C.Fransen, J.A.W.M.van.Eekelen, A.Pogromsky, M.A.A.Boon and I.J.B.F.Adan: A dynamic path planning approach for dense, large, grid-based automated guided vehicle systems; Computers and Operations Research, Vol. 123, pp. 1-10 (2020)
- [27] R. (M.) B. M. de Koster, T. Le-Duc and Y. Yugang: Optimal storage rack design for a 3-dimensional compact AS/RS; *International Journal of Production Research*, Vol. 46, No. 6, pp. 1495-154 (2008)
- [28] Kevin R. Gue, Kai Furmans, Zăzilia Seibold and Onur Uludağ: GridStore: A Puzzle-Based Storage System With Decentralized Control; *IEEE Transactions on Automation Science and Engineering*, Vol. 11, No. 2, (2014)
- [29] Kevin R. Gue: Very high density storage systems; *IIE Transactions*, Vol. 38, No. 1, pp. 79-90 (2007)
- [30] Ly Gia Bao, Truong Giang Dang and Nguyen Duy Anh: Storage Assignment Policy and Route Planning of AGVS in Warehouse Optimization; 2019 International Conference on System Science and Engineering (ICSSE), pp. 599-604 (2019)
- [31] J. Bartholdi and L. Platzman: Retrieval Strategies for a Carousel Conveyor; *IIE Transactions*, Vol. 18, No. 2, pp. 166-173 (1986)

- [32] Xiao Cai, Sunderesh S. Heragu and Yang Liu: Modeling Automated Warehouses Using Semi-Open Queueing Networks; *Handbook of Stochastic Models and Analysis of Manufacturing System Operations*, Springer, pp. 29-71 (2013)
- [33] I.F. Akyildiz and A. Sieber: Approximate analysis of load dependent general queueing networks; *IEEE Transactions on Software Engineering*, Vol. 14, No. 11, pp. 1537-1545 (1988)
- [34] Johanna Bolanos-Zuñiga, Jania Saucedo, Tomas Salais and Jose Antonio Marmolejo: Optimization of the Storage Location Assignment and the Picker-Routing Problem by Using Mathematical Programming; *Applied Sciences*, Vol. 10, No. 2 (2020)
- [35] Lin Xie, Nils Thieme, Ruslan Krenzler and Hanyi Li: Introducing split orders and optimizing operational policies in robotic mobile fulfillment systems; *European Journal of Operational Research*, Vol. 288, No. 1, pp. 1-38 (2020)
- [36] Bipan Zou, René B.M. De Koster, Yeming Gong and Xianhao Xu: Robotic Sorting Systems: Performance Estimation and Operating Policies Analysis; *Transportation Science*, Vol. 55, No. 2, pp. 1-26 (2021)
- [37] Warren H. Hausman, Leroy B. Schwarz and Stephen C. Graves: Optimal Storage Assignment in Automatic Warehousing Systems; *Management Science*, Vol. 22, No. 6, pp. 629-638 (1976)
- [38] Banu Y. Ekren and Sunderesh S. Heragu: Simulation based regression analysis for rack configuration of autonomous vehicle storage and retrieval system; Proceedings of the 2009 Winter Simulation Conference (WSC), pp. 2405-2413 (2009)
- [39] Sonja Otten, Ruslan Krenzler, Lin Xie and Lüneburg Hans Daduna Analysis of semi-open queueing networks using lost customers approximation with an application to robotic mobile fulfillment systems; *OR Spectrum* (2021)
- [40] N. Wu and M. C. Zhou: AGV Routing for Conflict Resolution in AGV Systems; *International Conference on Robotics & Automation*, pp. 14-19 (2003)
- [41] 小泉賢司, 増山繁: 梯子型走行経路を用いた AGV システムにおけるデッドロック回復問題の計算複雑さ, システム制御情報学会論文誌, Vol. 23, No. 5, pp. 91-104 (2010)
- [42] L. Kalinovic, T. Petrovic, D. Bogdan and V. Bobanac: Modified Banker's algorithm for scheduling in multi-AGV systems; *IEEE International Conference on Automation Science and Engineering*, pp. 351-356 (2011)
- [43] Dan Song, Yadong Li and Tengtao Song: Modified Banker's algorithm with dynamically release resources; 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), pp. 566-569 (2021)
- [44] S. A. Reveliotis: Conflict resolution in AGV systems; *IIE Transactions*, Vol. 32, pp. 647-659(2000)
- [45] Agha Ali Haider Qizilbash, Christian Henkel and Sanaz Mostaghim: Ant Colony Optimization based Multi-Robot Planner for Combined Task Allocation and Path Finding; 2020 17th International Conference on Ubiquitous Robots (UR), pp. 487-493 (2020)
- [46] A. Schadschneider and M. Schreckenberg: Cellular automaton models and traffic flow; *Journal of Physics A:Mathematical and Theoretical*, Vol. 26, No. 15, pp. 1-7 (1993)
- [47] 江崎貴裕, 今村卓史: ネットワーク上の非対称単純排他過程 (ASEP), 応用数理, Vol. 23, No. 5, pp. 7-14 (2013)

- [48] Neri, I, Kern N, and Parmeggiani, Totally asymmetric simple exclusion process on networks, *Phys. Rev. Lett.*, Vol. 107, 068702 (2011)
- [49] Ezaki, T and Nishinari, K: A balarlcnetwork for the asymmetric simple exclusion process, *J. Stat. Mech.* (2012)
- [50] Ezaki, T . and Nishinari, K: Exact solution of a heterogeneous multilane asymmetric simple exclusion Process; *Phys . Rev . E* , 84 (2011)
- [51] Ezaki, T . and N. ishinari, K . : Exact stationary distribution of an asymmetric simple excluion process with Langmuir kineticsand memory reservoirs; *J. Phys, A* , 45 (2012)
- [52] Chihiro Suwa and Kaoru Fujioka: A Cellular Automaton Model of Traffic with Railroad Crossing; 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), pp. 551-554 (2015)
- [53] Zhengtao Xiang, Juan Bao, Yujin Li and Li Xiong: A Cellular Automaton Traffic Model Considering the Influence of Driving State on Randomization; 2013 International Conference on Computational and Information Sciences, pp. 1158-1161 (2013)
- [54] 友枝 明保：「渋滞」を分析する数理モデルとしてのセルオートマトンとその応用, システム／制御／情報学会誌, Vol. 63, No. 7, pp. 271-277 (2020)
- [55] 和田 健太郎：道路ネットワークの渋滞マネジメントー基本原則と喫緊課題への応用ー, オペレーションズ・リサーチ, No. 8, pp. 421-428 (2020)
- [56] 石川翔太, 荒井幸代：渋滞低減に向けた路車間・車車間協調を実現する自動運転方策の学習法, 人工知能学会論文誌, Vol. 34, No. 1, pp. 1-9 (2019)
- [57] 奥田裕之, 田崎勇一, 鈴木達也：円滑な交通を実現する車両間調和のための制御技術, 計測と制御, Vol. 57, No. 3, pp. 185-192 (2018)
- [58] M. Forster, R. Frank, M. Gerla, T. Engel: A cooperative advanced driver assistance system to mitigate vehicular traffic shock waves; *Proceedings 2014 IEEE INFOCOM*, pp. 1968–1976 (2014)
- [59] M. Won, T. Park, S. H. Son: Toward mitigating phantom jam using vehicle-to-vehicle communication, *IEEE Transactions on Intelligent Transportation Systems*, Vol.18, No.5, pp.1313–1324 (2017)
- [60] T. Kim, K. Jerath: Congestion-Aware Cooperative Adaptive Cruise Control for Mitigation of Self-Organized Traffic Jams, *IEEE Transactions on Intelligent Transportation Systems*, pp. 1-12 (2021)
- [61] N. Wang, X. Wang, P. Palacharla and T . Ikeuchi :Cooperative Autonomous Driving for Traffic Congestion Avoidance through Vehicle-to-Vehicle Communications; *IEEE Vehicular Networking Conference (VNC)*, pp. 327-330 (2017)
- [62] 能登 紀泰, 奥田 裕之, 田崎 勇一, 鈴木 達也, 早川 聡一郎, 三田 誠一, 片山 正昭, 大日方 五郎：排他機構を組み込んだコンセンサス制御による協調的自動駐車システム, 計測自動制御学会論文集, Vol. 49, No. 11, pp. 986-993 (2013)
- [63] 坂井 康一, 小野 晋太郎, 杉町 敏之, 貝塚 勉, 鈴木 美緒, 平沢 隆之, 大口 敬, 須田 義大, 中野 公彦, 大石 岳史：自動運転車両を含めた高速道路の運用施策の事前評価のための仮想交通実験環境の要件にかかる研究, 生産研究, vol. 71, No. 2 (2019)
- [64] 飯田 克弘, 和田 泰明, 多田 昌裕, 筑後 智弘, 安 時亨, 澤田 英郎, 紀ノ定 保礼：ACC 車両の混

- 在比率が異なる交通流の安全性・円滑性評価, 交通工学論文集, vol. 4, No. 1 (2018)
- [65] 岡野 舜, 高山 宇宙, 三浦 清洋, 森本 章倫: レベル4の自動運転車導入における乗降環境を考慮した街路空間に関する研究, 交通工学論文集, vol. 6, No. 2 (2020)
- [66] Tung Thanh Phan, Long Bao Le and Dong Ngoduy: A Cooperative Space Distribution Method for Autonomous Vehicles at A Lane-Drop Bottleneck on Multi-Lane Freeways; *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 4, pp. 3710-3723 (2022)
- [67] Vishal Mahajan, Christos Katrakazas and Constantinos Antoniou: Crash Risk Estimation Due to Lane Changing: A Data-Driven Approach Using Naturalistic Data; *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 4, pp. 3756-3765 (2022)
- [68] Shukai Chen, Hua Wang and Qiang Meng: An Optimal Dynamic Lane Reversal and Traffic Control Strategy for Autonomous Vehicles; *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 4, pp. 3804-3815 (2022)
- [69] Peng Hang, Chen Lv, Chao Huang, Yang Xing and Zhongxu Hu: Cooperative Decision Making of Connected Automated Vehicles at Multi-Lane Merging Zone: A Coalitional Game Approach; *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 4, pp. 3829-3841 (2022)
- [70] Negar Mehr, Roberto Horowitz: How Will the Presence of Autonomous Vehicles Affect the Equilibrium State of Traffic Networks?; *IEEE Transactions on Control of Network Systems*, Vol. 7, No. 1, pp. 96-105 (2020)
- [71] Pavankumar Tallapragada and Jorge Cortés: Hierarchical-Distributed Optimized Coordination of Intersection Traffic; *IEEE Transactions on Intelligent Transportation Systems*, Vol. 21, No. 5, pp. 2100-2113 (2020)
- [72] Soojung Lee and Youngjun Lee: A Distributed Algorithm for Deadlock Detection under OR-Request Model; *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems* (1999) <https://dl.acm.org/doi/10.5555/829524.831042>
- [73] Vandana Kate, Akansha Jaiswal and Ambika Gehlot: A Survey on Distributed Deadlock and Distributed Algorithms to Detect and Resolve Deadlock; *2016 Symposium on Colossal Data Analysis and Networking (CDAN)* (2016)
- [74] Michel Raynal: Simple Deadlock Detection for the And-Communication Model; *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems* (2014)
- [75] Shigang Chen, Yi Deng, P. Attie and Wei Sun: Optimal Deadlock Detection in Distributed Systems Based on Locally Constructed Wait-for Graphs; *proceedings of 16th International Conference on Distributed Computing Systems*, pp. 613-619 (1996)
- [76] Maoning Chen, Yuangen Lu and Canrong Zhang: Deadlock-solving Traffic Control Methods for Automated Guided Vehicle Systems; *2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 1-7 (2021)
- [77] 村上誉, 石津健太郎, 川崎輝, 伊深和雄, フィリン スタニラブ, 松村武, 児島史秀: ローカル5Gの現状と今後の展望, *通信ソサイエティマガジン*, No. 55, 冬号 (2020)
- [78] 高玉 圭樹, 大谷 雅之: 大規模構造物組み立てにおける複数移動ロボットの主従関係交換によるデッドロック回避, 計測と制御, Vol. 52, No. 3, pp. 271-276 (2013)
- [79] 山崎公俊, 友納正裕, 坪内孝司, 油田信一: 未知形状の小型物体を自動で拾い上げる移動マニピュレータ, *ロボティクス・メカトロニクス講演会講演概要集 2006*, 2P1-D24 (2006)

- [80] A. Siberschatz, P. B. Galvin and G. Gagne: Operating System Concepts; *Wiley Publishing* (2005)
- [81] O. Ore: Theory of Graphs; *American Mathematical Society* (1962)

## 博士(工学) 学位申請 関連論文

(2022年6月現在)

	著者 題名 発表・発行掲載誌名 発表・発行年月日
投稿論文 (査読付き)	<ol style="list-style-type: none"><li>(1) 望月優加理, 澤田賢治, 新誠一, 通信可能範囲を考慮した分散協調制御システムに基づく迷路探索, システム制御情報学会論文誌, Vol.31, No.4, 2018</li><li>(2) 望月優加理, 澤田賢治, 新誠一, 分散協調制御システムに基づく迷路探索問題における通信手段と探索効率の関係, システム制御情報学会論文誌, Vol.32, No.3, 2019</li><li>(3) 望月優加理, 澤田賢治, ヘテロジニアスなエージェント群のグラフ探索における役割交換と探索効率の関係, システム制御情報学会論文誌, Vol.34, No.10, 2021</li><li>(4) Yukari Mochizuki, and Kenji Sawada, An analysis of expansion and reduction speeds of traffic jams on graph exploration, Artificial Life and Robotics, 2022</li></ol>