

駐車経路探索における双方向 A*法の有用性

羽鳥 友基(17X4117) ZHU YIWEI (18X4040) 藤田 颯(18X4119) 指導教員 五島 洋行

1. はじめに

日本の自動車産業は著しい発展を遂げ、自動車の保有率も急増している。その中で、自動車移動の効率性や交通環境負荷等の問題の一部として、駐車場の効率化は重要な社会的課題となっている。

本研究では、先行研究として最適な駐車スペースを発見し、提案するという駐車場最適化[1]–[3]を参考に、その続きとして、駐車スペースまでの最短経路の算出に注目した。その算出において双方向 A*法を提案し、その手法が算出時間において有用かどうかを他探索手法と比較する。そして、その結果から優れている手法を判断し、本研究における最適手法とする。

2.1 A*法

A*法 (A*Search Algorithm, A*) とは、ダイクストラ法のやり方を基に、各ノードにコストを設け、推定値を計算することで最短経路を発見するアルゴリズムである。経路をノードで表現し、始点から終点までの経路を計算し、その経路が最短であることを保証するアルゴリズムで、始点から終点までの間の障害物等を避けた経路も算出が可能である。推定コストには様々なパターンが存在するが、一般的には、ユークリッド距離を用いたヒューリスティック関数でのコスト計算が広く知られている。

2.2 単方向探索

単方向探索とは、始点から終点に向かい、一方向から探索する探索方法である。その中で A*法を含む 4 種を本研究で採用する。

- ・深さ優先探索 (Depth First Search, DFS) … 先に進めない行き止まりのノードに辿り着くまで経路を戻らず隣接ノードを進む探索方法。
- ・幅優先探索 (Breadth First Search, BFS) … 探索開始する頂点から近い順に探索する方法。
- ・ダイクストラ法 (Dijkstra's Algorithm) … ノードとエッジで構成されたグラフで、2 点間の距離を最短にする経路の候補をあげ、各ノードにコストを与え、その和が最小になるよう順次更新して経路の選択をする方法。

2.3 双方向探索

双方向探索 (Bidirectional search) とは、グラフ探索アルゴリズムの一種で、同時に 2 つの方向から探索を行う。一方は始点から終点に向かい探索し、もう一方は終点から始点に向かい探索して、両方の探索で交わったノードのうち、最小の総コストとなるノードが確定した場合、探索を終了とする。双方向探索は、同手法の単方向探索と比較して、探索範囲を狭め、計算量を少なくし、経路の算出時間を短縮する効果が見込まれる。

3. 使用データと実験環境

本実験で想定した駐車場は、大型ショッピングセンターや大型テーマパークの立体駐車場である。110×110(m)の駐車場を 2 つ連結させ、一部構造上迂回しなければならないルートを作り、以下図 1 のような複雑な駐車場構造を再現する。

駐車スペースのノード数：1148

総ノード数：8619

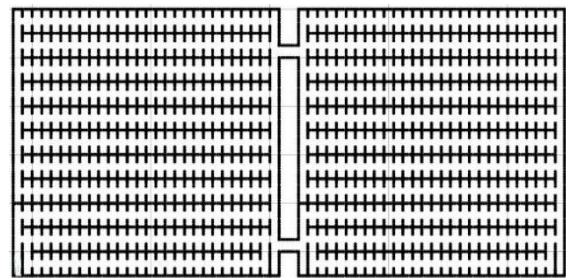


図 1: 駐車場マップ

実験環境は以下表 1 のようになっている。

表 1: 実験環境

OS	Windows10 Home 64-bit
プロセッサ	Intel® Core™ i3-8145U CPU @ 2.10GHz
実装 RAM	8.00GB
使用言語	Python 3.8.5 64-bit

4. 実験

本研究では各探索手法の比較において、最短経路の算出に要した計算時間の長短で性能を評価することとする。駐車場で距離を分散させ選択した10地点を終点に設定し、その終点までの最短経路の算出に要した各々の探索手法の計算時間を計測する。その結果を双方向A*法の計算時間と比較し、性能の評価をする。比較は、単方向探索4種で最も計算時間が短い1手法と双方向ダイクストラ法で行う。

図2より、4手法において10地点全てでA*法が最も計算時間が短いため、双方向A*法との比較に使用する単方向探索はA*法で行う。

図3より、10地点全てで双方向A*法による探索が最も計算時間が短い。そのため、他各手法よりも双方向A*法の方が計算時間の観点で優れている。なお、始点から終点までの移動経路が長い終点では、各手法と双方向A*法の計算時間の差は大きい。

5. おわりに

本研究では、駐車経路探索において双方向A*法が有用性のある手法であると推察し、有用性を検証した。各手法の算出時間を比較した結果、駐車経路探索において双方向A*法に有用性があることが示すことができた。評価実験より、双方向A*法の算出時間が最短となり、大型駐車場の最短経路探索において双方向A*法が有用であるといえる。また、始点から終点までの

最短経路が長いほど、双方向A*法と各手法の算出時間の差が大きいことから、駐車場が混雑して入り口付近に駐車できない状況下で、より双方向A*法が有用であると考えられる。また、今回の実験は、始点と終点が指定されていることから、単一頂点对最短経路問題とみなせる。よって、双方向A*法は単一頂点对最短経路問題においても有用性があるといえる。

今回の実験結果から、駐車場の最短経路算出を実際にセンシング技術や車車間通信技術等の駐車最適化システムと組み合わせ、実用化することを考えたとき、双方向A*法を採用し、高速化を図るのが良いと考えている。

参考文献

- [1] 孫為華, 剣持真弘, 山内由紀子, 柴田直樹, 安本慶一, 伊藤実, “待ち時間を短縮する駐車ナビゲーション,” 情報処理学会研究報告, Vol. 2011-DPS-148 No.7 September 2011.
- [2] 孫為華, 剣持真弘, 柴田直樹, 安本慶一, 伊藤実, “待ち時間を短縮する駐車ナビゲーション,” マルチメディア通信と分散処理ワークショップ論文集(DPSWS2013), December 2013.
- [3] W. Sun, N. Shibata, M. Kenmotsu, K. Yasumoto, M. Ito, “A Method for Navigating Cars in Multilevel Parking Facility,” J. Inf. Process. Vol.23 No.4, pp.488-496, July 2015.

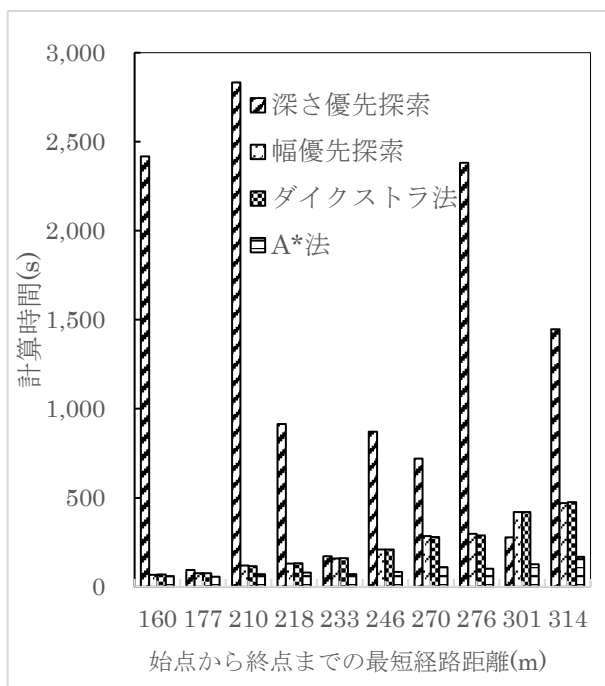


図 2: 単方向探索手法の計算時間

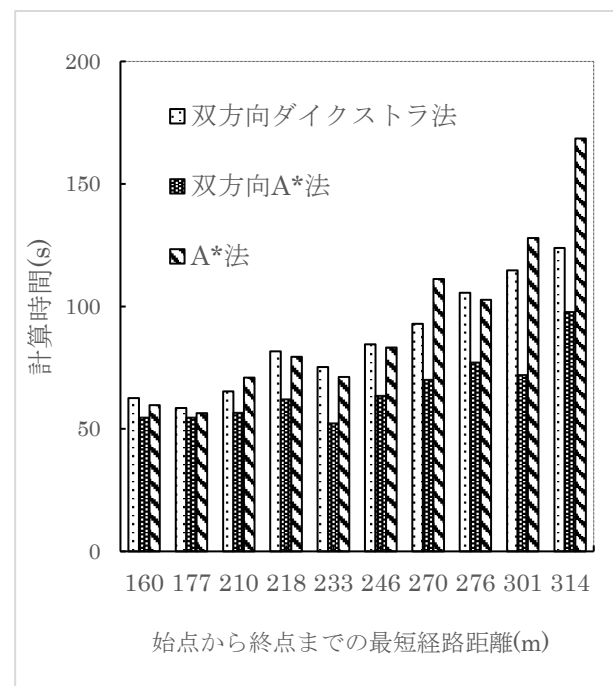


図 3: 3 手法の計算時間の比較

駐車経路探索における 双方向 A*法の有用性

法政大学 理工学部 経営システム工学科
経営数理工学研究室

17X4117 羽鳥 友基

18X4040 ZHU YIWEI

18X4119 藤田 颯

指導教員 五島 洋行 教授

学科名	経営システム工	学籍番号	17X4117, 18X4040, 18X4119
申請者氏名		羽鳥 友基, ZHU YIWEI, 藤田 颯	
指導教員名		五島 洋行	

論文要旨

論文題目	駐車経路探索における双方向 A*法の有用性
------	-----------------------

近年, 日本の自動車産業は著しい発展を遂げ, 自動車の保有率も急増している. そのため, 自動車移動の効率性 (渋滞) や交通環境負荷等の問題により, 駐車や輸送の効率化は重要な社会的課題となっている.

駐車場の問題における先行研究として, センシング技術や車車間通信技術といった技術があり, 最適な駐車スペースを見つけることへの研究が多くされている. しかし, 最適なスペースの発見をした後, そのスペースまでの最短経路探索について触れている研究は少ない.

そこで本研究では, 駐車場の混雑状況に焦点を当て, 駐車スペースまでの最短経路の算出において双方向 A*法を提案し, その手法が有用であるかどうかを他の探索手法と比較し, 検証する.

単方向探索としてダイクストラ法, 深さ優先探索, 幅優先探索, A*法, 双方向探索としてダイクストラ法を採用し, 各算出時間と双方向 A*法の算出時間と比較したところ, 駐車場における最短経路探索の計算速度において双方向 A*法を用いた探索が最も有用であるという結果が得られた.

本研究より, 駐車場の最短経路算出を実際にセンシング技術や車車間通信技術等の駐車最適化システムと組み合わせ, 実用化することを考えたとき, 双方向 A*法を採用し, 高速化を図るのが良いと考えている.

目次

第1章 序論.....	- 1 -
第2章 先行研究.....	- 2 -
第3章 基本方針.....	- 5 -
3. 1 先行研究の問題提起.....	- 5 -
3. 2 本研究の位置づけ.....	- 5 -
3. 3 仮説.....	- 6 -
3. 4 仮説検証.....	- 6 -
第4章 関連知識.....	- 7 -
4. 1 木構造.....	- 7 -
4. 2 深さ優先探索.....	- 7 -
4. 3 幅優先探索.....	- 8 -
4. 4 ダイクストラ法.....	- 9 -
4. 5 A* (A-STAR) 法.....	- 10 -
4. 6 双方向探索.....	- 13 -
4. 6. 1 双方向ダイクストラ法.....	- 14 -
4. 6. 2 双方向A* 法.....	- 14 -
第5章 駐車経路シミュレーション.....	- 17 -
5. 1 前提条件.....	- 17 -
5. 2 シミュレーション設定.....	- 17 -
5. 2. 1 駐車場構造.....	- 17 -
5. 2. 2 ノード探索.....	- 19 -
5. 2. 3 ノード管理.....	- 21 -
5. 2. 4 確定ノード導出.....	- 22 -
5. 2. 5 最短経路導出.....	- 22 -
5. 3 実験環境.....	- 23 -
第6章 評価実験.....	- 24 -
6. 1 評価方法.....	- 24 -
6. 2 探索結果.....	- 24 -
6. 3 比較1.....	- 28 -
6. 4 比較2.....	- 29 -

第 7 章 結論.....	- 30 -
参考文献.....	- 31 -
謝辞	- 32 -

第1章 序論

日本の自動車産業は著しい発展を遂げ、自動車の保有率も急増している。そのため、自動車移動の効率性や交通環境負荷等の問題により、駐車や輸送の効率化は重要な社会的課題となっている。本研究では、これらの問題の解決を図るべく、駐車場の混雑状況に焦点を当て、駐車場における最短経路の算出時間を双方向A*法を用いた探索と様々な探索法とで比較し、双方向A*法を用いた算出の有用性を検討する。

日本では駐車場内の渋滞は各地で頻繁に起きており、特に大型ショッピングモールや空港等の公共施設のような車の出入りの多い駐車場では、入場規制や整備員を置いているところが多くみられる。そのような駐車場において、店舗の入り口へ近い場所や買い物後にカートを横につけやすい場所等は人気が高く、そういった場所を探して周回する人は少なくない。この人気の差が駐車場内での渋滞を引き起こしている。そのような状況下で新しく入場したドライバーが、どのスペースが空いているのか、埋まっているのかを把握できず、既に起きている渋滞に巻き込まれることでさらに渋滞の規模は拡大していく。このような渋滞により客足の回転率が下がり、店舗側の収益にも大きな損失を及ぼす。したがって、駐車場で起きている渋滞の回避は顧客側と店舗側どちらにとっても重要な課題である。また、単に駐車場の渋滞は混雑することだけが問題ではなく、数十年前から世界中で大きな社会問題である二酸化炭素の排出による環境への悪影響にも関わってくる。

現在、上記課題の解決策として、実際に様々なシステムが導入されている。空車、満車を把握し、出入口前に可視化するシステムやスマートフォンで遠方からでも空き状況を確認できるシステムの導入により混雑を未然に防ぐなど様々な対策措置が取られている。

そのような様々なシステムが既に導入されたショッピングモールなどでも未だに、なるべく近くの駐車スペースを見つけるために周回している車が多く、渋滞しているケースが多く見られる。そこで、システムによって駐車スペースを最適に選択し、指定することで駐車箇所を固定し、そこまでの最短経路を算出し、混雑することなく行くことのできる最適ルートを提案することで無駄な駐車場周回を防ぎ、駐車場内の混雑を緩和できると考えた。

本研究では、先行研究として最適駐車スペースを発見し、提案するという駐車場最適化を参考に、その続きとして、その駐車スペースまでの最短経路の算出に注目した。その算出において双方向A*法が有用であるかどうかを他探索手法と比較し、その結果から優れている手法を判断し、採用する。

第 2 章 先行研究

駐車経路ナビゲーションの研究において主要となる技術は、センシング技術と車車間通信技術である。センシング技術とは、センサを使用して、物理的、化学的、または生物学的特性の量を検出して情報を取得し、付加価値の高い情報に変換する技術である。車車間通信技術とは、車両同士の無線通信により周囲の車両の情報（位置、速度、車両制御情報等）を入手し、必要に応じて運転者に安全運転支援を行う技術である。

センシング技術を活用した研究として、Y. Bi ら[1]を紹介する。

Step 1：駐車場内に無線センサネットワーク環境を用いることで、リアルタイムに駐車場の空車状況の情報を取得し、空車スペースを検知する。

Step 2：全ての駐車スペースと通路に対してノードを設定し、検知した空車スペースを個々の車両に別々に振り分ける。

この手法のメリットとして、個々のドライバーが 1 つの空車スペースを取り合うといったような現象を未然に防げることが挙げられる。しかし、駐車場が大規模になると無線センサにかかる費用が高くなるといった課題がある。

車車間通信技術を活用した研究として、R. Lu ら[2]を紹介する。

Step 1：車車間通信技術（VANET）や路車間通信技術（DSSS）を用いて駐車場の空車状況の情報を取得する。

Step 2：取得した情報をもとに各車両に対して電子駐車チケットを発行する。各電子駐車チケットは駐車スペースが指定されており、異なる駐車スペースには駐車ができない。

駐車スペースは渋滞が起きないように流量のバランスを重視して振り分けるため、駐車場内での渋滞緩和の効果が期待できる。しかし、ドライバーの行動特性や満足度を考慮していないので、ドライバーが不満を持つてしまう可能性が生じる。

ドライバーの満足度を考慮した駐車経路ナビゲーションとして、徳田ら[3]を紹介する。駐車場情報と車両情報をもとに駐車場の混雑状況を予測し、駐車場と車両との間で駐車場の利用に対する交渉を行う。

交渉により各車両は選好に適した駐車スペースを割り当てられるといったメリットが生じる。しかし、ドライバーが駐車場へ到着する時間が遅くなることで駐車場の利用効率が悪くなるという恐れがある。

[1]–[3]の問題点を踏まえて提案された、孫ら[4]–[6]を紹介する。本研究が最も参考にした研究であるため、特に重要である研究概要、駐車場構造、駐車経路導出方法について詳細に述べる

(i) 研究概要

大型ショッピングモールに多く採用されている立体駐車場をモデル化し、その環境下で駐車待ち所要時間を最小化する駐車場ナビゲーションを提案している。提案手法は以下の3つのステップで構成されている。

Step 1：駐車場内のサーバに提案手法を搭載した車両のナビゲーションシステムからアクセス可能であると仮定し、サーバはナビゲーションシステムから駐車場の統計利用情報と各ゾーンで駐車したか通過したかの最新の駐車センシング情報を収集する。

Step 2：Step 1 の情報をもとに、サーバは各駐車場にかかる時間、およびそのゾーンで駐車できる確率を推定し、これらのデータを各車両へ送信する。

Step 3：Step 2 の情報をもとに、車両は駐車場内のゾーンを一定数経由する全経路に対して駐車スペースを見つけるまでの時間期待値を計算し、時間期待値が最小となる経路を導出し、ドライバーに提示する。

(ii) 駐車場構造

駐車場構造を図1に示す。店舗入り口、駐車ゾーン、場内交差点、駐車場入り口の集合をそれぞれ、 CE , PZ , IN , G で表す。場内経路の集合を R とするとき、駐車場全体は、 $CE \cup PZ \cup IN \cup G$ をノード全体の集合とし、 R をリンクの集合とするグラフで表される。

駐車スペースは、1台の自動車を駐車するのに必要なスペースである。駐車ゾーンは、複数の駐車スペースと、隣接する駐車ゾーンに至る自動車用経路から構成される。駐車場は複数の駐車ゾーンと、各ゾーンから店舗入り口に至る歩行者用経路から構成される。

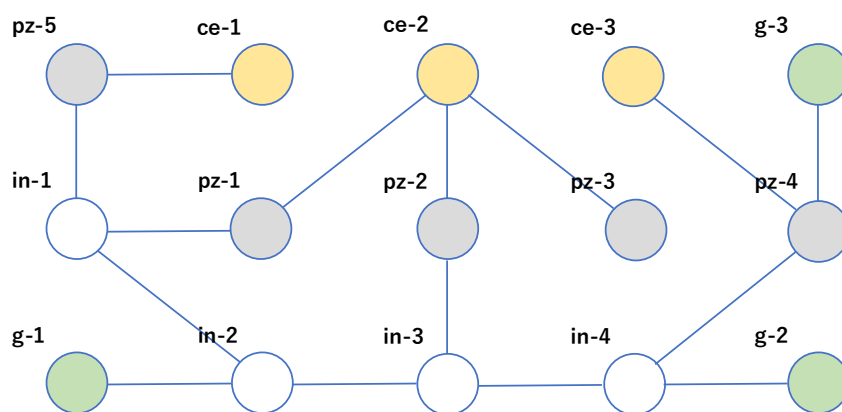


図1：駐車場構造

(iii) 駐車経路導出方法

経路導出は，中央サーバで動作するサーバ部と，各車両で動作する車両部で行われる．

1 サーバ部

サーバ部では，近隣ゾーン間を移動するのににかかった時間と，過去 30 分でそのゾーンに駐車できた割合から，ゾーンに駐車できる確率を計算する．サーバ部でのアルゴリズムの入出力は以下のようになっている．

入力：30 分以内に駐車場を訪れた車両がゾーン間を移動するのににかかった時間，各自動車がゾーンを通過したか駐車したかの情報，過去のそれらの情報，駐車場の地図

出力：ゾーン間を移動するのに必要な時間の期待値 E ，各ゾーンで駐車できる確率 P_{pz}

2 車両部

車両は入場後自身の位置を特定し，サーバと通信しながら駐車経路を算出する．

$$Exp_{rpr} = \sum_i^{i \in rpr} \left(P_{pzi} (TP_{ei} + TZ_{pzi}) \prod_k^{k < i} (1 - P_{pzk}) \right) \quad (1)$$

推薦駐車経路 rpr の期待値は Exp_{rpr} で表し，式(1)で示す． P_{pzi} と TZ_{pzi} ， TP_{ei} はそれぞれ，駐車ゾーン i で駐車できる確率，ゾーン i 内の移動時間，前のゾーンからゾーン i までの移動時間を表している．式(1)中の $P_{pzi}(TP_{ei} + TZ_{pzi})$ は，駐車ゾーン i に到達して駐車するまでの時間期待値を表しており， $\prod_k^{k < i} (1 - P_{pzk})$ は，駐車ゾーン i に駐車できる確率を表している．

入場後や駐車経路が失敗した場合は，サーバがアナウンスする parking information に基づき，自身の位置から最も駐車期待時間が短い駐車経路を算出する．車両部でのアルゴリズムの入出力は以下のようになっている．

入力：サーバ部の出力，駐車場の地図

出力：車両の現在地から 4 つのゾーンを通過する経路のうち，駐車できるまでの時間の期待値が小さな経路 3 つの中からランダムに選んで得た経路

第3章 基本方針

本章では、先行研究の問題提起を行い、本研究の基本方針を示す。

3. 1 先行研究の問題提起

先行研究では、センシング技術や車車間通信技術などを用いて空き駐車スペースを検知する研究や、駐車場内の混雑状況やドライバーの行動心理を踏まえて各車両に駐車スペースを振り分ける研究が多くなされてきた。しかし、各車両に駐車スペースを振り分けた後の経路探索まで触れている研究が少ない。

孫ら[4]–[6]は駐車経路算出を行っているが、算出における計算が煩雑である。理由としては、利己的なドライバー（提示されているナビゲーションを無視して駐車したいスペースに駐車するドライバー）の存在や、提案手法のシステムが搭載されていない車両が存在することを想定しているため、ナビゲーション通りに駐車できない可能性も考慮して立式しなければならないからである。その結果、駐車できるかどうかの確率を計算したり、駐車できるまでの時間期待値の計算をしたりする必要があり、計算が煩雑になる。

3. 2 本研究の位置づけ

本研究は、先行研究の問題を解消することに焦点を当てている。問題点は4. 1で述べたとおり、駐車経路算出における計算が煩雑であるという点である。この問題の原因は、ナビゲーション通りにドライバーが行動しないことを考慮しなければならないということである。

私たちは、自動駐車を想定することで、ナビゲーションを無視して駐車をする可能性を排除することができると考えた。ここで想定している自動駐車とは、近年注目されている自動バレーパーキング[7]である。自動バレーパーキングとは、ドライバーが駐車場で降車し、スマートフォンなどの端末を使って自動駐車のリクエストをすると、ドライバーが立ち会うことなく、無人で自動的に駐車してくれるサービスのことである。自動バレーパーキングは、各車両に特定の駐車スペースを振り分ける。したがって、駐車できるかどうかの確率を計算したり、駐車できるまでの時間期待値の計算をしたりする必要がなくなり、経路算出における計算は単純化される。経路探索手法においては、ダイクストラ法やA*法が知られており、それらの手法をもとに新たな手法が提案されている。

そこで、本研究では、各車両に振り分けた駐車スペースまでの最短経路算出において、有用性のある経路探索手法を検証することを目的とする。

3. 3 仮説

本研究は、駐車スペースを振り分けた後の経路探索を想定していることから、始点と終点は指定されているものとする。したがって、想定状況は単一頂点对最短経路問題（2 頂点对最短経路問題）とみなすことができる。単一頂点对最短経路問題は、一般的に単一始点最短経路問題のアルゴリズムを使用することが広く知られている。また、ヒューリスティック関数や、双方向探索を用いることで、算出時間が短くなると言われている。

そこで、私たちは、ヒューリスティック関数と双方向探索を兼ね備えた、双方向 A*法が有用性のある手法なのではないかと推考した。

したがって、本研究では、駐車経路探索における双方向 A*法の有用性を検証する。

3. 4 仮説検証

双方向 A*法に有用性があるかどうかを検証するために、一般的に広く知られている以下の探索アルゴリズムを用いて比較する。

（i）単方向探索

深さ優先探索，幅優先探索，ダイクストラ法，A*法

（ii）双方向探索

双方向ダイクストラ法

仮説検証は、駐車経路シミュレーションで行う。駐車経路シミュレーションの大まかな流れを図 2 に示す。駐車経路シミュレーションのもと、各手法による探索の算出時間を比較する。詳細は第 5 章で説明する。

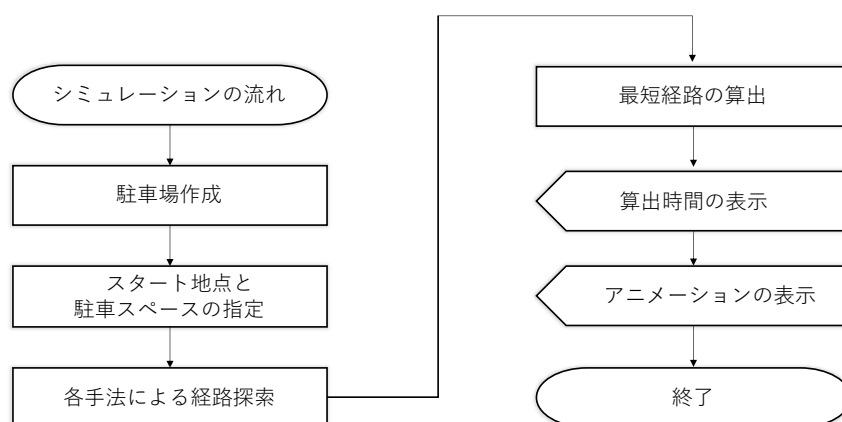


図 2：シミュレーションの大まかな流れ

第4章 関連知識

本章では、探索手法において重要なデータ構造の一種である木構造と、検証実験で用いる単方向探索、双方向探索の各手法について述べる。

4. 1 木構造

木構造とは、節点 (node) と枝 (edge) から構成されているデータ構造である。最上位の節点は根 (root) といい、最下位の節点は葉 (leaf) という。また、上の枝に繋がっている節点は親、下の枝に繋がっている節点は子である。木構造の仕組みを図3に示す。

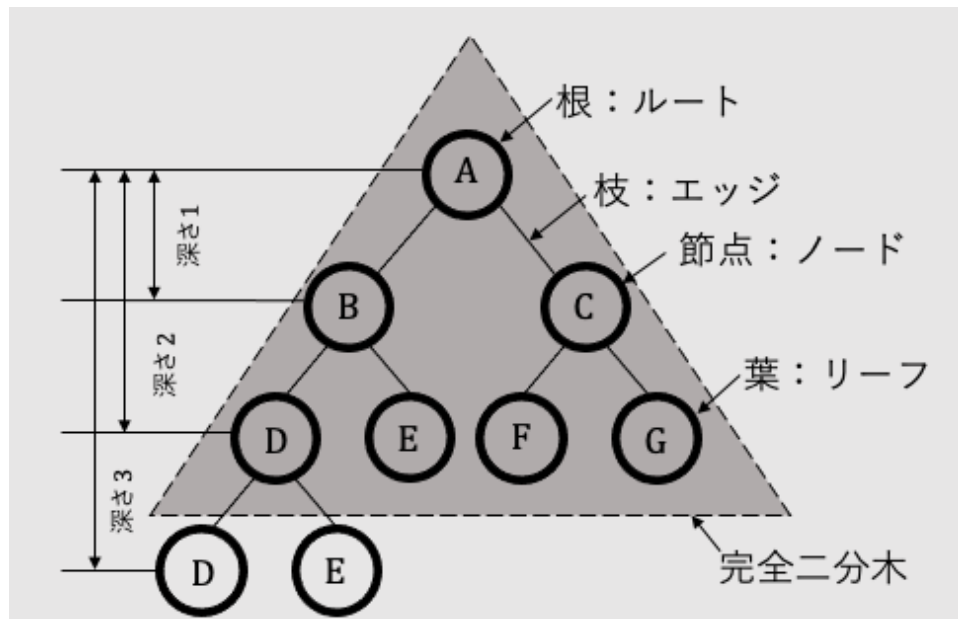


図3：木構造と名称

4. 2 深さ優先探索

深さ優先探索 (DFS / Depth-First Search / 縦型探索) とは、グラフを探索するためのアルゴリズムの一つである。探索を開始する頂点から LIFO (Last-In First-Out) 方式で次に進むノードを決定する。それ以上進めない末端 (木の場合は葉ノード、グラフの場合はすべての隣接ノードが探索済みの場合も含む) まで来たら、経路を遡って最初の未探索ノードへ進む。その先で末端に到達したら、再び経路を遡り…という手順を終了まで繰り返す。深さ優先探索の様子を図4に示す。また、図に示した数字は、探索の順番である。

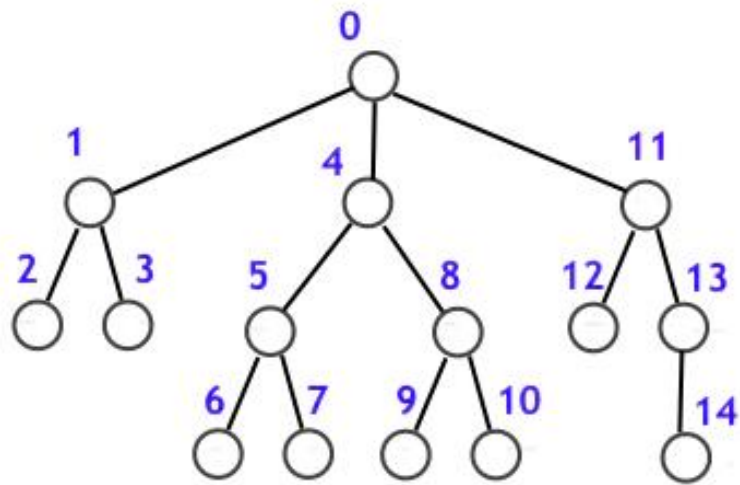


図 4：深さ優先探索の探索順

4. 3 幅優先探索

幅優先探索（BFS / Breadth-First Search / 横型探索）とは，グラフを探索するためのアルゴリズムの一つである．探索を開始する頂点から FIFO（First-In First-Out）方式で次に進むノードを決定する．幅優先探索の様子を図 4 に示す．また，図に示した数字は，探索の順番である．

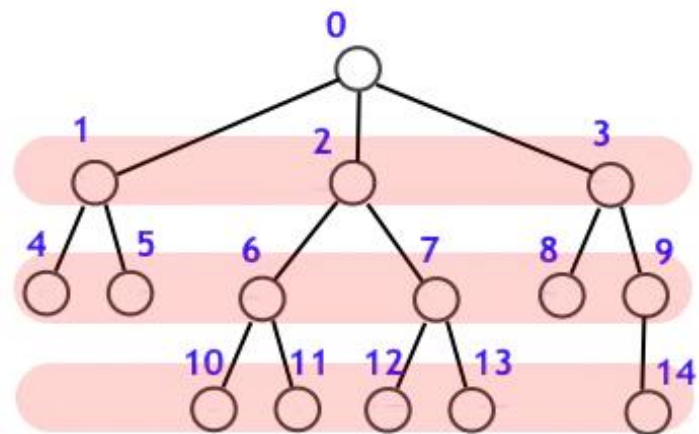


図 5：幅優先探索の探索順

4. 4 ダイクストラ法

ダイクストラ法 (Dijkstra's Algorithm) は 1959 年エドガー・ダイクストラによって考察された最短経路問題を求めるアルゴリズムである。ノードとエッジで構成されたグラフで、ある 2 点間の距離を最短にする経路の候補をあげ、各ノードに距離またはコストなどの値を与え、その和が最小になるよう順次更新して経路を選択する。ダイクストラ法はグラフ理論における辺の重みが非負数の場合の単一始点最短経路問題を解くことができるため、ナビゲーションの経路探索や交通問題の経路案内においてよく利用されている。

ノード集合を V 、エッジ集合を E とすると、グラフ G は、 $G = (V, E)$ と表せる。エッジの始点ノードを i 、終点ノードを j とおく。 $j \in V$ の各頂点に対して i, j を結ぶ辺があるとき、その辺を (i, j) 、エッジの重みを w とする。

ダイクストラ法は非常にシンプルで効率的な計算である。ダイクストラ法の手順は以下のとおりである。

Step 1 : 始点と初期値の設定

始点 d_i を 0 とし、その以外のノードまでの距離を全て無限大にする。

$$\text{if } \{i, i\}, d = 0$$

$$\text{if } \{i, j\} \notin E, d = \infty$$

Step 2 : 最短経路の確定

始点から、まだ辿っていないノード集合の中から距離が最も小さいノードを選び、その距離を始点から選択したノードまでの最短距離として確定する。

Step 3 : 一部の最短経路の更新

最短距離が確定したノード集合と隣接する頂点について、最短距離の候補 d_i をまとめる。候補の中から、最も d_i が小さいノードを最短経路が確定したノード集合に加える。 $d_i + w_{ij} < d_j$ ならば、 d_j の値に $d_i + w_{ij}$ を代入する。

Step 4 : 繰り返す

終了条件：未確定ノードの集合が空集合であること。

終了条件を判定し、条件を満たさない場合は、Step 2 から Step 4 の処理を繰り返す。ダイクストラ法の計算量は $O(V^2)$ である。

4. 5 A* (A-star) 法

A* (A-star) 法 (A*Search Algorithm, A*) とは、ダイクストラ法のやり方を基に、各ノードにコストを設け、推定値を計算することで最短経路を発見することのできるアルゴリズムである。経路をノードで表現し、開始地点から目標地点までの経路を計算する。その計算した経路が最短であるのかを保証するアルゴリズムであり、始点から終点までの間の障害物等を避けた経路も算出が可能である。

説明の例題として、以下のような 5×5 の正方形グリッドの格子地図を用意する。色が塗られた地点は通過できない障害物とし、S が始点、G が終点である。格子地図は図 6 に示す。

y \ x	0	1	2	3	4
0					
1		S			
2					
3					
4					G

図 6：例題 (A*法) の格子地図

step1：ノードデータを設定する。

x, y 座標の設定と各ノードの Open, Close 等のステータス (初期は None), 実コスト (始点から移動するための実際の距離), 推定コスト (終点まで移動コストがどのくらいかかるのか予測した距離), スコア, 経路算出に用いる親ノードポインタを定義する。

Step2：始点地点のパラメータを求める。

実コストを C , 推定コストを H , スコアを S とする。

始点地点の実コストは開始地点なので、以下ようになる。

$$C_{(1,1)} = 0$$

始点地点の座標を S_x, S_y , 終点地点の座標を G_x, G_y , 推定コストの x, y は d_x, d_y とする. 移動方向は8方向となるため, 推定コストは以下のようなになる.

$$d_x = G_x - S_x = 4 - 1 = 3, \quad d_y = G_y - S_y = 4 - 1 = 3$$

$$H_{(1,1)} = \sqrt{d_x^2 + d_y^2} = 3\sqrt{2}$$

スコアの値は, 実コストと推定コストの合計で計算できる.

$$S_{(1,1)} = H_{(1,1)} + C_{(1,1)} = 0 + 3\sqrt{2} = 3\sqrt{2}$$

親ノードポインタは開始地点なので null となる. 上記計算結果を図 7 に示す.

$y \backslash x$	0	1	2	3	4
0					
1		$C=0$ $H=3\sqrt{2}$ $S=3\sqrt{2}$			
2					
3					
4					G

図 7 : step2 (A*法) の説明図

Step3 : 探索開始

Step2 の計算式を用いて基準ノードである始点地点の周辺コストを設定する.

$\left\{ \begin{array}{l} \text{始点ノード(1,1)の周りを探索} \\ \text{スコアの値が最小のノードを基準ノードに変更} \\ \text{基準ノードが決まったら, 確定ノード集合に移動} \end{array} \right.$
 を繰り返す.

図 8 では, 確定ノードを青色, 基準ノードを緑色で表現している.

$y \backslash x$	0	1	2	3	4
0	C=1 H= $4\sqrt{2}$ S= $1+4\sqrt{2}$	C=1 H=5 S=6	C=1 H= $2\sqrt{5}$ S= $1+2\sqrt{5}$		
1	C=1 H=5 S=6	C=0 H= $3\sqrt{2}$ S= $3\sqrt{2}$			
2	C=1 H= $2\sqrt{5}$ S= $1+2\sqrt{5}$	C=1 H= $\sqrt{13}$ S= $1+\sqrt{13}$			
3					
4					G

図 8 : step3 (A*法) の説明図

Step4 : 最短経路の算出

Step3 で算出した確定ノードから最短経路を導出する.

親ノードが始点ノードになるまで終点ノードから親ノードを辿る.

図 9 では, 経路をオレンジ色で表現している.

$y \backslash x$	0	1	2	3	4
0	C=1 H= $4\sqrt{2}$ S= $1+4\sqrt{2}$	C=1 H=5 S=6	C=1 H= $2\sqrt{5}$ S= $1+2\sqrt{5}$	C=2 H= $\sqrt{17}$ S= $2+\sqrt{17}$	C=3 H=4 S=7
1	C=1 H=5 S=6	C=0 H= $3\sqrt{2}$ S= $3\sqrt{2}$		C=2 H= $\sqrt{10}$ S= $2+\sqrt{10}$	C=3 H=3 S=6
2	C=1 H= $2\sqrt{5}$ S= $1+2\sqrt{5}$	C=1 H= $\sqrt{13}$ S= $1+\sqrt{13}$		C=3 H= $\sqrt{5}$ S= $3+\sqrt{5}$	C=3 H=2 S=5
3	C=2 H= $\sqrt{17}$ S= $2+\sqrt{17}$			C=4 H= $\sqrt{2}$ S= $4+\sqrt{2}$	
4					G

図 9 : step4 (A*法) の最終算出結果

4. 6 双方向探索

双方向探索 (Bidirectional search) とは、グラフ探索アルゴリズムの一種で、同時に 2 つの方向から探索を行う。一方は始点から終点に向かい探索し、もう一方は終点から始点に向かい探索して、両方の探索で交わったノードのうち、最小の総コストとなるノードが確定した場合、探索を終了とする。双方向探索は、同手法の単方向探索と比較して、探索範囲を狭め、計算量を少なくし、経路の算出時間を短縮する効果が見込まれる。図 10 と図 11 で単方向探索と双方向探索の様子をそれぞれ表した。

ノード数を b 、最短経路を解くまでの木構造の深さを d とすると、単方向探索の計算量は $O(b^d)$ である、必要となる探索が深くなればなるほど、必要なメモリ量が指数的に増加する。双方向探索の場合、始点からの探索が必要となる探索計算量は $O(b^{\frac{d}{2}})$ となり、両方合わせて合計の計算量は $O(b^{\frac{d}{2}} + b^{\frac{d}{2}})$ となる。

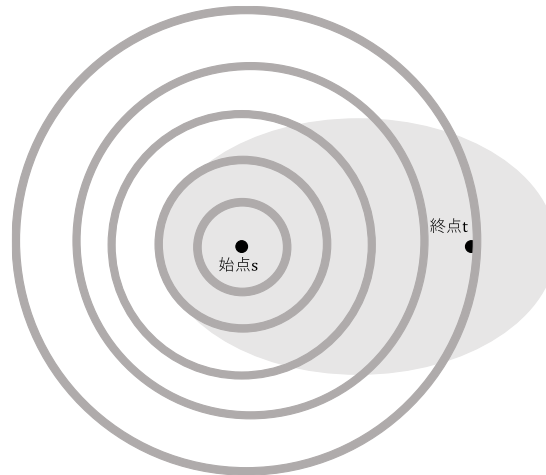


図 10：単方向探索の手順

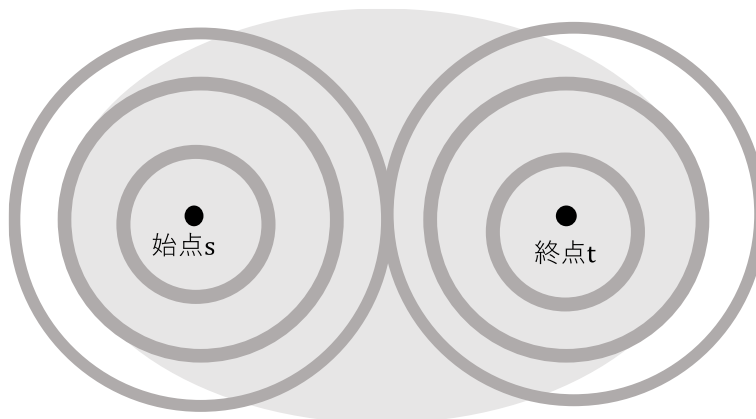


図 11：双方向探索の手順

4. 6. 1 双方向ダイクストラ法

双方向ダイクストラ法とは、ダイクストラ法をもとに、始点だけではなく終点からもダイクストラ法による探索を行うアルゴリズムである。

ダイクストラ法は、始点を中心に周囲を探索するため、終点に到達するまでに余分なノードを探索しなければならないといったデメリットがある。しかし、双方向ダイクストラ法は始点と終点から探索を行うことで、比較的効率よく探索を行うことができる。

4. 6. 2 双方向 A* 法

双方向 A*法とは、A*法をもとに、始点だけではなく終点からも A*法による探索を行うアルゴリズムである。

説明の例題として、以下のような 5×5 の正方形グリッドの地形を用意する。グレー色が塗られた地点は通過できない障害物とし、S が始点、G が終点である。例題の地形図を図 12 に示す。

y \ x	0	1	2	3	4
0					
1		S			
2					
3					
4					G

図 12：例題（双方向 A*法）の地形図

Step 1: ノードデータを設定

初期設定は単方向 A*法による探索と同様にノードデータを設定する。

Step 2: 始点ノードと終点ノードのパラメータを計算

始点ノード S のパラメータの計算は、単方向 A * 法と同様に以下のように計算される。

$$C_{(1,1)} = 0$$

$$H_{(1,1)} = 3\sqrt{2}$$

$$S_{(1,1)} = 3\sqrt{2}$$

終点ノード G のパラメータの計算は、終点ノードを始点ノードと見なし、始点ノードを終点ノードと見なして考えられる。逆方向での探索では、終点ノード G は開始地点なので、パラメータは以下になる。

$$C_{(4,4)} = 0$$

$$H_{(4,4)} = 3\sqrt{2}$$

$$S_{(4,4)} = 3\sqrt{2}$$

上記の流れは以下図 13 に示す。

y \ x	0	1	2	3	4
0					
1		C=0 H=3√2 S=3√2			
2					
3					
4					C=0 H=3√2 S=3√2

図 13：step 2（双方向 A*法）の説明図

Step 3：両地点から探索を開始

Step2 の計算式を用いて、基準ノードである始点ノード(1,1)と終点ノード(4,4)に隣接したノードのスコアをそれぞれ計算する。

{ 二つの基準ノードに隣接したノードのスコアを同時に計算
 両方のスコアの値が最小のノードを基準ノードに変更 を繰り返す。
 基準ノードが決まったら、確定ノード集合に移動

上記の流れを図 14 に示す。

$y \setminus x$	0	1	2	3	4
0	C=1 H= $4\sqrt{2}$ S= $1+4\sqrt{2}$	C=1 H=5 S=6	C=1 H= $2\sqrt{5}$ S= $1+2\sqrt{5}$		
1	C=1 H=5 S=6	C=0 H= $3\sqrt{2}$ S= $3\sqrt{2}$			
2	C=1 H= $2\sqrt{5}$ S= $1+2\sqrt{5}$	C=1 H= $\sqrt{13}$ S= $1+\sqrt{13}$			
3				C=1 H= $2\sqrt{2}$ S= $1+2\sqrt{2}$	
4				C=1 H= $\sqrt{13}$ S= $1+\sqrt{13}$	C=0 H= $3\sqrt{2}$ S= $3\sqrt{2}$

図 14 : Step 3 (双方向 A*法) の説明図

Step 4 : 最短経路の算出

Step3 で算出した確定ノードから最短経路を導出する。親ノードが始点ノードになるまで終点ノードから親ノードを辿る。図 15 では、経路をオレンジ色で表現している。

$y \setminus x$	0	1	2	3	4
0	C=1 H= $4\sqrt{2}$ S= $1+4\sqrt{2}$	C=1 H=5 S=6	C=1 H= $2\sqrt{5}$ S= $1+2\sqrt{5}$	C=2 H= $\sqrt{17}$ S= $2+\sqrt{17}$	
1	C=1 H=5 S=6	C=0 H= $3\sqrt{2}$ S= $3\sqrt{2}$		C=2 H=2 S=4	C=3 H=3 S=6
2	C=1 H= $2\sqrt{5}$ S= $1+2\sqrt{5}$	C=1 H= $\sqrt{13}$ S= $1+\sqrt{13}$		C=2 H= $\sqrt{5}$ S= $2+\sqrt{5}$	C=2 H= $\sqrt{10}$ S= $2+\sqrt{10}$
3	C=2 H= $\sqrt{17}$ S= $2+\sqrt{17}$			C=1 H= $2\sqrt{2}$ S= $1+2\sqrt{2}$	
4			C=2 H= $\sqrt{10}$ S= $2+\sqrt{10}$	C=1 H= $\sqrt{13}$ S= $1+\sqrt{13}$	C=0 H= $3\sqrt{2}$ S= $3\sqrt{2}$

図 15 : Step 4 (双方向 A*法) の説明図

第 5 章 駐車経路シミュレーション

5. 1 前提条件

前提条件を以下で述べる.

- ・ノード設定

- (i) 駐車場構造は非負の重み付き無向グラフで表し, 重みは隣接ノードの距離とする.
- (ii) 始点ノード (始点地点) と終点ノード (駐車スペース) は指定されている.

- ・ノード探索条件

- (i) 駐車場の範囲外のノードは探索不可とする.
- (ii) 障害物として設定されているノードは探索不可とする.
- (iii) 車体の動きも考慮して, 障害物に隣接しているノードも探索不可とする.

- ・探索終了条件

- (i) 単方向探索
終点ノードと探索済みノードが一致した場合, 探索を終了とする.
- (ii) 双方向探索
始点からの探索と終点からの探索で交わったノードのうち, 最小の総コストとなるノードが確定した場合, 探索を終了とする.

- ・ヒューリスティック関数

A*法と双方向 A*法のヒューリスティック関数は, 各ノードから終点までのユークリッド距離とする.

5. 2 シミュレーション設定

5. 2. 1 駐車場構造

本実験で想定した駐車場は, 大型ショッピングセンターや大型テーマパークの立体駐車場である. 110×110(m)の駐車場を 2 つ連結させ, 一部構造上迂回しなければならないルートを作り, 複雑な駐車場構造を再現する. 駐車場マップを図 16 に示す.

駐車スペース数: 1148, 探索可能ノード数: 8619

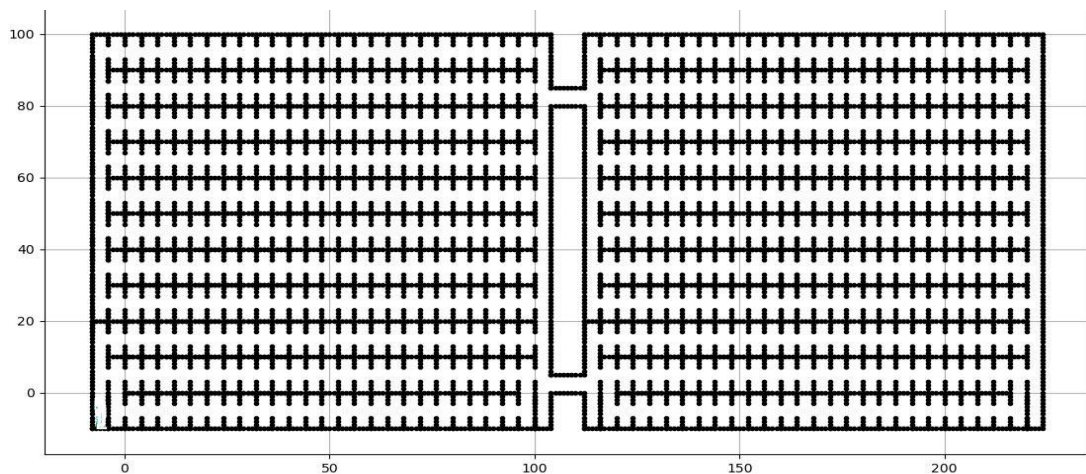


図 16：駐車場マップ

図 16 の駐車場マップを説明するため、簡易的にしたものを図 17 に示す。

始点ノードを左下の青点，終点ノードをもう一方の青点で表現する．黒点は障害物である．外枠の黒点は駐車場の範囲を表している．

終点ノードは駐車スペースにだけ存在する．駐車スペースは一台の自動車を駐車するために必要なスペースとし，図 17 の駐車スペースを拡大したものを図 18 に示す．

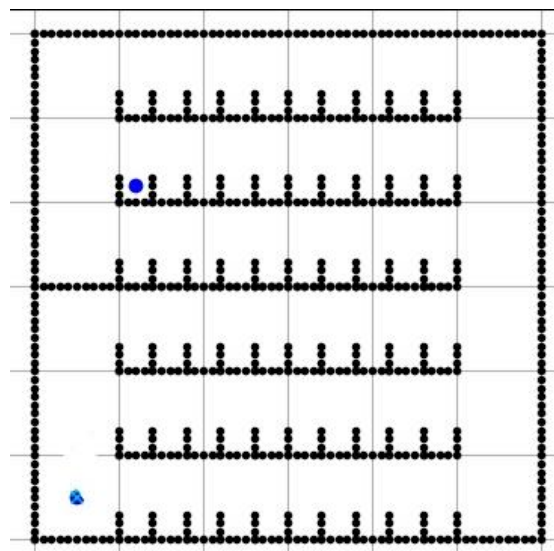


図 17：簡易駐車場マップ

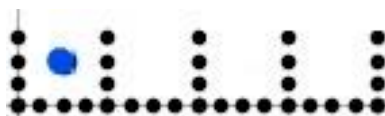


図 18：駐車スペース

5. 2. 2 ノード探索

表1で示したモーションモデルに従ってノードを探索する。モーションモデルとは、要素数3の配列が8つ組み合わさった二次元配列であり、配列[0]–[7]によって、8方向に対して探索（図19）を行うことができる。探索にあたって基準となるノードを基準ノードとすると、添え字[0]の要素は、基準ノードから x 軸方向に平行移動する値とする。添え字[1]の要素は、基準ノードから y 軸方向に平行移動する値とする。添え字[2]の要素は、始点ノードからのコスト（距離）とする。例えば、配列[0]のとき、現在のノードから x 軸方向に1平行移動させたノードを探索する。また、配列[7]のときは現在のノードから x 軸方向に1、 y 軸方向に1平行移動させたノードを探索する。なお、縦横移動の移動コストは1、斜め移動の移動コストはユークリッド距離を用いて $\sqrt{2}$ とする。

表1：モーションモデルにおける移動距離

	[0]	[1]	[2]
配列	($x, 0$)進む	($0, y$)進む	コスト(距離)
[0]	1	0	1
[1]	0	1	1
[2]	-1	0	1
[3]	0	-1	1
[4]	-1	-1	$\sqrt{2}$
[5]	-1	1	$\sqrt{2}$
[6]	1	-1	$\sqrt{2}$
[7]	1	1	$\sqrt{2}$

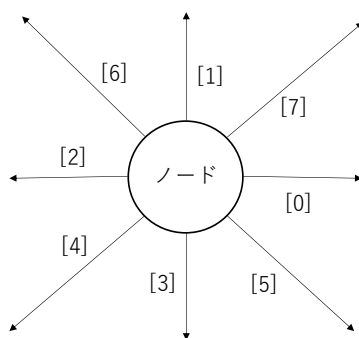


図19：モーションモデルにおける探索方向

ノード探索の流れ図を図 20 に示す。図 20 の挙動を説明する。まず、モーションモデルの配列[0]を用いて探索を行う。探索ノード（ x 座標, y 座標, 累計コスト, 親ノードが格納されたもの）を t とし、探索ノードのノード番号を n とする。 t がノード探索条件を満たすとき次の条件分岐に移る。close リストに n が存在しないとき、つまり n が未確定ノードであるとき次の条件分岐に移る。open リストに n が存在しないとき、 t を open リスト[n]に代入する。open リストに n が存在するとき、open リスト[n]のコストが t のコストと等しいまたは大きいとき、 t を open リスト[n]に代入する。この作業によって累計コストを最小のものに更新していく。これらの流れを配列[7]まで繰り返し行う。

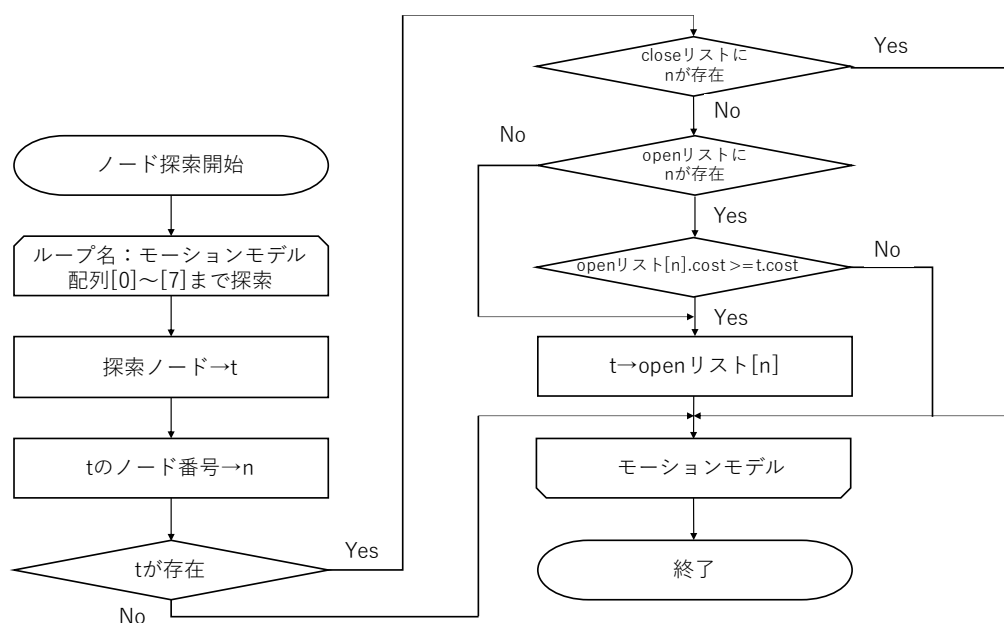


図 20：ノード探索の流れ図

5. 2. 3 ノード管理

始点からの最短経路が確定していないノードのリストを open リスト，確定したノード（探索済みノード）のリストを close リストとし，辞書型でデータを管理する．

open リスト，close リストの key 値にはノード番号，value 値には x 座標， y 座標，コスト（距離），親ノード番号が格納される．ノード管理の流れ図を図 21 に示す．

図 21 の挙動を説明する．はじめに open リストと close リストともに空の状態にする．始点ノードを open リストに代入し，始点ノードからノード探索を開始する．ノード探索は上記の 5. 2. 2 で述べたので省略する．探索したノードを open リストに格納する．open リストから各経路探索手法によって，確定ノード n を選定する．確定ノードの選定方法については，5. 2. 4 で述べる．確定ノード n を close リストに格納する．探索の終了条件を満たすまで，探索を繰り返し行う．

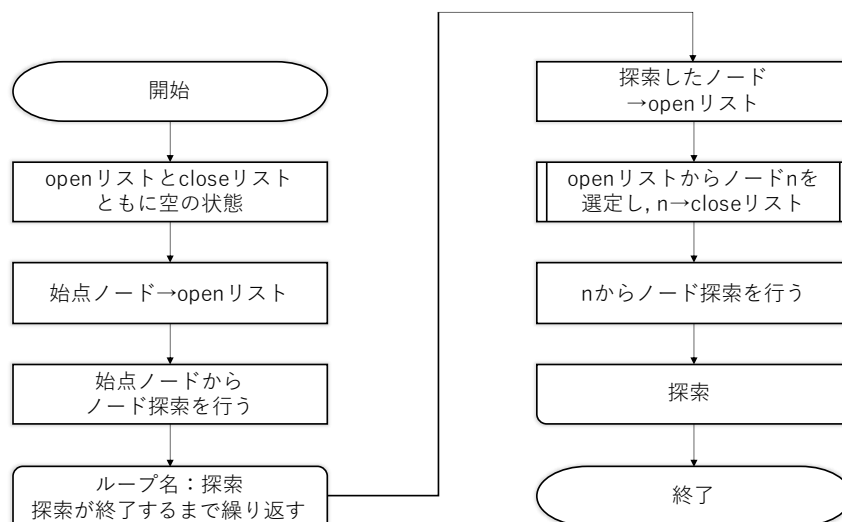


図 21：ノード管理の流れ図

5. 2. 4 確定ノード導出

それぞれの手法の確定ノードの導出方法を示す.

深さ優先探索: open リストの末尾に格納されているノードを確定ノードとする.

幅優先探索: open リストの先頭に格納されているノードを確定ノードとする.

(双方向) ダイクストラ法: open リストにある累計コストが最小であるノードを確定ノードとする.

(双方向) A*法: 累計コストとヒューリスティック値の合計値が最小であるノードを確定ノードとする.

5. 2. 5 最短経路導出

close リスト (始点からの最短経路が確定しているノードが格納されているリスト) を用いて, 終点ノードから親ノードを順に辿っていくことで最短経路を導出する. この導出の流れ図を図 22 に示す.

図 22 の挙動を説明する. はじめに終点ノードを path リストに格納する. 次に終点ノードのノード番号を n に代入する. close リストにある n の親ノードを path リストの末尾に入れる. このノードのノード番号を n に代入する. これらの流れを n の親ノードと始点ノードが一致するまで行う. 最後に, path リストの末尾から順に要素を表示することで, 最短経路を導出する.

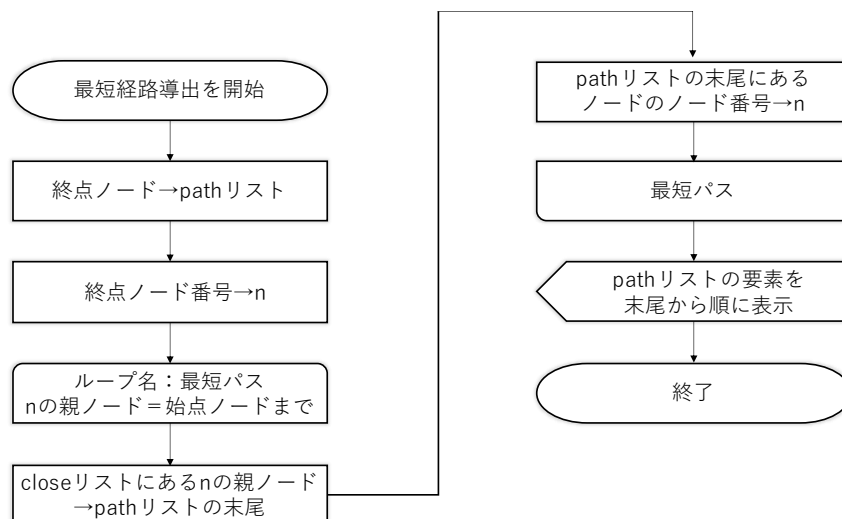


図 22 : 最短経路導出の流れ図

5. 3 実験環境

シミュレーション実験の環境は表 2 のとおりである.

表 2 : 実験環境

OS	Windows10 Home 64-bit
プロセッサ	Intel® Core™ i3-8145U CPU @ 2.10GHz
実装 RAM	8.00GB
使用言語	Python 3.8.5 64-bit

第 6 章 評価実験

6. 1 評価方法

本研究では各探索手法の比較において、最短経路の算出に要した計算時間の長短で性能を評価することとする。駐車場内で距離を分散させ選択した 10 地点を終点ノードに設定し、その終点までの最短経路の算出に要した各々の探索手法の計算時間を計測する。その結果から、各探索手法の計算時間を比較し、性能の評価をする。また、探索の様子をアニメーション化し、見つかった最短経路と探索済みノードの可視化を行う。

6. 2 探索結果

各手法の探索の様子を可視化する。以下の図 23 では、終点ノード (150, 78) の際の経路探索を可視化した図である。

深さ優先探索法は探索が行き詰まるまでひたすらに探索し、探索した中で経路を見つけるため、設定する終点ノードによって計算時間にばらつきが生まれた。

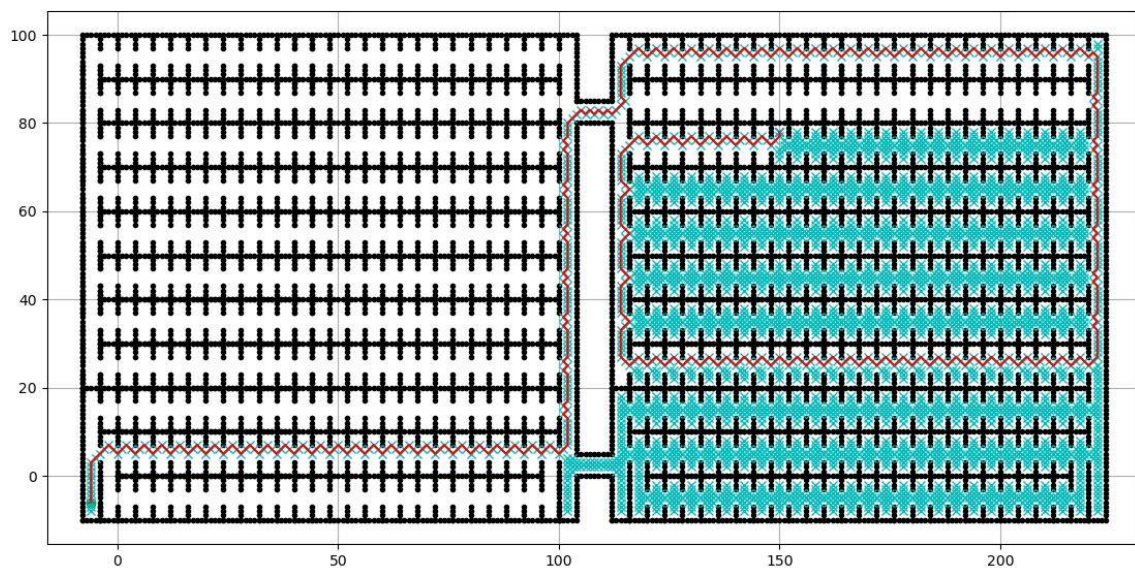


図 23：深さ優先探索の可視化

幅優先探索とダイクストラ法による探索では、全体的に似たような形状となった。終点ノード(150, 78)の計算では、誤差ではあるもののダイクストラ法による探索の方がわずかに探索済みノード数は少なかった。また、探索の流れは水が流れるように広がっていくような特徴があった。図 24 と図 25 に示す。

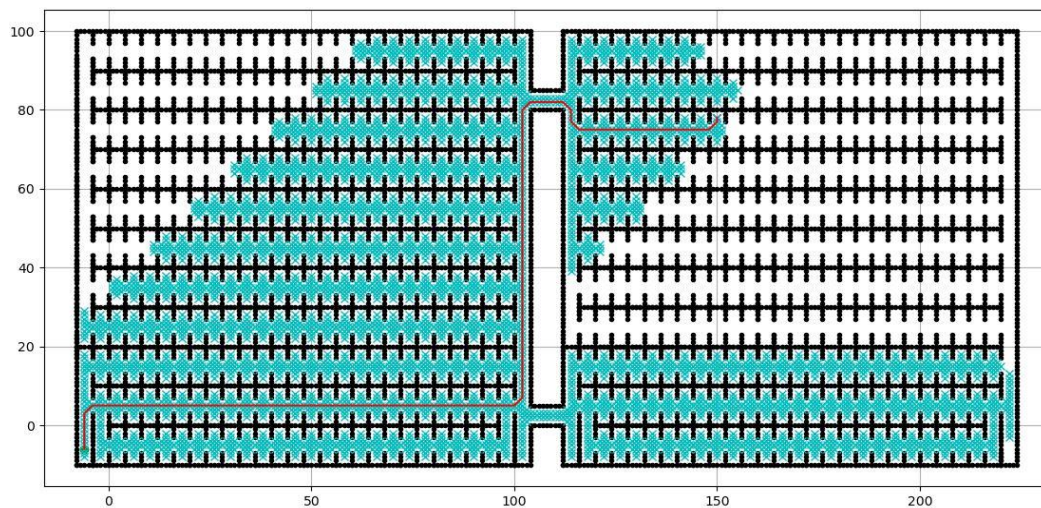


図 24：幅優先探索の可視化

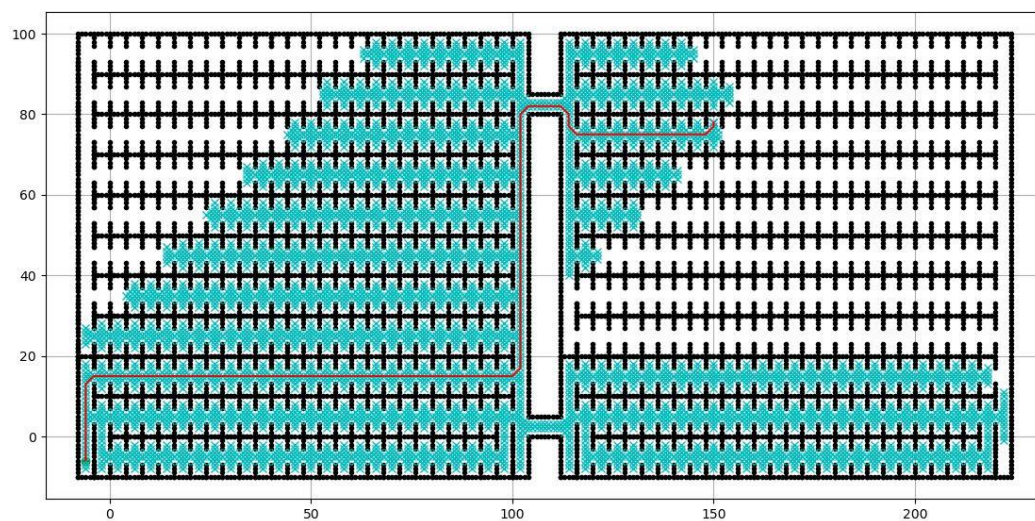


図 25：ダイクストラ法による探索の可視化

A*法による探索では，探索済みノード数は他探索法と比べて最も少なく，最適な最短経路を算出することができた．また，探索の流れは，終点に向かって一直線に探索が進んでいくような特徴があった．図 26 に示す．

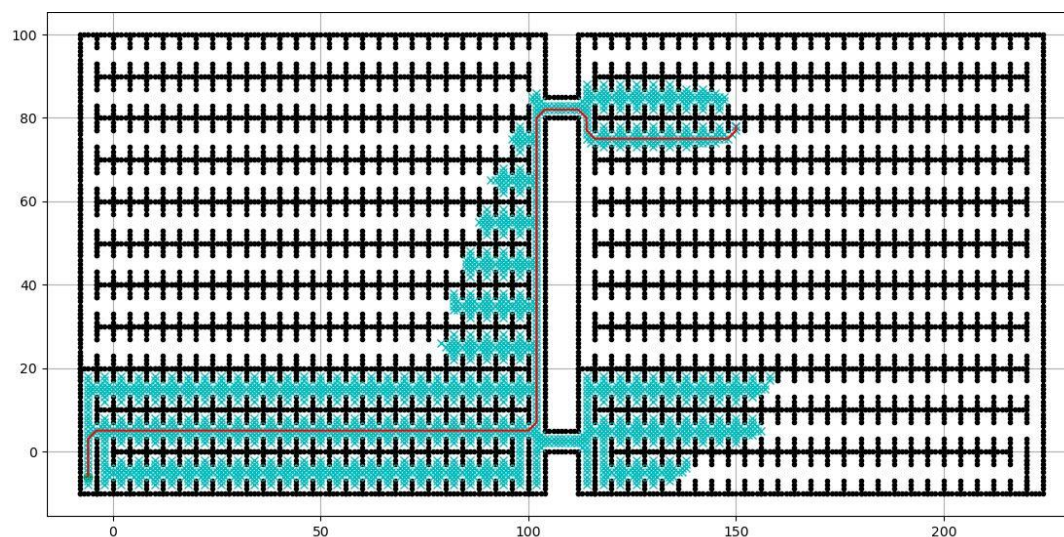


図 26： A*法による探索の可視化

双方向 A*法による探索では，探索済みノード数は単方向 A*法とあまり違いは見られないが，探索箇所の違いは見られた．図 27 に示す．

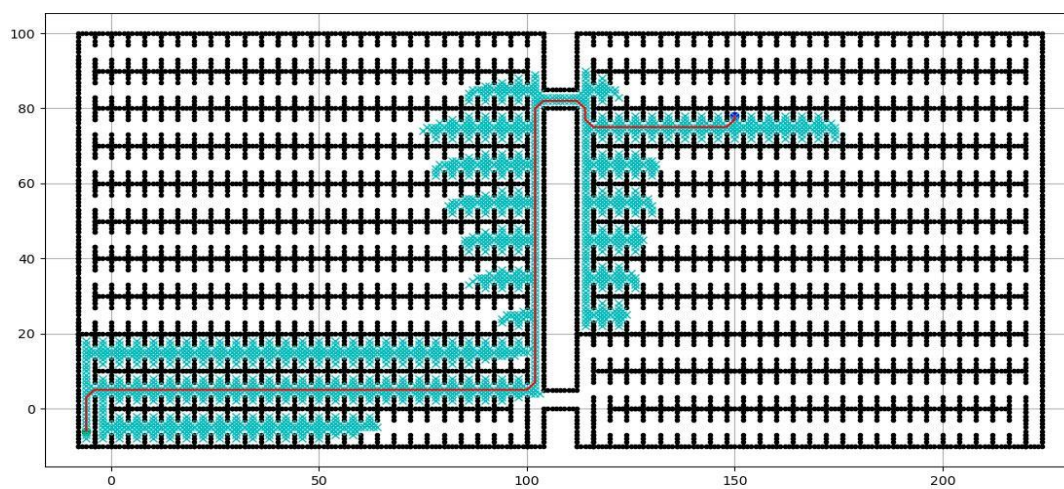


図 27： 双方向 A*法による探索の可視化

双方向ダイクストラ法による探索では、探索済みノード数は単方向ダイクストラ法と違いは見られなかったが、探索箇所の違いは見られた。図 28 に示す。

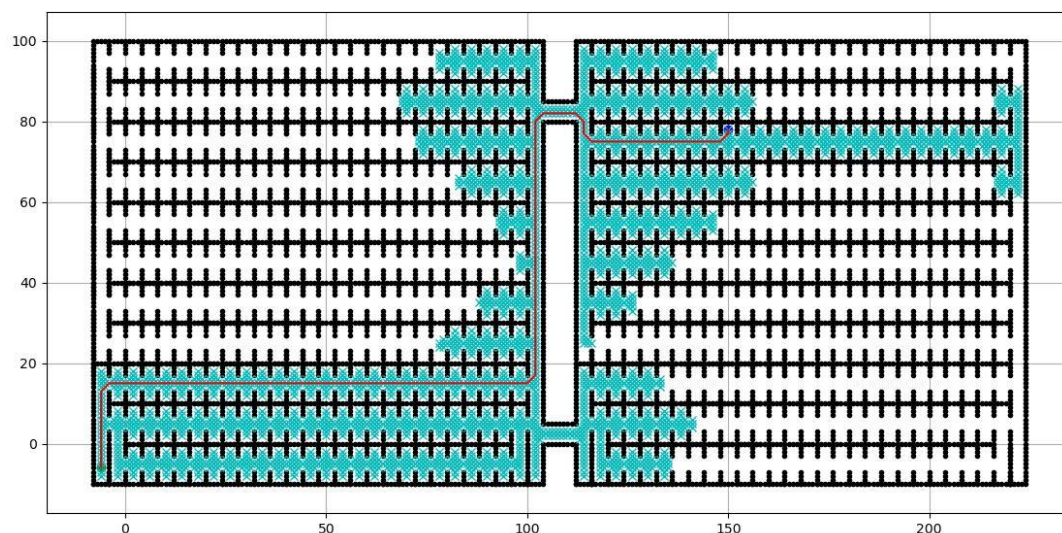


図 28：双方向ダイクストラ法による探索の可視化

図 29 から、4 つの方法の中で 10 試行中 10 回とも A*法による探索が最も計算時間が短かった。また、ダイクストラ法、幅優先探索の計算時間は、誤差はあるが、ほぼ同じ計算時間となった。深さ優先探索は、設定した終点地点によって計算時間にばらつきが見られた。以上の結果から、4 手法の中で A*法が最も有用であるといえる。

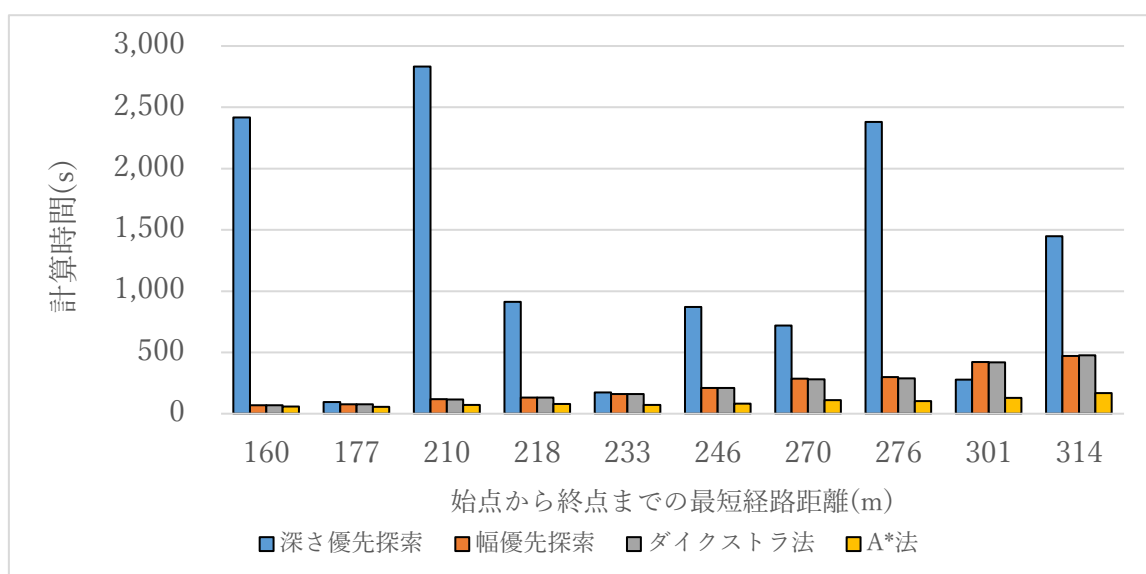


図 29：単方向探索 4 種類の計算時間比較

6. 3 比較 1

ここでは、単方向探索で最も計算時間が早かった A*法と本研究の趣旨である双方向 A*法を比較する。以下の図では、左から最短距離が短い順で並べてある。図 30 に示す。

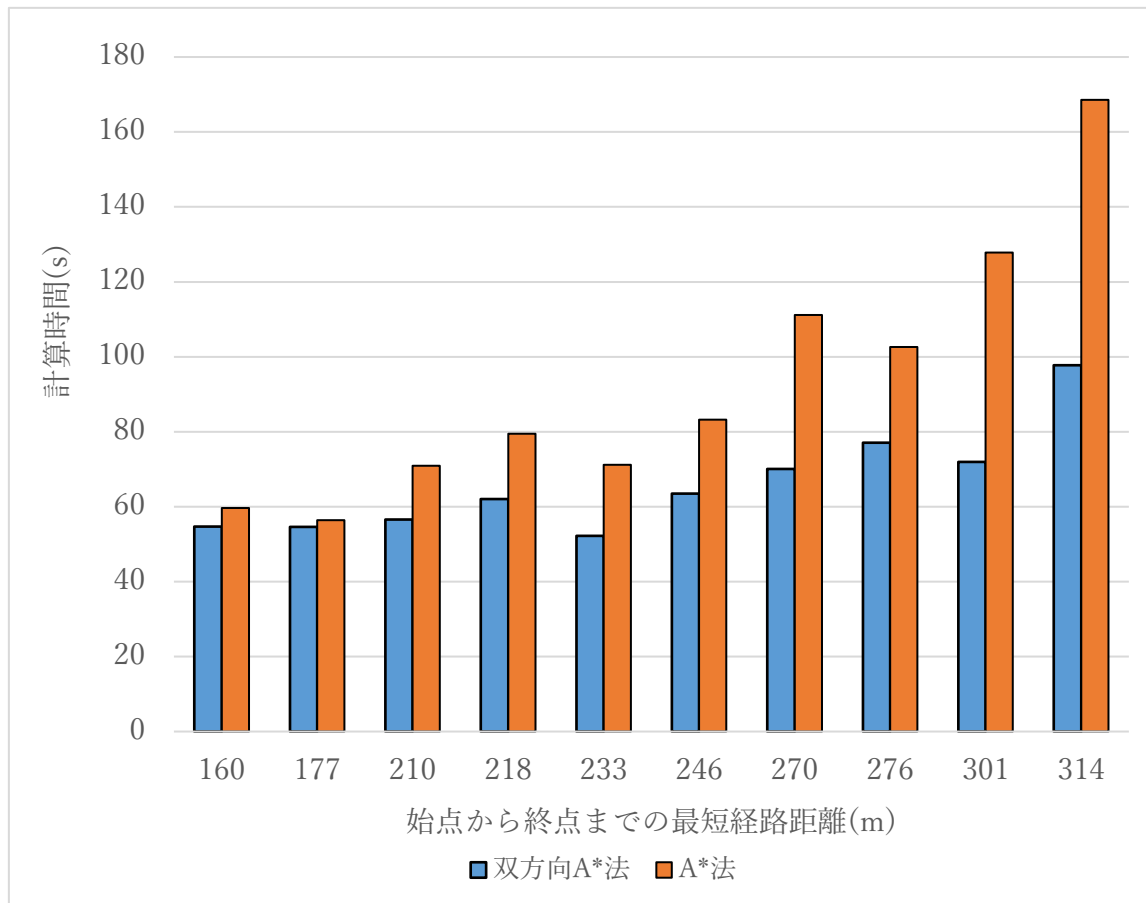


図 30：双方向 A*法と単方向 A*法の計算時間の比較

図 30 を見てわかるように、10 地点中 10 地点すべてで双方向 A*法による探索の方が計算時間は短い。そのため、単方向の 4 つの手法よりも双方向 A*法の方が計算時間の観点で優れているといえる。なお、始点地点から終点地点までの移動経路が長い地点では、通常の A*法と双方向 A*法の計算時間の差は大きくなる傾向がある。

6. 4 比較2

次に、一般的に双方向探索として用いられることの多い双方向ダイクストラ法を採用し、双方向ダイクストラ法で探索した結果と双方向 A*法とを比較する。こちらにおいても、左から最短経路が短い順に並べている。図 31 に示す。

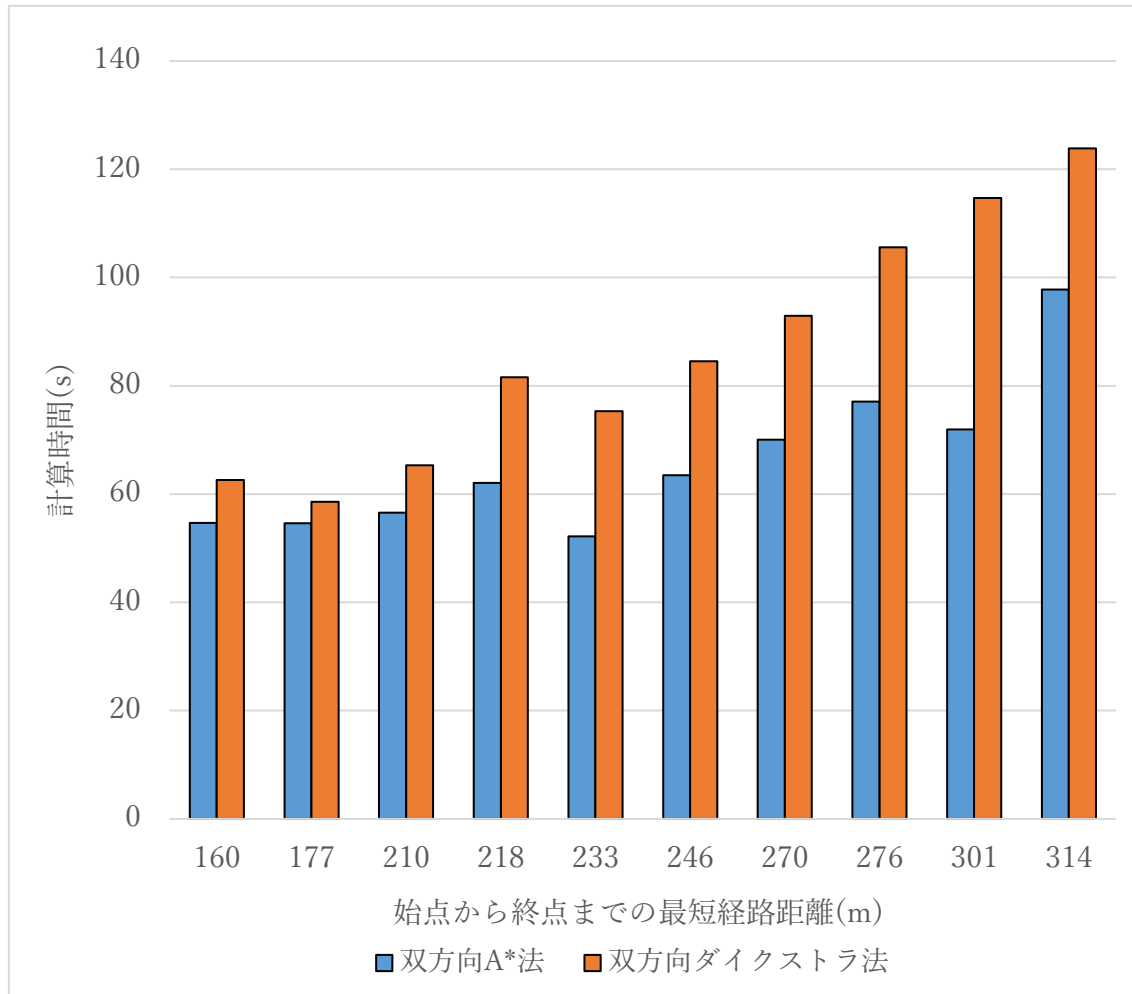


図 31：双方向 A*法と双方向ダイクストラ法の計算時間の比較

10 地点中 10 地点すべてで双方向 A*法による探索の方が計算時間は短い。そのため、双方向ダイクストラ法よりも双方向 A*法の方が計算時間の観点で優れているといえる。なお、始点地点から終点地点までの移動経路が長い地点では、双方向ダイクストラ法と双方向 A*法の計算時間の差は大きくなる傾向がある。

第7章 結論

本研究では、駐車場での渋滞を引き起こしている原因が、新しく入場したドライバーが、どのスペースが空いているのか、埋まっているのかを把握できず、既に起きている渋滞に巻き込まれることであると考え、駐車スペースの最適化とその駐車スペースまでの最短経路導出に着目した。駐車スペースの最適化の先行研究として、実際に様々なシステムが導入されており、空車、満車を把握し、出入口前に可視化するシステムやスマートフォンで遠方からでも空き状況を確認できるシステムの導入により混雑を未然に防ぐなど様々な対策措置が取られている。しかし、そのように駐車スペースの最適化に触れている研究は多いものの、駐車スペースまでの最短経路探索が少ないため、駐車スペースまでの最短経路を算出し、混雑することなく行くことのできる最適ルートを提案することで無駄な駐車場周回を防ぎ、駐車場内の混雑を緩和できると考えた。

上記から、駐車場内の渋滞問題を解決することを目的として、駐車場ナビゲーションの一部機能である、駐車スペースまでの最短経路探索に取り組んだ。私たちは、最短経路探索において双方向 A*法が有用であると仮説を立て、仮説を検証するために、ダイクストラ法、A*法、深さ優先探索、幅優先探索といった 4 種類の単方向探索との比較実験と、双方向探索として双方向ダイクストラ法との比較実験を行った。有用性のある手法とは、今回実験で用いる手法の中で、最短経路の算出時間が最も短い手法を指す。

実験の想定状況は、車両の現在位置から、システムによって指定された駐車スペースまでの最短経路探索である。駐車場構造は大型ショッピングモールなどの大型立体駐車場をモデルとして、110(m)×110(m)の駐車場を二つ連結させたものを用意した。

評価実験では、単方向探索 4 種の中では A*法の計算時間が最も短く、その A*法と双方向 A*法との算出時間を比較したが、双方向 A*法による算出の方が計算時間は短かった。さらに、単に双方向探索自体が優れているという結論を避けるため、同じ双方向探索として双方向ダイクストラ法との比較を行ったところ、やはり双方向 A*法による探索が最短という結果となり、大型駐車場の最短経路探索において双方向 A*法が有用であるといえる。また、探索開始地点から駐車スペースまでの最短経路が長ければ長いほど、双方向 A*法と比較手法の経路算出時間の差が大きいことから、駐車場が混雑して入り口付近に駐車できない状況下や移動経路が複雑になってしまう駐車場のときなどでは、より双方向 A*法が有用であると考えられる。

今回の実験は、始点と終点が指定されていることから、単一頂点对最短経路問題とみなせる。よって、双方向 A*法は単一頂点对最短経路問題においても有用性があると考えられる。

今回の実験結果から、駐車場の最短経路算出を実際にセンシング技術や車車間通信技術等の駐車最適化システムと組み合わせ、実用化することを考えたとき、双方向 A*法を採用し、高速化を図るのが良いと考えている。

参考文献

- [1] Y. Bi, L. Sun, H. Zhu, T. Yan, Z. Luo, “A parking management system based on wireless sensor network,” Proc. ACTA Automat. SINICA, Vol. 32, No. 6, pp. 38-45, November 2003.
- [2] R. Lu, X. Lin, H. Zhu, X. Shen, “SPARK: A New VANET-based Smart Parking Scheme for Large Parking Lots,” Proc. IEEE int. Conf. Comput. Commun. (INFOCOM) 2009, pp. 1413-1421, April 2009.
- [3] 徳田渉, L. Miguel, 金森亮, 伊藤孝行, “車両に対する自動交渉による駐車場割当手法の提案,” 情報処理学会研究報告, Vol. 2015-ICS-179 No.13, March 2015.
- [4] 孫為華, 剣持真弘, 山内由紀子, 柴田直樹, 安本慶一, 伊藤実, “待ち時間を短縮する駐車ナビゲーション,” 情報処理学会研究報告, Vol. 2011-DPS-148 No.7 September 2011.
- [5] 孫為華, 剣持真弘, 柴田直樹, 安本慶一, 伊藤実, “待ち時間を短縮する駐車ナビゲーション,” マルチメディア通信と分散処理ワークショップ論文集 (DPSWS2013), December 2013.
- [6] W. Sun, N. Shibata, M. Kenmotsu, K. Yasumoto, M. Ito, “A Method for Navigating Cars in Multilevel Parking Facility,” J. Inf. Process. Vol.23 No.4, pp.488-496, July 2015.
- [7] 一般財団法人日本自動車研究所, “平成30年度高度な自動走行システムの社会実装に向けた研究開発・実証事業：自動バレーパーキングの実証及び高度な自動走行システムの実現に必要な研究開発,” https://www.data.go.jp/data/dataset/meti_20200217_0512, February 2020. (閲覧日 2021年12月26日)