東京大学工学系研究科システム創成学専攻

修士論文

経路選択行動の学習を考慮した マルチエージェント交通流シミュレーション

Multi-Agent based Traffic Simulation
with Reinforcement Learning of Route Selection Behavior

2012 年3 月5 日

指導教員　吉村忍　教授

学籍番号　37-106316

内田英明

目 次

図 目 次

表 目 次

第1章　はじめに

小目次

第1章　はじめに

1.1　研究の目的と背景

道路交通は現代社会の基盤システムであると同時に大気汚染や交通渋滞などの問題を生んできた．これらの問題を解決するため様々な施策が提案されており，近年ではその評価に交通流シミュレーションが採用されることが多い．これは交通施策の実証実験には多大なコストが必要となり，代替手段としてのシミュレーションの需要が大きくなっているためである．しかしシミュレーションにおいて再現される交通流にはいくつかの注意すべき性質が存在する．その一つが本研究で扱う運転者の経路選択行動である．

多くのシミュレータにおいて運転者の経路選択はネットワークのコスト情報に基づいて行われる．一方，現実の運転者は限定された周囲の状況と経路に対するこれまでの知識に基づいてルートを選択する．この差異はシミュレーションによって再現される交通現象に大きく影響する．例えば，交通施策の実施によりある時点から道路ネットワークが変化するという条件のもとでシミュレーションを行うとすると，ネットワークのコスト情報を瞬時に利用できるモデルでは常に各運転者が最適な経路選択を行うため，交通状況は初めから定常的な状態に落ち着く．しかし，過去の走行経験に基づいて経路選択が行われるとすれば，知識を修正していく過程で過渡的な交通状況が生じると考えられる．交通施策の策定を行う自治体などにとって，この過渡期における現象を事前評価することは，充分な時間経過の後収束する定常状態を予測することと同様に重要である．

そこで本研究では，知的マルチエージェント型交通流シミュレータ MATES に強化学習に基づいたルーティングアルゴリズムである Q-routing を導入し，過去の経験に基づいた経路選択の実現を目指す．同時に，岡山市で検討されている路面電車の軌道延伸計画に関し，実施直後の過渡的な交通状況から定常状態に収束するまでをシミュレーションする．

1.2 本論文の構成

本論文の構成は以下のとおりである。まず 2 章において交通工学とそれに基づく交通流シミュレータの基礎理論についてまとめ，本研究で使用する知的マルチエージェント型交通流シミュレータ MATES の概要について述べる．また，本研究の問題設定を行う．3 章では強化学習の概要についてまとめる．4 章で本研究の提案手法である Q-routing の概要と，交通流に適用するにあたっての改良について説明し，関連研究との比較を行う．5 章では不規則格子の仮想環境でのシミュレーションと，現実の岡山市の路面電車軌道延伸計画に関するシミュレーションを行い結果を考察する．最後に 6 章において本研究における結論を述べる．

第2章　交通流シミュレーション

小目次

第2章　交通流シミュレーション

本章では，交通流シミュレーションの概要について述べる．まず 2.1 節において，交通流シミュレーションに求められている社会的役割について説明する．次いで，2.2 節では交通流の基本量についてまとめ，それぞれの関係と性質について説明する．2.3 節では現在提案されている 3 つの代表的な交通流モデルを紹介する．2.4 節において，交通流シミュレータを 2 つのモデルに分類し，交通流モデルとの対応とそれぞれの特性を説明する．2.5 節では本研究で使用する知的マルチエージェント型交通流シミュレータ MATES の概要を説明し，最後に 2.6 節で本研究の問題設定を行う．

2.1　シミュレーションの意義

自動車や道路，公共交通インフラの普及により今世紀の経済・産業は著しく発展したが，都市部における交通渋滞や交通事故が増大し社会問題化した．近年では大気汚染などの環境問題も表面化し，低炭素社会の実現に向けた交通施策策定の取り組みが各地で行われるようになっている．

従来，これらの交通施策は計測値に基づいた交通流理論の理論式により試算が行われたが，制約が多く，問題依存性・時間依存性の高い複雑な問題に関しては思うような結果が得られない場合があった．また，交通現象は周囲との相互作用による影響が大きい大規模な問題であることから，実際の道路を用いた社会実験を行うことはコストや安全面から現実的ではない．そんな中で，問題解決の有効な手段のひとつとして研究開発が進められているものに，本研究で扱う交通流シミュレータがある．

交通流シミュレータは現実の道路ネットワークや自動車を計算機上に写し取り，交通現象を仮想的に再現する．理論式では分析できない複雑さを考慮できるほか，社会実験のように大きなコストをかける必要がない．また，計算機の特性上繰り返し試算を行うことが容易であり，交通現象のように車両挙動や信号などの確率的な要素を多く含む問題に対し非常に効率的であると言える．加えて，可視化やデータマイニングの技術と組み合わせることで，施策の効果を住民や自治体の意思決定者が

容易に理解することが可能である．これまで交通工学の専門家が用いてきたマクロな指標（交通容量[1] やボトルネック容量[2] といったもの）を理解することなく，視覚的に解釈することができるためである．このように，交通流シミュレーションには多くの利点があり，交通施策の策定において大きな社会的需要が存在すると考えられる．

2.2 交通流の基本量とその性質

以下では，交通工学の基礎として，3 つの基本量とそれぞれ相互の関係，及び交通流の基本性質について説明する．

2.2.1 交通流の基本量

交通流の性質や個々のモデルの説明に先立ち，交通流の基本量を定義する．基本量は以下の 3 つである．

- 交通流量

  交通流量 q(例えば [台/h]) とは，ある時間内に計測地点を通過した車両数として定義される．時間 T(例えば [h]) あたりに通過した車両数を N([台]) とすると，交通流量 q は式 2.1 で表現される．

$$q = \frac{N}{T} \tag{2.1}$$

- 車両密度

  車両密度 k(例えば [台/km]) とは，ある時間に計測区間内に存在する車両数として定義される．計測区間長を L(例えば [km])，区間内に存在する車両数を N とすると，車両密度 k は式 2.2 で表現される．

[1]交通容量（traffic capacity）：一定の道路条件と交通条件の下で，ある道路の断面を一定の時間間隔内に通過することが期待できる最大車両台数と定義される [1].

[2]ボトルネック容量（bottleneck capacity）：対象道路区間のうちで交通容量が最小となっている地点をボトルネック (隘路) とし，そのボトルネックにおける交通容量と定義される [1]．つまり，交通量がこの値を超えると渋滞が発生する．

$$k = \frac{N}{L} \tag{2.2}$$

- 空間平均速度

  空間平均速度 v(例えば [km/h]) とは，ある時間に計測区間内に存在する車両の平均速度として定義される．車両 i の速度を v_i，区間内に存在する車両数を N とすると，空間平均速度 v は式 2.3 で表現される．

$$v = \frac{1}{N}\sum_{i=1}^{N} v_i \tag{2.3}$$

2.2.2 交通流の性質

高速道路上の単路など，道路形状によって交通流が乱されない連続流区間では，交通流量 q，車両密度 k，空間平均速度 v の 3 つの基本量の間に式 2.4 の関係が成り立つ．

$$q = k \times v \tag{2.4}$$

このように，前項で述べた 3 つの基本量は互いに影響しあっていることが知られている．また，任意の 2 つの基本量についての関係を Q-K 関係，Q-V 関係，K-V 関係，とそれぞれ表現し，このうち，Q-K 関係について観測値をプロットした散布図は Q-K 図または交通基本図と呼ばれる．図 2.1 は現実の高速道路における Q-K 図に，その概略を重ね合わせた．以下では，図 2.1 中に示した基本的な用語を定義する．

- 自由走行相

  自由走行相とは，Q-K 図において交通流量と車両密度の間に比例関係が成り立つ領域をさす．これは計測地点における車両の流れがスムーズで，車両密度が空間平均速度に影響を及ぼすことのない状態である．自由走行相においてはほとんどの車両が制限速度に近い速度で走行すると考えられるため，Q-K 図のプロットは拡散せず，傾きが空間平均速度 v に等しい線分を形成する．これは式 2.4 の変形により自明である．

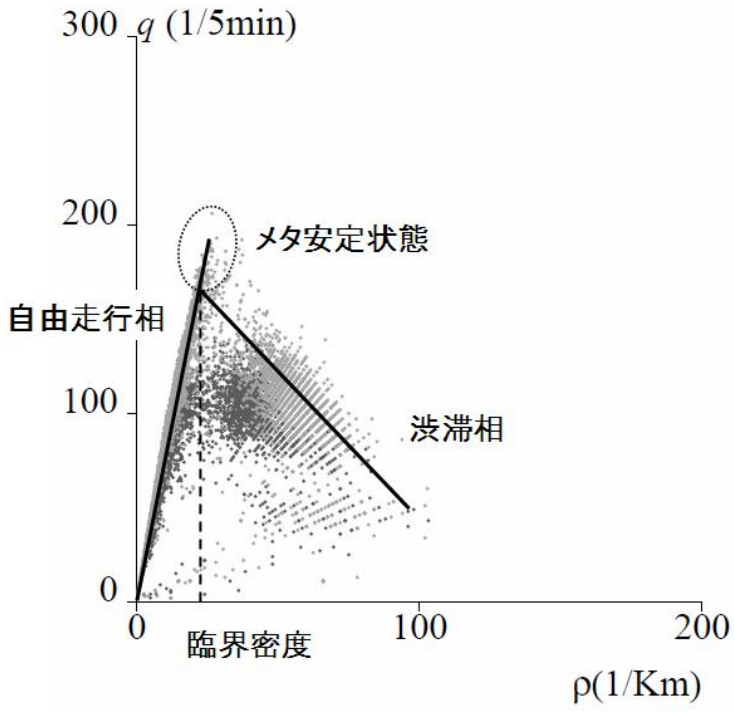

図 2.1: Q-K 図

- 渋滞相

  渋滞相とは，交通流量と車両密度の間に負の相関関係が成り立つ領域をさす．これは計測地点において混雑が発生し，車両密度の増加が空間平均速度の低下を促しているため，渋滞が発生したことを示している．

- 臨界密度

  臨界密度とは，自由走行相と渋滞相との相転移が発生する密度である．高速道路などの均一な道路環境ではほぼ一定の値をとるが，一般道においては制限速度や道路の幅員など様々な要因に依存するため一意に定まらない．

- メタ安定相

  メタ安定相とは，自由走行相における交通流量と車両密度の比例関係が，臨界密度以上においても成り立つ領域をさす．これは計測地点において混雑が発生し，本来渋滞が発生する状況にも関わらず，自由走行相と同程度の空間平均速度が実現されている状態である．

交通渋滞は，交通容量上のボトルネックにその地点の交通容量を超える交通需要が流入し生じる待ち行列の交通状態（渋滞車列），と定義される [2]．また，道路条件や交通条件によって常態的にボトルネックが生じる場合を「交通集中渋滞」，交通事故・車両事故などの突発事象によりボトルネックが生じる場合を「突発渋滞」，と呼ぶ．ただし，一般に渋滞を判断する技術的な基準は管理者によって異なり [2]，渋滞車列の延長・進行速度・継続時間などによって複合的に決定される，非常に曖昧なものである．

そこで本研究では，自由走行相における車両の空間平均速度が一定である点に注目する．空間平均速度の減少，つまり特定の経路における旅行時間が増大することを「渋滞」とし，旅行時間が自由走行時を上回る車両が存在すればその経路上に「渋滞が発生した」と考える．本論文では交通施策の予測・評価を目的とするため，突発事象を考慮しないが，本研究の枠組みに基づいて突発事象やそれに起因する突発渋滞を扱うことも可能である．

2.3 交通流モデル

交通流を解析する際に用いる交通流モデルは，一般的にマクロモデルとミクロモデルに大別できる．このうち，マクロモデルは交通流を流体近似して連続的に解析するのに対し，ミクロモデルは個々の車両の挙動を再現し，車両間の相互作用の結果として交通流を表現する．現在，マクロモデルとして流体モデルが，ミクロモデルとして追従モデルとセル・オートマトン（以下，CA : cellular automaton）モデルが知られている．それぞれ，時間や空間の表現によって表 2.1 のように分類することができる．

2.3.1 流体モデル

交通流の巨視的解析は 1955 年に Lighthill らによって，Q-K 関係を流体力学に基づいた記述による，Kinematic Wave theory[3] が提唱された．これら流体モデルは，交通流を交通流量，車両密度，空間平均速度などのマクロ量の関係式として記述している．このため，個別の車両の位置や挙動，内部状態などを表現することはできないが，連続な時間と空間を扱うことができる．

表 2.1: 交通流モデルの分類

	マクロモデル	ミクロモデル	
	流体モデル	追従モデル	CA モデル
空間	連続	連続	離散
時間	連続	離散/連続	離散

具体的には以下，式 2.5，式 2.6 に示す流体力学の原則，及び式 2.4 に示す交通流の制約条件に基づいている．

- 交通量についての連続式が成立する

$$\frac{\partial k(x,t)}{\partial t} + \frac{\partial q(x,t)}{\partial x} = 0 \tag{2.5}$$

- 空間平均速度は車両密度の関数として表現できる

$$v = v(k) \tag{2.6}$$

このとき，x は計測地点，t は計測時間を示す．また，Burgers 方程式や Navier-Stokes 方程式を適用したモデルも存在する [4]．同時に，方程式の超離散化によって，後述するミクロモデルとの関連についても議論されている [5]．

2.3.2 追従モデル

交通流の微視的解析は車両間の相互作用からモデルを構築しており，追従モデルと CA モデルが代表的である．交通流に特有の信号機等による密度の急激な変動に対応することが可能であり，より精度の高いシミュレーションが可能であると考えられる．特に追従モデルの場合，dt を微小にすることで近時的に連続な時間を扱うことができる．マクロモデルと比較して計算量が大きくなるという欠点があるものの，非常に優れた性質であると言える．

追従モデルは1953年にPipesによって提唱され[6]，個々の車両が，前方を走行する車両によって自らの挙動を制御するモデルである．具体的には，前方との車間距離や相対速度に応じた加速度の変更を行う．車両の位置や速度の関係は通常，微分方程式によって記述される．代表的な追従モデルであるChandlerのモデル[7]を式2.7に示す．

$$\frac{d}{dt}v_i(t+T) = a\left\{v_{i+1}(t) - v_i(t)\right\} \tag{2.7}$$

このとき，aは感応度と呼ばれ，速度差に対する車両の反応の鋭敏性を示すパラメータである．また，Tは応答のタイムラグを表し，車両の反応の遅れを表現する．現在，式2.7のような単純な線形単項型モデルから，指数関数型のNewellモデル，双曲線正接関数型の最適速度モデルなども提案されている[6]．

2.3.3 CAモデル

CAモデルでは非対称単純排除過程（以下，ASEP：Asymmetric Simple Exclusion Process）などが知られている[5]．ASEPをはじめ，CAは車線を1次元格子状に分割した離散空間で構成される．このとき，空間の最小要素をセルと呼び，各セルには車両が存在するかしないかの，2種類の内部状態が存在する．また，時刻$t+1$における内部状態は時刻tにおける自分自身，及び前方のセルの内部状態によって決定される．このため，CAモデルは追従モデルの空間を離散化したものとして捉えることができる．ASEPは以下のルールによって状態更新を行う．

1. 自己駆動
   進行方向に前進確率pで1セルだけ前進する
2. 排除効果
   前方セルに車両が存在する場合，新たな車両は前進できない

このルールは自動車を例に取ると，前方車両との車間距離が詰まってくると速度を落とすことに対応している．このとき，以上のルールにしたがってセルの状態を1stepごとに更新していくことによって，密度の高い状況下ではあたかも渋滞が後方に伝播していく様子を再現することが出来る．

CA モデルは複雑な挙動を車両の状態更新ルールに組み込むことが容易で，近年盛んに研究が行われているモデルである [9].

2.4　交通流シミュレータの分類

交通流シミュレータは車両の表現方法によって大きく 2 つのカテゴリに分類することができる．これまで述べてきた交通流モデルのうち，車両を流体近似するマクロモデルを採用するものをマクロシミュレータ，個々の車両を明示的に取り扱うミクロモデルを採用するものをミクロシミュレータと呼ぶ．これらのシミュレータは評価の対象とする現象の性質によって使い分ける必要がある．以下にその特徴と著名なシミュレータを簡単に紹介するが，紹介に当たっては, まず 1970 年代末に開発された，CONTRAM，SATURN，NETSIM といった，第 1 世代ともいうべきネットワークシミュレーションモデルに加え，1980 年代後半〜1990 年代の比較的新しいシミュレータも取り上げる.

2.4.1　マクロシミュレータ

マクロモデルは，一般にミクロモデルよりも計算量が小さく，よってマクロシミュレータでは広域な適用範囲でのシミュレーションが可能である．そのため，交通需要のトレンドを入力として与えることで交通渋滞の発生を再現することができる．そのため，交通施策の実行に際し事前におおまかな予測・評価を行うことができる．また，流体モデルによって交通流を再現するため数学的な性質が良く，道路ネットワークにおける均衡解を導出するなど，解析的な議論が容易である．かつてはこのマクロシミュレータが主流であり，多くのシミュレータが開発された.

以前の主流は Greenshields[10]，Greenberg[11]，Draw[12] らが導出した流体モデルを用いたマクロシミュレータであったが，最近では車両を離散的に扱うモデルも増えている．著名なシミュレータには以下のものがある.

SATURN

SATURN（Simulation and Assignment of Traffic in Urban Road Networks）は平面交差点における信号制御，右左折禁止などの交通規制の影響評価などを目的として，1979 年に英国 Leeds 大学で発表されたモデルである [13][14]．ネットワークはリンクとノードで構成され，交差点近傍はレーンのイメージを持つ．シミュレーション対象時間は 15〜30 分程度の間隔に分割され，各時間帯に OD（Origin-Destination）交通量を与える．交通量は流体近似され，各リンクへの到着が 1 信号サイクルの間の IN パターンとして入力される．リンク下流端では IN パターンに応じて車群の拡散を考慮した ARRIVE パターンが生成され，交通容量から求められる ACCEPT パターンにマスキングされて，リンク流出交通量の OUT パターンが計算される．捌け残った交通量は停止線部に待ち行列を形成する．この OUT パターンや待ち行列は次の単位時間でのフローの状態に影響するので，このような手続きを繰り返し，OUT パターンが収束した時点で，1 サイクルでの定常的なフローパターンを得るものである．SATURN は，動的な利用者均衡状態を再現することを目指している．そこで，信号のサイクルを全ての交差点で共通とし，1 サイクル分の定常状態をシミュレーションすることで，この過程で得られた交通量－遅れの関係を利用し利用者均衡配分を行う (SATURN では確率的均衡配分も実現している)．以前の時間間隔において捌け残った交通量がすでに決定された経路に固定されてしまうことや，過飽和時の密度管理が十分でないため，渋滞の延伸・解消が正しく再現されないという問題などがしばしば指摘される．

SOUND

SOUND は東大生研において開発された，過飽和状態の都市内高速道路を対象としたモデルである [15]．車両は一台ずつ表現され，設定された車頭間隔－速度曲線によって，各時刻でのリンク上の位置が計算される．この手法により，ボトルネック容量の違いによる渋滞状況の違いなども正確に再現することができるようになっているが, あくまでマクロな数値に従った制御である．SOUND は高速道路を対象としているため，詳細な車両の挙動は考慮されておらず，レーン変更や信号交差点などのモデル化は省略されている．また，SOUND は現在の旅行時間情報に従って，各車両が分岐点において確率的経路選択を行う，動的均衡配分を実現しており，首

都高速道路の交通状況の分析に適用されている．最近の研究では，旅行時間を予測するモデルを組み込み，予測時間情報の提供効果の評価を行った報告がある．

2.4.2 ミクロシミュレータ

ミクロシミュレータはより詳細な車両挙動を再現することが可能である．計算量が大きく広域には適用できないものの，交通流の合流や車線変更，追い越し行動などの挙動を再現し，相互作用によってこれらの行動が系に与える影響を観察することが可能である．単路部や交差点構造の局所的な変化が引き起こす交通現象の変化を予測・評価できる．

この分類に含まれる主なシミュレータには，交通流モデルに追従モデルを採用したものとして以下のものがある．一方，計算量の制約によって更に適用範囲の限定される CA モデルでは，統合的な交通流シミュレータが開発されている例は少ない．交通施策の予測・評価ではなく，単路部における交通流の振る舞いなど，より一般的な分野での研究が主である．

TRAF-NETSIM

NETSIM（Network Simulation Model）1970 年代初頭に米国 FHWA（Federal Highway Administration）によって開発されたモデルである．現在は交通シミュレーションの統合システムである TRAF[16] の一部を構成し，TRAF-NETSIM と呼ばれている [17]．ネットワークはノードとリンクで構成され，リンクはレーンのイメージを持つ．交通流の表現は車両 1 台ごとの挙動を微視的・確率的に再現するもので，各車両には車種のカテゴリ，加速度などの車両性能，ドライバーの行動類型 (受動的，ノーマル，能動的)，希望速度などの多数のパラメータが設定され，リンク上を追従走行する．そのほかにも交差点でのギャップ待ちやレーン変更の判断などの挙動についても多くのパラメータが用意されており，車両挙動の柔軟なモデル化が可能である．しかしながら，NETSIM では各車両は目的地の情報を持っておらず，リンクごとに設定された右左折分流率に従って，次に流入するリンクを決定している．したがって，交通状況に応じた経路選択を自身で行うことはできないため，面的な交通運用策を評価することは困難である．

INTEGRATION

INTEGRATION はカナダ Queen ’s 大学の Van Aerde と Waterloo 大学の Yagar らによって 1988 年に開発された，高速道路と一般街路を統合してシミュレーションを行うモデルである [18]．一般街路の信号制御と高速道路のランプ流入制限などの運用策を同時に一つのモデルで評価できるだけでなく，情報端末搭載車への情報提供なども評価することができる．道路ネットワークはノードとリンクから構成され，レーンのイメージはない．交通流のモデリングは車両を一台ずつ表現するミクロモデルであるが，追従ではなく待ち行列によって交通状況を再現するものである．各車両には目的地が設定されており，決められた出発時刻にネットワークに流入する．リンクは流入率，流出率，および車両密度の属性を持ち，リンク上の車両のスタック (Link Data Stack)，流出可能な車両のスタック (Departure Stack) を扱う．リンクに流入した車両は Link Data Stack に入れられ，現在のリンクの車両密度から求められる旅行時間を用いて，次のリンクに流出することができる時刻 (Next Scheduled Departure Time：NSDT) が設定される．シミュレーション時刻が車両の NSDT をこえると，その車両は Departure Stack に移され，リンク容量，信号現示，下流側の待ち行列，などの状態をもとに流出できるかどうかの判断がなされる．リンク容量は交通が渋滞しているかどうかによって変化する．このような計算手法で，SATURN で指摘されるような渋滞時の密度管理の問題は解消される．INTEGRATION では，現在のリンク旅行時間情報をもとに一定時間ごとに最短時間経路が更新され，各車両は常に目的地への最短経路を選択することで，動的な均衡状態を再現する．最新のバージョンでは，交通状況にかかわらず決まった経路を選択するもの，日常の交通状況より予測される旅行時間に従って経路選択するもの，システム最適の規範に従って経路選択するものなど，複数の道路利用者層を扱うことができるようになっている．

2.4.3 中間の（メゾ）シミュレータ

また，マクロシミュレータとミクロシミュレータの中間的なものとして，メゾスコーピックなシミュレータも存在する．

AVENUE

AVENUE(An Advanced & Visual Evaluator for Road Networks in Urban Areas)は，都市街路の交通運用策の評価を目的としたモデルである [19]．過飽和状態を考慮したハイブリッドブロック密度法という手法で交通流を再現し，交通状況に応じた経路選択挙動のモデルを内包している．これは，車両の位置を連続値で得ることはできないが，ある程度流動的な位置管理を行うことができるようになっている．また，レーン，信号，バス停など，市街路の交通状況に影響する施設を明示的にモデル化すると同時に，各種交通規制，バス専用レーン，リバーシブルレーンなど詳細な交通運用策を設定することが可能である．

2.5 MATES

本節では，我々が開発を行なっている知的マルチエージェント型交通流シミュレータ MATES (Multi-Agent based Traffic and Environment Simulator) [21][22][23][24] について説明する．

2.5.1 知的マルチエージェントモデル

MATES の基本設計として,「複雑なものを過度に簡略化せず，複雑なまま扱う」というものがある．本質的に社会的な行動の集合である交通現象は，それ自体が複雑系としての性質を持っており，MATES はその性質を自然な形で扱うことを目的としたシミュレータである．交通流シミュレータにおける理論的背景は前章で既に述べたとおりで，流体モデル，追従モデル，CA モデルなど多くの種類があり，それぞれに長所と短所がある．MATES はその中で複雑なままの車両挙動を扱うことのできる追従モデルを採用しており，ミクロシミュレータとして分類することができるが，既存のシミュレータにない特徴として知的マルチエージェントモデルの枠組みを有する．エージェントとは自律した個々の主体のことを指し，エージェントが「知的」であるというのは適応的に行動することができるものであるとする．適応的であるとは以下の三つの条件を満たすものを指す.

- 即応性
  エージェントは，環境の変化に対して即座に反応した行動をとることができる．
- 目的指向性
  エージェントが持つ目的に向かって，積極的な行動をとることができる．
- 社交性
  エージェントは他のエージェントと通信を行うことができる．

ある環境において，意思決定主体 (エージェント) がそれぞれ自律した判断の下行動した場合，相互作用の結果として創発的な振る舞いが生じるようなシステムを，知的マルチエージェントシステムと呼ぶことにしている．ここでは，システム全体として一つの目的を達成したり，その結果がより良いものであったりする必要はないという立場を取る．

知的マルチエージェントの枠組みでは，現象を記述するために以下の 2 つを規定することが必要である．

- エージェント
- 環境

MATES では車両をエージェントとし，エージェントの活動する場所を，つまり道路空間を環境として定義している．ただし環境には道路そのものだけではなく，信号や標識，路上の障害物，周囲の建築物や場合によっては法律的な制限まで含まる．更に，自分以外のエージェントも環境に含まれることが重要である．エージェントとは環境から情報を取得し，状況を判断して行動を決定し，実際に行動することで環境に対して働きかける．車両 A が動くということは車両 B の環境の変化を意味し，車両 B が動くということは車両 A の環境の変化を意味する．ここに相互作用が生まれる．この相互作用の総和として，複雑な現象が創発される．MATES のエージェントと環境のモデルを図 2.2 に示す．

2.5.2 知的エージェント

MATES では，車両をエージェントとして扱う．通常はエージェントを「自ら考えて自律的に行動する主体」とみなすため，正確には運転者がエージェントにあた

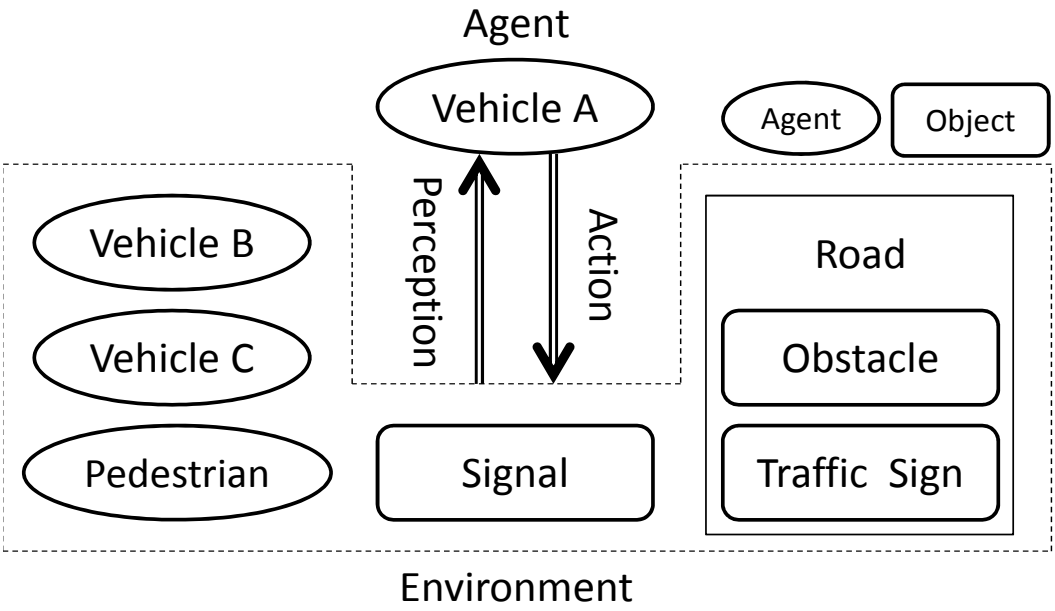

図 2.2: MATES のマルチエージェントモデル

るが，運転者と車両の動きを切り離す必要がないため，このような形をとる．

エージェントの挙動のうち，行動そのものに関しては，さほど複雑な物理現象は考慮しないものとして構わない．例えばブレーキを踏めば，自動車はあるパラメータに従って減速するが，このときブレーキの機械的構造からその効き方を計算することは，交通流シミュレータにとって本質的ではない．つまり，運転者がどの程度ブレーキを踏んだらどのような効果が生まれるかという情報は単純なマッピングとして定義できる．

しかし，ブレーキを踏むのかアクセルを踏むのか，またどの程度踏むのかといった判断は単純に定義することはできない．前の車両が減速したら自分も減速するといった条件反射的なルールが適用される範囲は広いが，これはいくつかのパラメータが関わってくる問題であり，また必ずしもそのルールが成立するとは限らない．従来のシミュレータでは，ドライバーはかなり単純なモデル化が行われてきたため，経路の選択などはそもそも考えられていないものが多く，考えられていたとしても，それは全体の均衡をとるような理論に従うものであったり，出発地と目的地を与えると単純に推定されたりするものがほとんどであった．運転者が経路を選択するプロセスには何らかの思考が加わるため，そこには個人の嗜好が反映されなければな

らない．従って，個々のエージェントがそれぞれの立場で判断し，その結果としての行動を起こすという経過を捉えるためには，エージェントをそれぞれ独立の存在として定義する必要がある．このときに最も重要なのは,「自律的に」行動することが可能でなければならないことである．全体に起こる現象を再現するための行動規範を組み込むのではなく，交通現象はあくまで創発によって起こるものとする．

以下にエージェントが満たすべき自律性を挙げる．これらの行為を自律的に行うことができれば，道路交通を適切に再現できると考える．

- 道路ネットワーク上での自律性
  - プランニング (出発地と目的地)
  - 経路の探索
  - 経路の選択
- 道路上を運転する能力
  - 交通規則に関する知識とその反映
  - 速度決定
  - 車線変更・合流・分岐
  - 交差点での他エージェントを意識した右左折
  - ある道路内での経路決定

2.5.3 仮想道路環境

仮想道路環境の役割

現実の道路環境との区別を行う意味で，モデル化を行いコンピュータ上に再現された道路環境を仮想道路環境と呼ぶことにする．

シミュレーションにおいては，適用可能性を限定しないためにもその計算効率を初めから考えたモデル化を行うことが必要になる．これは交通流シミュレータにおいても例外ではなく，モデル化とデータ構造やアルゴリズム，そして計算効率は密接な関係を持つ．交通流シミュレータでは，他の知的マルチエージェント手法を利

用したシミュレーションと異なり，エージェントの数が非常に多くなる．MATES において目標としている規模は数百から数十万，場合によっては数百万のエージェントが登場する領域である．

逆に連続体のシミュレーションなどと比べると，再現すべき現象には多くの制約がかかっている．具体的には，車両エージェントは車道上を走行するという仮定があり，2 次元，または 3 次元空間全てを考慮する必要はないといえる．ただし，これ以上の制約を設けることは，MATES 構築の本来の意図に反することになる．これは，シミュレーションを行う上での定量性は確保する必要があるということである．また仮想道路環境はあくまでエージェントが存在する環境であるから，エージェントからの視点で，他のエージェントや道路，横断歩道がどう見えるか，ということに注意しなければならない．

また，エージェントが信号を見ることを可能にするためには，環境が信号の色を保持するだけでなく，信号を見るという行為の見返りとして適切な情報として提供する必要がある．本来であればエージェントは自身で環境が保持する情報を自律的に収集すべきであるが，個々のエージェントには知り得るべき情報と知り得ない情報がある．環境は全ての情報を保持しているため，そのエージェントが知り得るべき情報であるのか判断する必要が生じるが，この判断はエージェントが自身で行うべきではない．そこでエージェント-エージェント間での情報のやり取り，また環境-エージェント間での情報のやり取りをプロトコルという形で定義した．

環境は，プロトコルに従ったエージェントからの要請をうけて，適切な情報を提供する，または適切な変化を起こす必要がある．例えば，車両エージェントは各ステップにおいて標識の情報を得ようと環境に要請するが，このとき，そのエージェントから見える範囲内の標識であるかどうかを環境側で判断して，もし見える範囲内であれば提供する．これは本研究で扱う経路選択についても同様であり，仮想道路環境は情報を保持すると共に，プロトコルに従った返信を行う，計算機で言うサーバのような役割を果たすと考えるとよい．以下では，環境がどのように情報を保持するかを中心に述べる．

仮想走行レーン

MATESでは，自動車の走行車線にはオブジェクト指向道路モデルを採用する．オブジェクト指向道路モデルでは，全ての道路構造を仮想走行レーンという最小単位で表している（図2.3）．このモデルは一般的に有向グラフの構造を持っている．道路をネットワークとして捉えた場合には，走行レーン毎に走行できる向きが決まっていることが普通である．そのため，単純に接続状態を記述するだけでは不完全であり，有向グラフと同じく，方向を持つことが必要である．仮想走行レーンは自動車のモデル化と密接に関わった道路のモデル化である．運転の操作は以下のように分類できる．

- 操舵とその結果の車線幅内の左右振れ及び，隣接車線への移動
- 加減速操作とその結果としての速度変化
- 灯火器及び音響による情報伝達

この分類の上で車線幅内の左右振れを簡略化し，操舵は道路のレーンに従って行われるものと仮定する．自動車の運転をするときには運転者は予定軌道を頭に描き，それに沿うように運転する．これは直観的には，線路を走る列車と似たモデルである．仮想走行レーンには全体の長さや前後の仮想走行レーンとの接続情報，道路が持つ様々な付帯情報などを保持させ，これらを知的エージェントに対して提供する．仮想走行レーンを作成することにより，車の操舵の計算を省くと同時に，本来は2次元の連続値である車両の位置を，離散的な構造と1次元の連続値へと変換できる．レーンの持つ情報は表2.2のような定義を行うことができる．仮想走行レーンモデルを初めに提案し用いているMITRAM[30]では，仮想走行レーン同士の接続情報テーブルを保持する方法で道路ネットワークを表現している．しかしMATESでは，基本的には仮想走行レーンの概念を踏襲しているが，広域への適用と挙動の精緻化を求めて，新しい概念を加えている．それがレーン束オブジェクトの概念である．

レーン束オブジェクト

仮想走行レーンは人間の直感的な道路の概念とは多少離れたものになっている．交通主体は人間であるから，車両エージェントからの始点を考えるときには，人間が

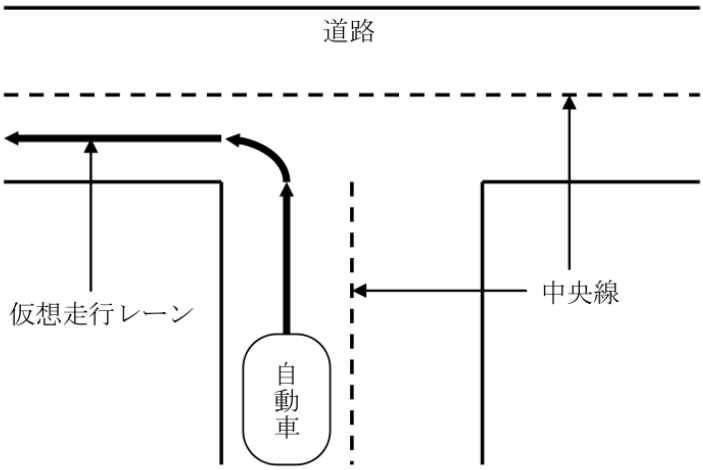

図 2.3: 仮想走行レーン

認識できる情報を過不足なく提供することが必要である．そこで MATES では，仮想走行レーンを接続して全ての道路構造を表すのではなく，中間の概念を置くことにしている．これがレーン束オブジェクトである．レーン束オブジェクトとは，具体的には単路と交差点の二つを指す.

レーン束オブジェクトは仮想走行レーンと，それを接続するコネクタという最小構成要素からなり，レーンとコネクタはリンク-ノードの関係にある．コネクタは仮想走行レーンの始点または終点であり，流入と流出の二つの方向を保持している．またレーン束オブジェクトはそれ自身が一つ上位のリンク-ノード関係を形成しており，入れ子のデータ構造を持っている．そして地図全体で一意な番号を持ち，マクロ的に見たネットワークの構成単位に当たる．レーン束オブジェクトを導入する利点を次のように考える.

- エージェントは基本的に近傍の情報を利用する場合が多い．つまり，最も回数の多い問い合わせに対する探索空間を限定することができる.
- ドライバーが経路選択をする際に，論理的な接続情報として利用する最小単位は単路と交差点であるから，それより詳細な情報を用いて探索を行うことで，計算量を抑えることができる.

レーン束オブジェクトは，コネクタの列を保持する「境界」を持つ．単路であれ

表 2.2: 仮想走行レーンに付帯する情報の例

仮想走行レーン	
形状	始点座標（ベクトル）
	終点座標（ベクトル）
	道路長
	道路勾配
レーン情報	識別番号
	レーン上に存在するエージェントの集合
	制限速度

ば境界は 2 箇所に存在し，十字路であれば境界は 4 箇所存在する．単路は交差点の保持する境界を結ぶものであり，これらの持つ境界は交差点の持つ境界と本質的に重複する．つまり，境界の保持するコネクタは必ず 2 つのレーンオブジェクトに共有されていることになる．

階層型道路ネットワーク

道路の接続情報と位置の情報を階層化することによって，道路構造を離散化することが可能になる．しかしレーン束オブジェクトは中間層として仮想走行レーン，コネクタの保持のほかに，より上位層との接続情報を持たなければならない．それはレーン束オブジェクト同士の接続関係と，単路や交差点としての付加情報である．レーン束オブジェクトとしての交差点は入力データの中で一意な識別番号が付けられる．

複数の単路や交差点の情報を取り扱うためには，これらをまとめて保持する「入れ物」が必要である．そこで導入されたのが地図クラスである．

また，MATES の道路環境には信号や歩行者など含まれることも忘れてはならない．これらの情報も単路や交差点といったレーン束オブジェクトに付加されている．

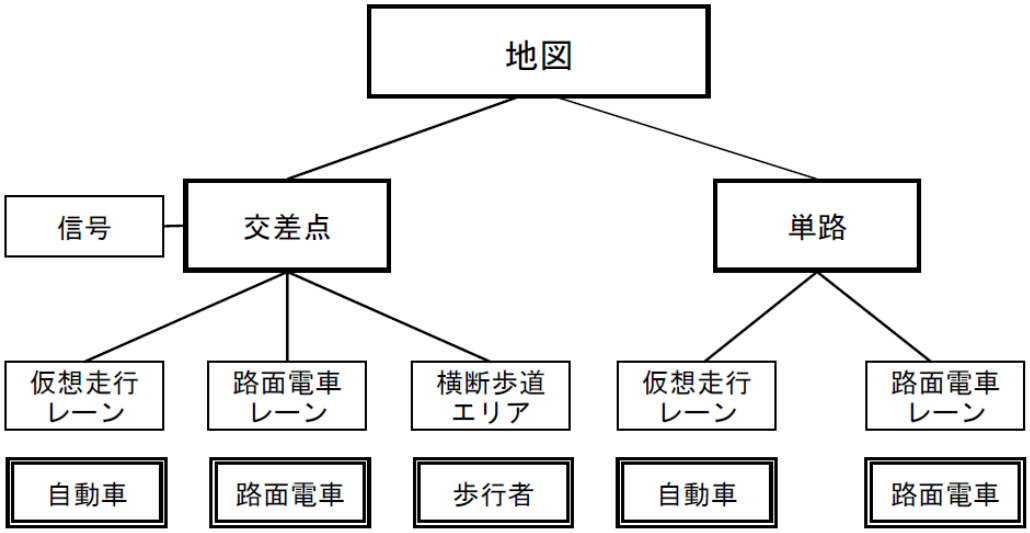

図 2.4: 階層型道路ネットワークとエージェント

MATES の持つ階層型道路ネットワークは図 2.4 のように表される．各環境に属するエージェントは，必要なだけ上位のカテゴリにさかのぼることによってあらゆる情報を取得することができる．

2.6 過渡状態の交通現象

本節では，本研究で問題とする交通現象について考える．道路ネットワークの変更が生じる場合に，シミュレータ上で再現される過渡状態の交通現象である．以下では，道路ネットワークの変更によって想定される交通現象と，その再現における既存のシミュレータの限界について説明し，本研究の問題設定を行う．

2.6.1 道路ネットワークの変更

ここまで述べてきたとおり，交通流シミュレータは交通現象を再現し，交通施策の予測・評価に役立てることが求められている．解決すべき交通問題によって策定・実施すべき施策の性質は変化するが，例えば渋滞の発生しやすい道路に対して新たにバイパスを整備し，交通需要の分散を図ることが考えられる．また最近では，ト

ランジットモールと呼ばれる，公共交通機関のみ進入・運行が許可されている歩車共存道路の導入によって，交通需要の転換を意図する施策も存在する．これらの施策に共通するのは，既存の道路ネットワークに対し，前者はリンクの追加，後者はリンクの除去という変更を加えている点である．また，ここまで劇的な変化ではないものの，道路の拡幅や事故や工事による車線閉塞は日常的に経験する事例であり，リンクの流量を増減させるという点においてはネットワークの変更に当たる．

ここで，日常的に同じ道路ネットワークを，同じ目的地に向かって走行する運転者を仮定する．通勤ドライバーなどがこれに該当する．このような運転者は通常，これまでの経験によって走行する経路はほぼ固定化されており，当日の状況や信号のタイミングなどによって多少の変動が起こる程度である．また，このような運転者が多く存在する環境では，道路ネットワーク上の交通流も日々安定した状態を保つと考えられる．

しかし，あるタイミングで交通施策が実施され，道路ネットワークに変更が生じたとする．このとき，ネットワークの変更によって生じた新たな環境に運転者が即座に対応することは困難である．これまで頼りにしていた過去の経験に従うことはできず，新たに生じた環境全体を容易に俯瞰することもできないためである．また，走行経験を重ねていく過程で運転者は徐々に新たな環境に慣れていくが，他の運転者にも同様のことが言えるため日々環境が変化していく可能性も示唆される．ネットワークの変更によって生じる以上のような現象は次第に落ち着き，時間の経過と共に以前とは異なる，しかし安定した状態に収束すると考えられる．

本研究では以降，道路ネットワークにおける交通流の状態を次の 2 つに分類して議論することとする．

- 定常状態
  道路ネットワークに関する知識が十分に浸透し，交通流が安定している状態．個々の運転者の走行経路もほぼ固定化していると考えられる．
- 過渡状態
  定常状態に対し外乱（ネットワークの変更）が生じた状態．交通流が不安定になり，運転者の走行経路も容易に変動すると考えられる．

2.6.2 既存のシミュレータの限界

交通施策の策定について考えた場合，前項における過渡状態の現象を事前評価することは，充分な時間経過の後収束する定常状態を予測することと同様に重要である．しかし，既存のシミュレータにおいて過渡状態を再現することは困難である．

マクロシミュレータにおける過渡状態

マクロシミュレータの代表的なモデルである流体モデルは，密度管理で交通流を表現できるための仮定が設けられている．多くの時間帯において定常交通流が期待され，信号などもほとんどない高速道路のような環境においてはよい再現性が期待できるが，それでもジャンクションでの流入流出などの現象が入って来た場合には，かなり複雑な理論や統計処理によるパラメータ推定が必要になる．このような問題は，流体モデルという個々の車両の動きを再現できないモデルを採用するマクロシミュレータにとって必然的なものである．換言すれば，これらのモデルは観測されるマクロな指標が従うルールを理論的に考察したものであり，シミュレーションによってその理論の裏付けをとることができる可能性はあっても，その理論の枠組を越えるものには適用できない．定常状態を前提としたこのモデル化では，当然ながら過渡状態を議論することは不可能である．

ミクロシミュレータにおける過渡状態

マクロシミュレータと同様，ミクロシミュレータにおいても過渡現象を議論することは困難である．過渡現象が発生する要因としては次の 2 点が考えられるが，どちらも車両が自ら思考して行動する「自律性」を満たさなくては実現することができないためである．

- 経験依存：過去の走行経験により経路を選択する
- 部分観測：周囲の限定的な環境のみ観測可能である

はじめに経験依存について考えてみよう．多くのミクロシミュレータでは車両が全て均一な行動規範を持つ．このようなモデル化では車両ごとに異なった走行経験を蓄積することが出来ない．確率的に経路を変化させることで多様性を表現するこ

とはできるが，本来車両の数だけ異なった走行経験が存在するため，それぞれが独立に経路を決定すべきである．結果的にシミュレータで観察される交通現象が現実をよく再現していても，全体に起こる現象を再現するために行動規範が組み込まれていたのでは本末転倒である．これでは，まだ施行されていない交通施策の予測において，過渡状態を再現することはできない．交通現象はあくまで創発によってボトムアップに生じるべきである．

また，部分観測についても考えてみる．運転者は出発地と目的地を決定してから経路選択を行うが，通常その途中の交通状況について正確に知ることはできない[3]．そのため，道路ネットワークにおいて過去の走行経験から逸脱するような状況が発生したとしても，運転者がそのことを認識するのは周辺に到達してからということになる．しかし，現在のミクロシミュレータにおける経路選択はネットワーク全体に渡って各リンクのコストを取得できることが前提であり，ある地点での道路ネットワークの変更を全運転者が瞬時に認識するモデルとなっている．結果として，運転者はネットワークの変更が起こった直後から安定した経路選択を実現できてしまい，現実と乖離した挙動を引き起こす．これでは過渡状態を再現することができず，不完全なモデル化であると言わざるを得ない．

2.7 本研究の問題設定

上述したとおり，道路ネットワークの変更を伴う交通施策の評価は非常に重要である．しかし，その際に考慮すべき過渡状態の交通現象は扱いが困難で，既存の交通流シミュレータはどれも十分な再現性を持ち得ない．そこで本研究では，この過渡現象を再現するモデルを新たに提案し，交通施策の評価に適用することとする．

この問題は，車両の自律性をモデル化することにより解決可能である．特に，過渡状態の再現に関しては車両の経路選択におけるモデル化が重要である．本研究では本節で説明した知的マルチエージェント型交通流シミュレータ MATES を用いる．MATES は知的マルチエージェントモデルの枠組みで設計されており，ここに強化学習による経路選択アルゴリズムを導入する．

[3]近年カーナビの普及がめざましいが，総務省の調査によればそれでも全車両の半数以下である．また，提供される情報の質も，必ずしも高いとは言えない．そのため，やはり運転者は経験依存な経路選択を行う．

第3章　強化学習

小目次

第3章　強化学習

本章では強化学習の概要について述べる．まず 3.1 節において強化学習の前提とするモデルについて説明する．次いで 3.2 節では実際の学習アルゴリズムをまとめ，それぞれの特性を明らかにする．最後に，3.3 節では行動選択の手法について紹介する．

3.1　強化学習のモデル

未知の環境において最適な制御則を試行錯誤的に獲得する枠組みのひとつに強化学習（Reinforecement Learning）がある．強化学習における意思決定者（以下，エージェント）には，状態入力に対する正しい出力を明示した教師信号が存在せず，報酬と呼ばれるスカラーの情報のみが与えられる．エージェントはこの報酬の期待総和を最大化することを目的とし，学習を繰り返す．強化学習の概念を図 3.1 に示す．

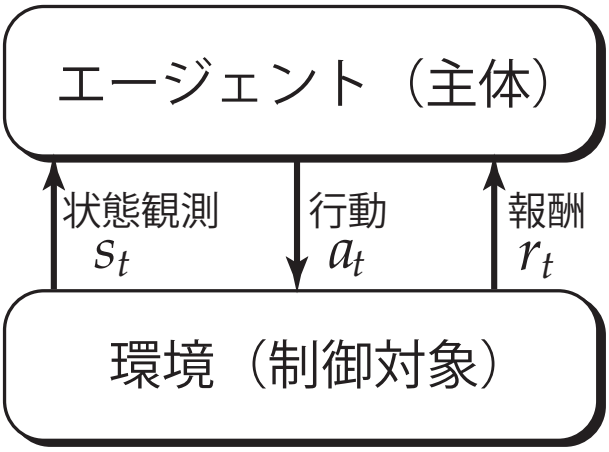

図 3.1: 強化学習の概念図

強化学習において，エージェントは制御対象である環境との間で図 3.2 の試行を繰り返す．このとき，環境の状態集合を $\mathcal{S}$，エージェントの行動集合を $\mathcal{A}$ とする．

強化学習は機械学習（Machine Learning）の分類の一つである．機械学習とは，人間が日々の経験から知識を獲得していく過程をコンピュータによって再現しようとい

1. エージェントはある時刻 t において環境の状態観測 $s_t \in \mathcal{S}$ に応じた意思決定を行い，行動 $a_t \in \mathcal{A}$ を出力する．
2. 行動出力により環境は $s' = s_{t+1}$ へ状態遷移し，その遷移に応じた報酬 r_t をエージェントに与える．
3. 時刻 t を $t+1$ に更新しステップ 1 へ戻る．

図 3.2: 強化学習のアルゴリズム

う試みから生じたもので，大きく，教師あり学習・教師なし学習・強化学習の三つに分類できる．ニューラルネットワークに代表される教師あり学習 (Supervised learning) では，事前に与えられた教師データ（入出力ペア）をもとに望ましい出力を学習し，判別問題・回帰問題に応用される．また教師なし学習 (Unsupervised learning) は期待される出力が未知であり，入力データのみが与えられる．そこからデータの構造を抽出するために学習を行い，クラスタリングなどに応用される．

一方，強化学習は教師信号こそ与えられないものの，環境から報酬を得てエージェントの期待報酬を最大化し，その結果としてエージェントの行動の選択規範を獲得する．強化学習の他の学習則と異る特徴は，正しい行動を教示されるのではなく，取った行動を事後に評価することで学習を行う点である．つまり，教師なし学習に近い入力条件下で教師あり学習の問題を解くことが必要となる．現実の経路選択行動においては，エージェントの運転挙動の多様性や複雑なネットワーク構造から，行動規範を数理的に求めることは困難であることが多い．そこで，シミュレーションとあわせ，この強化学習の手法を適用する事が有効であると考えられる．

3.1.1 マルコフ決定過程

マルコフ決定過程 (以下，MDP : Markov Decision Process) は，状態遷移にマルコフ性を持つ確率システムの動的最適化のための数学モデルである．マルコフ性とは状態 s' への遷移が，直前の状態 s と行動 a のみに依存し，それ以前の状態及び行動とは無関係であるという性質である．

時刻 t の状態 s_t においてエージェントがある行動 a_t をとると，時刻 $t+1$ でのエー

ジェントの状態は s_{t+1} へと遷移するが，マルコフ性を持たない Non-MDP では時刻 $t+1$ の状態と報酬はそれ以前に起こった全ての事象が関係すると考えられるため，次の式 3.1 のようになる．

$$P_r\left\{s_{t+1}=s', r_{t+1}=r' | s_t, a_t, r_t, s_{t-1}, a_{t-1} r_{t-1}, \cdots, r_1, s_0, a_0\right\} \tag{3.1}$$

一方，一般に強化学習では，離散的な入出力，離散的な時間ステップ，そして環境のマルコフ性が仮定されている．マルコフ性を持つ環境下ならば，時刻 $t+1$ の状態と報酬は直前の状態と行動のみによって決定するため，次の式 3.2 のように表せる．

$$P_r\left\{s_{t+1}=s', r_{t+1}=r' | s_t, a_t\right\} \tag{3.2}$$

このように，マルコフ性を仮定すると，現在の状態と行動から次の時刻の状態と報酬を予測することができ，さらに繰り返し計算により，全ての将来の状態と報酬を予測することができる．

もちろん，これらは理想的な環境を仮定した場合にのみ成り立つものであり，実世界においてはマルコフ性が保証されない場合がほとんどである．例えばエージェントの経路選択においても，センサの測定精度やノイズの分解能の問題から，環境から与えられる状態入力は必ずしも十分とは言えない．ただし，確率的な状態観測しか行い得ない場合のモデルとして部分観測 MDP（POMDP：Partially Observable MDP）が提案されており，そのような環境下でも強化学習によって一定の近似解を得ることができるとされている [20].

3.1.2 価値関数

強化学習では報酬を最大化することで学習を行う．そのために，ここでは現在の状態，もしくは行動がどのくらい良いのかを測る関数として，価値関数というものを考える．この価値関数は将来にわたって得られる報酬として定義する．

エージェントの行動は方策 π によって決定されるが，方策 π とは，状態 s で行動 a をとることであり，$\pi(s,a)$ と表す．この方策 π のもとで，状態 s の価値は，状態価値関数 $V^{\pi}(s)$ として以下の式 3.3 ように定式化できる．

$$V^{\pi}(s) = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^{k}r_{t+k+1}|s_t = s\right\} \tag{3.3}$$

同様に，方策 π のもとで，状態 s において行動 a を取ることの価値は，行動価値関数 $Q^{\pi}(s,a)$ として以下の式 3.4 ように定式化できる．

$$Q^{\pi}(s,a) = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^{k}r_{t+k+1}|s_t = s, a_t = a\right\} \tag{3.4}$$

ここで $\gamma(0 \leq \gamma \leq 1)$ は割引率を表しており，状態価値関数及び行動価値関数はどちらも，遠い将来に得られる報酬ほど割引いて評価した，報酬の総和の期待値である．

状態価値関数 $V^{\pi}(s)$ と行動価値関数 $Q^{\pi}(s,a)$ の違いは，行動 a を考慮するかどうかのみである．$V^{\pi}(s)$ の場合は状態 s において，方策 π に従い次々に行動した場合にどのような報酬が得られるか，$Q^{\pi}(s,a)$ の場合は，状態 s において行動 a をとった後，方策 π に従った場合にどのような報酬が得られるかを表している．ここで，$Q^{\pi}(s,a)$ における最初の行動 a は方策 π に関係の無いものである．

このことを念頭に最適方策 π^{*} を考えると，最適価値関数 $V^{*}(s)$，及び $Q^{*}(s,a)$ はそれぞれ式 3.5，3.6 のように表せる．

$$V^{*}(s) = \max_{\pi} V^{\pi}(s) \quad for\ all\ s \in S \tag{3.5}$$

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a) \quad for\ all\ s \in S,\ for\ all\ a \in A \tag{3.6}$$

最適方策は複数存在する可能性があるが，全ての最適方策は唯一の最適状態価値関数 $V^{*}(s)$ と最適行動関数 $Q^{*}(s,a)$ を共有する．これらは式 3.7，3.8 のように変形することができる．

$$
\begin{aligned}
V^*(s) &= \max_{a \in A(s)} Q^{\pi^*}(s,a) \\
&= \max_a E_{\pi^*} \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a \right\} \\
&= \max_a E_{\pi^*} \left\{ r_{t+1} + \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_t = s, a_t = a \right\} \\
&= \max_a E_{\pi^*} \left\{ r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a \right\} \qquad (3.7)
\end{aligned}
$$

同様に，

$$
Q^*(s,a) = E_{\pi^*} \left\{ r_{t+1} + \gamma \max_{a_{t+1} \in A(s_{t+1})} Q^*(s_{t+1}, a_{t+1}) | s_t = s, a_t = a \right\} \qquad (3.8)
$$

3.2 学習アルゴリズム

強化学習には様々なアルゴリズムが提案されているが，代表的な手法として TD 法，Sarsa，Q 学習などが知られている．本節ではこれらについて説明する．

3.2.1 TD 法

TD（Temporal Difference）法は，時刻 t における状態 s_t の価値 $V^\pi(s_t)$ を，行動 a_t をとった場合その行動によって得る報酬 r_{t+1} と行動後の状態価値 $V^\pi(s_{t+1})$ を用いて更新する．この更新式は以下の式 3.9 によって表される．このとき，$\alpha (0 < \alpha \leq 1)$ は学習率を表しており，学習率は状態価値関数の更新の度合いを決定する．

$$
V^\pi(s_t) \leftarrow V^\pi(s_t) + \alpha \left\{ r_{t+1} + \gamma V^\pi(s_{t+1}) - V^\pi(s_t) \right\} \qquad (3.9)
$$

ここで，前節の状態価値関数の定義である式 3.3 より以下の式 3.10 が導出できる．

$$
V^\pi(s_t) = r_{t+1} + \gamma V^\pi(s_{t+1}) \qquad (3.10)
$$

1. $V(s)$ の各要素の初期化
2. 時刻 $t \leftarrow 0$，状態 $s \leftarrow s_0$
    1. 方策 π により行動 a を取得
    2. 行動 a を実行し，報酬 r と次の状態 s_{t+1} を観測
    3. 更新式 $V(s) \leftarrow V(s) + \alpha \{r + \gamma V(s_{t+1}) - V(s)\}$ の実行
    4. 時刻 $t \leftarrow t + 1$，状態 $s \leftarrow s_{t+1}$
    5. 施行が終端するまでステップ 1 へ戻る

図 3.3: TD 法のアルゴリズム

このことから，式 3.9 は，時刻 t に行動したことによって新たに推定した状態価値（$r_{t+1} + \gamma V^{\pi}(s_{t+1})$ の項）と時刻 t 以前までに推定された過去の状態価値（$V^{\pi}(s_t)$ の項）の差分をとることに等しい．これを TD 誤差と呼ぶ．この更新式では過去の推定値のみを利用していることから，方策 π によって最終的にどのような総和報酬が得られるかを観測する必要がないため，オンライン学習を行うことができるという特性を持つ．図 3.3 に TD 法のアルゴリズムを示す．

3.2.2 sarsa

前項の TD 法は状態価値関数 $V^{\pi}(s)$ を推定するアルゴリズムであった．しかし，制御問題などではエージェントの行動出力が問題となるため，行動価値関数 $Q^{\pi}(s, a)$ を学習することが望ましい．このような場合，sarsa（state-action-reward-state-action）による学習が有効である．sarsa は，時刻 t における状態 s とその時にとり得る行動 a における行動価値 $Q^{\pi}(s, a)$ を，行動 a をとった場合にその行動によって得る報酬 r_{t+1} と，行動後の状態 s_{t+1} において方策 π に従った場合に選択すると考えられる行動 a_{t+1} における行動価値 $Q^{\pi}(s_{t+1}, a_{t+1})$ を用いて更新する．この更新式は以下の式 3.11 によって表される．

1. $Q(s,a)$ の各要素の初期化
2. 時刻 $t \leftarrow 0$，状態 $s \leftarrow s_0$
3. 方策 π により行動 a を取得
    1. 行動 a を実行し，報酬 r と次の状態 s_{t+1} を観測
    2. 方策 π により行動 a_{t+1} を取得
    3. 更新式 $Q(s,a) \leftarrow Q(s,a) + \alpha\{r + \gamma Q(s_{t+1}, a_{t+1}) - Q(s,a)\}$ の実行
    4. 時刻 $t \leftarrow t+1$，状態 $s \leftarrow s_{t+1}$，行動 $a \leftarrow a_{t+1}$
    5. 施行が終端するまでステップ 1 へ戻る

図 3.4: sarsa のアルゴリズム

$$Q^{\pi}(s_t, a_t) \leftarrow Q^{\pi}(s_t, a_t) + \alpha\{r_{t+1} + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) - Q^{\pi}(s_t, a_t)\} \tag{3.11}$$

式 3.11 では TD 法における TD 誤差と同様の差分を用いて更新を行なっていることがわかる．

また，sarsa において一般的な方策 π について学習を行う意義は薄いため，報酬を最大化するような（greedy な）方策をとって最適な行動価値観数 Q^{π} を求めることが多い．ただし，greedy な方策では全ての状態 s と行動 a を選択することができないため，適宜ランダム性を取り入れた ϵ -greedy 選択を用いる．この ϵ -greedy 選択については後述する．そして，学習の進行に応じて greedy な方策に漸近させることで，Q^{π} を求めることが期待できる．図 3.4 に sarsa のアルゴリズムを示す．

3.2.3 Q 学習

Q 学習（Q-Learning）は，sarsa と同様に行動価値関数 $Q^{\pi}(s,a)$ を推定するが，その方策 π に関わらず最適な Q^{*} を直接学習することができる．このことから，sarsa が方策 on 型学習と呼ばれるのに対し，Q 学習は方策 off 型学習と呼ばれる．Q 学習

は，時刻 t における状態 s とその時にとり得る行動 a における行動価値 $Q^{\pi}(s,a)$ を，行動 a をとった場合にその行動によって得る報酬 r_{t+1} と，行動後の状態 s_{t+1} において最も高い行動価値の大きい行動 a_{t+1} を選択すると考えて，$\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$ を用いて更新する．この更新式は以下の式 3.12 によって表される．

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left\{ r_{t+1} + \gamma \max_{a_{t+1} \in A(s_t+1)} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right\} \quad (3.12)$$

式 3.12 において，$r_{t+1} + \gamma \max_{a_{t+1} \in A(s_t+1)} Q(s_{t+1}, a_{t+1})$ の項には $Q^*(s_t, a_t)$ を代入することが直感的な更新式の形である．しかし，実際には $Q^*(s_t, a_t)$ を知ることができないため，最適行動関数 $Q^*(s,a)$ の変形式 3.8 の右辺を代入している..

また，Q 学習は MDP 環境下において全ての状態行動対 (s,a) が十分な回数現れた場合，学習率 α に関する以下の条件式 3.13，3.14 を満たせば，必ず $Q^*(s,a)$ に収束することが知られている．

$$\sum_{t=0}^{\infty} \alpha_t \rightarrow \infty \quad (3.13)$$

$$\sum_{t=0}^{\infty} \alpha_t^2 < \infty \quad (3.14)$$

そのため，方策 π はランダムでも理論的には問題なく収束する．ただし，収束を早めるための手法として sarsa と同様に ϵ -greedy 選択を用いたり，後述する Boltzmann 選択などのソフトマックス手法を利用する．図 3.5 に Q 学習のアルゴリズムを示す．

3.3 行動選択手法

本節では，強化学習の方策 $\pi(s,a)$ に当たる行動選択について説明する．シンプル手法としては，価値関数が最大の状態 s もしくは行動 a を与える greedy 選択，ランダムな出力を行うランダム選択などがあるが，ここではその中間的な手法である ϵ -greedy 選択と Boltzmann 選択を扱う．

1. $Q(s,a)$ の各要素の初期化
2. 時刻 $t \leftarrow 0$，状態 $s \leftarrow s_0$
    1. 方策 π により行動 a を取得
    2. 行動 a を実行し，報酬 r と次の状態 s_{t+1} を観測
    3. 全ての $a_{t+1} \in A(s_{t+1})$ に対し $Q(s_{t+1}, a_{t+1})$ を検索し，最大値 $\max_{a_{t+1} \in A(s_t+1)} Q(s_{t+1}, a_{t+1})$ を取得
    4. 更新式 $Q(s,a) \leftarrow Q(s,a) + \alpha \left\{ r + \gamma \max_{a_{t+1} \in A(s_t+1)} Q(s_{t+1}, a_{t+1}) - Q(s,a) \right\}$ の実行
    5. 時刻 $t \leftarrow t+1$，状態 $s \leftarrow s_{t+1}$
    6. 施行が終端するまでステップ 1 へ戻る

図 3.5: Q 学習のアルゴリズム

3.3.1 ϵ -greedy 選択

ϵ -greedy 選択は，確率 $1-\epsilon$ で greedy な選択を，小さな確率 $\epsilon (1 \gg \epsilon > 0)$ でランダムな選択を行う手法である．方策 $\pi(s,a)$ は以下の式 3.15 で与えられる．

$$\pi(s,a) = \begin{cases} \pi(s, a = argmax_{a \in A(s)} Q(s,a))) & \text{確率 } 1-\epsilon \\ \pi(s, a \neq argmax_{a \in A(s)} Q(s,a))) & \text{確率} \epsilon \end{cases} \tag{3.15}$$

この手法は確率 ϵ で探索的な行動を出力するため，バランスよく学習を行うことが可能である．$\epsilon = 0$ で greedy 選択，$\epsilon = 1$ でランダム選択に一致する．sarsa では学習の進行に応じて $\epsilon \to 0$ に漸近させることで，十分な探索の後に最適行動価値関数 Q^* に収束すると考えられる．また，Q 学習においてはランダム選択を適用しても問題ないため，ϵ が学習期間を通して一定であってもかまわない．しかし，適切な値に調整することで学習の収束を早めることが可能である．

3.3.2 Boltzmann 選択

ソフトマックス手法の代表的な手法として Boltzmann 選択がある．ソフトマックス手法とは式 3.16 で与えられるソフトマックス関数に従って行動の選択確率を等級付けするものである．

$$\frac{\exp(x_i)}{\sum_{k=1}^{n}\exp(x_k)},\ i=1,2,\cdots,n \tag{3.16}$$

Boltzmann 選択はソフトマックス関数としてボルツマン分布の式 3.17 を使用する．ここで温度係数 $T(T>0)$ は方策の性質を左右する重要なパラメータである．Boltzmann 選択の方策 $\pi(s,a)$ の式 3.18 を変形することで，次のような性質を見出すことができる．

$$\frac{\exp(x_i/T)}{\sum_{k=1}^{n}\exp(x_k/T)},\ i=1,2,\cdots,n \tag{3.17}$$

$$\pi(s,a)=\frac{\exp(Q(s,a)/T)}{\sum_{a_k\in A(s)}\exp(Q(s,a_k)/T)} \tag{3.18}$$

$T \to \infty$ のとき，

$$\begin{aligned} \pi(s,a) &= \frac{1}{\sum_{a_k \in A(s_a)} 1} \\ &= \frac{1}{n} \end{aligned} \tag{3.19}$$

$T \to 0$ のとき，

$$\begin{aligned} \pi(s,a) &= \frac{\exp(Q(s,a)/T)}{\sum_{a_k \in A(s),\, a_k \neq a} \exp(Q(s,a_k)/T) + \exp(Q(s,a)/T)} \\ &= \frac{\exp(Q(s,a)/T)}{\frac{\sum_{a_k \in A(s),\, a_k \neq a} \exp(Q(s,a_k)/T)}{\exp(Q(s,a)/T)} + 1} \\ \text{ここで, } \beta &= \frac{\sum_{a_k \in A(s),\, a_k \neq a} \exp(Q(s,a_k)/T)}{\exp(Q(s,a)/T)} \text{ とすると，} \\ \pi(s,a) &= \frac{1}{\beta + 1} \end{aligned} \tag{3.20}$$

まず式 3.19 より，温度係数 T は無限大に近付くとランダム選択に一致することがわかる．また式 3.20 より，温度係数 $T = 0$ で $Q(s,a) \geq Q(s,a_k)$ のとき $\beta \to 0$ のため，$\pi(s,a) \to 1$，$Q(s,a) < Q(s,a_k)$ のとき $\beta \to \infty$ のため，$\pi(s,a) \to 0$ である．以上より，行動 $a_k (a_k \in A(s))$ の中で最大の行動価値を持つ行動 a を確率 1 で選択することから greedy 選択に一致する．

第4章　経路選択における強化学習

小目次

第4章　経路選択における強化学習

本章ではQ-routingとその改良について説明する．はじめに4.1節において本研究の問題設定を再確認し，次いで4.2節においてQ-routingの概要を説明する．4.3節ではMATESに実装する上で追加した改良点について述べる．4.4節では既存の交通流シミュレータとの比較を行う．最後に4.5節では既存の経路探索手法を紹介し，Q-routingと比較する．

4.1　過渡状態を再現するための要件

第2章において，道路ネットワークの変更を伴う交通施策の評価は非常に重要であるが，その際に考慮すべき過渡状態の交通現象は扱いが困難で，既存の交通流シミュレータはどれも十分な再現性を持ち得ないことを説明した．そこで本研究では，この過渡現象を再現するモデルを新たに提案し，交通施策の評価に適用することとする．この問題は車両エージェントの自律性をモデル化することにより解決可能であると考えられるため，本研究では過渡状態を再現するための要件として以下の4点を満たす．

1. 経験依存：過去の走行経験により経路を選択する
2. 部分観測：周囲の限定的な環境のみ観測可能である
3. 多様性：以上2点により，エージェントの経路は多様性を持つ
4. 説明可能性：経路選択について説明可能なログを出力する

本研究ではこの要件を満たすものとして強化学習法に基づく経路選択モデルを採用する．

はじめに，経験依存に関して，強化学習においては価値関数で行動の期待報酬値を保存しているため，過去の走行経験を利用でき，(1)の条件を満たせる．次に部分観測ついては，道路ネットワークのコスト情報が瞬時に取得可能な条件下で経路選択を行うことの問題点は既に指摘した．強化学習法ではエージェントの状態入力を限定することで容易に(2)の条件を満たすことができる．以上2点は過渡状態の発

生に必要であると考えられる要件であった．続く多様性に関しては，合理的に意思決定するエージェントが，グローバルな情報に頼らず経験依存・部分観測を満たして行動する限り，自ずと (3) の条件を満たすことができる．岡山市の路面電車延伸計画のような交通施策についてシミュレーションを行う場合，経路選択を決定する過程は説明可能なものでなくてはならない．確率的な要素の大きい発見的手法に較べ強化学習の行動と報酬は因果関係が明確であり，得られる価値関数はスカラー値のログをシミュレーションのステップ毎に出力することができ，(4) の条件を満たすことができる．

4.2 Q-routing

4.2.1 Q-routing の概要

Q-routing とは強化学習の一つである Q 学習の枠組みに基づいた自律分散型の経路選択アルゴリズムである [26]．この手法はパケットルーティングの分野において提案され，多様なネットワーク環境への適応力が高いことで知られている [26][27][28]．Q-routing では，各ノード x が隣接ノード y とデスティネーションノード d の組に対してそれぞれ行動価値関数 $Q_x(y,d)$ を持ち，これを基に経路を決定する．ここで行動価値関数 $Q_x(y,d)$ とは，x から隣接ノード y を経由してデスティネーションノード d に到達するまでにかかる送信時間の推定値を表す．送信時間はノードにおける待ち時間とリンク上をパケットが移動する時間で占められるが，その大部分はノードにおける待ち時間である．Q-routing の概要を図 4.1 に示す．

4.2.2 価値関数の更新

各ノード x において，パケットが移動する隣接ノード y は行動価値関数 $Q_x(y,d)$ が最小となるものを選択する．パケットは y に移動すると直ちに $Q_x(y,d)$ の更新に必要となる q_y をローカルに情報交換する．q_y は y からデスティネーションノード d までの推定配送時間の最小値であり，式 (4.1) に従って与えられる．

また，ノード x の待ち行列における待ち時間と x から y までのリンクにおける送信時間を t_x としたとき，$Q_x(y,d)$ の更新式は式 (4.2) のようになる．またこのアルゴ

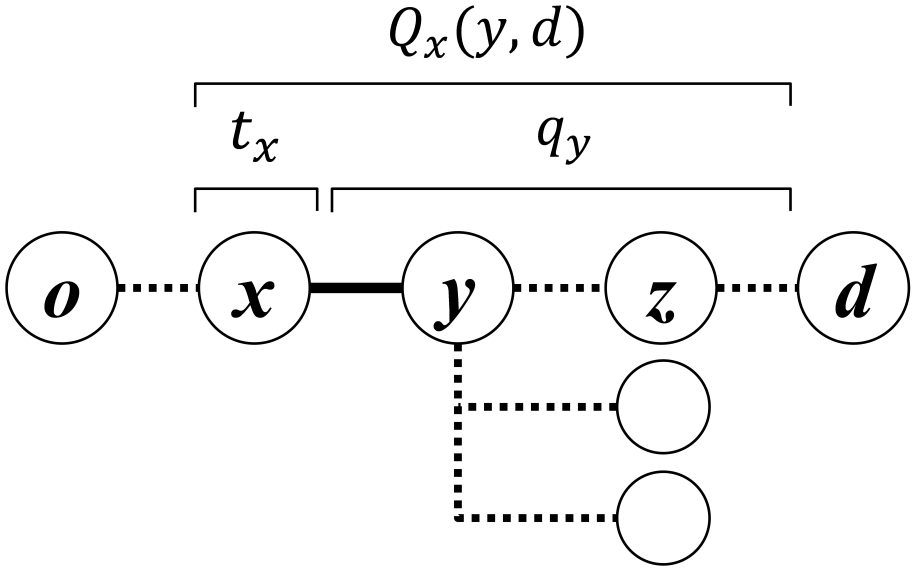

図 4.1: Q-routing の概略

リズムを図 4.2 に示した.

$$q_y = \min_{z \in \text{neighbors of } y} Q_y(z,d) \tag{4.1}$$

$$Q_x(y,d) \longleftarrow Q_x(y,d) + \alpha(t_x + q_y - Q_x(y,d)) \tag{4.2}$$

ここで α は学習率 $(0 < \alpha \leq 1)$ を表す．この式は観測された最新の推定配送時間に $Q_x(y,d)$ を漸近させるため，輻輳が発生するようなノードの $Q_x(y,d)$ は増大しホップ先として選択されにくくなる．このため輻輳が生じ易いトポロジをもつネットワークでもそれを回避するようにルートを決定する．また，パケットが d に到着した際には $q_y = 0$ をとるため，デスティネーションノードに近いノードから順により正確な $Q_x(y,d)$ に更新される．

また，式 (4.2) は前章で示した Q 学習の更新式に類似しているが，Q-routing では扱う問題を経路選択に限定しているため，報酬の最小化を目的としている点と割引率 γ が定義されていない点が異なっている．送信時間を扱う場合，時間は割引現在価値などを導入する必要がなく，どのタイミングにおいても等価であると考えることができるからである．

1. $Q_x(y,d)$ の各要素の初期化
2. 時刻 $t \leftarrow 0$，状態 $s \leftarrow x_O$（Origin ノード）
    1. 方策 π により隣接ノード y を取得
    2. 隣接ノード y に移動し，送信時間 t_x と次の状態 $z \in \mathrm{neighbors\,of}\, y$ を観測
    3. 全ての z に対し $Q_y(z,d)$ を検索し，最小値 q_y を取得
    4. 更新式 $Q_x(y,d) \longleftarrow Q_x(y,d) + \alpha(t_x + q_y - Q_x(y,d))$ の実行
    5. 時刻 $t \leftarrow t+1$，状態 $x \leftarrow y$
    6. 施行が終端するまでステップ 1 へ戻る

図 4.2: Q-routing のアルゴリズム

4.3 MATES への実装

本研究では，上述した Q-routing の交通流シミュレーションにおける経路選択行動への適用を提案する．パケットルーティングと交通流シミュレーションにおける経路選択の違いは，後者ではエージェントが物理的なボリュームを持つことと，ネットワークの性質が異なることである．通信ネットワークではノードでの待ち時間が送信時間を規定したが，ノードを交差点，リンクを単路と考えると，道路ネットワークでは単路における移動時間が旅行時間に相当する．待ち行列の取り扱いと自律的に走行するエージェントの経路選択は異なるため，Q-routing を交通流シミュレーションに応用するために新たな改良を行う必要がある．そこで以下の事項に留意しながら MATES への実装を行った．

4.3.1 意思決定のタイミングの変更

パケットと異なり車両には物理的な制約が存在する．例えば次の交差点を右折するような経路を選択した場合，交差点に進入する以前に右折可能なレーンに車線変更しておく必要がある．そのため図 4.1 に示した従来の Q-routing のように直近の隣

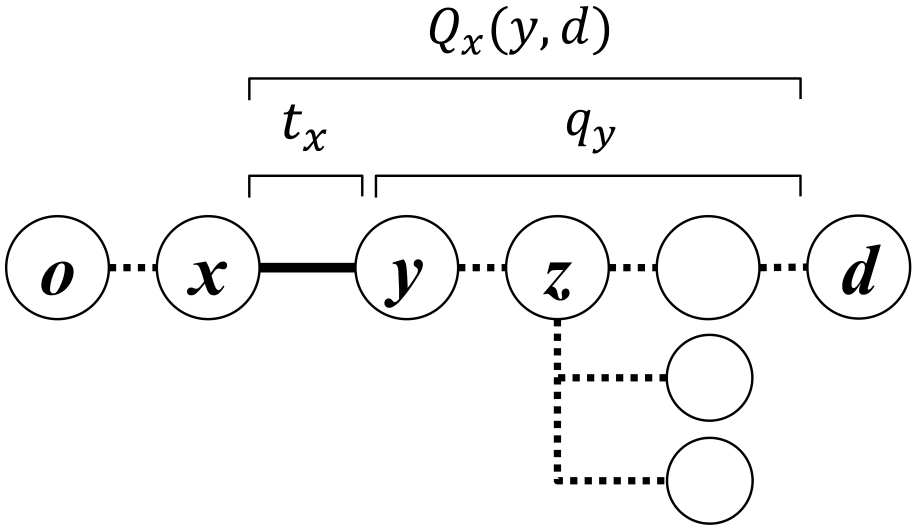

図 4.3: 改良後の Q-routing の概略

接ノード y を選択するのではなく，更に 1 つ先の交差点 z に関して意思決定を行う必要がある．

行動価値関数 $Q_x(y,d)$ を，あるエージェントが交差点 x から隣接交差点 y を経由して目的地 d に到着するまでの推定旅行時間と再定義した新たなモデルを図 4.3 に示す．旅行時間 t_x が交差点ではなく単路にかかる点や，意思決定を行うための行動価値関数の最小値を隣接交差点 y ではなく，その先の z から取得する点が図 4.1 のモデルとは異なっている．

4.3.2 滞留発生時の予測更新

道路交通において右折待ちや信号待ちによる滞留は一般的に見られる現象である．しかし Q-routing における $Q_x(y,d)$ の更新は交差点に到達した時点で行われるため，単路内での滞留はこの更新のタイミングを遅らせることにつながる．これは本来渋滞している経路へのルーティングを誘発し，最悪の場合はグリッドロックを引き起こす可能性がある．さらに，信号制御によるリンク（単路）内の滞留は通信ネットワークで想定されていないため，新たに考慮すべき点である．本研究では混雑の状況を素早く反映することを目的に，滞留の先頭にいるエージェントの行動価値関数の更新タイミングを前倒しすることを考える．先頭のエージェントは青信号になる

と同時に交差点へ侵入できるため，残りの赤信号の点灯時間を予め見積に含んだ予測旅行時間を行動価値関数の更新に使うこととする．その後，青信号になり真の旅行時間が確定した時点で，実測値を用いた行動価値関数の補正も行う．更新式を式 (4.3)-(4.5) に示す．

予測旅行時間 $\tilde{t_x}$ は，滞留の先頭に存在するエージェントが赤信号により停車し滞留が開始した時点での移動時間 t'_x と，信号待ち時間の期待値の和によって見積る．信号待ち時間の期待値は赤信号点灯時間である $RedSignal$(今回は 70 秒に設定) の 1/2 とした．また，最終的に確定する単路の旅行時間 t_x による補正は式 (4.5) のとおりである．式 (4.5) を式 (4.3) に代入することで本来の更新式である式 (4.2) に一致する．

滞留の先頭で赤信号により停車

$$Q_x(y,d) \longleftarrow Q_x(y,d) + \alpha(\tilde{t_x} + q_y - Q_x(y,d)) \tag{4.3}$$

$$\tilde{t_x} = t'_x + \frac{RedSignal}{2} \tag{4.4}$$

青信号になり交差点に進入

$$Q_x(y,d) \longleftarrow Q_x(y,d) - \alpha(\tilde{t_x} - t_x) \tag{4.5}$$

4.3.3 道路ネットワーク変化時の価値関数補完

交通施策の評価を行う場合，道路ネットワークの変化について扱う必要がある．ネットワークの変化は表 4.1 に示した 4 つが考えられる．トランジットモールについては 2.6.1 項ですでに述べた．

本研究では，施策の適用前の学習結果である行動価値関数を適用後のネットワークのシミュレーションで初期値として用いる．この時，車線数の増減は行動価値関数の要素数に変化がなく，単路の削除は該当する要素も同時に削除することで問題

表 4.1: 交通施策の事例

ネットワークの変化	交通施策
車線数の増加	道路の拡張，右折レーンの新設
車線数の減少	事故・工事による車線閉塞
単路の増加	バイパスの新設
単路の削除	トランジットモールの導入

なく行動価値関数を読み込むが，単路が増加するケースでは新たな要素が発生する．例えば 2 つの交差点 a と b の間に単路が新設された時，交差点 a から交差点 b を経由して目的地 d へ向かう際の行動価値関数 $Q_a(b,d)$ は未知である．施策適用後のシミュレーション時にこの要素の初期値を極端な値に設定してしまうと学習に悪影響を及ぼす懸念があるため，ここでは交差点 b における推定旅行時間の最小値 q_b を代入する．式 (4.1) によれば，この値は真の $Q_a(b,d)$ よりも ab 間の旅行時間 t_a だけ小さく，この経路が本来よりも僅かに選択されやすくなっている．t_a についてはどのエージェントも走行経験を持たないため，学習初期に選択確率を高めておくことで早期に正確な推定値を得ることができる．

また，実験を行うに当たって新たに ϵ-greedy 選択と Boltzmann 選択を実装し，それぞれの導入を検討した．エージェントにランダムな行動選択の機会が与えられることによって選択可能なルートの価値関数を万遍なく更新するため，局所解への収束を回避でき，同時に現実に起こり得るランダム性をモデル化することにもなつながる．また，学習が収束した後に道路状況が変化するようなケースでも，新たな経路を容易に学習しなおすことが可能となる．ここまでの内容を，MATES における Q-routing のシミュレーションフローとして図 4.4 に示す．

次章以降で行う実験に向け，現在 nMATES の経路選択で採用している A*に従うエージェントと Q-routing に従うエージェントが共存する形での実装を行った．また，シミュレータ上では多数のエージェントが存在するため，それらが個別に学習すると全体の挙動が安定しない．この同時学習の問題を解決するため，共通の目的地を持つエージェント群は価値関数を共有するものとした．この設定により，エー

ジェントは自らの走行経験以外の要素（例えばエージェント同士の情報交換による影響や，その街・道路ネットワークにおける暗黙知の形成）を考慮でき，明示的にエージェントの協調行動を誘発させる必要なく学習を収束させることが可能である．

4.4 既存のシミュレータ

本研究ではここまで述べてきたQ-routingをMATESに実装する．本節では，既存の知的マルチエージェント型の交通流シミュレータにおいて，学習などを応用した経路選択を行っているものについて紹介する．

そもそも，交通流シミュレータの経路選択行動は(1)経路選択モデルを外生するもの，(2)経路選択モデルを内包するもの，の2つに大別することができる．(1)のモデルは発生交通量などを与える際，同時に走行経路を設定するものである．このような設計ではシミュレーションのリアルタイム性は向上するものの，エージェントの経路選択が限定されてしまい交通状況を動的に反映することができない．(2)のモデルではエージェントの行動モデルに経路選択が組み込まれており，シミュレーションの状況に応じて経路選択を行うことができる．多くは経路の一般化費用を算出するものであり，従来のMATESに実装されていたA*もこのモデル群に含まれる．

一方，交通流シミュレータに学習モジュールを組み込んだ例としてはMATSim (Multi-Agent Transport Simulation) [29] がある．MATSimは，各エージェントの1日の行動と学習をシミュレートすることで交通状況の変化を再現する．MATSimにおいて，エージェントはODや経由地，出発・到着時刻や滞在時間を1日ごとの行動計画プランとして保持し，日々のシミュレーションを繰り返す．そしてその結果を基に各エージェントが学習を行い，1日のプランとシミュレーション結果との誤差から評価値を算出して，その値が最良のプランを実行する．同時に，観測に基づく各ルートの予想通過時間などから新たなプランを作成することもできる．

また，学習ではなくファジィ推論による経路選択を組み込んだ例としてMITRAM (MIcroscopic model for analyzing TRAffic jaMs in the city area)[30] がある．MITRAMではドライビングシミュレーションによる被験者実験やアンケート調査からファジィ推論則を生成し，AHP及び多項ロジットモデルと組み合わせた経路選択を行う．得られた出力は現実の交通データによる検証によって調整される．これによりエージェ

ントの多様性をあいまいさによって表現し，学習に比べ普遍的な (走行経験によって更新されない) 経験則に基づいた経路選択を実現する．

このように経路選択一つをとっても様々なモデルが提案さている．そのなかで，今回扱う Q-routing には走行しながらリアルタイムに学習と経路選択を行うという特徴がある．MATSim のように出発の時点で走行プランを作成する必要がないため，例えば交通事故による突発的な渋滞に対しても状態入力を基にエージェントが自律的な走行を継続する．また，強化学習ベースであることから，何故その経路が選択されたかを定量的に解析することが可能である．MITRAM ではファジィ推論則等を用いている運転者の経験則も，Q-routing ではスカラーの価値関数として表現されるため扱いが容易である．

4.5 経路選択アルゴリズムにおける Q-routing の位置づけ

定式化された任意の探索問題が与えられたときに，その解を求めるアルゴリズムを探索アルゴリズムという．適正な形で実装された探索アルゴリズムはどのような探索問題でも解くことができるため，歴史的に，この技術は一般的な問題解決を行うための知能の基盤となる技術と考えられてきた．本研究では対象問題が車両エージェントの経路選択であるため，これらを特に経路選択アルゴリズムと称して説明を行う．

経路選択アルゴリズムはオフライン探索とオンライン探索の 2 つに分類することができる．オフラインという言葉は，エージェントが環境から切り離されて問題解決をする状況を表現している．実際，オフライン探索においては，環境に関する全ての情報が事前にエージェントに与えられる．エージェントは十分に時間をかけて経路選択を行い，最適解を見つけ，その後に解を実行する．つまり，経路選択は 1 度だけ徹底的に行われ，その後，出発地点から目的地点に至る一連の行動が計画的に実行される．一方，オンラインという言葉は，エージェントが環境中に存在し，両者が相互作用し得る状況を表現している．実際，オンライン探索においては，環境に関する情報の全てがが事前にエージェントに与えられるのではなく，エージェントと環境が相互作用することによって，環境情報が少しずつエージェントに開示さ

れることを想定している．そして，現在までに知ることができた部分観測に基づく局所的な探索と，その結果得られる不完全な解から決定される行動の実行を交互に繰り返す．したがって，1つ1つの行為はその場で動的に決定されるものであり，事前に計画的に定まっているものではない．特に，局所的な先読み探索に使用できる時間が（比較的小さな）一定時間であるようなオンライン探索を実時間探索と呼ぶ．

以下では，オフライン探索として従来のMATESにも実装されていたA*を，オンライン探索としてRTA*，LRTA*を説明し，Q-routingと共に本研究の問題設定における位置づけを示す．

4.5.1 A*

現在，研究が最も盛んな探索アルゴリズムのひとつにA*がある．A*はヒューリスティック関数という見積もりの値を利用することで，効率的に探索を行うことのできるアルゴリズムである．A*において，ノードxからノードyまでの最短旅行時間を$g(y)$，ノードyからノードdまでの最短旅行時間を$h(y)$と定義すると，ノードxからyを経由してdへ到着するような経路の最短旅行時間$f(y)$は式(4.6)で表される．ここで，$g(y)$はノードyまでの過去の経験を表現し，$h(y)$はノードy以降の将来の見積もりを表現するため，取り得る行動の中から$f(y)$の小さいものを優先的に探索することで，素早く解を得ることを目的としている．

$$f(y) = g(y) + h(y) \tag{4.6}$$

経路選択における具体的なアルゴリズムは図4.5のようになる．

1. $g(y)$，$h(y)$ の各要素の初期化
2. オープンリストとクローズドリストの初期化
3. オープンリストに初期ノード (Origin ノード) を追加
    1. オープンリストから $f(y)$ が最小のノードを選択
    2. 旅行時間 $g(y)$ を取得
        1. $g(y)$ によって $f(y)$ を算出し，y がクローズドリストにあり $f(y) \geq h(x)$ なら y をオープンリストに追加しない．
        2. それ以外の場合には y をオープンリストに追加する．
    3. 施行が終端するまでステップ 1 へ戻る

図 4.5: A*のアルゴリズム

また，A*は，リンクのコストが非負であり，ヒューリスティック関数が許容的であれば，以下の 2 つの性質を持つことが示されている．

- 完全性：解が存在するとき，その解を見つけることが保証されている．
- 最適性：複数の解が存在するとき，最良の解を見つけることが保証されている．

ただし，交通流シミュレータにおける経路選択においてリンクのコストは非負であるため，実際にはヒューリスティック関数が許容的であればこれらの性質を持つことになる．許容的であるとは，$h(y)$ が非負で，実際の最短旅行時間 $h^*(y)$ を超えない，つまり楽観的な見積もりであることを言う．式 (4.7) で表される．

$$h^*(y) \geq h(y) \geq 0 \tag{4.7}$$

このように，A*は探索中の全ての局面を記憶するので，莫大な節点数を持つ探索空間で経路を求めるときは，メモリの使用量が大きくなるという問題があった．現在，この問題を解決するために A*には様々なアプローチから改良が加えられ新たな

アルゴリズムが提案されている．表 4.2 に A*に関連した既存の経路選択アルゴリズムをまとめ，オフライン探索とオンライン探索に分類した．次項では，このうちの基礎的なオンライン探索に分類される，RTA*と LRTA*のアルゴリズムについて説明する．

表 4.2: 経路選択アルゴリズムの分類

	経路選択アルゴリズム
オフライン探索	A*
	AA* (Adaptive A*)[31]
	GAA* (Generalized Adaptive A*)[32]
	D*Lite[33]
	IDA* (Iterative-Deepening A*)[34]
	AWA* (Anytime Weighted A*)[35]
	PRA* (Parallel Retracting A*)[36]
オンライン探索	RTA* (Real-Time A*)[37]
	LRTA* (Learning Real-Time A*)[37]
	RTAA* (Real-Time Adaptive A*)[38]

4.5.2 RTA*

RTA*は基礎的なオンライン探索アルゴリズムである．RTA*は A*と同様に，ノード x からノード y までの最短旅行時間を $g(y)$，ノード y からノード d までの最短旅行時間を $h(y)$ とし，その和 $f(y)$ を利用して経路を決定する．オンライン探索の性質上，アルゴリズムの実行が進むにつれて，$h(y)$ の値が動的に変更されていく点が異なる．RTA*はヒューリスティック関数が許容的であれば，完全性を持つ．経路選択における具体的なアルゴリズムは図 4.6 のようになる．ヒューリスティック関数の更

新には本来最も小さい値を利用すべきだが，2 番目に小さい値 $f(\tilde{z})$ によって旅行時間を過大に見積もっておくことにより，ノード y を再訪した際の探索を効率化している．RTA*はこの性質により 1 回目の探索である程度質の良い解を実行できるが，繰り返し探索を行うとヒューリスティック関数が過大になり，解は次第に劣化していく可能性がある．

1. $g(y)$，$h(y)$ の各要素の初期化
2. 状態 $x \leftarrow O$（Origin ノード）
    1. $f(y)$ が最小の隣接ノード y を取得
    2. 隣接ノード y に移動し，旅行時間 $g(y)$ と
       隣接ノード $z \in$ neighbors of y を観測
    3. 全ての z に対し $f(z)$ を検索し，2̌ 番目に小さい値 $f(\tilde{z}) = \mathrm{secondmin}_{z \in \text{neighbors of } y} f(z)$ を取得
    4. $h(y) \leftarrow f(\tilde{z})$ に更新
    5. 時刻 $t \leftarrow t+1$，状態 $x \leftarrow y$
    6. 施行が終端するまでステップ 1 へ戻る

図 4.6: RTA*のアルゴリズム

4.5.3 LRTA*

LRTA*は RTA*と同じく基礎的なオンライン探索アルゴリズムである．アルゴリズムは図4.7のとおりで，1 点を除いて RTA*と同じである．しかし，LRTA*はヒューリスティック関数が許容的であれば完全性を持つ他，収束性を持つことも知られている．

- 収束性：繰り返し探索を十分に行うとヒューリスティック関数が真の値に収束する

LRTA*ではヒューリスティック関数の更新に $f(z)$ の最小値を利用する．これによりRTA*のように旅行時間を過大に見積もることがなくなるため，ヒューリスティック関数が収束する．ただし，1回目の探索で実行する解は一般的にRTA*よりも悪く，繰り返し探索を行うなかで次第に改善していく．

1. $g(y)$，$h(y)$ の各要素の初期化
2. 状態 $x \leftarrow O$（Origin ノード）
    1. $f(y)$ が最小の隣接ノード y を取得
    2. 隣接ノード y に移動し，旅行時間 $g(y)$ と
       隣接ノード $z \in$ neighbors of y を観測
    3. 全ての z に対し $f(z)$ を検索し，
       2番目に小さい値 $f(\tilde{z}) = \min_{z \in \text{neighbors of } y} f(z)$ を取得
    4. $h(y) \leftarrow f(\tilde{z})$ に更新
    5. 時刻 $t \leftarrow t+1$，状態 $x \leftarrow y$
    6. 施行が終端するまでステップ1へ戻る

図 4.7: LRTA*のアルゴリズム

4.5.4 定常状態と過渡状態の再現性

ここまで，Q-routing，A*，RTA*，LRTA*の説明を行った．その内容をまとめると表4.3のようになる．

まず提案手法のQ-routingについて，学習の過程で過渡状態を再現できるだけでなく，収束後は定常状態の交通流を再現することが可能である．一方，代表的なオフライン探索であるA*は，最適性が保証された解が常に得られるため定常状態こそ再現できるが，環境に関する全ての情報が事前にエージェントに与えられるため過渡状態を再現することはできない．RTA*はオンライン探索でありながら1回目の探索で良い解を得ることができるものの，ヒューリスティック関数を過大に見積もり続けるため解が安定しない．そのため，本研究における問題設定では定常状態も過

渡状態も再現することができない．最後に LRTA*について，Q-routing と同様繰り返し探索の過程で過渡状態を再現できるだけでなく，収束後は定常状態の交通流を再現することが可能である．

これらの性質を基に，次章の実験には Q-routing の比較対象として A*と LRTA*を用いる．これは定常状態の再現性に関しては A*が，過渡状態の振舞いに関しては LRTA*が Q-routing の参照解と考えられるためである．

表 4.3: 定常状態と過渡状態の再現性

	定常状態	過渡状態
Q-routing	○	○
A*	○	×
RTA*	×	×
LRTA*	○	○

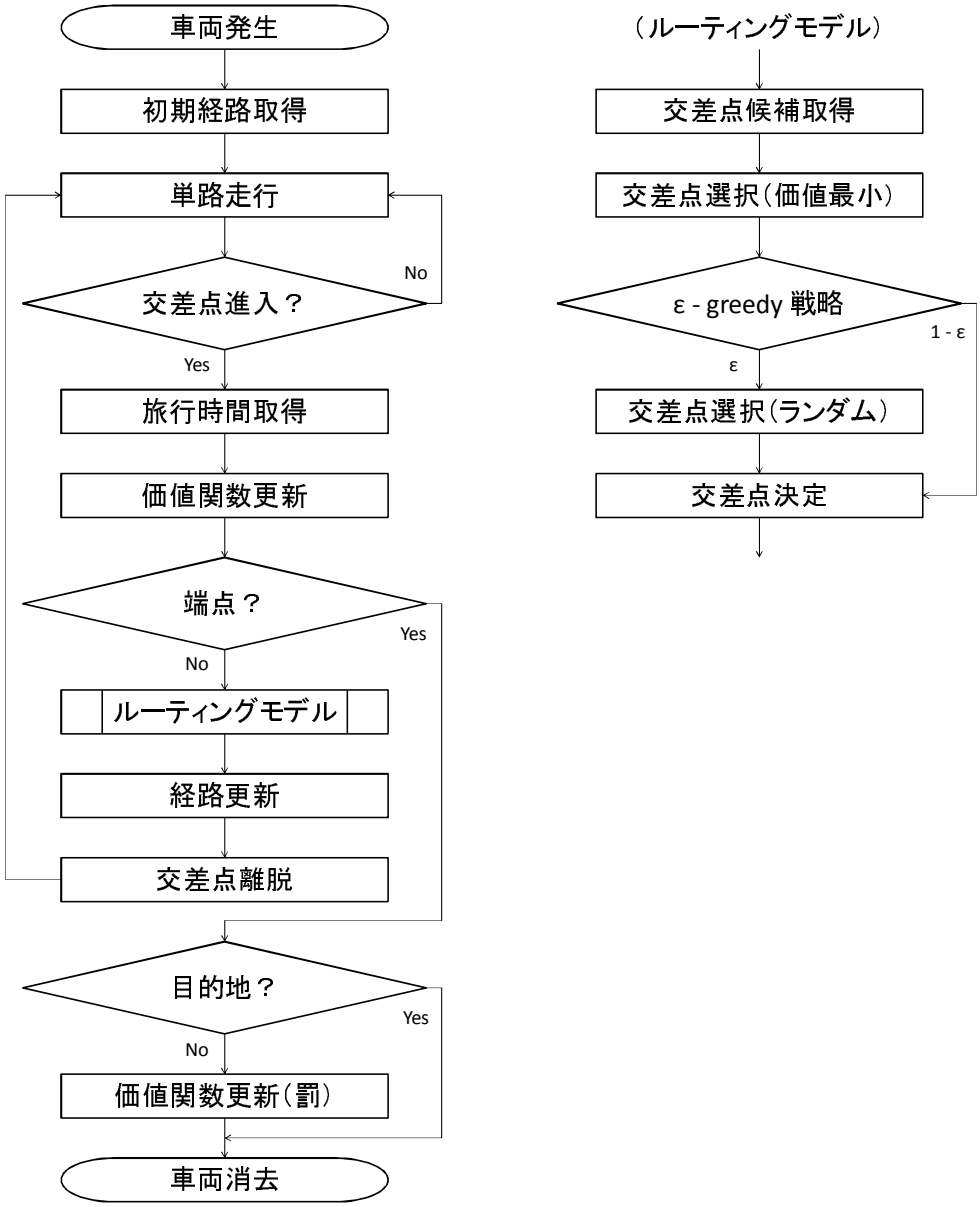

図 4.4: Q-routing のフローチャート

第5章　実験と考察

小目次

第5章　実験と考察

本章では提案手法によるシミュレーションを行う．5.1 節では前章で述べた Q-routing による経路選択モデルの性能検証のために，不規則格子の道路ネットワークを用いた予備実験を行う．交通流の基本的な条件下で既存手法との比較を行い，結果について考察する．更に 5.2 節では，現実の交通施策として岡山市で検討されている路面電車の軌道延伸計画を取り上げ，交通流が実施直後の過渡状態から定常状態に収束するまでをシミュレーションする．

5.1　実験1：不規則格子でのシミュレーション

5.1.1　実験環境

前章で述べた Q-routing による経路選択モデルの性能検証のために，先行研究 [26] を参考に不規則格子の道路ネットワークを用いた予備実験を行った．図 5.1 に道路ネットワークのトポロジを示す．リンクの距離は全ては 200[m] である．端点となる各ノードからエージェントが流入し，ランダムな目的地を持った車両がそれぞれ等しい任意の交通量で発生する．エージェントは目的地に到着した時点でネットワークから消去され，発生から到着までの時間を旅行時間として記録する．このネットワークの特徴として左右のネットワークをつなぐ中継リンクが 10-11 と 26-27 の 2 本に制限されている点が挙げられる．このため，全てのエージェントが最短経路を選択すれば中央部の中継リンク 26-27 において渋滞が発生する．そこで，このリンクを含む推定旅行時間が長い場合には上部の中継リンク 10-11 へ迂回する経路選択を行う必要がある．

5.1.2　条件設定1：静的な学習

これまで経路選択の目的関数が旅行時間の最小化であることを前提に説明を行なってきたが，本項ではまず旅行距離の最小化を目的とした実験を行う．道路状況によっ

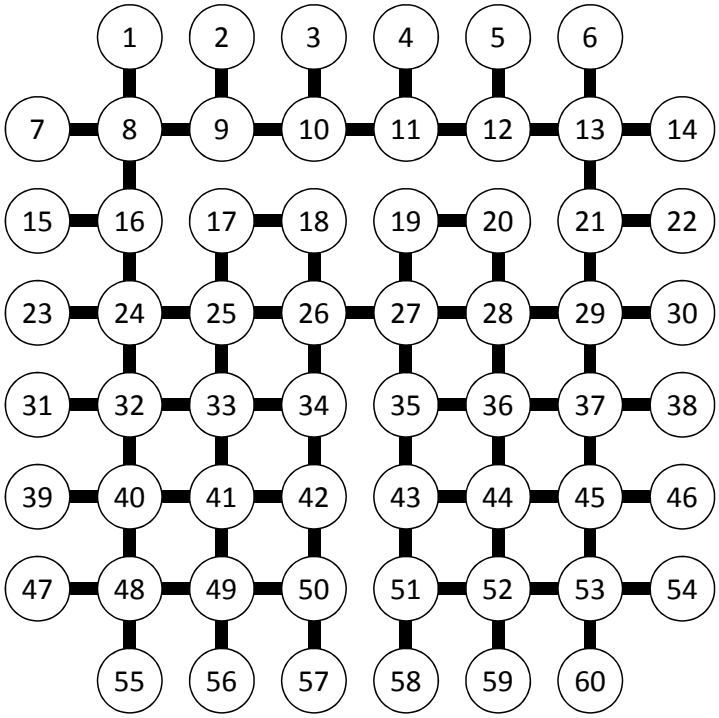

図 5.1: 実験に使用した道路ネットワーク

て Dynamic（動的）に変化する旅行時間に比べて，旅行距離は Static（静的）であることから，学習アルゴリズムを含んだ経路探索手法の性能検証として重要である．

実験では A*，LRTA*，Q-routing の各手法を比較する．Q-routing については行動選択手法として ϵ-greedy 選択と Boltzmann 選択の比較も行う．全ての手法の目的関数を旅行距離の最小化とする．また，LRTA*のヒューリスティック関数 $h(x)$ はユークリッド距離を $[0, 1)$ の範囲におさまるよう正規化した値で初期化する．通常，LRTA*はユークリッド距離やマンハッタン距離によって初期化するが，不規則格子の環境と今回の静的な条件設定においてはヒューリスティック関数の収束値を事前に与えてしまうためである．Q-routing のルーティングテーブル $Q_x(y, d)$ も同様に $[0, 1)$ で正規化したユークリッド距離で初期化し，学習率 $\alpha = 0.3$，$\epsilon = 0.05$（ϵ-greedy 選択の場合），温度係数 $T = 200$（Boltzmann 選択の場合）とした．

発生交通量が各端点で 50[台/h] とし，信号制御は行わなかった．制御無しの場合は信号の現示は常に青であるが，MATES ではエージェントがお互いに衝突を回避する行動をとる．

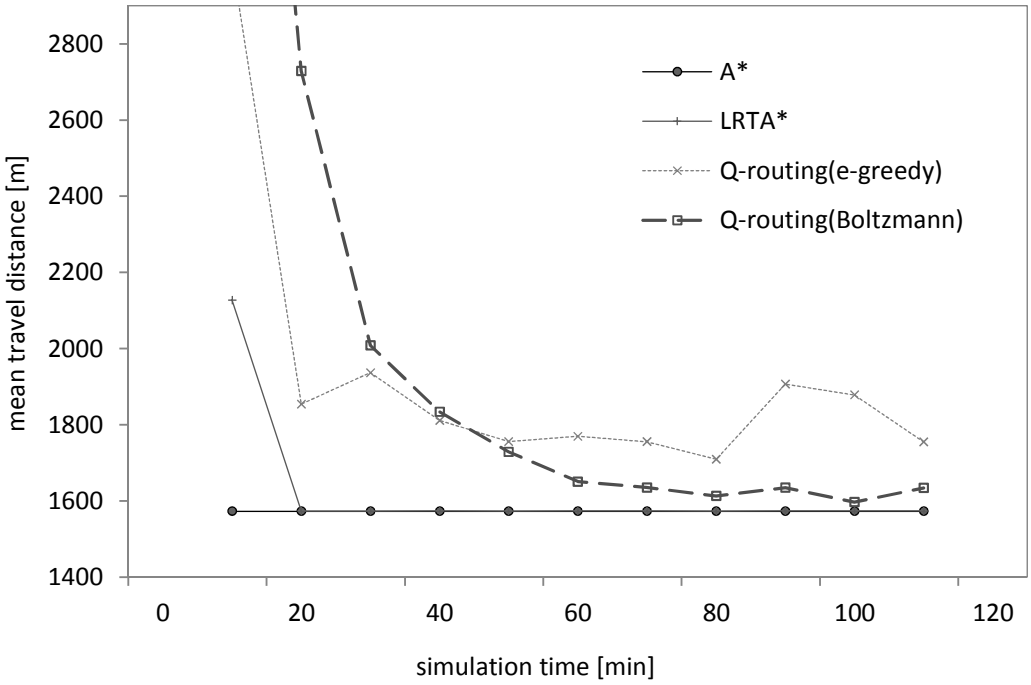

図 5.2: 各手法の平均旅行距離

5.1.3 結果と考察

図 5.1 のノード 1 を出発地，ノード 56 を目的地としたエージェントの平均旅行距離を図 5.2 に示す．縦軸は 10 分ごとの平均旅行距離 [m]，横軸は MATES 内のシミュレーション時間 [min] である．道路ネットワークが格子であるため最短距離経路（以降，本条件設定では単に最短経路と呼ぶ）は複数存在し，その距離は 1600[m] であるが，交差点内は斜めに走行することが可能であるため，実際の出力値は約 1570[m] となる．A*は事前に全リンクのコスト情報を取得するオフライン探索手法であるため，静的な条件ではエージェントは必ず最短経路を走行する．LRTA*はシミュレーション開始後 20[min] で最短経路に収束し，その後も安定した．行動選択の異なる Q-routing 同士を比較すると，Boltzmann 選択を用いた場合の収束が良く，ϵ-greedy 選択に比べて安定して収束していることがわかる．

学習アルゴリズムを含んだ手法を比べると LRTA*の収束が最も良くなった．ヒューリスティック関数の初期値が良質であるため，学習率 α を定義せず直近の経験をそのまま行動に反映させる LRTA*が有効であったことによる．また，静的な条件下での不

規則格子の問題空間が単純なため，ノードの幾何的な接続関係を考慮せずQ-routingに比べ状態数が小さく抑えられる点も収束が早かったことの理由である．

ϵ-greedy選択とBoltzmann選択はどちらも収束までに60[min]程度要している．これは，ノード1を出発地，ノード56を目的地としたエージェントの最短経路が，リンク24-25，32-33，40-41，48-49，のどれかを経由する4通りに分かれるためである．各経路の距離が正確に一致することから，探索的な行動が誘発され，結果としてLRTA*に比べ収束が困難であったと考えられる．ただし，両手法の学習が一定程度収束した60[min]以降，ϵ-greedy選択の平均旅行時間は不安定に推移する一方，Boltzmann選択は安定している．これは，次善の行動に対する選択確率の重み付けに起因するものである．例えば，学習が十分に収束した時点でのノード8における行動選択を考えてみると，最良の行動はノード16へ直進することであり，ノード8からノード16を経由してノード56へ向かう場合の最短距離は1400[m]なので，ルーティングテーブルは$Q_x(y,d) = Q_8(16,56) = 1400$，同様に次善の行動は$Q_x(y,d) = Q_8(9,56) = 3000$となっているはずである．このとき$\epsilon = 0.05$の$\epsilon$-greedy選択ではノード8が選択される確率は95%であるのに対し，温度係数$T = 200$のBoltzmann選択ではほぼ100%となる．つまり，ϵ-greedy選択は収束の度合いに関係なくランダムな選択をし続けるが，Boltzmann選択は収束によってgreedy選択の性質に近づくため，より安定した学習曲線を描くのである．端的な例として，ϵ-greedy選択の場合の，シミュレーション時間90[min]におけるあるエージェントの走行経路を図5.3に示す．ルーティングテーブルの値は真の値に収束しているにも関わらず，エージェントの経路に反映されない可能性を示している．同じ時間帯におけるBoltzmann選択の場合にはこのようなエージェントは存在しなかった．

また，この性質の違いは最終的に収束した経路にも現れている．図5.4〜5.7にそれぞれA*，LRTA*，Q-routingのϵ-greedy選択，Boltzmann選択の主な経路を示す．上から順にシミュレーション時間10[min]，20[min]，30[min]，120[min]（実験終了）時点での経路である．赤いほどその経路を選択するエージェントが多く，黄色が薄くなるほど少ない．ここで，A*は当然学習を行わないため不変であるが，LRTA*とϵ-greedy選択も最短経路のうちの1つに早期に収束している．一方で図5.4のBoltzmann選択では4つある最短経路の全てにエージェントが分散したまま安定していることが分かる．この4つの経路の距離は正確に一致するものの，学習中の乱数の影響など

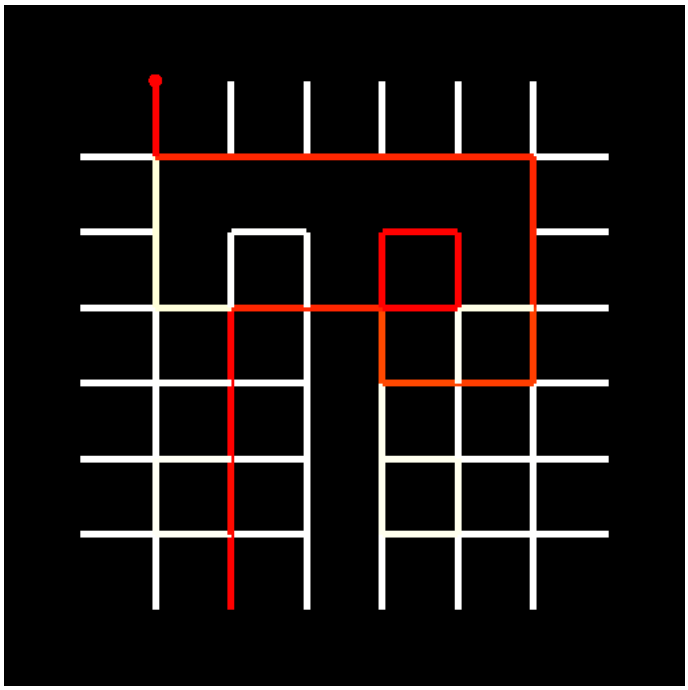

図 5.3: ϵ の影響を受け迷走するエージェントの例

によりルーティングテーブルの推定値には誤差が生じることがあるが，ϵ-greedy 選択などではそれを過大に反映した結果，エージェントの経路選択に多様性が失われている．しかし，ソフトマックス関数によって選択確率を算出する Boltzmann 選択では等距離の経路には同程度の選択確率が振り分けられるため，エージェントの経路選択の多様性を保ったまま学習が収束している．このことから，学習アルゴリズムを含んだ手法の中でも Boltzmann 選択が最も現実のエージェントに近い挙動を示していると考えられる．

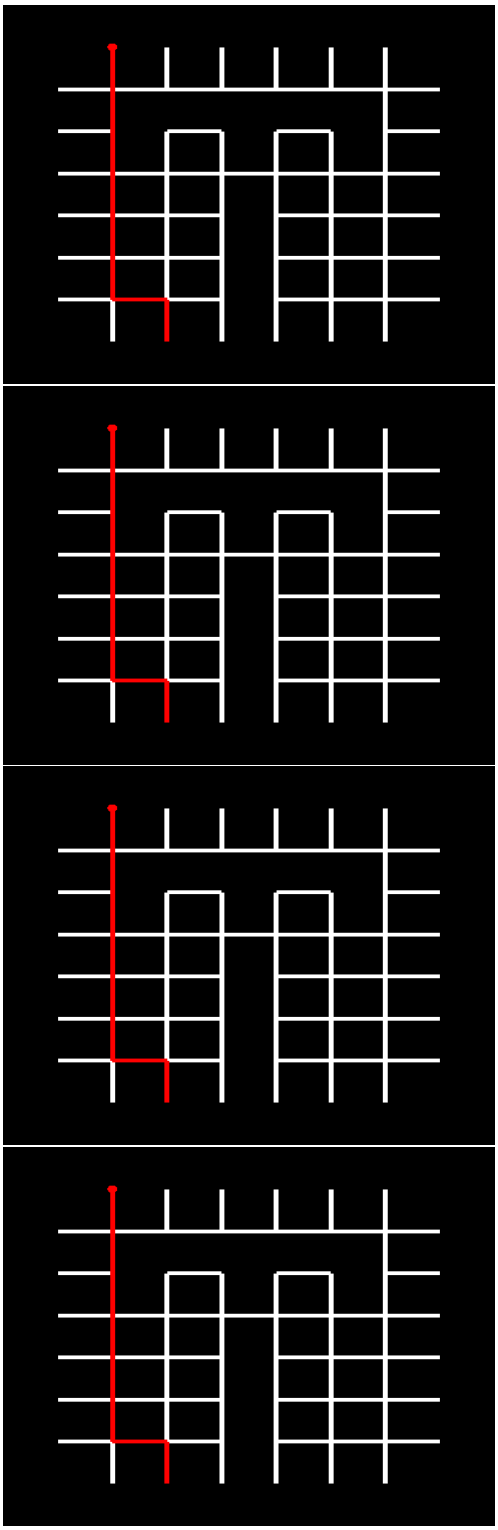

図 5.4: A*の経路の遷移

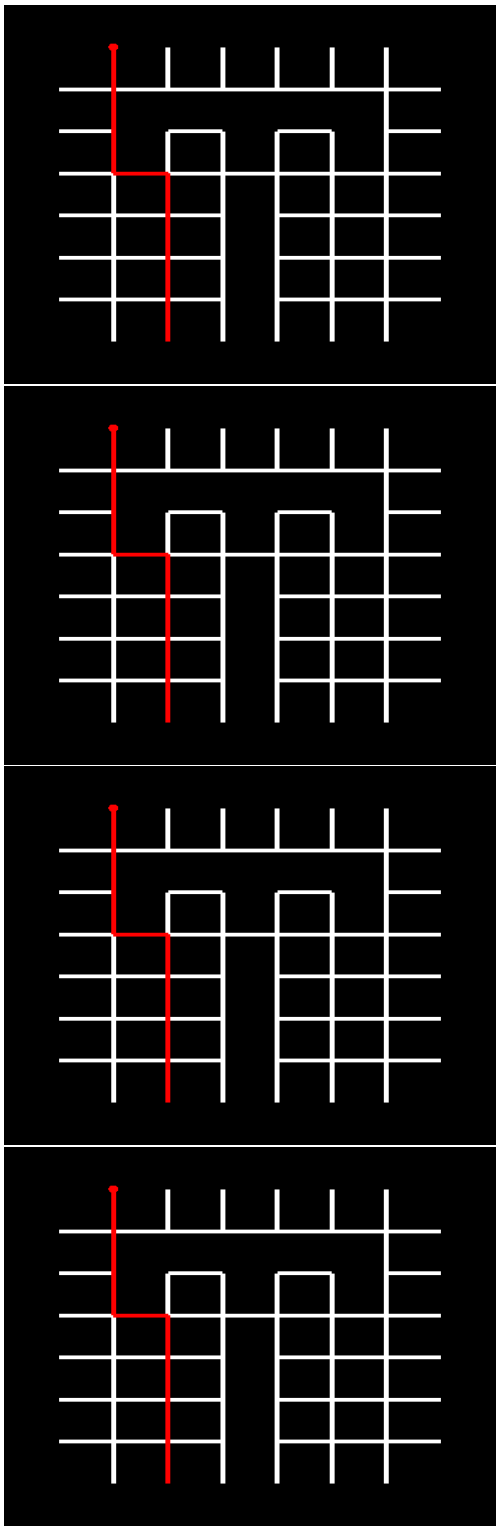

図 5.5: LRTA*の経路の遷移

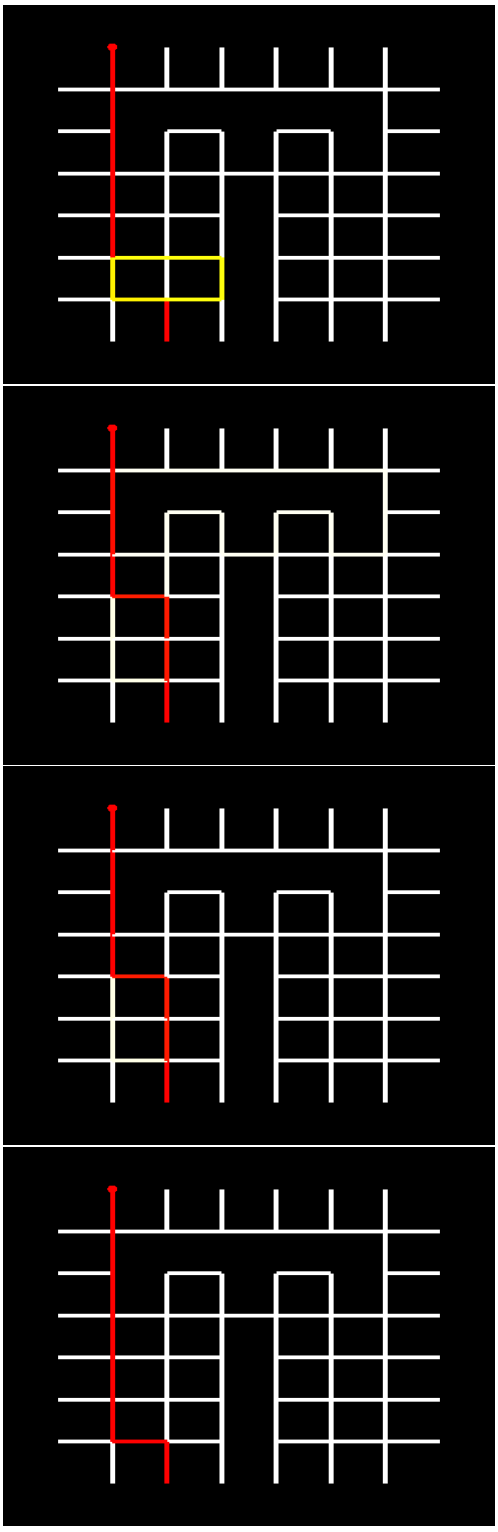

図 5.6: Q-routing(ϵ-greedy 選択) の経路の遷移

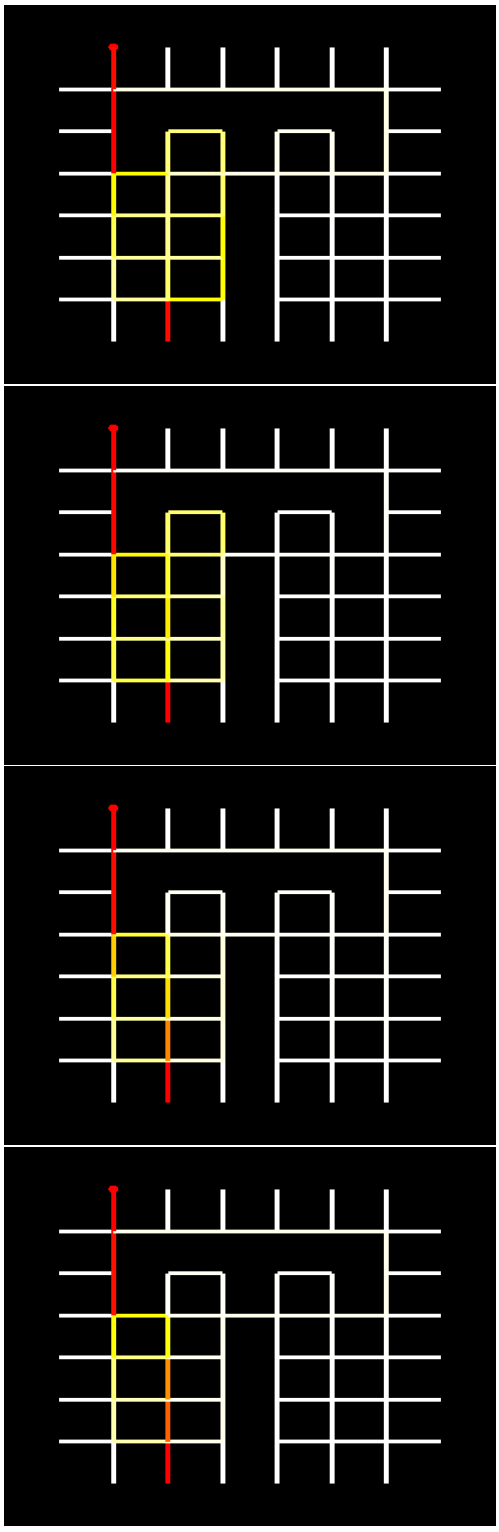

図 5.7: Q-routing(boltzmann 選択) の経路の遷移

5.1.4 条件設定 2：動的な学習

本項では，道路状況によって Dynamic（動的）に変化する旅行時間を最小化する学習を行う．実験では同様に A*，LRTA*，Q-routing（ϵ-greedy 選択，Boltzmann 選択）の比較を行う．LRTA*のヒューリスティック関数 $h(x)$ と Q-routing のルーティングテーブル $Q_x(y,d)$ はユークリッド距離で初期化する．学習率 $\alpha = 0.3$，$\epsilon = 0.05$（ϵ-greedy 選択の場合），温度係数 $T = 200$（Boltzmann 選択の場合）とし，発生交通量が各端点で 30[台/h]，300[台/h] とした．また，Q-routing の提案された通信ネットワークでは存在しなかった，信号制御による対流の影響への適応可能性を検証するため，各交差点において信号制御を有り，無しとした 2 つケースを想定した．制御有りの場合の信号パラメータはサイクル長 140 秒，各現示は交差する両方向ともに赤 70 秒，青 55 秒，右矢印 10 秒，黄色 5 秒とし，オフセットは設定しなかった．

5.1.5 結果と考察

図 5.8〜図 5.11 に各ケースの平均旅行時間の推移を示す．図 5.8 は発生交通量 30[台/h] で信号制御無し，図 5.9 は発生交通量 300[台/h] で信号制御無し，図 5.10 は発生交通量 30[台/h] で信号制御有り，図 5.11 は発生交通量 300[台/h] で信号制御有り，である．横軸はシミュレーション時間 [min]，縦軸は 10 分毎の平均旅行時間 [sec] である．

全体的な傾向として，静的な学習の場合と同様，A*は常に安定して定常状態と呼べる交通流を実現している．また LRTA*は比較的早く学習が収束し，やはり安定した交通流を実現している．Q-routing はどちらの行動選択の場合も学習曲線が類似しており，収束までに要するシミュレーション時間がほぼ同じである．一方，発生交通量や信号制御の有無によっては傾向が異なるものもある．

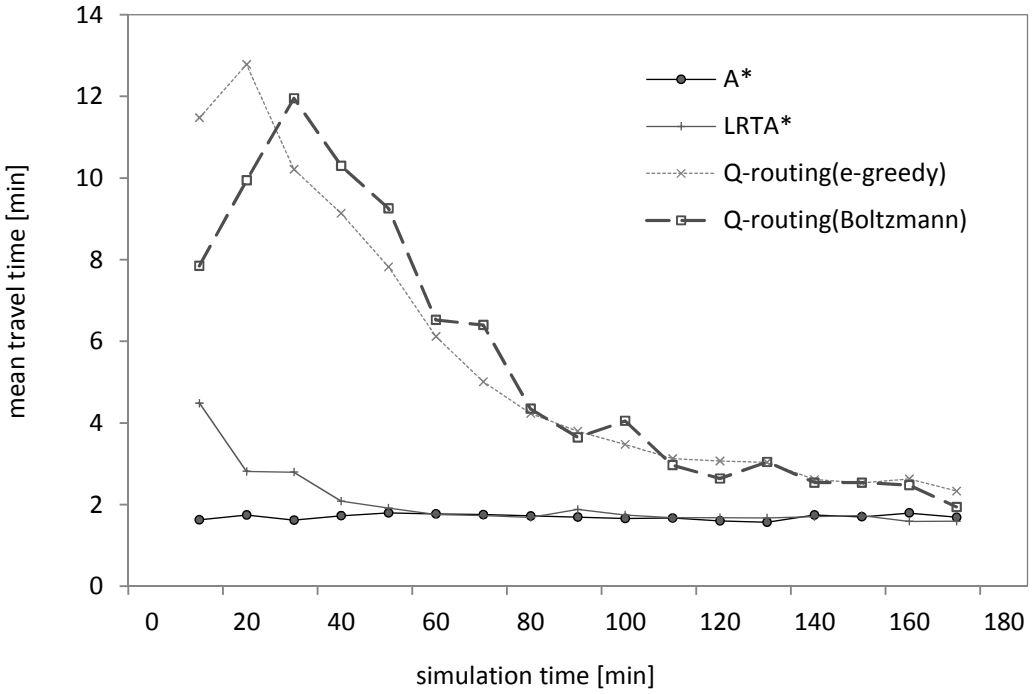

図 5.8: 発生交通量 30[台/h]，信号制御無しのケースの平均旅行時間

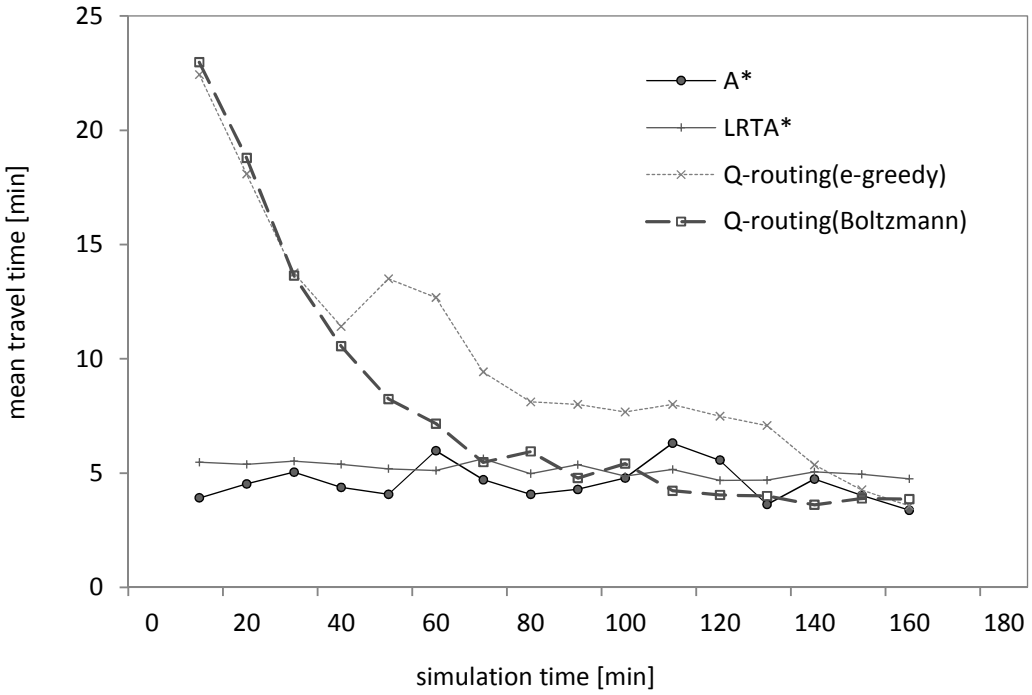

図 5.9: 発生交通量 300[台/h]，信号制御無しのケースの平均旅行時間

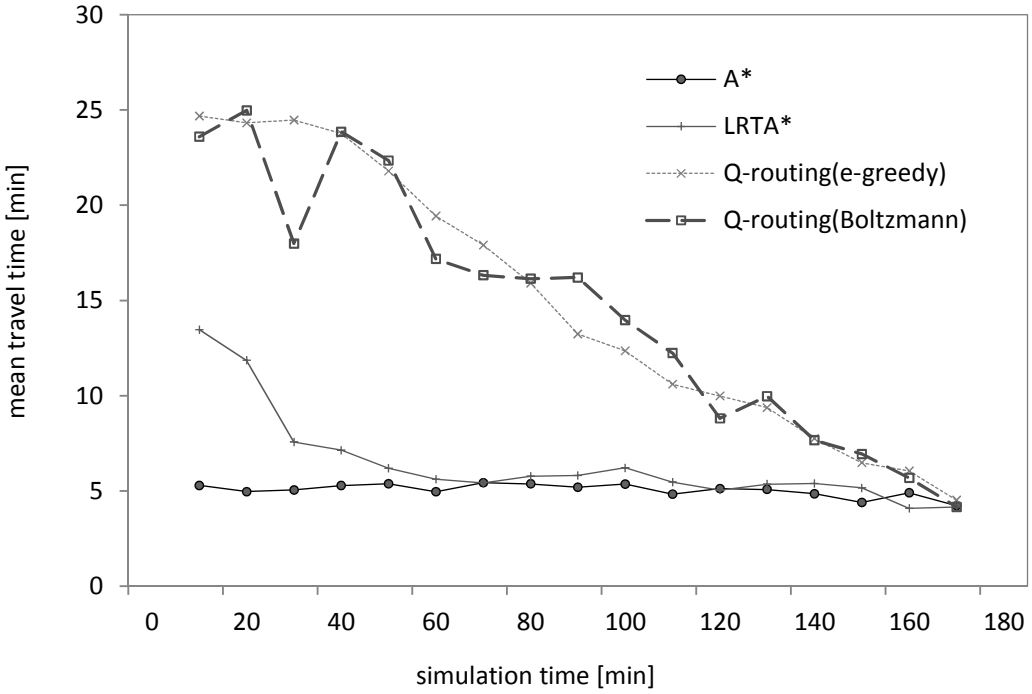

図 5.10: 発生交通量 30[台/h]，信号制御有りのケースの平均旅行時間

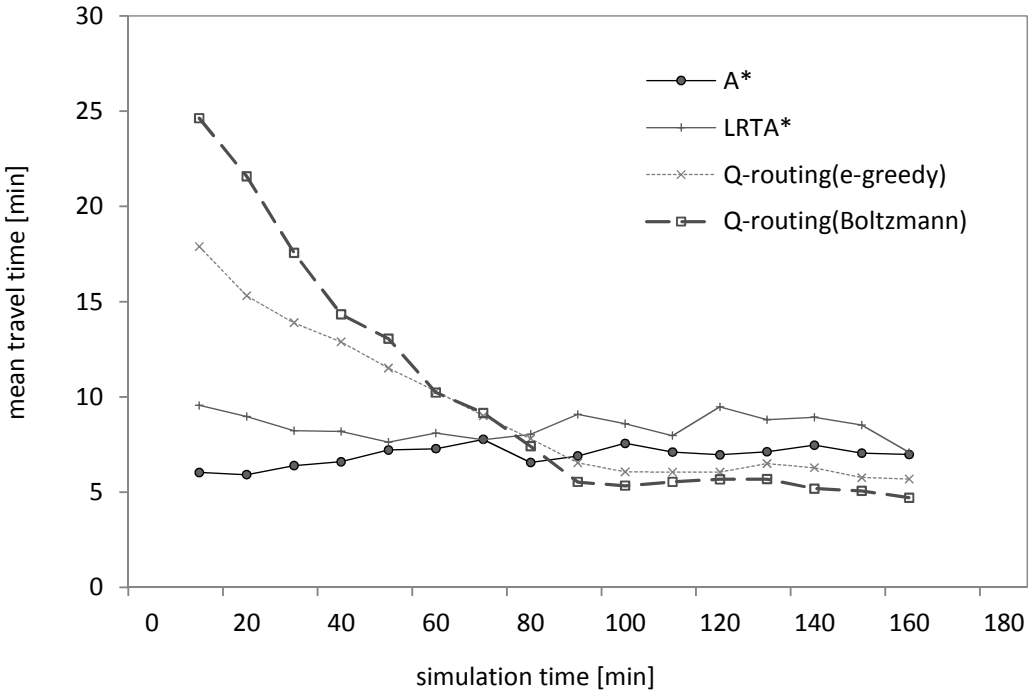

図 5.11: 発生交通量 300[台/h]，信号制御有りのケースの平均旅行時間

はじめに発生交通量の違いによる影響を考える．図 5.8 と図 5.9，図 5.10 と図 5.11 をそれぞれ比較した場合，Q-routing の平均旅行時間の収束値は A*と同程度であるが，図 5.11 においては A*より小さくなることがわかる．A*はエージェントが発生した時点での最短時間経路（以降，本条件設定では単に最短経路と呼ぶ）を選択することが数理的に保証されているにも関わらず，Q-routing の学習結果がこれを上回るのは興味深い現象である．

この現象には 2 つの側面があると考えられる．1 つは渋滞による旅行時間の増加である．エージェントが A*によって選択する最短経路はあくまで出発時点でのものであり，オフラインの探索であるため走行中に発生した渋滞に応じて経路を変えることはない．そのため，発生交通量が少なく平均旅行時間が小さい条件では出発時点での最短経路をそのまま走行しても問題ないが，平均旅行時間が大きくネットワークに留まる時間が長期化するほど環境が変化する可能性が大きくなってしまう．一方，Q-routing はオンラインに探索を行うため，走行中に発生した渋滞に応じて適応的に再探索を行うことが可能であある．

また，2 つ目はエージェントの経路選択の多様性である．条件設定 1 の静的な学習でも述べたが，エージェントが Boltzmann 選択により学習を行うと，経路選択の多様性が保たれるという特徴があった．本実験の実験環境である図 5.1 の道路ネットワークは，左右のネットワークをつなぐ中継リンクが 10-11 と 26-27 の 2 本に制限されているため，発生交通量が増加するとこの 2 本に交通流を分散させる必要がある．しかし，Boltzmann 選択以外の方法では，同じ目的地を持つエージェントは否応なくどちらか片方のリンクに集中せざるを得ない．発生交通量 300[台/h]，信号制御有りのケースのエージェントの経路を以下の図 5.12 と図 5.1.6 に示す．LRTA*では経路が中央のリンク 26-27 に集中しているが，Boltzmann 選択ではリンク 10-11 にも分散している．

このような理由から，交通量の多い場合には Q-routing，とりわけ Boltzmann 選択の学習結果が A*よりも良くなる場合があると考えられる．

続いて，信号制御による制御の影響について考える．本研究では Q-routing に対し，信号制御による滞留発生に適応するため価値関数の予測更新を導入している．このため，信号制御有りの場合でも図 5.10，5.11 に示したとおり学習が収束することがわかる．信号のタイミングによって経路をローカルに変更することに相当する

挙動であるが，一定の効果があることがわかる．特に図 5.11 では Q-routing の学習が収束して以降，旅行時間が一貫して A*や LRTA*よりも安定して小さくなっている．交通量が多い場合にはより滞留が発生しやすくなるため，それに対応した経路選択を行う手法が優位である．これらのことから，Q-routing が信号制御に対しロバストに適応することが分かる．

5.1.6 まとめ

本実験では，改良した Q-routing の基本的な性質を確かめ, 経路選択モデルとしての精度検証を行った．その結果，充分な時間経過の後収束する定常状態は，数理的に最適性が保証されている A*と同等程度の旅行時間となった．また，静的な学習では Boltzmann 選択を行なった場合に経路の多様性を確保できることを示した．動的な学習では交通量の増加や信号制御に対する適応などのケースを取り上げたが，その全てで学習が収束し，現実の交通状況に近い条件に対してもロバストであることを示した．このことから Q-routing は，本来の目的である交通施策の仮想社会実験を実施する際に，過渡期の表現を可能にするだけでなく，定常状態についても従来モデルと同等の妥当性を主張することができる．

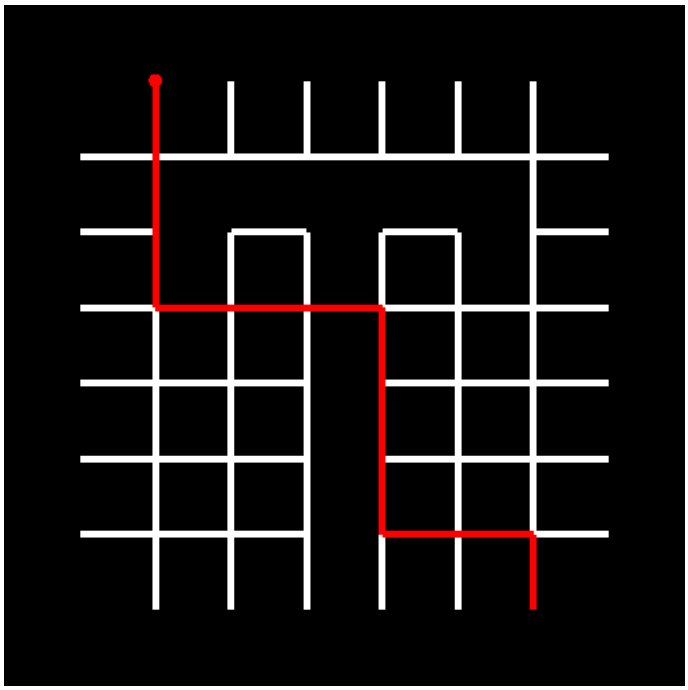

図 5.12: LRTA*におけるエージェントの経路（180[min]，ノード 1 からノード 60 方向）

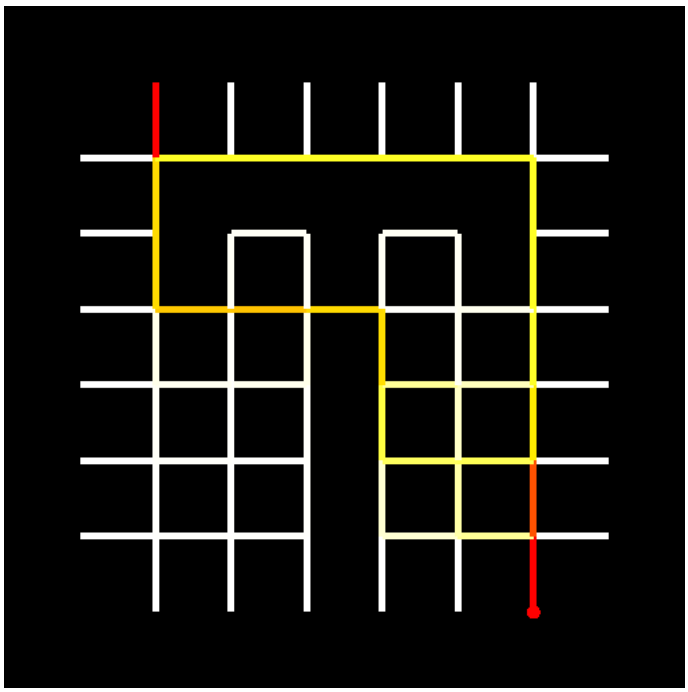

図 5.13: Boltzmann 選択におけるエージェントの経路（180[min]，ノード 1 からノード 60 方向）

5.2　実験2：路面電車の軌道延伸シミュレーション

5.2.1　岡山市における路面電車軌道延伸計画

2010年4月，岡山商工会議所は市中心部の路面電車の軌道延伸・環状化を核に地域活性化を図る「人と緑の都心1kmスクエア構想」の推進・実現への協力を岡山市に要望した．また同5月には岡山電気軌道を傘下に持つ両備グループが同様の構想をまとめ，さらに，路面電車と都市の未来を考える会 (RACDA) は後楽園など観光施設へのアクセス向上を含めた複数の環状化案を提起している [39]．これらの概略を図5.14に示す．路面電車延伸は岡山市において以前から繰り返し提案されてきたものの，様々な問題から計画が難航していたものである．しかし，近年の富山ライトレールの成功などから路面電車の有効性が再認識されるようになった．一方で，このような大規模な交通施策については事前に行われるべき社会実験についてもコストが高いと予想されるため，計算機による仮想社会実験の需要が存在すると考えられる．

5.2.2　延伸による影響

路面電車延伸の影響としては (1) 路面電車の軌道敷設による主要道の車線数減少，(2) 路面電車への乗換による交通量の減少，の2点を想定した．(1) については，延伸後のシミュレーションで延伸経路の車線数を片側1車線ずつ減少させることを考える．また (2) については，近年路面電車が導入された富山市の富山ライトレールを事例として，自動車から路面電車に利用を乗換えると予想される交通量を次のように算出した．

はじめに，富山市が行ったLRT化の整備効果調査 [40] より，富山ライトレール以前に運営されていたJR富山港線の平日の利用者が2331人，開業1年間で自動車から乗換えた利用者が572人であった．岡山市の路面電車は平日1時間当り約600人の利用者が存在するため自動車からの乗換えは147人とすることができる．道路交通センサスによれば平日に走行する乗用車の平均乗員数は1.33[人/台] であることから，路面電車延伸による交通量の減少は1時間当り111台となる．

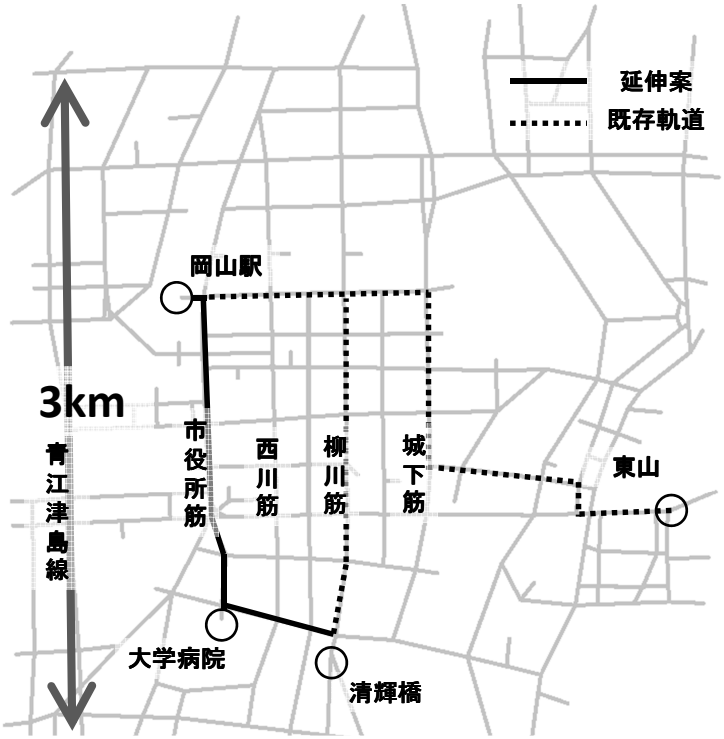

図 5.14: 路面電車延伸案 (岡山市中心部)

5.2.3 条件設定 1：軌道延伸前のシミュレーション

路面電車軌道延伸計画を扱うため，図 5.14 に示した岡山市の中心部の道路ネットワークを実験環境として使用する．領域は約 3km 四方であり，262 ノード (交差点)・381 リンク (単路) である．主要な信号 4 箇所については実測値を使用した．また，発生交通量は 2001 年に行われた社会実験のデータを基に設定し，平日の平均的な 2 時間を再現するため時間当たりの発生交通量は一定とした．本実験では，実験 1 の結果を参考に Q-routing のなかでも Boltzmann 選択による行動選択を採用する．Q-routing のルーティングテーブル $Q_x(y,d)$ はユークリッド距離で初期化し，学習率 $\alpha = 0.3$，温度係数 $T = 1000$ である．

また，規模の大きなネットワークで実験を行うにあたり学習を行うエージェントを限定した．図 5.14 に図示した大学病院〜岡山駅や清輝橋〜岡山駅といった OD を持つエージェントは路面電車延伸の影響を直接的に受けるため，Q-routing による学習を行い，その他のエージェントは A*によって経路選択を行うこととした．

5.2.4 結果と考察

図 5.15 は図 5.14 に示した岡山駅から清輝橋へ向かうエージェントの平均旅行時間を示している．縦軸は 10 分毎の平均旅行時間 [min]，横軸はシミュレーション時間 [min] である．また，横軸には学習エージェントの発生数も併記した．本実験では共通の目的地を持つエージェント間では価値関数を共有する設定としていることから，走行経験の蓄積が高速化される．そのため，学習が収束に至るまでのシミュレーション時間を計測するだけでは，過渡的な交通現象の継続時間が実際にどの程度のタイムスケールに相当するかを示すことができない．そこで，経路の学習を行うエージェントの発生数を観察することで現実の運転者の行動と関連付けることとした．

例えば，図 5.15 では学習の収束がシミュレーション時間 120[min] 程度であるが，この時間を現実の現象に読み替えることはできない．しかし学習を行うエージェントの発生数が 600[台] ということは，600 回の走行経験の後に交通流が定常状態に安定したということになる．シミュレーションの上では 600[台] のエージェントが 1 回づつ走行しているが，これを 1[台] の現実の運転者が 600 回の走行経験を積むことと解釈すれば，通勤などで 1 日 1 回の走行を想定すると約 600 日で定常状態が現れると考えることができる．

図 5.15 を見ると，シミュレーション開始直後は学習がほとんど進んでいない状態のため旅行時間が長くなっているが，100 分経過後にはほぼ収束している様子が確認できる．ネットワークにエージェントが飽和してから交通流が定常状態に遷移するまでに，約 400 台のエージェントが学習を行なっている．

また，このシミュレーションにおける定常状態がどれだけ岡山市の現状をどの程度再現しているかについて検証を行なった．以下に，2001 年に行われた交通社会実験の際に行われた交通量調査の結果と本実験の結果を表 5.1 に示す．2001 年以降，岡山駅の東西を結ぶバイパスが建設されるなど道路ネットワークに変更が生じているため，2011 年現在のネットワークで行なった本実験の結果と必ずしも一致するとは限らないが，交通量は概ね同様の傾向を示している．

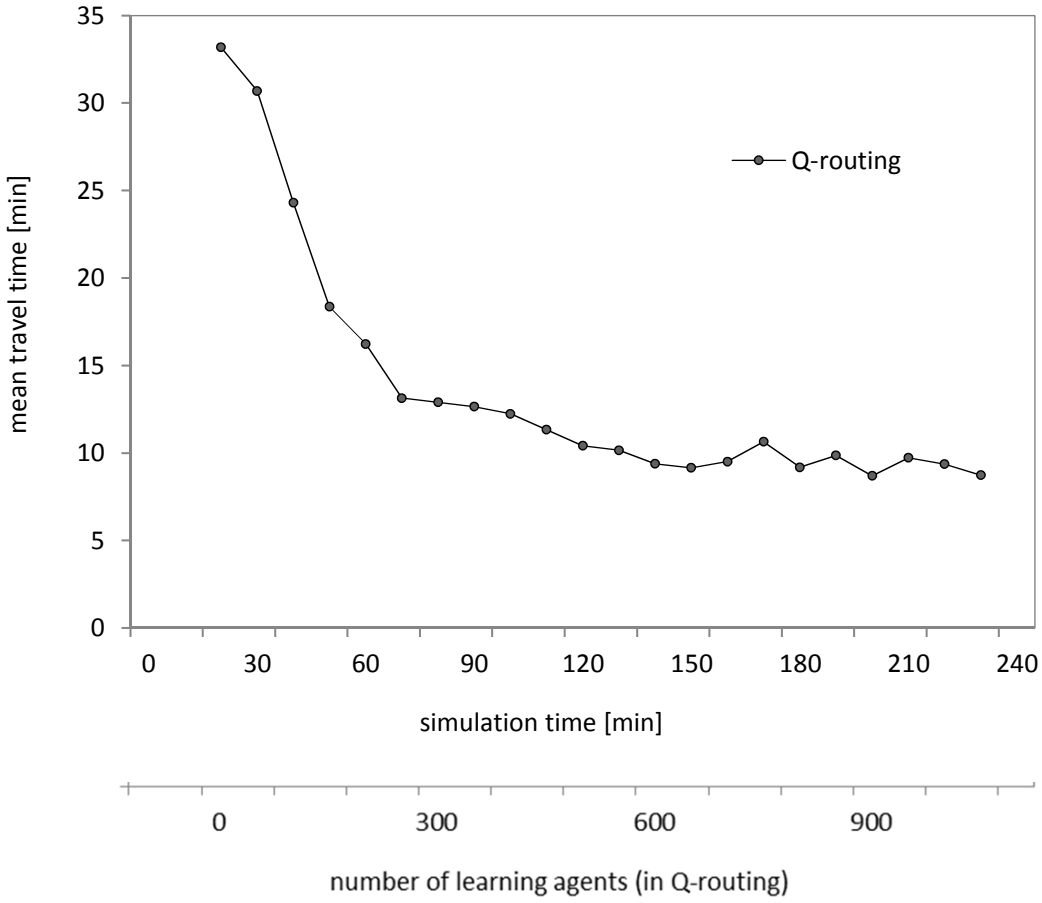

図 5.15: 路面電車延伸前の平均旅行時間

表 5.1: 交通量調査とシミュレーション結果の比較

	2001 年 交通量調査 [台/h]	シミュレーション [台/h]
青江津島線	2122	1935
市役所筋	2583	2533
西川筋	946	1050
柳川筋	2220	1792
城下筋	1192	1184

5.2.5　条件設定 2：軌道延伸後のシミュレーション

次に延伸後の道路ネットワークでシミュレーションを行った．道路ネットワークの形状は変化せず，軌道延伸が予定されている経路の車線数を 1 車線削減した．発生交通量は延伸前のシミュレーションと同様であるが，5.2.2 項より延伸による自動車からの乗り換え利用者を仮定し，延伸経路上の交通量を 1 時間あたり 100 台減少させた．

また，エージェントの過去の走行経験を反映させるため，条件設定 1 の延伸前のシミュレーションにおいて得られた行動価値関数 $Q_x(y,d)$ を読み込んだエージェントを作成した．これにより，延伸前の道路ネットワークに関する走行経験を持ったエージェントが，延伸後の道路ネットワークで走行するという状況を再現することができる．実験ではこの行動価値関数を読み込む場合（learned）と読み込まない場合（reset），そして各要素の値を半分に割り引いて読み込んだ場合（defective）でシミュレーションすることにより，経験を継承することの効果を観察する．

その他は条件設定 1 と同様である．

5.2.6　結果と考察

同様に岡山駅から清輝橋へ向かうエージェントの平均旅行時間を図 5.16 に示す．縦軸は 10 分毎の平均旅行時間 [min]，横軸はシミュレーション時間 [min] 及び学習エージェントの発生数とした．

結果を比較すると，learned，defective，reset の順に旅行時間が小さい．しかし，learned においても旅行時間の増大が見られ，学習終了時点と比較し最大 30%程度となっている．これには延伸による道路ネットワークの変化が影響していると考えられる．結局，延伸前の走行経験はあくまで別のネットワークにおけるものであり，延伸後はバイアスにも成り得ることを示している．実際，延伸によって図**??**の岡山駅〜大学病院などで混雑が発生し，それにあわせてエージェントが再度，学習を行なったことが原因である．

ただし，reset と learned を比較すると後者の交通流は安定しており，走行経験の継承によって軌道延伸前後のシミュレーション結果を連続的に捉えることが可能になった点は重要である．これまでの，A*をはじめとしたネットワーク全体の情報に

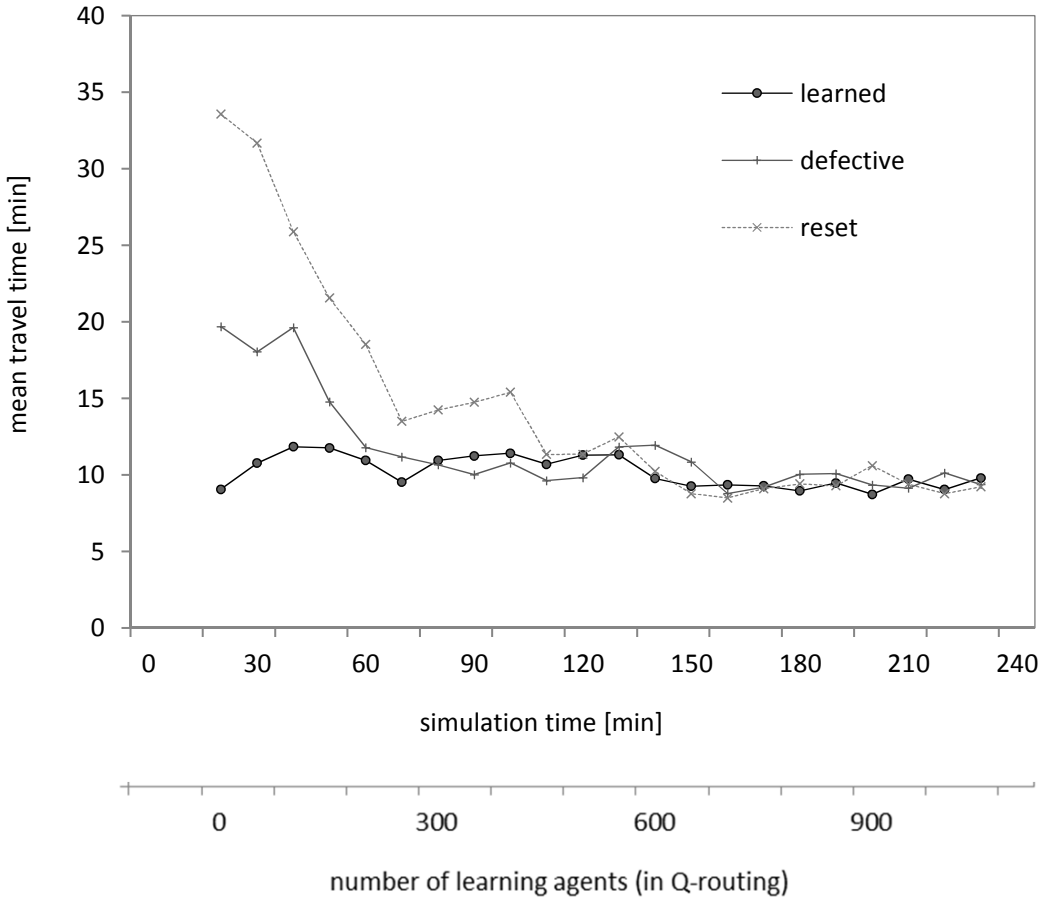

図 5.16: 路面電車延伸後の平均旅行時間

アクセスすることを前提としたアルゴリズムでは，軌道延伸前後それぞれの定常状態を評価することしかできなかった．また，Q-routing を採用した場合でも，reset のように走行経験の継承を行わなければ学習初期の環境同定に大きなコストが必要となり，意味のあるシミュレーションを実行することができないためである．

そこで，各時間帯ごとに learned のケースで発生している現象を以下にまとめる．

非飽和期 (0〜20 分)

ネットワークが大規模であるため，シミュレーション開始直後はエージェントが道路に行き渡っていない状況が継続する．このためを図 5.16 中には結果をプロットしていない．また，この間エージェントは各自自由走行している状態である．

再学習期 (20〜150 分)

エージェントが飽和し，路面電車延伸により狭くなった大学病院〜岡山駅の経路が混雑するようになると，延伸前のネットワークで学習済み価値関数から推定される旅行時間と，実際の旅行時間が次第に乖離し始める．それに合わせて新たな迂回経路の学習が始まり，試行錯誤の過程で更に渋滞の程度が増す．この時の，岡山駅から清輝橋へ向かうエージェントの経路の遷移を，シミュレーション時間 20[min]〜160[min] までの 20[min] ごとに図 5.17，5.18 に示す．

ただし再学習は局所的な経路にとどまり，延伸経路との相互作用の弱い OD についてはこれまでの価値関数を流用することが可能であるため，learned の過渡状態における渋滞現象は reset ほど深刻なもには進展しない．

収束期 (150 分〜)

迂回経路の再学習が収束し，定常状態に遷移する．旅行時間は 3 つのケースで同程度である．また収束のタイミングも同じである．

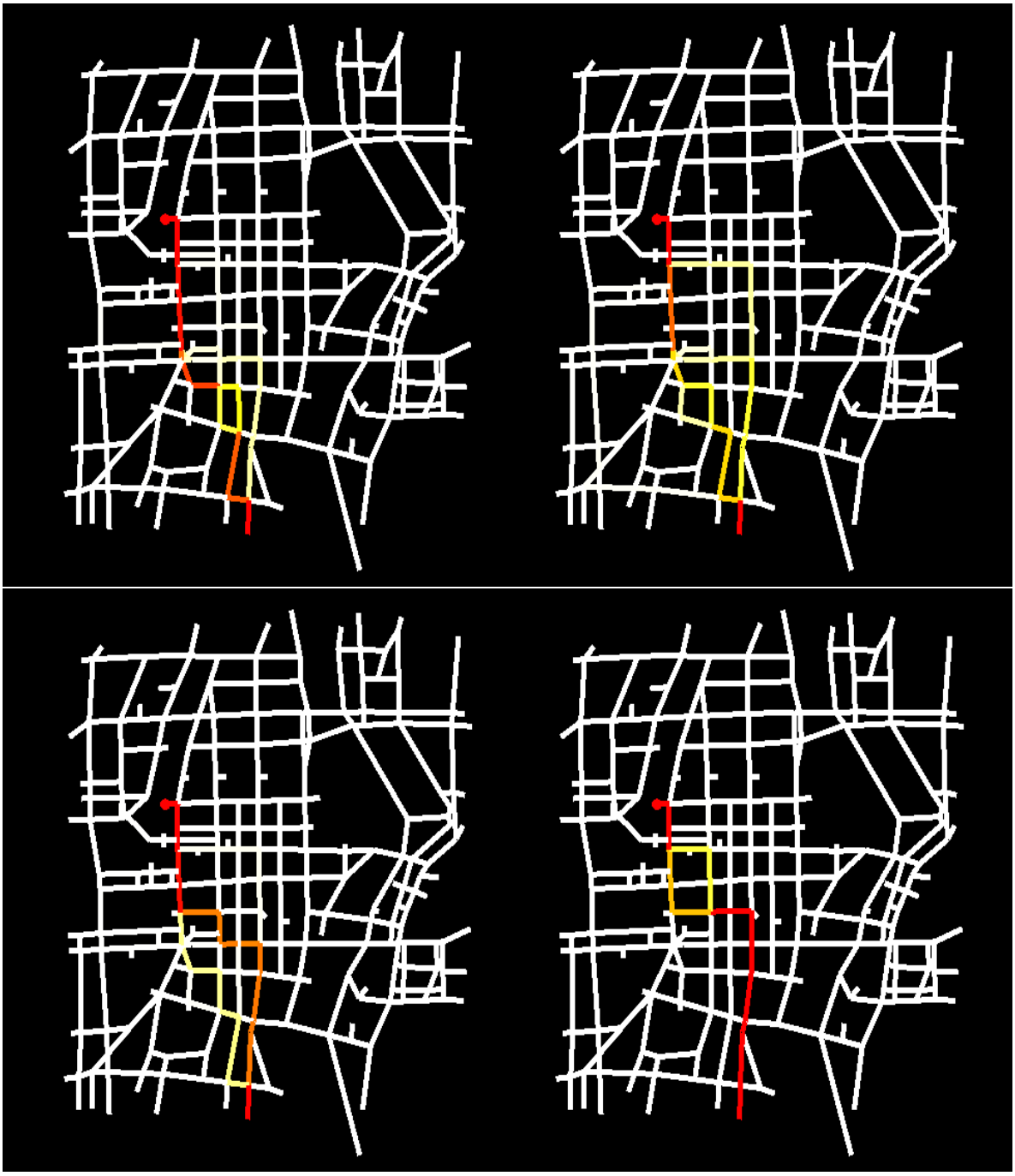

図 5.17: 岡山駅から清輝橋へ向かうエージェントの経路の遷移 (20[min]〜80[min])(左上，左下，右上…の順)

図　5.18:　岡山駅から清輝橋へ向かうエージェントの経路の遷移 (100[min]〜160[min])(左上，左下，右上…の順)

再学習期にあたる過渡的な渋滞は，岡山駅から清輝橋へ向かうエージェントの場合シミュレーション上で約 120 分間継続したが，この期間は実際のタイムスケール

を定量的に示すものではない．ただし，継続期間を学習エージェントの発生台数に読み替えると約 600 台となることがわかる．これは出発地点から到着地点までの移動，つまり Q-routing の価値関数を 600 回更新したことに相当する．また，延伸前後のシミュレーションを比較すると，収束した平均旅行時間は延伸前の Q-routing と延伸後の learned がそれぞれ 9.0 分と 9.2 分であった．この数値は延伸の影響を最も大きく受ける OD の一つで計測したものであり，今回の計画が長期的には交通渋滞を引き起こしにくいものであることを示している．

同様にして，以下に延伸前の旅行時間（定常状態），延伸後の旅行時間（定常状態），延伸後の旅行時間（過渡状態）を代表的な OD においてまとめた（表 5.2）．また各 OD において，延伸による影響（延伸前後の定常状態における旅行時間の差），延伸による過渡的な影響（延伸前の定常状態と，延伸後の過渡状態の旅行時間の差），をまとめ，過渡状態の継続時間も併記する（表 5.3）．過渡状態の継続時間については，再学習時の学習曲線がフラットで旅行時間に明確な変化の見られないケースに関して”-”と表記した．

その結果，岡山駅から清輝橋へ向かうエージェントと同様，今回の計画が長期的に交通渋滞を引き起こすものでないことを示した（表 5.3，”延伸による影響”の列参照）．特にいくつかのエージェントでは旅行時間が減少する場合もあった．また，計画の実施により過渡的な渋滞が発生する可能性についても，殆どの地点で 1〜3[min] 程度であった（表 5.3，”延伸による過渡的な影響”の列参照）．

一方，岡山駅から大学病院へ向かうエージェントの旅行時間は倍程度に上昇しており，局所的には慢性的な渋滞が発生する可能性を示した．また，過渡的な渋滞に至っては瞬間的に従来の 4 倍以上の旅行時間を要する可能性があった．更に過渡状態自体が他の OD い比べて長期間継続する結果となった（表 5.3，”過渡状態の継続時間”の列参照）．このため，計画を実施する際にはこの区間の渋滞を考慮し，情報提供などによって交通量を抑制する必要があるかも知れない．ただし，この区間に路面電車が延伸されることを考えると，むしろ路面電車の利用を促進するものであると解釈できる．本研究の設定では自動車から路面電車に乗換える利用者数は所与であるため，更なる検討が必要である．

表 5.2: 各 OD の定常状態・過渡状態における旅行時間

OD		延伸前の旅行時間 [min]	延伸後の旅行時間 [min]	
出発地	目的地	定常状態	定常状態	過渡状態
岡山駅	清輝橋	9.0	9.2	11.1
清輝橋	岡山駅	8.9	8.1	10.4
岡山駅	東山	11.0	11.9	14.0
東山	岡山駅	9.6	9.2	11.8
岡山駅	大学病院	10.9	18.2	46.3
大学病院	岡山駅	5.0	4.5	7.3

表 5.3: 各 OD の定常状態・過渡状態における旅行時間

出発地	目的地	延伸による 影響 [min]	延伸による 過渡的な影響 [min]	過渡状態の 継続時間 [min]
岡山駅	清輝橋	+0.2	+2.1	120
清輝橋	岡山駅	-0.8	+1.5	-
岡山駅	東山	+0.9	+3.0	150
東山	岡山駅	-0.4	+2.2	-
岡山駅	大学病院	+7.3	+35.4	180
大学病院	岡山駅	-0.5	+2.3	90

5.2.7　まとめ

本実験では，提案した改良 Q-routing による経路選択を利用し，岡山市における路面電車軌道延伸計画のシミュレーションを行った．延伸前の道路ネットワークにおける学習結果を読み込むことにより，走行経験を継承したエージェントを作成し，延伸直後の過渡状態を考慮したシミュレーションを行った．結果として，走行経験を継承したエージェントの旅行時間は概して小さく，それ自体は多分に有用な知見を含むことが分かった．しかし一方で，前の走行経験がバイアスとなり，新たな環境でで再学習し適応するまでには時間を要することも示した．

これまで多くの交通流シミュレータでは交通施策の計画に対し，長期的視点に立った定常状態の予測のみで意思決定を行っていた．しかし実際には本実験で示したような短期的で複雑な過渡状態が存在する可能性があるため，定常状態と過渡状態の 2 つの交通現象をどちらも考慮した上での実行判断が必要であると言える．上述した知見は，A*などネットワーク全体の情報にアクセスすることを前提とした手法では得られないものであり，Q-routing の有効性を示していると考えられる．

第6章　おわりに

第6章　おわりに

本研究ではより現実的な交通施策の評価を目標とし，運転者の過去の走行経験を経路選択に反映させるため，交通流シミュレーションにエージェントの学習機能を実装した．実装に際し，通信ネットワークと交通ネットワークの相違点に着目しながら改良を加えた．

続いて不規則格子での予備実験を行い，Q-routing による経路選択が過渡状態の交通状況を再現するだけでなく，定常状態の再現性についても A*と同等の性能を示すことを確認した．現実の交通施策として岡山市の路面電車軌道延伸計画を扱った実験では，複雑なネットワーク・信号制御といった環境下でもロバストに学習が収束することを示すと同時に，従来のシミュレータでは再現できなかった，エージェントのバイアスによる延伸後の過渡的な渋滞現象を確認することができた．また，この現象は比較的小規模で，時間経過の後渋滞による影響は観察されなくなることを示した．

今後の課題としては，過渡的な渋滞が収束し定常状態に至る過程で，経路選択がどのように遷移していったかを更に詳細に解析することが挙げられる．これにより，学習の収束を促進する有用な情報（価値関数）の抽出などを目指す．また，その情報を利用した情報提供により，渋滞の収束を早める・ピーク時の旅行時間を短縮することなども考えられる．

参考文献

[1] 大蔵泉:「交通工学」, コロナ社,1993.

[2] 交通工学編集委員会:「交通工学」,vol. 47, No. 1, 2012.

[3] J. Lighthill, G. B. WMtham:“ On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads ”, roceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, Volume 229, No. 1178, pp. 317-345, (1955)

[4] 山田剛:“ 拡張結合型 Burgers 方程式による多車線交通流モデルの提案 ”, Reports of RIAM Symposium，No. 19 ME-S2，(2008).

[5] 西成活祐:“ 交通流のセルオートマトンモデルについて ”, 応用数理, vol. 12, No. 2, pp. 26-37, (2002).

[6] 杉山雄規:“ 交通流の物理 ”, ながれ 22, pp. 95-108, (2003).

[7] Chandler, R. E., Herman R. and Montroll, E. W.: Traffic Dynamics: Studies in Car Following, Oper. Res., Vol.6, pp.165-184, (1958).

[8] S.Darabha and K.R.Rahagopal:”Intelligent cruise control systems and traffic flow stability ”, Transportation Research Part C, No.7, pp.329-352, (1999).

[9] 玉城龍洋, 安江里佳, 北栄輔:“ セル・オートマトンによる自動車専用道路の交通シミュレーション ”情報処理学会論文誌. vol. 46, sig. 10, pp. 30-40, (2005).

[10] B. D. Greenshields: ”A Study in Highway Capacity ”, Highway Research Board Proceedings, Vol.14, (1995).

[11] H. Greenberg: ”An Analysis of Traffic Flow ”, Operations Research, Vol.7, No.1, (1959).

[12] D. R. Draw: ”Deterministic Aspects of Freeway Operations and Control ”, Texas Transportation Institute, Research Report, (1965).

[13] J. D. Bolland, M. D. Hall, D. Van Vilet:SATURN: A Model for the Evaluation of Traffic Management Schemes, Institute for Transport Studies Working Paper 106, Leeds University, (1979).

[14] M. D. Hall, D. Van Villet, L. G. Willumsen:SATURN - A Simulation- Assignment Model for the Evaluation of Traffic Management Schemes, Traffic Engineering & Control, Vol. 21, pp.168-176, (1980).

[15] 吉井稔雄, 桑原雅夫, 森田綽之:都市内高速道路における過飽和ネットワークシミュレーションモデルの開発, 交通工学, Vol.30, No.1, pp.33-41, (1995).

[16] C. C. Liu: Integrated Network Modeling with TRAF, Preprint at the Second Multinational Urban Traffic Conference, (1991).

[17] A. K. Rathi, A. J. Santiago: Urban Network Simulation: TRAF-NETSIM Program, Transportation Engineering, Vol.116, No.6, pp.734-743, (1992).

[18] M. Van Aerde, S. Yagar: Dynamic Integrated Freeway / Traffic Signal Networks: A Routing-Based Modeling Approach, Transportation Research A, Vol.22A, No.6, pp.445-453, (1988).

[19] 堀口良太, 片倉正彦, 赤羽弘和, 桑原雅夫:都市街路網の交通流シミュレータ -AVENUE-の開発, 第 13 回交通工学研究発表会論文集, pp.33-36, (1993).

[20] Lovejoy, W. S.: A Survey of Algorithmic Methods for Partially Observed Markov Decision Processes, Annals of Operations Research 28, pp.47-65 (1991).

[21] YOSHIMURA S: MATES : Multi-Agent Based Traffic and Environment Simulator-Theory, Implementation and Practical Application, Computer Modeling in Engineering and Sciences 11(1), 17-25, 2006.

[22] H. Fujii, T. Sakurai, S. Yoshimura:Virtual Social Experiment of Tram Railway Extension Using Multi-Agent-Based Traffic Simulator, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.15, No.2, 226-232, 2011.

[23] 吉村 忍 , 西川 紘史 , 守安 智:知的マルチエージェント交通流シミュレータ MATES の開発, シミュレーション 23(3), 228-237, 2004-09-15.

[24] 藤井 秀樹 , 仲間 豊 , 吉村 忍:知的マルチエージェント交通流シミュレータ MATES の開発 : 第二報:歩行者エージェントの実装と歩車相互作用の理論・実測値との比較, シミュレーション 25(4), 274-280, 2006-12-15.

[25] Hart, P. E.; Nilsson, N. J.; Raphael, B.:"A Formal Basis for the Heuristic Determination of Minimum Cost Paths", IEEE Transactions on Systems Science and Cybernetics SSC4(2), 100-107, 1968.

[26] J. Boyan , M. L. Littman:Packetrouting in dynamically changing networks: a reinforcement learning approach, Advances in Neural Information Processing Systems, volume 7, 671-678, 1994.

[27] Choi, S., Yeung, D.-Y.:Predictive Q-routing: A memory-based reinforcement learning approach to adaptive trac control, Advances in Neural Information Processing Systems 8 (NIPS8), 945-951, 1996.

[28] Said, H., Abdelhamid, M. and Yacine, A.: K-Shortest Paths Q-Routing: A New QoS Routing Algorithm in Telecommunication Networks, *Lecture Notes in Computer Science*, Vol.3421, pp.164-172 (2005).

[29] Balmer, M., Meister, K., Rieser, M., Nagel, K. and Axhausen, K. W.: Agent-based simulation of travel demand: Structure and computational performance of MATSim-T, the 2nd TRB Conference on Innovations in Travel Modeling, 2008.

[30] 立本 真治 , 本多 中二:微視的道路交通シミュレータ MITRAM への経路選択モデルの導入と検証, 知能と情報 18(4), 586-597, 2006-08-15.

[31] S. Koenig and M. Likhachev. Adaptive A* [poster abstract]. In Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems, pages pp.1311-1312, (2005).

[32] Sun, X., Koenig, S., Yeoh, W.: Generalized adaptive a*. In: Int. Foundation for Autonomous Agents and Multiagent Systems AAMAS 2008, pp.469-476, (2008).

[33] Koenig, S., Likhachev, M.: Improved fast replanning for robot navigation in unknown terrain. In: Proc. of the Int. Conf. on Robotics and Automation, pp.968-975, (2002).

[34] Korf, R.E.: Depth-first iterative-deepening: An optimal admissible tree search. Artif. Intell. 27, pp.97-109 (1985).

[35] Hansen, E.A., Zhou, R.: Anytime heuristic search. J. Artif. Intell. Res (JAIR) 28, pp.267-297, (2007).

[36] Evett, M., Hendler, J., Mahanti, A., Nau, D.: Pra*: massively parallel heuristic search. Technical report, (1991).

[37] Korf, R.E.: Real-time heuristic search. Artif. Intell. 42(2-3), pp.189-211, (1990).

[38] Koenig, S., Likhachev, M.: Real-time adaptive a*. In: AAMAS 2006, pp.281-288. ACM Press, New York (2006).

[39] RACDA かわら版, vol 74, 新・路面電車環状化　岡山商工会議所新提案の意義, 2010-05-02.

[40] 富山市、国土交通省: 富山港線 LRT 化の整備効果調査, 2006.

謝辞

本論文は多くの方々からご指導，ご協力をいただいた結果として完成させることができたと思っています．

指導教員である吉村先生には，研究を進める上での姿勢や心構えについて大変多くのご指導を賜りました．また，研究テーマの選定にあたっても多くの示唆を頂きました．准教授の和泉先生には，ゼミでの議論など様々な場面でお世話になりました．人工知能学会や JAWS でも貴重な経験ができました．助教の藤井さんには MATES に関する全てを教わりました．藤井さんがいなければ，私のコードが動く日はついに来ることがなかったでしょう．遅くまで原稿の修正に付き合っていただき，ご迷惑をおかけしました．ありがとうございます．千葉大学の荒井先生には卒論に引き続きご指導いただきました．3 年間に渡り見守っていただき，大変感謝しています．

また，岡様をはじめとする RACDA の皆様と岡山電気軌道株式会社の皆様には，たくさんの有益なご意見を頂きました．本研究を進めるにあたり，大変参考になりました．ありがとうございました．

秘書の井上さんには，学会の事務手続きなどすべてお願いしていました．ご迷惑をおかけしたこともありましたが，大変お世話になりました．研究員の河合さんには毎晩のように夕食に連れて行ってもらいました．本郷キャンパス周辺の食事処と，研究室の空気を教わりました．杉本さんには普段の研究室やゼミの場でいろいろなことを教えていただきました．進学当初は PC の筐体を開いたことのなかった私も，なんとか 2 年間やっていくことができました．技術職員の川手さんの出勤時間が，研究室に泊まった日のペースメーカーになっていました．入口付近で寝ていて申し訳ありませんでした．室谷さんにはゼミでたくさんのご意見をいただきました．私の理解不足で答えられない場面も多くありましたが，非常に勉強になりました．淀さんとは普段の研究室やゼミの場でいろいろなことを教えていただきました．また，計算力学技術者資格試験の際にはお世話になりました．博士課程の片岡さんも良く夕

食をご一緒しました．スタッフの方々とはまた違ったアドバイスをいただきました．

南さんには，お忙しい中ゼミでの発表を引き受けていただいたりとご迷惑をおかけしました．卒業された犬塚さんには，逆にゼミの日程調整で我侭を聞いていただきました．鈴村さん，白根さんには飲み会やゼミ合宿のことなど，年の近い先輩ならでわの内容についてアドバイスをいただきました．お陰様で楽しい研究室ライフが過ごせました．

同期の遊佐君には本当に助けられました．元来おおざっぱな性格の私がなんとかやってこれたのは遊佐君の御陰です．一ヶ月間電中研に通ったのは，今や良い思い出です．博士課程でも頑張ってください．後輩の宮崎君には自宅を鍋企画のために提供してもらいました．食べるだけ食べてごめんなさい．皆川君には直前に修論の添削をしてもらいました．柴田君とはJAWSの会場で完全にアウェーな雰囲気の中，一緒に夕食を食べたことを覚えています．迫村君とは席も近く，雑談に付き合ってもらいました．王君のゼミ発表はとても勉強になりました．研究頑張ってください．私も英語力をどうにかしようと思います．友部君とは卒論の頃からよくラーメンを食べに行きました．また食べに行きましょう．MATES頑張ってください．

卒論生の三目君，渡辺君，蔵本君，池田君とは半年ながら楽しい時間が過ごせました．非常に個性的な面々で，話していて飽きなかったです．卒業してからもそれぞれの道を邁進してください．

最後に，修士課程での生活を支えてくれた家族に感謝します．