

学習アルゴリズムを導入した 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 Machine Learning

内田 英明*¹ 藤井 秀樹*¹ 吉村 忍*¹
Hideaki UCHIDA Hideki FUJII Shinobu YOSHIMURA

*¹東京大学大学院工学系研究科システム創成学専攻

Department of Systems Innovations, School of Engineering, the University of Tokyo

This paper proposes a routing algorithm based on reinforcement learning framework for the the Multi-Agent based Traffic and Environment Simulator MATES. In MATES, all cars search the shortest route. However, we consider routing algorithm based on learning focused on the agent's driving experiences. Firstly, the paper describes the Q-routing, a packet routing algorithm, and how to apply this algorithm to traffic simulation. We, then, implement new routing algorithm to improve the Q-routing. In this paper, we also report some experiments using a irregular grid network with various loads or signal control, and good performance result: Q-routing is robust to the heavy traffic and signal control.

1. はじめに

近年の道路交通には多くの問題が指摘されており、様々な交通施策が提案されている。交通現象は典型的な社会系の複雑現象であり、人間生活とも密接に関連していることから、繰り返し実証実験を行うことは時間・コストの両面から非現実的である。このような複雑システムの挙動を予測するため、実験の代替としてシミュレーションの利用に対する需要が高まっている。これは計算機の技術的進歩により、今まで以上に複雑・大規模なシミュレーションが可能となったためである。

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である。多くのシミュレータにおいて、運転者はネットワークについての膨大な情報にアクセスすることができるため容易に最短ルートを探索することができるが、現実の運転者は、周囲の状況と道路状況に対するこれまでの知識や経験に基づいてルートを選択すると考えられる。そこで本稿では、マイクロ交通流シミュレータ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 [Boyan 94] を導入し、過去の経験を基にローカルな情報のやりとりのみで実現する経路選択を提案する。また、既存の手法との比較を行いその性質を考察する。

2. 交通流シミュレータ

2.1 MATES

本研究では、著者らが開発している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) [Yoshimura 06][Fujii 11][吉村 04][藤井 06] に Q-routing を実装する。

MATES では自動車などの交通主体をエージェントとし、その活動場所である道路空間を環境と定義する (図 1)。このマルチエージェント型のフレームワークにのっとることで、前方

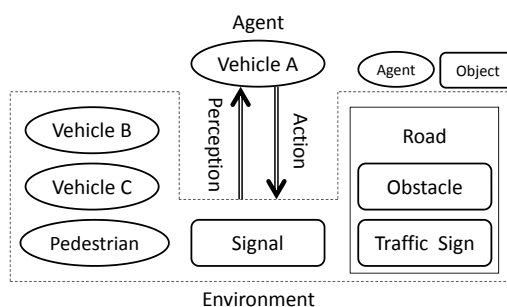


図 1: MATES のマルチエージェントモデル

車両との車間距離を計測し追従するなどのマイクロなルールから、相互作用の結果としてのマクロな交通現象を観察することができる。

MATES では現在 A*[Hart 68] によるルーティングを採用している。A*とはダイクストラ法のルート探索に推定値 (一般的にはユークリッド距離など) を与えて改良したものである。また、各ルートのコスト算出にパラメータを設定し、距離や旅行時間、車線数に応じた重みづけを行うことが可能である。しかしこの A*によるルーティングは、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る。そのため、ある地点での渋滞発生を瞬時に全エージェントが認識し迂回するなど、現実の運転者と乖離した挙動を引き起こす可能性がある。そこで本研究では、限定的な情報のみでルート探索を行う Q-routing を MATES に組み込んだ。

2.2 既存のシミュレータ

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) [Balmer 08] 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、その値が最大のプランを実行す

連絡先: 内田英明, 東京大学大学院工学系研究科システム創成学専攻, 〒113-8656 東京都文京区本郷 7-3-1 工学部 8 号館 422 号室, (03) 5841-6994, uchida@save.sys.t.u-tokyo.ac.jp

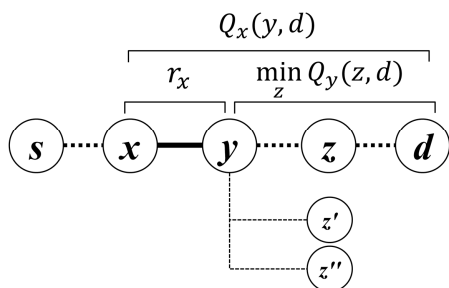


図 2: Q-routing の概略

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．[Balmer 08]

3. Q-routing

3.1 Q-routing の概要

Q-routing とは強化学習の一つである Q-learning の枠組みに基づいた自律分散型のルーティングアルゴリズムである [Boyan 94] 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 [Boyan 94][Choi 96][藪内 04]．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価値関数 $Q_x(y, d)$ を持ち，この Q テーブルを用いてルーティングを決定する．ここで $Q_x(y, d)$ 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．ただし，この送信時間の大部分はノードにおける待ち行列での待ち時間で占められる．

3.2 価値関数の更新

各ノード x において，パケットをホップする隣接ノード y は価値関数 $Q_x(y, d)$ が最小となるものを選択する．パケットは y にホップするとただちに $Q_x(y, d)$ の更新に必要な 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 (1) に従って与えられる．

$$q_y = \min_{z \in \text{neighbors of } y} Q_y(z, d) \quad (1)$$

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遅延時間の和を r_x としたとき， $Q_x(y, d)$ の更新式は式 (2) のようになる．またこの更新のモデルを図 2 に示した．

$$Q_x(y, d) = Q_x(y, d) + \alpha(r_x + q_y - Q_x(y, d)) \quad (2)$$

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可能性がある．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より正確な $Q_x(y, d)$ に更新される．

3.3 MATES への実装

本研究では上述した Q-routing の交通流への適用を提案する．そこで価値関数 $Q_x(y, d)$ 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，と定義する．パケットルーティングと交通流における最も大きな違いは，ノード（交差点）における待ち行列の待ち時間ではなくリンク（単路）における移動時間によって目的地までの旅行時間が規定される点である．これらのことから，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った．考慮すべき事項は以下の 2 点である．

- ルーティングにおける車線変更の制約

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．そのため従来の Q-routing のように直近の隣接ノードを選択するのではなく，更に 1 つ先の交差点に関して隣接交差点を取得し意思決定を行う必要がある．

- 滞留による価値関数への影響

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．そのため，正常なルーティングを妨げないように学習率を高く設定し混雑の状況を素早く反映する必要はある．

また，実装に当たっては新たに ϵ -greedy 戦略を導入した．これはエージェントが高い確率 $1 - \epsilon$ で価値関数を最小化するルートをとる低い確率 ϵ でその他のルートをランダムに選択するものである．エージェントにランダムな選択の余地が生まれることで選択可能なルートの価値関数を万遍なく更新するため，学習に要する時間を短縮することができる．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，新たな経路を再び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を図 3 に示した．

以下で行う実験では，MATES 上に存在するエージェント全てが Q-routing を行い価値関数を更新していく．本来この価値関数はエージェントごとに独立したものであるが，シミュレーション時間の短縮のため，同じ目的地を持つエージェント間で共有できる設定とした．これは，日々蓄積される走行経験を現実的なシミュレーション時間で再現できることを意図したものであり，価値関数を共有するエージェント群は，それぞれが異なった日に走行する同一のエージェントであると見做せる．また，定常的な交通流を再現するために発生交通量はシミュレーションの全時間に渡って一定とした．

4. 実験

4.1 実験設定

先行研究 [Boyan 94] を参考に irregular grid の仮想環境を用いて実験を行った．図 4 にネットワークのトポロジを示す．端点となる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

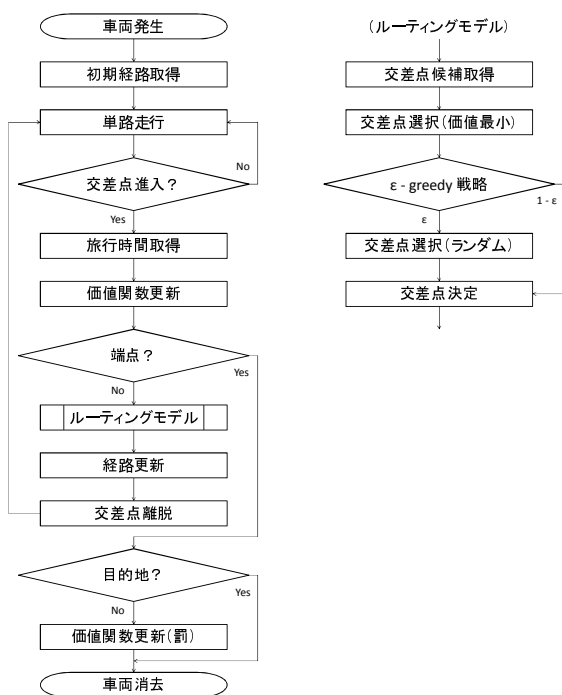


図 3: Q-routing のフローチャート

リンクが 10-11 と 26-27 の 2 本に制限されており、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。そのため、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 に迂回するルーティングを行う必要がある。

実験では $Q_x(y, d)$ の初期値を 0 とし、予備実験の結果から学習率 $\alpha = 0.5$, $\epsilon = 0.05$ とした。

4.2 発生交通量に対する比較

はじめに、各交差点における信号制御の無い状態で実験を行った。このとき、Q-routing によるルーティングを導入したモデルとの比較として以前から MATES で採用されている A*を用いた。A*は距離コストを最小化する A*-distance と旅行時間コストを最小化する A*-time を比較対象とした。発生交通量は各端点で 100, 500[台/h] とした。

図 5~7 に各ケースの旅行時間の推移を示す。横軸をシミュレーション時間 [min]、縦軸を 5 分毎の平均旅行時間 [sec] としてプロットしたものである。まず図 5 に示した交通量の少ない 100[台/h] のケースでは、A*-time・A*-distance とともに平均旅行時間が一定であり渋滞が発生していないことが分かる。一方 Q-routing はシミュレーション開始直後に平均旅行時間が増大し、学習が進展するにつれ開始後 90 分程度までには一定の収束が見られる。学習収束後の Q-routing が A*と比較し依然として旅行時間が大きいのは ϵ -greedy 方策を採用したためであり、 $\epsilon = 0.05$ の割合で $Q_x(y, d)$ が最小ではない交差点を選択するためであると考えられる。この実験において推定された $Q_x(y, d)$ を用い、新たに $\epsilon = 0$ とした実験を行った結果が図 6 のようになることからこの点を確認することができる。

また、交通量が多い 500[台/h] のケースを図 7 に示す。ここでは A*-time が最も平均旅行時間が短く、A*-distance は時間が進展するに連れ旅行時間が増大していくことがわかる。これは単に最短距離を選択するだけではこのネットワークの特徴である、中央部の中継リンク 26-27 への交通集中により渋滞

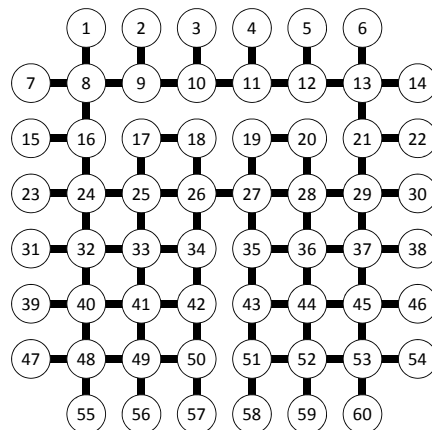


図 4: 実験環境

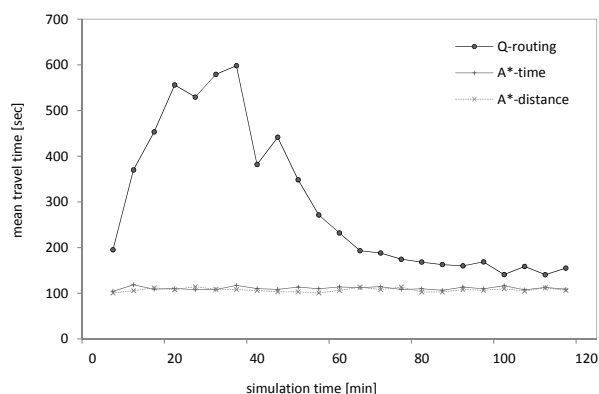


図 5: 発生交通量 100[台/h] のケースの旅行時間

が発生するためである。一方 Q-routing は 100[台/h] のケースに比べ収束に時間がかかるものの、A*-distance よりも短い値に収束し、交通量の増加に対してロバストであるといえる。

4.3 信号制御に対する比較

次に各交差点において信号制御を行ったケースの実験をおこなった。制御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。ここでも Q-routing の比較対象として A*-distance と A*-time を用い、発生交通量は各端点で 100[台/h] とした。

図 8 に各ケースの旅行時間の推移を示す。これまでと同様、横軸をシミュレーション時間、縦軸を 5 分毎の平均旅行時間としてプロットしたものである。A*-time はネットワーク全体の情報を活用できるため、信号機導入下でも渋滞が発生せず一定して良好な平均旅行時間を示した。また A*-distance はシミュレーションが進むに連れ旅行時間が増大した。これは発生交通量 100[台/h] で信号制御を行わないケース（図 5）で見られなかった現象であることから、信号待ちによるエージェントの滞留が進み渋滞が起きたことがわかる。A*-distance は信号の有無に関わらず最短距離の経路をとるため、このような現象が起こったと考えられる。

一方 Q-routing はシミュレーションが進展するに連れ旅行時間が減少し、開始 90 分頃からは A*-time と同等の良好な旅行時間が得られた。これは学習によって信号制御のサイクル自体を価値関数に取り込んだことによる。また、図と異なり収束

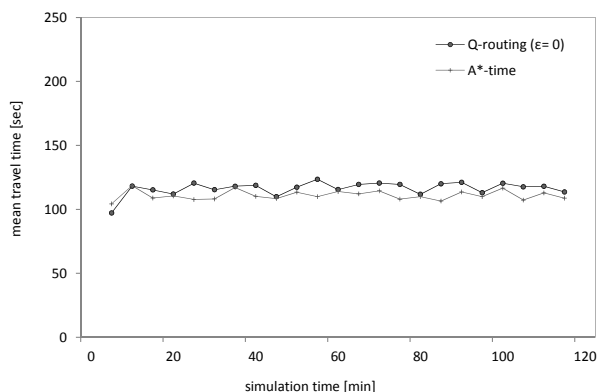


図 6: 発生交通量 100[台/h], $\epsilon = 0$ としたケースの旅行時間

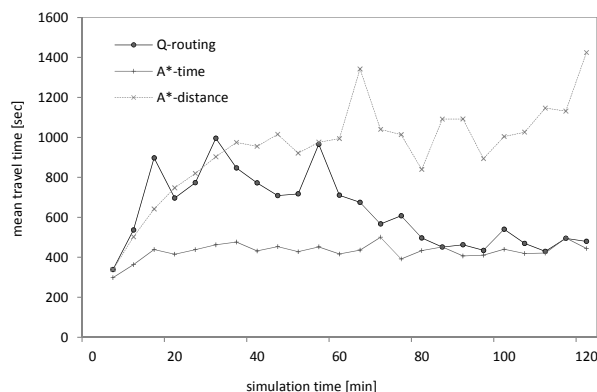


図 8: 信号機制御ありとしたケースの旅行時間

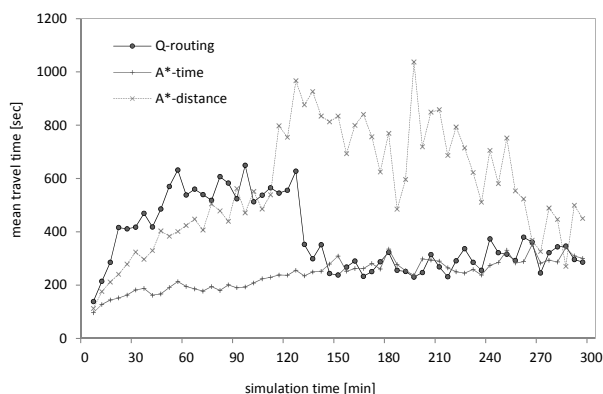


図 7: 発生交通量 500[台/h] のケースの旅行時間

後の平均旅行時間が Q-routing と A*-time で変わらないのは、Q-routing の ϵ の影響が信号待ちによって打ち消されたものと考えられる。このことから、信号機によって動的に変化する交通流に対しても Q-routing がロバストに適應することが示された。

5. 結論

本研究では Q 学習を応用した Q-routing を、もともと提案されたパケットルーティングではなく交通流の経路選択に応用した。いくつかの変更を加えることで、本来知られてきた混雑環境下でのロバスト性だけでなく、信号機による制御にも適合した。Q-routing は価値関数と直近の単路の旅行時間のみを利用するが、この 2 つはそれぞれ経路に対する長期的な経験と短期的な経験に相当する。これら限定された情報のみでルーティングを行うことから、交通流シミュレーションの経路選択としてより人間らしい意思決定規範となる可能性がある。今後の課題として、事故や災害によるネットワークの変化や交通施策によるシステムの変化に対して、その変化を運転者がどのように受容し知見を改めるかといったシミュレーションを行うことが挙げられる。

参考文献

[Yoshimura 06] 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.

[Fujii 11] 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.

[吉村 04] 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.

[藤井 06] 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.

[Hart 68] 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.

[Boyan 94] 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.

[Choi 96] 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.

[藪内 04] 藪内 佳孝, 加藤 昇平, 犬塚 信博: 強化学習に基づいたルーティングアルゴリズムの一手法, 2004 年度人工知能学会全国大会 (第 18 回) 論文集, 2004.

[Balmer 08] 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.