

大規模マルチエージェント交通流シミュレーションのための 階層的経路探索手法

福田 隼馬^{1,a)} 阿部 和規^{1,b)} 藤井 秀樹^{1,c)} 山田 知典^{1,d)} 吉村 忍^{1,e)}

受付日 2017年11月22日, 採録日 2018年4月4日

概要: マルチエージェントシステムを採用する交通流シミュレーションでは, 車両エージェントの個性を反映した挙動を再現できる. しかし大規模な道路ネットワークを対象としたシミュレーションを実施する場合, 経路探索に要する計算時間がボトルネックとなる. 本研究では階層化された道路ネットワークを用いた経路探索手法を提案する. このとき, 道路ネットワークの階層化に用いる属性と経路探索に用いる属性とを区別することで, 単一の階層道路ネットワークを用いてエージェントごとの多様な経路探索結果を出力できるようにした. 東京都心部の道路ネットワークを対象とした実験により, 提案手法を用いることで経路探索のクエリ処理時間を従来手法の1.5~2%まで短縮できることを示した.

キーワード: 経路探索アルゴリズム, 階層ネットワーク, エージェントシミュレーション, 交通システム

Layered Route Search Method for Large-scale Multi-agent-based Traffic Simulation

SHOTA FUKUDA^{1,a)} KAZUKI ABE^{1,b)} HIDEKI FUJII^{1,c)}
TOMONORI YAMADA^{1,d)} SHINOBU YOSHIMURA^{1,e)}

Received: November 22, 2017, Accepted: April 4, 2018

Abstract: Traffic simulations adopting multi-agent systems can reproduce behaviors reflecting the characteristics of each car agent. However, in the case of simulation targeting a large-scale road network, the computational time required for route search becomes a bottleneck. In this paper, we proposed a novel route search method using layered road network. By dividing the property used for layering road network and the properties used for route search, the proposed method outputs various result on a unique layered network. Through the experiments on the road network at Tokyo central, it was demonstrated that the proposed method can reduce the total query processing time to 1.5~2% of that by the conventional method.

Keywords: route search algorithm, layered network, agent-based simulation, traffic system

1. 緒言

交通は大規模で複雑なシステムであり, 交通施策が及ぼす効果を現実世界における実験によって事前に検証することは困難である. その代替として交通流シミュレータを用

いた仮想的な社会実験がさかんに実施されている [1].

交通は多くの自律的な行動主体が相互作用する現象であるため, 車両エージェントの個性を反映した挙動を再現できるマルチエージェントシステムを用いたモデル化が有効である. エージェントとして実装された自家用車やバス, タクシーなどは, 周囲の環境や自分以外のエージェントを知覚し, その状況に応じて行動する. あるエージェントの行動の影響は周囲のエージェントへと伝播し, その相互作用によって交通現象を再現することができる. MATSim [2] や SUMO [3], Megaffic [4] などが既存のマルチエージェント交通流シミュレータとしてあげられる. 著者らが開発す

¹ 東京大学
The University of Tokyo, Bunkyo, Tokyo 113-8656, Japan
a) s.fukuda@save.sys.t.u-tokyo.ac.jp
b) k.abe@save.sys.t.u-tokyo.ac.jp
c) fujii@sys.t.u-tokyo.ac.jp
d) tyamada@sys.t.u-tokyo.ac.jp
e) yoshi@sys.t.u-tokyo.ac.jp

る ADVENTURE_Mates [5], [6] もその 1 つであり, 実都市を対象とした交通施策の評価における実績を持つ [7], [8].

個々の車両挙動を詳細に表現する交通流シミュレータが抱える課題の 1 つは計算コストの大きさである. このうち, 走行中の車両挙動に関する処理は情報参照範囲が空間的に限定されるため領域分割型の並列化が有効であるが [4], [9], [10], 一方で, 経路探索処理では大域的な道路ネットワーク情報を参照するうえ, 最短経路を逐次処理によって確定させるといった性質を持つため, クエリ単位の並列化は容易であっても, 各クエリをさらに細分化し並列処理することは困難である. すなわち, 並列化による高速化には限界があるといえる.

アルゴリズムの改良によって経路探索を高速化する手法の 1 つとして, 道路ネットワークの階層化と段階的な簡略化が提案されている [11], [12]. 経路探索アルゴリズムの計算量は多くの場合ノード数 $|V|$ と, リンク数 $|E|$ によって決定される. たとえば基礎的な経路探索アルゴリズムである Dijkstra のアルゴリズム [13] では $O(|E| + |V| \log |V|)$ とされる. そのため, ネットワークのノード数とリンク数を適切に抑えることができれば, シミュレーションにおける各車両エージェントの経路探索時間を削減できる. 同時に, 大規模な道路ネットワークを対象とした経路探索では細街路を含んだ厳密な最短経路探索は現実的ではなく, 個々のドライバの道路ネットワークの知覚レベルに依存した経路選択が求められる.

本研究では既存の階層化された経路探索手法を応用して, 高速で, かつ, 多様性を表現可能な経路探索手法を提案する. 提案手法では道路ネットワークの階層化に用いる属性と経路探索に用いる属性とを区別することで, 単一の階層道路ネットワークを用いてエージェントごとの多様な経路探索結果を出力できるように工夫した.

以下, 2 章では関連研究と本研究の位置づけを明確にし, 3 章で手法の具体的な提案を行う. 4 章で東京都心部の大規模な道路ネットワークを対象とした数値実験例により本提案手法の有効性を示し, 5 章で結言を述べる.

2. 関連研究と本研究の位置づけ

2.1 経路探索アルゴリズム

経路探索問題を扱う際, 道路ネットワークはグラフ $G = (V, E)$ として扱われる. V はノードの集合, E はリンクの集合である. 道路ネットワークの場合, ノードは交差点, リンクは交差点どうしを接続する単路部に相当する. リンク $e \in E$ には, それぞれ非負のコスト $c_e \in \mathbb{R}_+$ が付与される. c_e には e に該当する単路部の長さやその単路部を通過するのに要する時間 (リンク旅行時間) などが用いられることが多い.

最短経路探索^{*1}では, スタートノード $s \in V$ からターゲットノード $t \in V$ に向かう様々な経路を探索し, 経路に含まれるリンクの総コストが最も小さい経路を出力する. 道路ネットワークの規模が大きい ($|V|$ や $|E|$ が大きい) 場合, 取りうるすべての経路を探索すると計算時間が膨大になってしまうため, Dijkstra のアルゴリズムをはじめ, 探索空間を制限する多くの手法が提案されてきた.

A*アルゴリズム [14] は Dijkstra のアルゴリズムの拡張のうち初期に提案されたアルゴリズムの 1 つである. スタートノード s からの総コスト $g(u)$ とターゲットノードまでの推定コスト $h(u)$ の和 $f(u) = g(u) + h(u)$ が最小となるノード u を優先的に探索する. 通常, 最短距離の経路を出力したい場合にはヒューリスティック関数と呼ばれる $h(u)$ として u, t 間のユークリッド距離が用いられる. ヒューリスティック関数を用いることで Dijkstra のアルゴリズムよりも探索空間を狭められること, ヒューリスティック関数しだいであるが, 一般に経路探索のための特別な前処理が必要ないことが利点としてあげられる. また, A*アルゴリズムをさらにオンライン探索へ拡張した手法として RTA* (Real Time A*) アルゴリズムや LRTA* (Learning Real Time A*) アルゴリズムなども提案されており [15], これらは高速に 1 つの経路を出力できる. ただし経路の最適性や学習に要する時間に課題があり, そのまま交通流シミュレーションに用いるのは適切でない.

経路探索アルゴリズムは双方向探索, すなわち s から t へ向かう探索 (順方向探索) とともに t から s へ向かう探索 (逆方向探索) を同時に行うことで, 探索空間をさらに狭められる [16], [17]. 多くの場合, 双方向探索の探索空間は単方向探索の半分以下となる. 双方向探索は Dijkstra のアルゴリズム, A*アルゴリズムおよびそれらから派生したアルゴリズムの計算量のオーダを減じるものではないが, 次節で述べる階層的経路探索アルゴリズムを含め, 様々な経路探索アルゴリズムで利用される技術である.

2.2 階層的経路探索

2.2.1 Highway Hierarchy

highway hierarchy [18] は, スタートノード s とターゲットノード t が十分に離れている場合, その最短経路は一部の重要なリンクを経由するという知見に基づく手法である. パラメータ H を設定し, s, t 間の最短経路 $P_{s,t}^* = (s, \dots, u, v, \dots, t)$ を構成するリンク $e = (u, v)$ のうち, $u \notin N_H(t)$ かつ $v \notin N_H(s)$ を満たすものを highway link とする. ここで $N_H(s)$, $N_H(t)$ はそれぞれ s, t から近い順に H 個のノードを集めた集合である. すべての s, t の組について求

*1 一般的に最短経路探索と呼ばれるが, コストとしてリンクの長さではなくリンク旅行時間を用いた場合は, スタートノードからターゲットノードまでの旅行時間を最小化する経路が得られる. 本論文では便宜上, コストが最小となる経路を総称して最短経路と呼ぶ.

めた highway link を接続することで、元のネットワークよりも簡略化された第1レベルの highway network を構築する。構築された highway network に対して同様の操作を繰り返すことによって、さらに上位の highway network を再帰的に構築できる。

階層的に構築したネットワークを用い、Dijkstra のアルゴリズムで双方向に探索することで最短経路を出力する。このとき、下位のネットワークから上位のネットワークへと探索が進行することは許容するが逆を認めないことで探索空間を制限する。最上位のネットワークのノード群を利用してさらに高速な経路探索を行う手法 [19] や、動的な経路探索を可能とした手法 [20] も存在する。

2.2.2 Contraction Hierarchy

contraction hierarchy [21], [22] は、ノードを縮約する操作を繰り返し行うことによってネットワークを階層化する手法である。ノード u を縮約するとは、 u を経由する他の2つのノード間の最短経路を探索し、その経路を短絡するリンクで置き換えることを意味する。各ノードには縮約のための重要度が与えられ、重要度の高いノードから重要度の低いノードへ順に縮約が行われる。

経路を探索する際は、重要度が低いノードから重要度が高いノードのみへと向かう探索を双方向に行う。リンクコストが変化するような動的なネットワークの場合に対応した手法 [23] では、ネットワーク全体でなくコストが変化したリンクにかかわるサブネットワークを組み替えることで、短い再構築時間で経路の最適性を保つ。

highway hierarchy, contraction hierarchy のいずれのアルゴリズムも、ネットワークがリンクコストを含めて不変あるいは変化が小さく、かつ多様性がないことを前提とし、大規模な前処理によってきわめて短いクエリ処理時間を達成する。これはカーナビゲーションシステムのように短時間のうちに画一的な最適経路を出力するような問題に向くものの、現実の交通現象の再現を考えた場合、自動車ドライバーごとに異なるであろうリンクコストの認識をアルゴリズムに組み込むのは困難である。ドライバーの個性に応じて別々の階層ネットワークを構築しなければならない、かつ個性が多様化するにつれ1つの階層ネットワークにおけるクエリ処理頻度が減少することにより、クエリ処理時間の短縮よりも前処理時間の増大が支配的となるためである。

2.2.3 道路ネットワーク特有の階層的経路探索

中村ら [12], [24] は道路ネットワークがほぼ平面グラフであり、かつ幹線道路と細街路とが分類されている（あるいは容易に分類できる）という特徴を利用して単純化ネットワークを構築した（図1）。

単純化ネットワークは幹線道路を表す幹線リンクと集約リンクから構成される。集約リンクとは、幹線リンクによって形成される多角形のうち2辺間を結ぶ経路をまとめた仮想リンクであり、1つの集約リンクが細街路の集合に

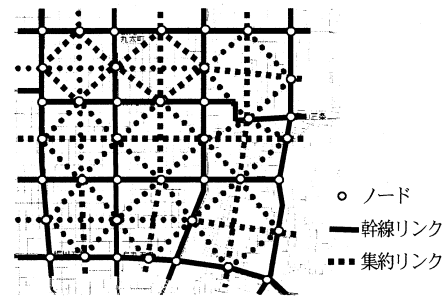


図1 単純化ネットワークの例 [12]

Fig. 1 Example of simplified network [12].

相当する。これは、自動車ドライバーが経路探索する際に幹線道路そのものを経路の要素として直接的に認識する一方、出発地および目的地周辺以外の区間において細街路を「ある幹線道路から別の幹線道路までの経路」として集約的に認識するという仮定に基づく。

中村らは単純化ネットワークを提案するとともに、ドライバーが経路を選択する際の効用関数を、経路に含まれる1車線・2車線・3車線以上の道路の総延長および経路の旅行時間、左折・右折回数、通過する信号交差点の数の重み付け線形和で定義することを提案し、2003年に名古屋都市圏でインターネット ITS 協議会が調査したタクシープローブデータを用いてその重みを推定している。

この手法による交通流シミュレーションの高速化の程度は Mita ら [25] が実証しているが、階層化が2層に限られるため大規模道路ネットワークへの適用が困難であること、また細街路は必ず幹線道路に囲まれていなければならないという制約があることが問題点として考えられる。

2.3 提案する階層的経路探索手法の特徴

本研究では、マルチエージェント交通流シミュレーションによって現実の交通現象を再現する場合に適用可能な経路探索アルゴリズムを提案する。具体的にはリンクに付与される情報のうち1つを「経路探索を行うどのエージェントにとっても同一の属性（以降、固定属性と呼ぶ）」、残りを「経路探索を行うエージェントごとに異なることを許容する属性（以降、可変コストと呼ぶ）」に分けたうえで、固定属性を用いて道路ネットワークを階層化し、そのうえで可変コストを用いた経路探索を実現する。

2.2.1 項および 2.2.2 項にあげた手法の問題点に対し、提案手法は同一の階層ネットワークであってもエージェントの個性を反映した可変コストによって多様な経路探索結果を出力しうる点でメリットがある。

本論文においては固定属性として車線数を利用する。これは、車線数が大きい道路を優先して通るという先行研究の知見 [12], [24] を利用するためであるが、別の固定属性、たとえば高速道路・国道・県道・市道のような道路種別を利用することもできる。固定属性として車線数あるいは道

路種別を用いる場合、それらの情報は OpenStreetMap [26] などから取得できる。2.2.3 項の手法と異なり、3 層以上の階層化も可能であり、シミュレーションの高速化への寄与が期待できる。さらに、新たに提案するネットワークの補正を併用することで幹線道路ネットワークの切断にも対処する。

なお、本研究で想定する交通流シミュレーションのプロセスは、エージェントがシミュレータ上に生成され、エージェントにとってのスタートノード、ターゲットノードおよび可変コストが与えられるたびに経路探索のクエリが生じるものであり、スタートノード、ゴールノード、可変コストの組に対し経路探索結果を保存し、それを再利用するものではない。これは仮に可変コストの組合せが無限であっても経路探索結果を出力するための条件である。

3. 提案手法

本章では道路ネットワークの階層化手法と、そのうえでの経路探索手法について述べる。

3.1 道路ネットワークの階層化

3.1.1 上位ネットワークの抽出

リンク e が持つ片側車線数のうち小さいものを l_e とする。ただしいずれかの片側車線数が 0 となる一方通行リンクは $l_e = 0$ ではなく $l_e = 1$ として扱う。 $l_e \geq n$ ($n \in \mathbb{N}$) を満たすリンクの集合 E_n による辺誘導部分グラフを $G_n = (V_n, E_n)$ と表記し、これをレベル n のネットワークと呼ぶこととする。最下層に位置する、もとの道路ネットワークは G_1 であり、 G_1 から順に上位ネットワーク G_n ($2 \leq n \leq N$) を抽出する ($V_1 \supseteq V_2 \supseteq \dots \supseteq V_N$, $E_1 \supseteq E_2 \supseteq \dots \supseteq E_N$)。なお N は階層化を制御するためのパラメータとして与える n の上限である。

同時に、車線数を考慮したリンクコスト \tilde{c}_e を式 (1) によって与える。 c_e は真のリンクコストであり、本論文ではリンクの両端点間のユークリッド距離として与えるが、リンク旅行時間で与えてもよい。 w_e は車線数が少ない場合の「リンクの選びにくさ」を表す係数であり、ここでは、先行研究によって求められた値 [12], [24] を w_e ($l_e \geq 3$) で正規化した値を用いる (式 (2))。

$$\tilde{c}_e = w_e c_e \quad (1)$$

$$w_e = \begin{cases} 1 & (l_e \geq 3) \\ 1.607 & (l_e = 2) \\ 2.768 & (l_e = 1) \end{cases} \quad (2)$$

w_e は地域性やドライバーが持つ道路ネットワークの知識などに依存するが、精密に求めるためには先行研究と同じくプローブデータを調査すればよい。本研究は w_e の与え方そのものを提案するのではなく、ドライバーの個性として

複数の w_e を許容する経路探索を提案するものであり、個性として与えられる w_e と出力される経路との関係は 4 章で議論する。実用上は、得られるシミュレーション結果と現実で得られる交通量などの観測値とを比較し、試行錯誤的に許容可能な w_e を推定する必要がある。

3.1.2 上位ネットワークの補正

車線数をもとにしてリンクを抽出するだけでは、多くの場合上位ネットワーク G_n ($2 \leq n \leq N$) は分断され非連結*2となる。後述する経路探索手法では 2.2.1 項で述べた手法と同様に上位ネットワークから下位ネットワークへの探索を禁じるため、上位ネットワークが非連結であると経路探索に失敗するケースが生じる。そこで、本研究では以下に示す手順にしたがって G_{n-1} から G_n へとリンクを追加することにより非連結を解消することを試みる。この補正は G_2 から順に G_N まで行う。

Step 1: ネットワーク G_n の連結成分のうち、ノード数が最大でないものを 1 つ選択する。これを $G_n^{cc} = (V_n^{cc}, E_n^{cc})$ とする。

Step 2: ノード $s \in V_n^{cc}$ を 1 つ選び、 s をスタートノードとして G_{n-1} 上でネットワーク修正用経路探索 (後述) を行う。得られた経路を構成するノードおよびリンクを保存する。

Step 3: ノード数が最大でないすべての連結成分の処理が終わっていないければ新たな G_n^{cc} を選び、Step 2 を繰り返す。

Step 4: Step 2 の繰返しによって得られたノード、リンクをそれぞれ V_n , E_n に追加する。

Step 5: 再びネットワークの連結性を調査し、ノード数が最大でない連結成分に含まれるすべてのノードおよびリンクをそれぞれ V_n , E_n から除外する。これにより G_n が連結であることを保証する。

Step 2 で用いるネットワーク修正用の経路探索を以下に示す。なおここで用いる関数 $d(s, u)$ はスタートノード s からノード u までの探索により確定した最短経路 $P_{s,u}^* = \langle s, \dots, u \rangle$ 上のリンクコストの総和であり、これを単にコストと呼ぶこととする。本研究では式 (2) によって重み付けされたリンクの両端点間のユークリッド距離をリンクコストとして用いた。また $h(s, u)$ は $P_{s,u}^*$ に含まれるノード数であり、これをホップ数と呼ぶこととする。ただし、ノード u が G_n においてスタートノード s と同じ連結成分に含まれている場合には $d(s, u) = h(s, u) = 0$ として扱う。つまり、以下の手順ではスタートノード s が含まれる連結成分からノード u までの最小コスト $d(s, u)$ およ

*2 本研究における道路はプログラム上は無向リンクとして実装され、3.1.1 項に記したように 1 つのリンクが持つ l_e の値は 1 つであるが、内部では 2 方向の片側車線数を保持しているため、一方通行の情報は失われない。したがって有向グラフにおける弱連結と同様の状況も発生しうるが、便宜上、そのような状況を含めて「非連結」と表記する。

び最小ホップ数 $h(s, u)$ を Dijkstra のアルゴリズム [13] によって求めている. ここで, どの程度の範囲まで分断されたサブネットワークどうしを連結するか制御する閾値として, ホップ数の上限 H_{max} をパラメータとして与える.

- Step 2a:** スタートノード s をキューに入れる.
- Step 2b:** キューが空なら Step 2f へ, 空でなければ $d(s, v)$ が最小のノード v をキューから取り出す.
- Step 2c:** $h(s, v) < H_{max}$ であれば Step 2d へ, そうでなければ Step 2b へ.
- Step 2d:** G_{n-1} における v の隣接ノード w について, v を経由した場合の $d(s, w)$ と $h(v, w)$ を算出する. w を初めて訪れるか, あるいはすでに訪れている場合でもそれまでに求めた $d(s, w)$ および $h(v, w)$ を下回るようであれば値を更新し w をキューに入れる.
- Step 2e:** v のすべての隣接ノードの処理が終わるまで Step 2d を繰り返し, Step 2b へ.
- Step 2f:** s が含まれる連結成分 G_n^{cc} からのコストが確定した G_{n-1} 上のノードのうち, レベル n において G_n^{cc} 以外の連結成分に含まれるものがあれば, そのノードまでの最短経路に含まれるノードおよびリンクを G_n に追加するノード, リンクとして出力する.

3.2 階層化されたネットワークを用いた経路探索

本研究で提案する経路探索の概要は, 2.2 節で述べた手法と同様である. すなわち, 探索を進めるにつれ上位の疎なネットワークに移らせることによって経路探索に用いるネットワークの規模を徐々に縮小させ, 経路探索の高速化を図る. このとき, やはり既存手法と同様に双方向の探索を行うことで, スタートノードからの順方向探索とターゲットノードからの逆方向探索が必ず単連結の上位ネットワークで出会うことを保証する.

以下に詳細を述べる. なお以下の手順において「探索を1つ進める」とは, A*アルゴリズムを用い, キューから $f(n)$ が最小のノードを1つ取り出し, 取り出したノードが G_n において隣接するすべてのノードの $f(n)$ を計算・更新し, 更新したノードをキューに加える操作を指す.

- Step 1:** 探索の対象となるネットワークのレベルを $n = 1$ として探索を開始する.
- Step 2:** スタートノード s を順方向探索のキューへ, ターゲットノード t を逆方向探索のキューに入れる.
- Step 3:** 順方向探索が G_{n+1} に含まれるノードにたどり着いていない場合, 順方向に探索を1つ進める.
- Step 4:** 逆方向探索が G_{n+1} に含まれるノードにたどり着いていない場合, 逆方向に探索を1つ進める.
- Step 5:** 双方から探索済みのノードがあれば探索を終了し, 経路を確定する.
- Step 6:** 双方からの探索がともに G_{n+1} に含まれるノードにたどり着いている場合, それらのノードを新たなス

タートノード, ターゲットノードとし, キューをクリアしたうえで $n \leftarrow n + 1$ として Step 2 へ. そうでなければ, n をそのままにして Step 3 へ.

4. 実験

前章で提案したアルゴリズムを用いて経路探索を行い, 出力される経路の性質と計算時間について典型的な経路探索手法である A*アルゴリズム [14] と比較する. 対象としたのは図 2 に示す東京都心部の道路ネットワークであり, これは OpenStreetMap [26] から取得したデータをもとに構築した.

4.1 道路ネットワークの階層化

図 2 に示したネットワークに対し, 階層化レベルの上限を $N = 3$ とし, ホップ数の上限をそれぞれ $H_{max} = 10, 30, \infty$ とした場合の階層ネットワークのノード数, リンク数を表 1 に示す. なお, 以降は階層化されたネットワークを LN (Layered Network), 階層化される前のネットワークを nLN (non-LN) と表記する. 階層化によってノード数, リンク数が削減されているが, H_{max} が大きい場合はレベ



図 2 東京都心部の道路ネットワーク
Fig. 2 Road network in Tokyo central.

表 1 階層ネットワークの諸元
Table 1 Specification of layered network.

	H_{max}	n	num. of nodes (ratio to nLN)	num. of links (ratio to nLN)
nLN	—	(1)	16,710 (1.00)	25,432 (1.00)
LN	10	2	3,521 (0.21)	3,704 (0.15)
		3	1,234 (0.07)	1,276 (0.05)
	30	2	3,881 (0.23)	4,159 (0.16)
		3	2,613 (0.16)	2,764 (0.11)
	∞	2	4,043 (0.24)	4,386 (0.17)
		3	2,903 (0.17)	3,085 (0.12)

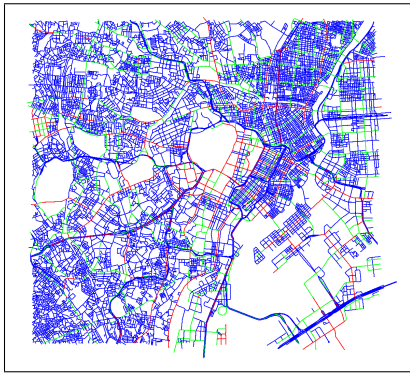


図 3 レベル 1 のネットワーク (G_1)
Fig. 3 Level 1 network (G_1).

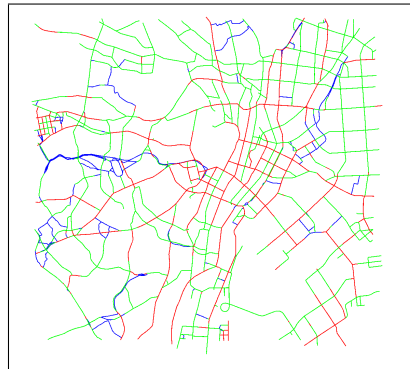


図 4 レベル 2 のネットワーク (G_2)
Fig. 4 Level 2 network (G_2).

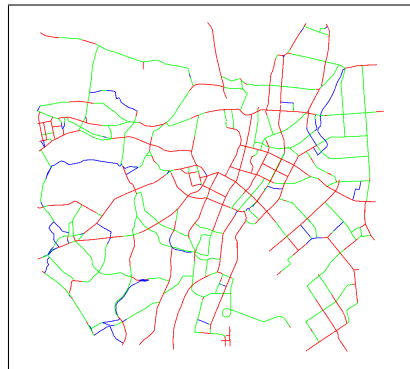


図 5 レベル 3 のネットワーク (G_3)
Fig. 5 Level 3 network (G_3).

ル 2 とレベル 3 の差が小さいことが分かる。

また、階層化の例として $H_{max} = 30$ の場合の G_1 から G_3 をそれぞれ図 3, 図 4, 図 5 に示す。図中の青, 緑, 赤のリンクはそれぞれ車線数が 1, 2, 3 以上であることを表す*3。たとえば図 5 中の青いリンクは, 3.1.2 項で述べた手法によって G_3 に追加されたリンクであることを表す。

4.2 経路探索の設定と結果

図 2 の道路ネットワークから無作為に 23,716 ペアの端

*3 OpenStreetMap に含まれる誤りをそのまま反映している可能性があるが, 経路探索手法の性能を比較するうえで悪影響がないと判断した。

点を抽出し*4, すべてのペアの間の経路探索を行い, その結果を比較する。ここで, エージェントの多様性を反映するため, 細街路を選択する傾向の強い順にエージェント群 A_1, A_2, A_3 を用意し, エージェント群 A_n ($n = 1, 2, 3$) に対するリンク e のコスト $\tilde{c}_{e,n}$ を式 (3), 式 (4) から式 (6) で定義した。

$$\tilde{c}_{e,n} = w_{e,n}c_e \quad (n = 1, 2, 3) \quad (3)$$

$$w_{e,1} = w_e \quad (\text{defined in Eq. (2)}) \quad (4)$$

$$w_{e,2} = \begin{cases} 1 & (l_e \geq 3) \\ 2.5 & (l_e = 2) \\ 4 & (l_e = 1) \end{cases} \quad (5)$$

$$w_{e,3} = \begin{cases} 1 & (l_e \geq 3) \\ 5 & (l_e = 2) \\ 10 & (l_e = 1) \end{cases} \quad (6)$$

従来の単方向探索である A* アルゴリズム, 双方向探索とした A* アルゴリズム, および, $N = 3, H_{max} = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, \infty$ とした提案手法を用いて経路探索を行った際の, エージェント群ごとの総クエリ処理時間 T_{total} および出力された経路の総コスト C_{total} , ネットワーク階層化に要した前処理時間 T_{pre} の比較を表 2 に示す。あわせて, 単方向探索の A* アルゴリズムと比較した場合の総クエリ処理時間, 総コストをそれぞれ図 6, 図 7 に示す。図中の Mono-A*, Bi-A* はそれぞれ単方向探索の A* アルゴリズム, 双方向探索の A* アルゴリズムを表す。ここで双方向探索の A* アルゴリズムとは階層化の影響を調査するために 3.2 節で述べた経路探索を nLN 上で行ったものであり, 最適経路を保証する手法ではないことに注意されたい。

階層化を行うことで総クエリ処理時間は単方向探索の A* アルゴリズムと比較して 1.5~2% 程度まで低減されており, 提案手法が交通流シミュレーションの計算時間低減に大きく寄与する可能性を示唆している。

本研究で提案する階層的経路探索は, リンクに付与される情報のうち 1 つを固定してネットワークを階層化し, そのうえでエージェントごとに異なるリンクコストを用い, エージェントごとに異なる経路を出力するものである。表 2 に示した総コスト C_{total} は重み $w_{e,n}$ が付与された総コストであったが, $H_{max} = 10$ とした場合に出力された経路の実際の距離の総和は, 細街路の選択に最も寛容なエージェント群 A_1 で 2.385×10^8 [m], 最も不寛容なエージェント群 A_3 で 2.619×10^8 [m] であり, 10% 弱の差が生じた。細街路をより積極的に取り入れたエージェント群が短い経

*4 ネットワークに含まれる端点は 613 であり, スタートノードとターゲットノードとして取りうる組合せの総数は $613 \times 612 = 375,156$ であるが, そのうち約 1/16 を選んだ。

表 2 各アルゴリズムを用いた経路探索の結果
Table 2 Route search results using each algorithm.

algorithm	agent group A_1		agent group A_2		agent group A_3		T_{pre} [sec]
	T_{total} [10^3 sec]	C_{total} [10^8 m]	T_{total} [10^3 sec]	C_{total} [10^8 m]	T_{total} [10^3 sec]	C_{total} [10^8 m]	
monodirectional-A*	39.578	2.890	47.894	3.785	66.860	6.661	0
bidirectional-A*	17.974	2.901	21.734	3.803	22.755	6.680	0
proposed method, $H_{max} = 10$	0.905	3.247	0.960	4.270	1.023	7.413	19
20	0.840	3.326	0.863	4.402	0.918	7.701	26
30	0.890	3.070	0.928	4.057	0.971	7.086	38
40	0.905	3.051	0.977	4.053	1.029	7.139	57
50	0.954	2.988	1.029	3.953	1.089	6.952	80
60	0.947	2.971	1.025	3.929	1.161	6.894	103
70	0.933	2.957	1.014	3.903	1.120	6.878	133
80	0.929	2.952	1.016	3.893	1.121	6.856	157
90	0.900	2.956	1.000	3.899	1.089	6.872	188
100	0.939	2.948	1.068	3.888	1.124	6.837	219
∞	0.889	2.949	0.952	3.889	1.049	6.832	327

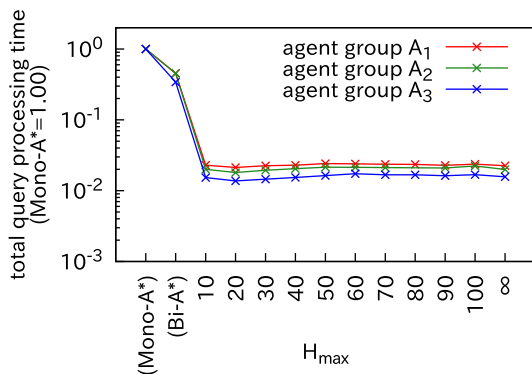


図 6 総クエリ処理時間の比較

Fig. 6 Comparison of total query processing time.

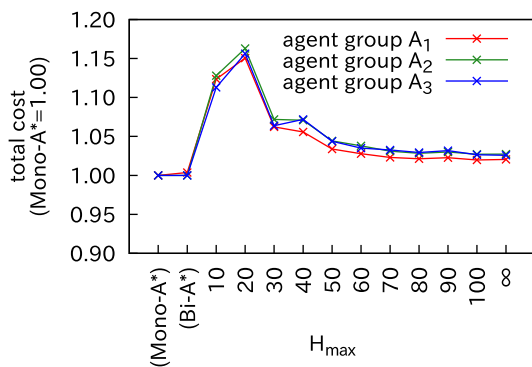


図 7 総コストの比較

Fig. 7 Comparison of total cost.

路を出力できた点も妥当な結果である。

提案手法は最短経路を保証しないアルゴリズムであるが、最短経路が保証された単方向探索の A* アルゴリズムと比較し、得られた経路の総コストが 2~15% 程度悪化していることも見てとれる。

4.3 考察

ここで、提案手法によって得られる経路の例を個別に示し、その質を定性的に評価する。図 8 は左から順に最短経路、 $H_{max} = 100$ とした場合のエージェント群 A_1 の経路、エージェント群 A_3 の経路である。図中の経路を構成する青、緑、赤のリンクは図 3 から図 5 と同じく、それぞれ車線数が 1, 2, 3 以上であることを表す。

最短経路は経路全般にわたって車線数 1 から車線数 3 以上のリンクまで頻繁に乗り換えており、人間のドライバーが選択する経路としては不自然といえる。一方でエージェント群 A_1 および A_3 の経路は経路長は長いものの、車線数 2 以上のリンクで構成されたシンプルな経路であり、全体として見ると実際にこのような経路を選択するドライバーがいても不自然ではないと考える。あわせてエージェント群 A_1 と A_3 が経路の中間で異なるリンクを選択していることも分かる。ただし A_3 の経路では 3 車線以上のリンクを重視するあまり、地図の中央付近で必要以上に右左折を繰り返しており、不自然な挙動であるといえる。パラメータの調整が必要である。

図 9 は左から順に最短経路、それぞれ $H_{max} = 20$, $H_{max} = 100$ とした場合のエージェント群 A_1 の経路である。最短経路の付近で G_2 , G_3 が十分に形成されず、上位ネットワークを使うことで逆に不自然な迂回が生じている。実際にこのような経路を選択するドライバーがいる可能性は否定できないが、特に $H_{max} = 20$ では迂回が顕著であり、 $H_{max} = 100$ で改善するものの、それでも一部で不必要に 3 車線以上のリンクを選択している。何らかの指標によって選択可能な上位ネットワークに制限を持たせることで問題を解決できる可能性があるが、今後の課題としたい。

上記のように解決すべき課題はあるものの、提案手法はエージェントごとに出力する経路の多様性を認める前提で

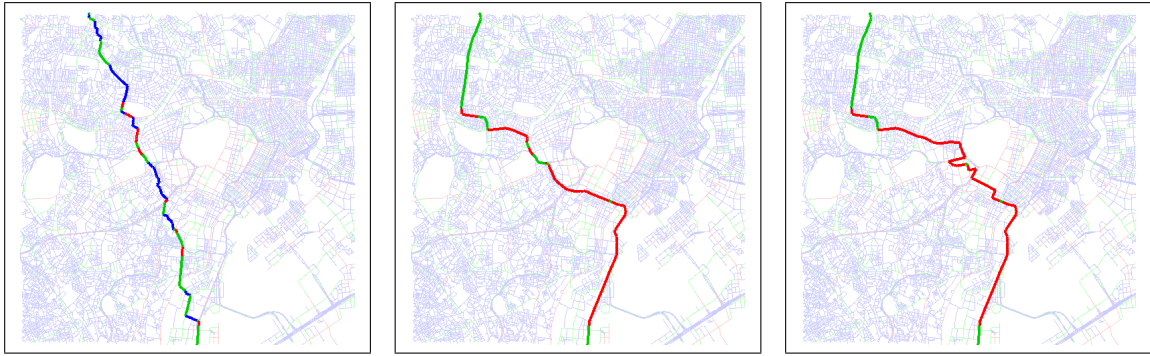


図 8 最短経路およびエージェント群 A_1, A_3 が選択する経路 ($H_{max} = 100$)
 Fig. 8 Shortest route and routes selected by agent group A_1, A_3 ($H_{max} = 100$).

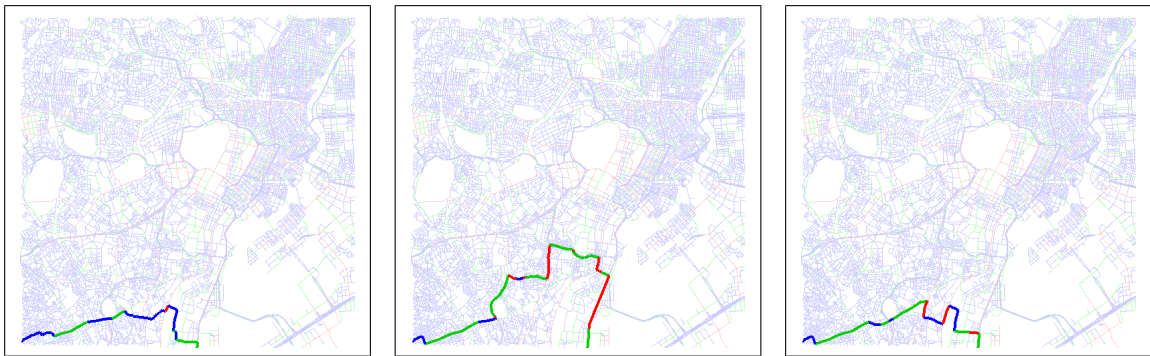


図 9 最短経路およびエージェント群 A_1 が選択する経路 ($H_{max} = 20, 100$)
 Fig. 9 Shortest route and routes selected by agent group A_1 ($H_{max} = 20, 100$).

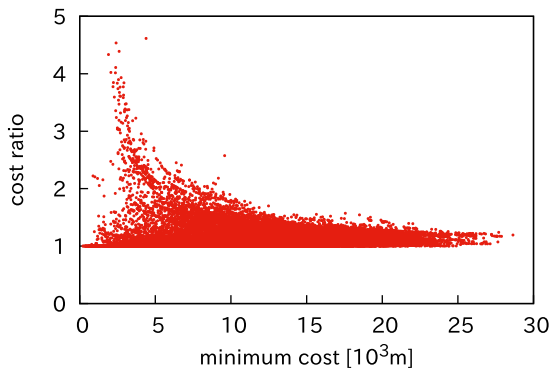


図 10 最小コストとコスト比の関係 ($H_{max} = 20$)
 Fig. 10 Minimum cost and cost ratio ($H_{max} = 20$).

あるため、最短経路に対して経路長が伸びるのは当然であり、図 7 に示すとおり、本論文で用いた条件においてはその程度が現実的な範囲におさまっている。

図 10 は、スタートノードとターゲットノードを結ぶ最短経路のコスト（これを最小コストと表記する）およびそれに対して提案手法で $H_{max} = 20$ とした場合にエージェント群 A_1 が選択する経路のコストの比をクエリごとに示したものである。最小コストが小さいクエリではコスト比が 4 を超えるような迂回を含む経路が少数ながら存在するが、最小コストが大きくなるにつれ、ほぼすべての経路のコスト比が 1 へと近づく。

つまり、スタートノードから遠く離れたターゲットノ

ドまでの探索であれば、最短経路の経路長に対して極端に長い迂回を含む経路を出力することがなくなる。実際には対象とする道路ネットワークにも依存するが、提案手法は大規模道路ネットワークにおいて長距離の経路を探索する場合に特に有効といえる。短距離（スタートノードとターゲットノードの直線距離が近い場合）のクエリは従来手法で処理し、計算時間増加の支配的要因となる長距離のクエリは提案手法で処理するという利用法も考えられる。

なお提案手法により得られる経路は H_{max} を大きくすることで最短経路へと近づく。 H_{max} が大きいほど上位ネットワークの補正に要する時間が長くなるが、表 2 に示すとおり H_{max} を無限にした場合であってもただか 330 秒程度であり、クエリ処理時間の短縮効果を十分に得られる。

5. 結言

本研究では、大規模マルチエージェント交通流シミュレーションへの応用を志向し、エージェントの個性を考慮可能な階層的経路探索手法を提案した。車線数を固定属性として上位ネットワークを抽出・補正することで階層ネットワークを構築し、探索が進行するにつれ上位のネットワークに移るような探索アルゴリズムを双方向的に用いて経路探索を行う。東京都心部の道路ネットワークに対し提案手法を適用したところ、異なるリンクコストを与えたエージェント群の個性を反映しつつ、従来手法に対して計

算時間を 1.5~2%程度に低減できたことを確認した。

本論文では固定属性として車線数を用い、経路探索で考慮するリンクコストも経路の距離という静的な属性であったが、動的な交通状況において旅行時間を考慮することはもちろん重要である。信号などによる一時的な滞留や比較的短時間で解消するような渋滞を考慮する場合、滞留や渋滞の影響を含めたリンク旅行時間を可変コストとして利用することができる。一方で長時間にわたる渋滞によってリンクに対する人々の選好が変動するような場合には、固定属性に反映し、新たに階層ネットワークを構築しなおすこともできる。提案手法を発展させたこれらの設定が計算時間あるいはシミュレーション結果に及ぼす影響については今後調査が必要である。

提案手法では内部で A*アルゴリズムを利用しているが、これは A*アルゴリズムでしか階層的な探索を実現できないことを主張するものではない。ネットワークの階層化と階層化されたネットワークにおける経路探索は独立しているため、Dijkstra のアルゴリズムや A*アルゴリズムから派生した種々の単層的な経路探索アルゴリズムを階層的に用いることも可能であり、高速かつエージェントの個性を考慮可能な経路探索という本研究の目的により良く合致するアルゴリズムの模索は今後も継続する予定である。

また提案手法は、リンクが持つ固定属性は各エージェントにとって共通という前提のもと、単一の上位ネットワークを構築しているが、これは現実の交通システムに対する過度な単純化となるおそれがある。たとえば、大型車と小型車の違いを考慮に入れる際、小型車は進入できるが大型車が進入できない道路が存在することを考えなければならぬ。この場合、車種ごとに階層ネットワークを構築すること、あらゆる車種が通行できる道路のみを上位ネットワークとして抽出すること、あるいは、探索が上位にしか進まない制約を緩和し、部分的に下位ネットワーク探索を許容することで非連結の上位ネットワークにおいても探索可能とすることなどの工夫が考えられるが、これらについても今後検討する。

謝辞 本研究は JSPS 科研費 15H01785 の助成を受けたものである。

参考文献

[1] 交通工学研究会 (編): 交通シミュレーション活用のススメ, 丸善 (2012).

[2] Balmer, M., Axhausen, K. and Nagel, K.: Agent-based demand-modeling framework for large-scale microsimulations, *Transportation Research Record: Journal of the Transportation Research Board*, No.1985, pp.125-134 (2006).

[3] Daniel, K., Jakob, E., Michael, B. and Laura, B.: Recent development and applications of SUMO - Simulation of Urban MObility, *International Journal On Advances in Systems and Measurements*, Vol.5, No.3&4, pp.128-138

(2012).

[4] 鈴木豊太郎, 小野寺民也, 今道貴司, 加藤 整, 井手 剛: X10 言語を用いた超並列大規模交通流シミュレーション, *PROVISION*, Vol.72, pp.74-80 (2012).

[5] 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, Vol.23, No.3, pp.228-237 (2004).

[6] Yoshimura, S.: MATES: Multi-Agent based Traffic and Environment Simulator - Theory, implementation and practical application, *CMES: Computer Modeling in Engineering and Sciences*, Vol.11, No.1, pp.17-25 (2006).

[7] 内田英明, 藤井秀樹, 吉村 忍, 荒井幸代: 道路ネットワークの変化に対する経路探索の学習, 情報処理学会論文誌, Vol.53, No.11, pp.2409-2418 (2012).

[8] Fujii, H., Uchida, H. and Yoshimura, S.: Agent-based simulation framework for mixed traffic of cars, pedestrians and trams, *Transportation Research Part C: Emerging Technologies*, Vol.85, pp.234-248 (2017).

[9] Cameron, G.D.B. and Duncan, G.I.D.: PARAMICS—Parallel microscopic simulation of road traffic, *The Journal of Supercomputing*, Vol.10, No.1, pp.25-53 (1996).

[10] 小橋敏浩, 文屋信太郎, 藤井秀樹, 吉村 忍: 知的マルチエージェント交通流シミュレータ MATES の並列化, 日本計算工学会論文集, Vol.2010, 20100003 (2010).

[11] 丹羽寿男, 吉田雄二, 福村晃夫: 道路網の階層的表現にもとづく経路探索アルゴリズムと地図情報システムへの応用, 情報処理学会論文誌, Vol.31, No.5, pp.659-666 (1990).

[12] 中村俊之, 吉井稔雄, 北村隆一: 全ての道路リンクに相当する単純化ネットワーク作成手法の構築, 土木計画学研究・論文集, Vol.23, pp.441-446 (2006).

[13] Dijkstra, E.W.: A note on two problems in connexion with graphs, *Numerische Mathematik*, Vol.1, No.1, pp.269-271 (1959).

[14] Hart, P.E., Nilsson, N.J. and Raphael, B.: A formal basis for the heuristic determination of minimum cost paths, *IEEE Trans. Systems Science and Cybernetics*, Vol.4, No.2, pp.100-107 (1968).

[15] Korf, R.E.: Real-time heuristic search, *Artificial intelligence*, Vol.42, No.2-3, pp.189-211 (1990).

[16] Pohl, I.: Bi-directional and heuristic search in path problems, PhD Thesis, Department of Computer Science, Stanford University (1969).

[17] Ikeda, T., Hsu, M.-Y., Imai, H., Nishimura, S., Shimoura, H., Hashimoto, T., Tenmoku, K. and Mitoh, K.: A fast algorithm for finding better routes by AI search techniques, *IEEE Vehicle Navigation and Information Systems Conference*, pp.291-296 (1994).

[18] Sanders, P. and Schultes, D.: Highway hierarchies hasten exact shortest path queries, *European Symposium on Algorithms*, pp.568-579, Springer (2005).

[19] Bast, H., Funke, S., Sanders, P. and Schultes, D.: Fast routing in road networks with transit nodes, *Science*, Vol.316, No.5824, p.566 (2007).

[20] Schultes, D. and Sanders, P.: Dynamic highway-node routing, *Experimental Algorithms*, pp.66-79 (2007).

[21] Geisberger, R., Sanders, P., Schultes, D. and Delling, D.: Contraction hierarchies: Faster and simpler hierarchical routing in road networks, *Experimental Algorithms*, pp.319-333 (2008).

[22] Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D. and Werneck, R.F.: Route planning in transportation networks, *Algorithm Engineering*, pp.19-80, Springer (2016).

- [23] Geisberger, R., Sanders, P., Schultes, D. and Vetter, C.: Exact routing in large road networks using contraction hierarchies, *Transportation Science*, Vol.46, No.3, pp.388-404 (2012).
- [24] 中村俊之, 吉井稔雄, 北村隆一: 単純化ネットワークを用いた経路選択モデルの構築と適用, 土木学会年次学術講演会講演概要集第4部, Vol.60, pp.71-72 (2005).
- [25] Mita, N., Uchida, H., Fujii, H. and Yoshimura, S.: Speedup of dynamic route search for large-scale microscopic traffic simulation, *Journal of Advanced Simulation in Science and Engineering*, Vol.4, No.1, pp.31-43 (2017).
- [26] The OpenStreetMap Foundation: OpenStreetMap, 入手先 (<https://www.openstreetmap.org>) (参照 2017-11-13).



福田 隼馬

2017年東京大学工学部システム創成学科卒業。現在、同大学大学院工学系研究科システム創成学専攻修士課程在学中。学士（工学）。個のモデリングに関する研究に従事。



阿部 和規

2014年東京大学工学部システム創成学科卒業、2016年同大学大学院工学系研究科システム創成学専攻修士課程修了。現在、同専攻博士後期課程在学中。修士（工学）。交通流シミュレーションのデータ処理、推定・最適化に関する研究に従事。日本機械学会、交通工学研究会会員。



藤井 秀樹 (正会員)

東京大学大学院新領域創成科学研究科人間環境学専攻博士後期課程修了。同大学人工物工学研究センター特任助教等を経て、現在、同大学大学院工学系研究科システム創成学専攻講師。博士（環境学）。マルチエージェントシステムの交通流シミュレーションへの応用等に関する研究に従事。人工知能学会、日本シミュレーション学会、交通工学研究会等各会員。



山田 知典

東京大学工学部システム量子工学科卒業、同大学大学院工学系研究科修士課程、および、博士課程修了。博士（工学）。理化学研究所、日本原子力研究開発機構、東京大学人工物工学研究センター等を経て、現在、同大学大学院工学系研究科システム創成学専攻准教授。大規模シミュレーションに関する研究に従事。日本機械学会、日本計算工学会等各会員。



吉村 忍

東京大学工学部原子力工学科卒業、同大学大学院工学系研究科原子力工学専門課程修士課程および博士課程修了。工学博士。東京大学大学院工学系研究科助教授等を経て、現在は同研究科システム創成学専攻教授。知的シミュレーションの研究開発と工学分野、社会・環境分野への応用に関する研究・教育に従事。日本計算力学連合会長、日本機械学会理事・フェロー、国際計算力学連合常任理事・フェロー等。