

卒業論文

オンデマンド実時間配車サービス SAVS における待ち 時間の分散最適化の検討

公立はこだて未来大学
システム情報科学部 複雑系知能学科
知能システムコース 1017116

西川 和真

指導教員 平田 圭二

提出日 2021 年 1 月 26 日

BA Thesis

Consideration on Distributed Optimization of Waiting Time in SAVS

by

Kazuma Nishikawa

School of Systems Information Science, Future University Hakodate
Intelligent Systems Course, Department of Complex and Intelligent Systems

Supervisor: Keiji Hirata

Submitted on January 26, 2021

Abstract— Smart Access Vehicle Service (SAVS), an on-demand real-time vehicle dispatching service, controls public transportation with a vehicle dispatching system that can dispatch vehicles in real time in response to passenger demand and automatically perform optimal route search. Conventional SAVS uses an algorithm called the sequential optimal insertion method for dispatching vehicles. Although the sequential optimal insertion method can minimize the average time to pick up and drop off a passenger at a destination after a demand is generated, the waiting time varies greatly among passengers, and individual differences in service quality may occur. In this study, we improved the sequential optimal insertion method and investigated a vehicle dispatching algorithm that minimizes the variance of waiting time. We compared the variance of the waiting time between the sequential optimal insertion method and the improved algorithm by simulation, and confirmed that the variance of the waiting time of the improved algorithm becomes smaller than that of the existing algorithm by introducing a certain number of vehicles.

Keywords: SAVS, Public Transport Service, Dispatch Algorithm

概要:

オンデマンド実時間配車サービス Smart Access Vehicle Service (SAVS) は、公共交通機関を配車システムによって制御することで、乗客の乗車要求 (デマンド) に対しリアルタイムな配車を行い、最適経路探索を自動で行える。従来の SAVS における配車は逐次最適挿入法と呼ばれるアルゴリズムを用いている。逐次最適挿入法はデマンドが発生してから乗客を目的地に送迎する時間の平均時間を最小化できるが、乗客間で待ち時間が大きくばらつき、サービスの質に個人差が生じる可能性がある。本研究では、逐次最適挿入法の改良を行い、待ち時間の分散を最小化する配車アルゴリズムを検討した。シミュレーションによって逐次最適挿入法と改良アルゴリズムの待ち時間の分散について比較し、その結果から改良アルゴリズムの車両待ち時間のばらつきが、一定以上の車両台数の導入によって既存アルゴリズムの待ち時間のばらつきより小さくなることを確認できた。

キーワード: SAVS, 公共交通サービス, 配車アルゴリズム

目次

第 1 章	序論	1
1.1	背景	1
1.2	オンデマンド実時間配車サービス SAVS	2
1.3	配車システムの課題	2
1.4	研究目標	3
1.5	研究の有用性	3
第 2 章	関連研究	5
2.1	Dial-a-ride 問題における関連研究	5
2.2	逐次最適挿入法	7
第 3 章	提案手法	9
3.1	改良における方針	9
3.2	改良アルゴリズムの説明	10
第 4 章	シミュレーションの環境と構成	12
4.1	開発環境	12
4.2	使用する道路網	12
4.3	プログラム構成	13
4.4	函館市の公共交通利用者における関連研究	13
第 5 章	シミュレーションの実行条件	15
5.1	パラメータ	15
5.2	使用した PC のスペック	16
5.3	デマンド発生時間	16
第 6 章	実験と考察	18
6.1	車両待ち時間の変動係数	19
6.2	車両待ち時間の平均値・標準偏差	21
6.3	車両移動時間の平均値・標準偏差	24

6.4	考察	27
第 7 章	結言	28
7.1	結論	28
7.2	今後の課題	28
参考文献		31

第 1 章

序論

本章では、移動サービスをめぐる近年の社会における動向と、フルデマンド型公共交通サービス Smart Access Vehicle Service (SAVS) の説明を行う。また、SAVS に用いられる配車アルゴリズムについて説明し、本研究における目的を明らかにする。

1.1 背景

1.1.1 日本と世界における自動車社会の現状

現在、日本は既存の自動車社会からの脱却を求められている。理由の一つは、高齢化社会や大都市集中による地域の過疎化である。この問題は世界の国々と比べ早く深刻化している。過疎地域では、都会のように鉄道やバスの本数は充実しておらず、公共交通による移動は利便性が高いとは言えない。これは公共交通の利用者の減少によって、事業者がサービスの採算性を保つことが難しくなり、本数を減らさざるを得ないことが一因している。そのため、過疎地域では移動を自家用車へ頼らざるを得ない。しかし、近年の高齢者ドライバーによる交通事故の多発から、高齢者が自ら運転免許を返納することは珍しくなくなった。公共交通の本数も少なく、運転することも憚られる過疎地域の高齢者は、移手段の乏しさという問題に直面している。この問題を解決するためには、自家用車に頼り切った社会から脱却し、利便性が高い新たな公共交通サービスを生み出すことが求められる。

自家用車に頼った社会からの脱却を実現するため、国土交通省は Mobility as a Service (MaaS) と呼ばれる移動サービスの推進を行っている [1]。MaaS は、社会に存在する多様な移手段を一元的に取り扱い、サービスとして提供する概念である。タクシーや鉄道などの移手段、予約・決済などを統合し、移動の利便性向上や公共交通の有効活用を実現する。MaaS は、2014 年に Sonja Heikkila によって案され [2]、フィンランドの首都ヘルシンキに交通改革をもたらした。その結果、ヘルシンキ行政と民間企業が連携し、2016 年に「Whim」 [3] という移動サービスのアプリが施行された。Whim はエンドユーザー（乗客）に対し、公共交通とレンタカーなどの移動サービスを統合し、その支払いを定額料金制で提供している。このサービスによって、ヘルシンキの公共交通の

利益増，ドアツードアでの移動実現などの利点をもたらした。世界各国は，こうした MaaS の実装による社会への利点を Whim によって確認し，それぞれの国に適した MaaS の開発・実装を進めている。

1.2 オンデマンド実時間配車サービス SAVS

日本における MaaS の一つとして，Smart Access Vehicle Service (SAVS) [4] の実証実験・社会実装が進んでいる。SAVS は，タクシーの利便性とバスの経済性を併せ持つ，デマンド応答型公共交通の一種である。配車アルゴリズムが乗客の乗車要求（デマンド）に対し，需要に即した乗り合い車両の配車決定を行うことができる。SAVS は株式会社未来シェアが中心となって開発を進め，これまで函館において 3 度の実証実験を成功させている [5]。実証実験を経て，群馬県でデイサービスの送迎を行う「福祉 Mover」[6]，岩手県でデマンド型乗合バス「しわまる号」[7] など，実際に公共交通としてのサービスを開始している

表 1.1: 函館市での SAVS 実証実験（文献 [5] の表 1 を基に作成）

回数	年月	期間	時間帯	SAV 台数	備考
1	2013/10	7 日間	7:30~19:00	5	半自動での運用テスト
2	2014/4	1 日間	11:00~18:00	16	完全自動リアルタイム配車に成功
3	2015/5	4 日間	12:00~19:30	20	学会参加者を対象とした実験



図 1.1: リアルタイム便乗配車

1.3 配車システムの課題

SAVS は，逐次最適挿入法と呼ばれる配車アルゴリズムを用いている。このアルゴリズムは，新しいデマンドが発生したとき，まだ達成されていない既存のデマンドに生じる遅延を最小にした，新デマンドへの配車を行うことができる。これにより，デマンド全体の平均的な待ち時間を最小化することができる。しかし，逐次最適挿入法は平均以外の統計量（分散・尖度・歪度）は最適化さ

れていない。そのため、デマンドの発生頻度や車両台数によっては、乗客個々人で待ち時間や移動時間にばらつきが生じる可能性がある。待ち時間や移動時間がばらつくことは、乗客個々人にサービスの質の不公平さをもたらす可能性がある。

以上より、本研究では逐次最適挿入法を改良し、デマンド全体の車両待ち時間の分散を最小化するアルゴリズムを検討する。車両待ち時間や移動時間の分散を最適化することによって、移動サービスの変動性・不安定さを抑えることが可能になる。

1.4 研究目標

本研究の目標は、デマンド全体の車両待ち時間の分散を最小化するアルゴリズムを検討し、乗客個々人の待ち時間のばらつきを小さくすることである。改良によるアルゴリズムの変化を図 1.2 に示す。図 1.2 は 3 名の乗客 (A・B・C) が乗車要求をしたとき、逐次最適挿入法と改良アルゴリズムによってそれぞれ配車された際の待ち時間を示す。逐次最適挿入法がデマンドを処理したとき、乗客 A が 3 分、B が 2 分で済む代わりに、C が 5 分になるデマンド群が存在する。改良アルゴリズムは、例えば乗客 A を 3.2 分、B を 3 分に増やし、C を 3.6 分の待ち時間に抑えるような配車を行い、デマンド全体の待ち時間の分散を最小化する。アルゴリズムを改良することによって、サービスの分配方法を変容させるということである。

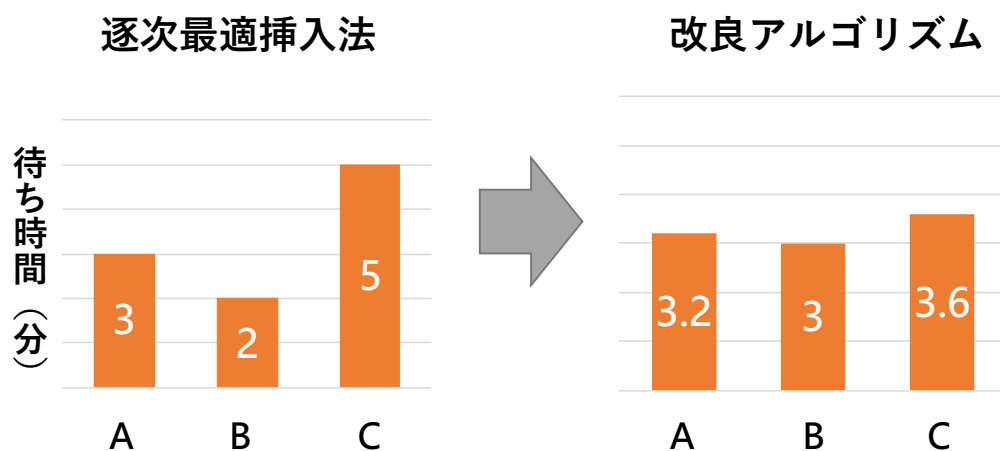


図 1.2: 改良によるサービスの変化

1.5 研究の有用性

アルゴリズムを改良しサービスの分配方法を変えることによって、乗客全体の待ち時間を少し大きくする代わりに、本来長く待つ必要があった乗客の待ち時間を減らすことができる。これにより、乗客個々人の待ち時間の不公平さを可能な限り抑えることができる。特に通勤通学などの利用者と利用する時刻がある程度固定されている状況で活かされる。発生するデマンドが固定され、逐

次最適挿入法の配車のみを行い続けることで、他の乗客より長く待つことが常態化している乗客が存在することが考えられる。この状況が長く続けば、乗客個々人の受け取るサービスの差は徐々に広がる。そこで、改良アルゴリズムは乗客個々人の待ち時間の分散を最小にし、サービスを可能な限り平等に分配する。これにより乗客個々人の受け取るサービスの差を解消し、サービスにおける最小不幸を小さくできる。

サービスを平等に分配することによって、利用者は車両の待ち時間の目安をつけやすくなる。例えば、食事宅配サービスは食事を届ける際の鮮度を計算しやすくなるため、SAVSの移動サービスを利用して食事を届けることができる。

また、SAVSを用いて移動サービスをデザインする際、デザインするサービスの自由度を向上させることができる。先述のように、改良アルゴリズムは逐次最適挿入法とは異なるサービスの分配を行う。利用できるアルゴリズムが増えることによって、導入地域の特性に合わせた配車アルゴリズムの選択を行うことができる。逐次最適挿入法によるSAVS導入をベースとしてデータを収集し、個々人に待ち時間の差が顕著に現れる場合は改良アルゴリズムを用いる、といった使い分けができるようになり、サービスの自由度を向上させることができる。

このような移動サービスを提供するため、車両待ち時間の分散を最適化できる配車アルゴリズムの開発を行う。具体的には、新たなデマンドが発生したとき、過去に配車した際の待ち時間との分散が最小となる車両と走行ルートを選べるよう、逐次最適挿入法を改良する。

有効性の検証方法として、交通流シミュレータを用い、改良アルゴリズムが逐次最適挿入法より車両待ち時間のばらつきを小さくできるか評価する。また、それに伴い移動時間や便乗率に変化があったかどうかを示し、改良アルゴリズムの利点と課題点を明らかにする。

第 2 章

関連研究

一般的に、複数の目的地と目的地間の距離が与えられ、各地点を巡回する最適な経路を求める問題は「巡回セールスマン問題」[8]と呼ばれ、組み合わせ最適化問題の一種とされる。これは、目的地を最短で巡る最適解を求めるとき、計算量が指数関数的に増加する問題である。配車で最適解を求めようとすると、配車計算に膨大な時間がかかり、サービスとして成立しない。そのため、効率よく最適解を求められる、もしくは最適解に近い準最適解を求められるアルゴリズムの研究が進められている。特に、デマンドバスのように乗客同士の便乗を考慮し、順序関係（乗車は降車の前に行わなければならない）などの制約を設け、乗客送迎の最適経路を求める問題を「Dial-a-ride 問題」と呼ぶ[9]。本章では、Dial-a-ride 問題における関連研究と、SAVS の配車アルゴリズムである逐次最適挿入法の特徴を説明する。

2.1 Dial-a-ride 問題における関連研究

Dial-a-ride 問題における配車計算の方法は、挿入法 (Insert-based method) と呼ばれるアルゴリズムが主に用いられている [10]。挿入法とは、すでに配車計算が完了した既存デマンドの乗車・降車位置の記号列に、まだ配車が行われていないデマンドの乗車・降車位置の記号を様々なパターンで繰り返し挿入することで、実行可能なルートを構築する方法である [11]。挿入法の例を図 2.1 に示す。乗車・降車位置の記号列を本研究では経由地点リストと呼称する。 O_1 はデマンド 1 における乗車地点を示し、 D_1 はデマンド 1 における降車地点を示す。新デマンドの乗車地点を O_k 、降車地点を D_k とする。新デマンドが発生したとき、 O_k と D_k を経由地点リストへそれぞれ挿入し、ヒューリスティックによって設定した目的関数によって評価をし、最適な配車パターンを探索する。挿入法の利点は、高速であること、最適解に近い準最適解を安定して求められること、実装が簡単な点が挙げられる [11]。

挿入法の以外の配車計算の方法として、内村ら (2002) はデマンドバスの配車システムをリアルタイムで実行するために、遺伝的アルゴリズムを巡回路探索に採用し、乗車効率の向上について検討した [12]。遺伝的アルゴリズムとは、まず個体 (解の候補) をランダムに複数用意し、評価関数により各個体の評価値を計算する。そして評価値の低い個体を淘汰、高い個体同士で組み合わせを

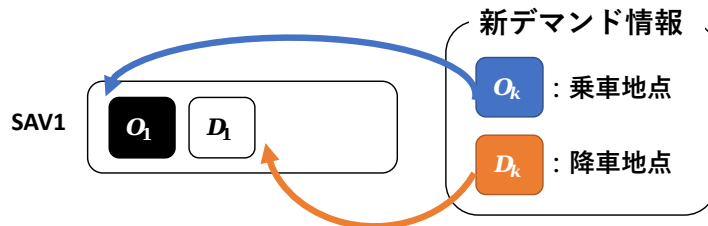


図 2.1: 挿入法の例

変化させ、新たな個体を生成する。これを複数回繰り返す行うことで、評価値の高い解を探索する手法である。内村らは、遺伝的アルゴリズムにおける個体を、乗車地点と降車地点の記号列とし、列自体を1つの解として評価した。図 2.2 に経由地点リストの例を示し、遺伝的アルゴリズムによる巡回路探索を説明する。最初に、経由地点リストをランダムに $P_1 \sim P_n$ まで生成し、これを個体として評価する。評価の次に、評価値が低い経由地点リストを候補から外す（淘汰）。この例では P_6 , P_8 をはじめとした経由地点リストが残っている。残った経由地点リスト同士をランダムに組み合わせ、リスト同士の記号列を部分的に入れ替えさせる（交叉）。この例では P_6 と P_8 の末尾3つの記号を互いに交叉させ、新たな世代 C_6 と C_8 が生成される。そして再生成された経由地点リストを再度評価する。この手順を繰り返すことによって、最終的に乗車効率の高い巡回路が探索できるとされる。

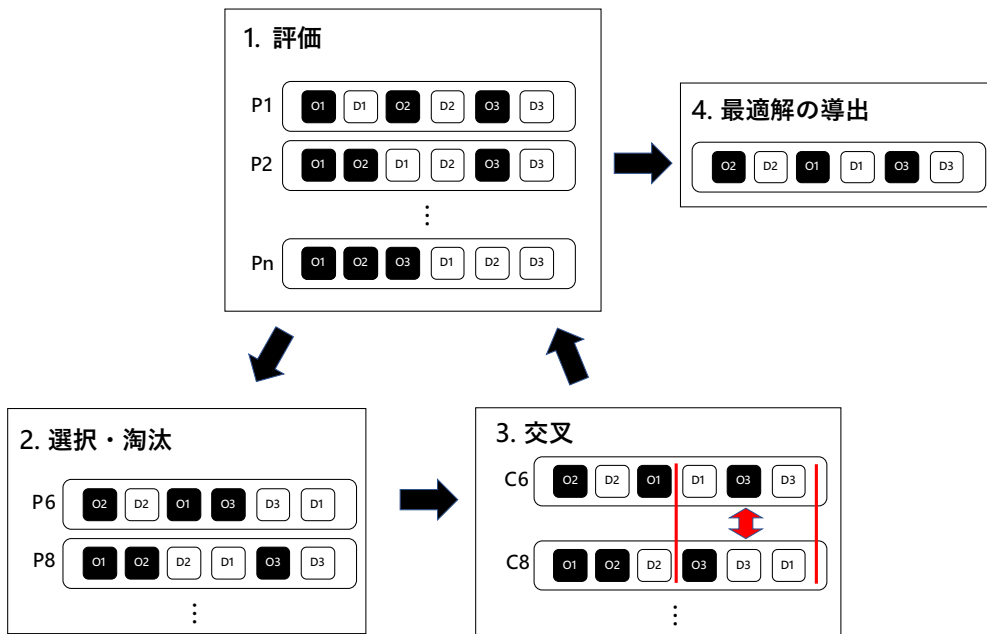


図 2.2: 遺伝的アルゴリズムによる巡回路探索

2.2 逐次最適挿入法

野田ら（2008）は，SAVS の利便性を調査するため，固定路線バスとデマンドバスの比較を行った [13]．その結果，デマンドバスに大きな需要があれば，採算性を維持しつつ固定路線バスより高い利便性を提供できることを示唆した．この研究ではデマンドバスの配車アルゴリズムとして，逐次最適挿入法が用いられている．逐次最適挿入法は，先述した挿入法の一つであり，高速な計算速度，安定して準最適解を求められるといった挿入法の特徴がある．目的関数は，「新デマンドを挿入した場合の，既存デマンドへ生じる遅延」であり，これが最小となる車両・挿入順になるよう配車を行う．逐次最適挿入法を図 2.3 に示し，野田らの先行研究 [13] から一部引用して説明する．

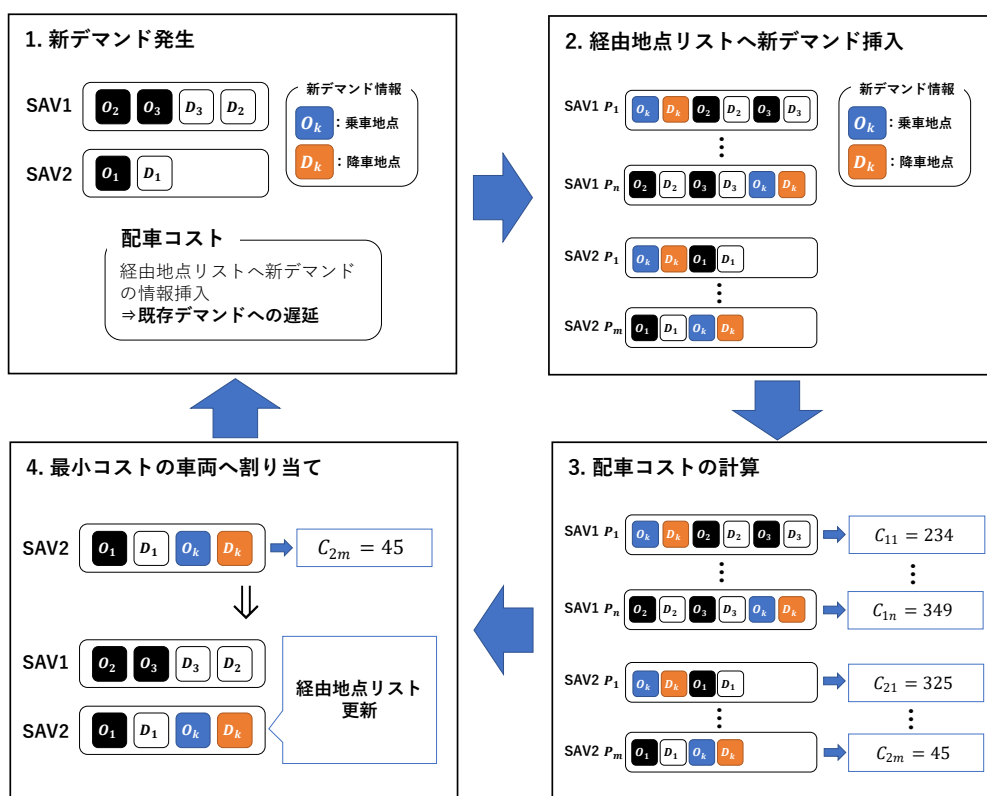


図 2.3: 逐次最適挿入法の概略図

- 1). 新デマンドが発生したとき，各車両の経路地点リストを取得する．このときの新デマンドの乗車地点を O_k ，降車地点を D_k とする．
- 2). 各車両は新デマンドの乗車・降車地点を，全ての挿入パターンで経路地点リストに挿入する．この例では i 番目の挿入パターンを P_i とし，SAV1 の挿入パターンは $P_1 \sim P_n$ の n 個，SAV2 は $P_1 \sim P_m$ の m 個存在する．
- 3). 全挿入パターンにおける，配車コストを求める．車両 1 の i 番目の配車コストを C_{1i} とし，SAV1 の配車コストは $C_{11} \sim C_{1n}$ の n 個，SAV2 は $C_{21} \sim C_{2m}$ の m 個存在する．配車コス

トは新デマンドを挿入したことによる、既存のデマンドへの「遅延時間」、新デマンドの「達成予定時刻」の二つの総和とする。ただし、挿入によって既存または新しいデマンドの締切時刻を超過してしまう場合は、その挿入は配車候補から除外する。

- 4). 全ての車両・挿入パターンの中で、最もコストが小さいものを求め、その車両に新デマンドを割り当てるよう経由地点リストを更新する。

第 3 章

提案手法

本章では，車両待ち時間の分散を最適化するための提案手法として，逐次最適挿入法を改良した配車アルゴリズムの概要を述べる．改良アルゴリズムは，配車コストを既存デマンドと新デマンドの待ち時間の分散とし，これが最小となる車両・デマンドの挿入順で配車を行う．配車コストを計算する際，過去に配車された待ち時間を参照する必要があるため，システム全体ですべての車両に渡って待ち時間を記録するリストを新たに作成した．

3.1 改良における方針

改良アルゴリズムは，逐次最適挿入法を改良して検討を行った．逐次最適挿入法を改良する利点として，以下の 2 点が挙げられる．

- 1). 最小化問題として応用が可能である．
- 2). 挿入法の利点を引き継ぐことができる．

1 について，逐次最適挿入法の配車コストは，新デマンドを挿入した車両・挿入順を説明変数とした目的変数と捉えることができる．この目的変数が最小となる説明変数の組み合わせ（車両・挿入順）を最終的に選び，配車を行う．よって，目的変数が「最小化したい数値」になっていれば，コスト計算方法を変更することで様々な最小化問題に応用することができる．改良アルゴリズムの目的は，デマンド全体の待ち時間の分散を小さくすることである．つまり，改良アルゴリズムにおいて新デマンドが発生したとき，最小化したい数値は，新デマンドの待ち時間と既存デマンドの待ち時間の分散である．よって，改良アルゴリズムは，目的変数（配車コスト）を「既存デマンドと新デマンドの待ち時間の分散」へ変更する．

2 について，逐次最適挿入法は挿入法の種類であることは前節にて説明した．挿入法の利点である計算速度の速さ，安定した準最適解導出はリアルタイム配車に不可欠である．よって，乗車地点と降車地点の記号列に基づき配車を行う，挿入法の基本的なアルゴリズムを引き継ぐこととした．

1 と 2 から，挿入法の基本的なアルゴリズムの流れは変えず，配車計算の方法を変更する，という改良の方針を取ることにした．

3.2 改良アルゴリズムの説明

以下に逐次最適挿入法への主な改良点を示す。

- 1). 新たに、過去に配車したデマンドの待ち時間を記録していく、待ち時間リストを設定した。
- 2). 配車コストは、新デマンドの待ち時間を待ち時間リストに追加したときの、リスト全体の分散とした。

1は、まず既存デマンドとの分散を計算するためには、過去に配車したデマンドにおける待ち時間を参照する必要がある。そのため、改良アルゴリズムによる配車を行う前に、待ち時間リスト \mathbf{W} を作成する。詳細図を図 3.1 に示す。この待ち時間リスト \mathbf{W} は、すでに配車が完了したデマンドの待ち時間を記録し、新デマンドの配車コストを計算するために用いられる。

2については、デマンドが発生した際、新デマンドの車両待ち時間をリストに追加し、リスト全体の分散を配車コストとする。新デマンドを配車する際、この配車コストが最も小さい車両・デマンド情報の挿入パターンで配車を行う。これにより、新デマンドの待ち時間と既存デマンドの待ち時間の分散が最小になる配車が行えると推測される。

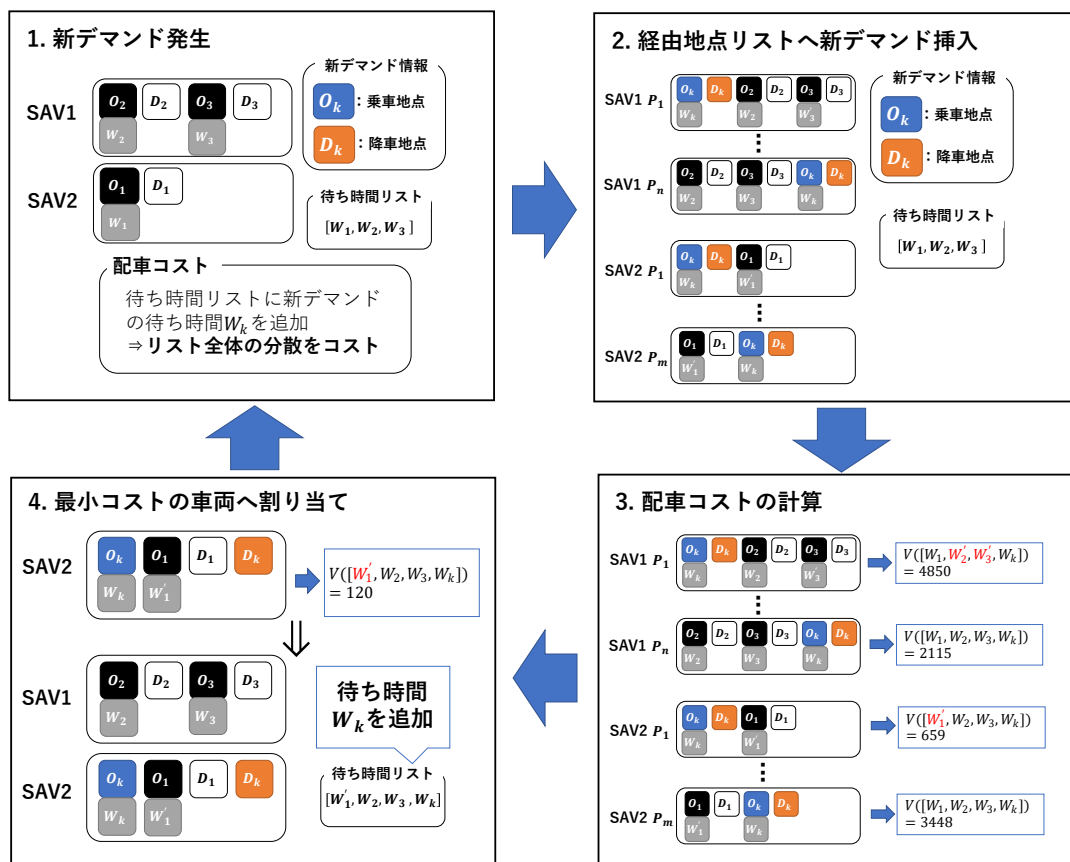


図 3.1: 改良アルゴリズムの詳細図

- 1). 新デマンドが発生したとき、各車両の経由地点リストと待ち時間リストを取得する。このときの新デマンドの乗車地点を O_k 、降車地点を D_k とする。待ち時間リストにおける W_1 はデマンド 1 における待ち時間を示す。
- 2). 各車両は新デマンドの乗車・降車地点を、全ての挿入パターンで経由地点リストに挿入する。この例では i 番目の挿入パターンを P_i とし、SAV1 の挿入パターンは $P_1 \sim P_n$ の n 個、SAV2 は $P_1 \sim P_m$ の m 個存在する。新デマンドが挿入されたことにより、既存デマンドの待ち時間が変動する可能性がある。デマンド i の待ち時間が変動したとき、変動した待ち時間を W'_i とする。
- 3). 全挿入パターンにおける、配車コストを求める。配車コストの算出は、まず挿入によって変動した既存デマンドの待ち時間 W'_i を待ち時間リストに反映し、新デマンドにおける車両待ち時間 W_k を待ち時間リストに追加する。そしてリスト全体の分散を算出し、これを配車コストとする。ただし、挿入によって既存または新しいデマンドの締切時刻を超過してしまう場合は、その挿入は配車候補から除外する。
- 4). 全ての車両・挿入パターンの中で、最もコストが小さいものを求め、その車両に新デマンドを割り当てるよう経由地点リストを更新する。この時、新デマンドの挿入によって他のデマンドの待ち時間が変動した場合、その変動分を待ち時間リストに反映する。

以上の改良アルゴリズムによって、新デマンドの待ち時間と既存デマンドの待ち時間の分散が最小になる配車が行えると推測する。また、配車計算の方法を変更したことで、逐次最適挿入法のデマンド全体の平均所要時間を最小にするという特徴を失い、平均所要時間が長くなることも推測できる。

第 4 章

シミュレーションの環境と構成

本章では、開発環境、使用する道路網やシミュレーションのプログラム構成について説明する。また、デマンド発生頻度を設定するため、函館の公共交通利用者における関連研究を述べる。

4.1 開発環境

本研究では、交通流シミュレータとして Simulation of Urban Mobility (SUMO) を用いる [14]。SUMO は、ドイツ航空宇宙センター (DLR) が中心となって開発した、オープンソースの交通流シミュレータである。SUMO が提供する通信ツールである Traffic Control Interface (TraCI) を用いることで、実行中のシミュレーションにアクセスでき、車両の情報を取得し、その挙動を操作することができる。TraCI を用いることでリアルタイムによる配車計算の結果を、即時にシミュレーション内に適用することが可能である。また、先行研究 [15] においてコミュニティが活発な点やメンテナンスが十分になされていることが評価されている。これらの点から、交通流シミュレータとして SUMO を用いることにした。

4.2 使用する道路網

シミュレーションに用いる道路を図 4.1 に示す。道路網の構築は、SUMO に標準搭載されている道路網の自動生成機能で行った。道路の大きさは 5 km 四方の正方形であり、100 m 間隔で網の目状の道路網が構築されている。左側通行であり、右側と左側は共に一車線である。車両待機場所は、デマンドが割り当てられていない車両が待機、または割り当てデマンドが空になった車両が回送する地点である。なお、回送途中にデマンドが割り当てられる場合があり、その場合は回送を途中で中止し乗車地点へ向かう。

このような簡易的な道路をシミュレーションに用いる理由は、シミュレーション内の状況把握が容易である点と、地理的な条件を排除する点である。本研究は逐次最適挿入法と改良アルゴリズムの配車結果による、移動時間の基本的な差異を分析する。地理的条件を最初から設定してしまうと、アルゴリズムの特性を十分に反映しない恐れがあるため、地理的条件を排除した。

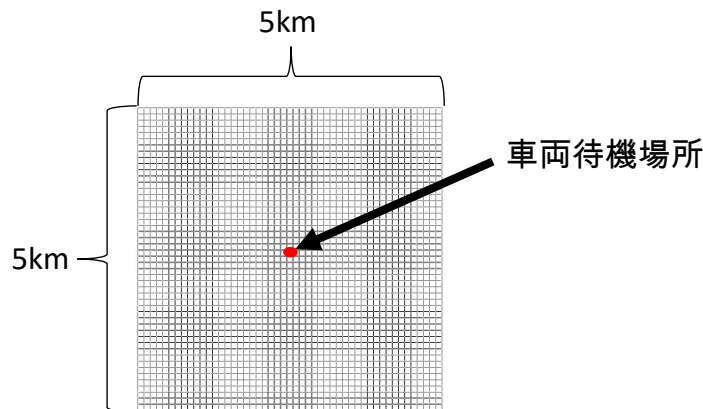


図 4.1: 使用する道路網

4.3 プログラム構成

プログラム構成は、広域地方に向けた SAVS シミュレーションを行った岩田 [16] のプログラム構成を参考に行っている。SUMO による車両シミュレーションと、配車計算を行う外部プログラムの情報を、TraCI を用いて相互通信させている。

SUMO はシミュレーションにおける動的な役割を担う。車両を加速度や減速度などに従い走行させ、外部プログラムからの指示による車両の走行ルートの変更を反映させる。また、車両の現在地や時刻情報などを外部プログラムへ出力する。

外部プログラムは静的な役割を担い、Python によって記述されている。デマンドが発生する時刻や乗車位置が記録されたファイルを読み取り、SUMO 内の時刻と同時刻になったとき、デマンド情報を配車計算プログラムへ出力する。配車計算プログラムは受け取ったデマンド情報と現在走行している車両の既存デマンドの情報から、配車アルゴリズムによるコスト計算を行い、SUMO への走行ルート変更の指示や csv ファイルへ実行結果の出力を行う。

4.4 函館市の公共交通利用者における関連研究

公共交通シミュレーションを行う上で、利用者人数の予測は重要な要素である。利用者人数の予測を実世界に即したものにするために、平田ら [17] の関連研究における公共交通利用者の予測結果を参考にする。この論文は、SAVS の実用実現性を検討するために行った、フィジビリティ実験の取り組みについてまとめている。その中で、全域人流モデルを構築した。このモデルは、まず公共交通のサービスレベル（所要時間や費用）を設定し、最尤推定法により市民が交通手段を選択する際の選択確率を推定し構築した。そして函館市における公共交通利用者数の予測を行った。函館市の曜日別公共交通利用者数を図 4.2a に、時間帯別公共交通利用者数の予測結果を図 4.2b に、論文中から引用し示す。本研究では、利用者数の予測をこの論文の予測結果を参考にし、デマンド発

生頻度の設定を行う。利用者数が最も多い時間帯は14時台の250人である。この利用者数をデマンド発生頻度が最も高いものとして扱い、デマンド発生頻度を設定する。

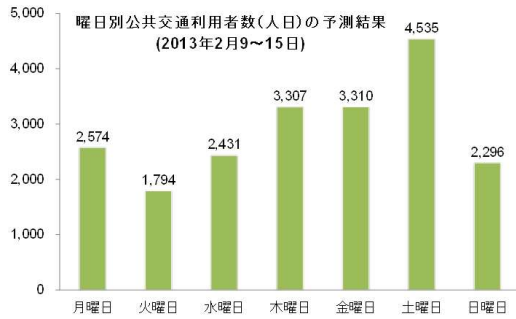


図4 曜日別公共交通利用者数の予測結果

(a) 曜日別公共交通利用者数

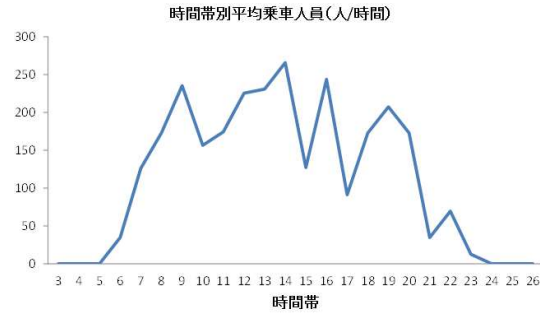


図5 時間帯別公共交通利用者数の予測結果

(b) 時間帯別公共交通利用者

図 4.2: 公共交通利用者数の予測結果 (文献 [17] の図 4 より引用)

第 5 章

シミュレーションの実行条件

本章では、シミュレーションにおいて設定した実行条件とパラメータの値について述べる。また、デマンド発生をランダム性の強い事象にするために行った 2 つの手法について説明する。

5.1 パラメータ

シミュレーションで用いたパラメータとその値について表 5.1 に示す。

表 5.1: シミュレーションにおけるパラメータ

パラメータ	パラメータの値
シミュレーション時間	18000s
デマンド発生頻度	1/135, 270, 405, 540, 675[回/s]
車両台数	5, 10, 15, 20 台
車両の速度	30[km/h]
最大搭乗人数	8 名
最低移動距離	4000m
乗降に要する時間	60s

5.1.1 デマンド発生頻度と発生位置のスケール変換

本研究では、函館市民の公共交通による移動手段が、SAVS へ移り変わった状況を想定する。そのため、デマンド発生頻度の設定は、関連研究において言及した公共交通利用者の予測結果のうち、時間帯別公共交通利用者の予測結果（図 5）を用いて行った。この予測結果は、函館市内のデータを予測したものであり、予測規模の面積は約 15km^2 である。図 5 から、利用者が一番多い時間帯は 14 時台であり、約 250 人が利用していることが分かる。これは、平均で 14.4 秒につき 1 人が公共交通を利用していることになる。この結果は計算の簡略化のため、15 秒につき 1 人の利

用とする。これを図 7 に示した 5km^2 の道路網に面積比でスケールダウンすることによって、デマンド発生頻度を設定した。 15km^2 と 5km^2 の面積比は 1 : 9 となる。例として、15 秒を 9 倍することによって 135 秒となるため、 5km^2 内で 15km^2 内の利用者を反映すると、135 秒に 1 人利用者が発生するということになる。また、デマンドの乗車・降車地点の分布は一様分布となっており、2 地点間は最低 4km の移動距離を持つことを条件とした。

5.2 使用した PC のスペック

使用 PC のスペックを表 5.2 に示す

表 5.2: PC スペック

スペック	詳細
メーカー	GTUNE
PC 名	NEXTGEAR-NOTE (NG-N-i7931PA1)
CPU	2.6GHz, 6 コア
メモリ容量	32GB
OS	Windows10 Home 64bit

シミュレーション 1 回あたりに要する計算時間は車両台数、デマンド発生頻度により変動し、平均して 20~30 分を要した。

5.3 デマンド発生 の 時間

デマンドの発生頻度については、前節のように設定を行ったが、実世界のデマンドは、いつ起こるかわからないランダム性の強い事象である。このランダム性を表現するために、デマンド発生期間を指数分布で、一定期間の平均デマンド発生回数をポアソン分布により推定した。指数分布とポアソン分布によるデマンド発生を行うことで、改良アルゴリズムが実世界のようなランダム性の強いデマンド発生に対しても、分散の小さい配車ができるかを検証する。

5.3.1 指数分布によるデマンド発生

デマンド発生期間の設定として、指数分布を用いた。指数分布とは、統計における連続確率分布の一種である。単位時間あたりに λ 回事象が発生したとき、次に事象が生起する期間の確率を示す。そのため、デマンド発生頻度が指数分布に従うということは、ある単位時間あたりでデマンドが λ 回発生したとき、次にデマンドが発生する期間の確率は指数分布に従うことを意味する。例として、デマンド発生頻度が $1/135[\text{回}/\text{s}]$ のとき、つまり単位時間を 1 時間のとき、27 回デマンドが発生する頻度のときを挙げる。あるデマンドが発生したとき、次にデマンドが発生するまでの

期間の確率密度関数と累積分布関数は図 5.1 のようになる。生起する期間が 0 に近づくほど、確率密度関数は大きくなる。これは指数分布における無記憶性と呼ばれ、次の事象が起こるとき、前の事象が発生したか否かという情報は関与しない性質のことである。そのため、完全なランダムで発生する事象の確率分布であると捉えることができる。この性質を利用し、デマンド発生期間を、デマンド発生頻度ごとに、指数分布に従う乱数によって生成させた。

5.3.2 ポアソン分布によるデマンド発生

デマンドが一定期間における平均発生回数を設定するため、ポアソン分布を用いた。ポアソン分布とは、統計における離散確率分布の一種である。ある期間に λ 回事象が発生したとき、次の期間で k 回発生する確率を示す。デマンド発生頻度がポアソン分布に従うということは、あるデマンドがある期間に λ 回発生したとき、次の期間で k 回発生する確率はポアソン分布に従うことを意味する。1 時間に 10 回デマンドが発生する頻度のとき、ポアソン分布による k 回発生する確率は図 5.2 のようになる。この確率分布を基に、10 分ごとにデマンドが平均何回発生するかを、デマンド発生頻度に従い、ポアソン分布に従う乱数によって生成させた。

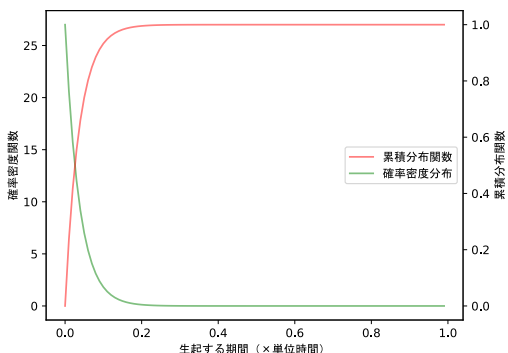


図 5.1: 指数分布の確率密度関数と累積分布関数

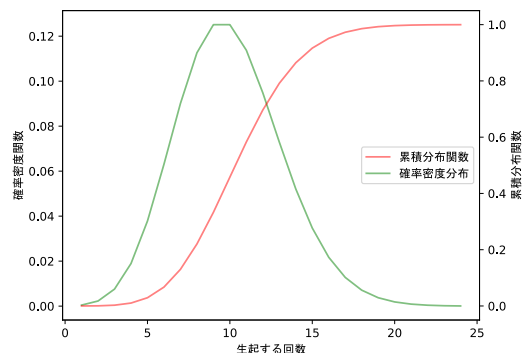


図 5.2: ポアソン分布の確率密度関数と累積分布関数

第 6 章

実験と考察

本章では、先述した条件におけるシミュレーションによって得られた、車輛待ち時間の変動係数・車輛待ち時間・車輛移動時間・便乗率・デマンド遂行率の値を示し、考察を行う。シミュレーションは、デマンド発生頻度ごとに、指数分布とポアソン分布によるデマンド群を事前に 3 通り生成させた。全てのシミュレーションは、全く同じデマンド発生パターンのデマンド列の下で実行される。つまり、デマンド発生頻度 $1/135[\text{回}/\text{s}]$ などにはデマンド発生期間・回数が別々な 3 通りのデマンド群が存在するということである。これは、ランダムに発生するデマンドにおいて、車両台数・デマンド発生頻度の条件における、尤もらしい値を取得するためである。車両台数 10 台・デマンド発生頻度 $1/135[\text{回}/\text{s}]$ の変動係数を求める流れを、図 6.1 に示す。まず指数分布によりランダムに 3 通りのデマンド群を生成、すべての車両台数・デマンド発生頻度の条件で、3 通りのデマンド群をシミュレーションし、それぞれの待ち時間を取得する。取得した待ち時間が 3 通りあるため、3 つの待ち時間の平均値を取り、その平均値を車両台数・デマンド発生頻度の条件における尤もらしい待ち時間とし、最終的に示すデータとした。

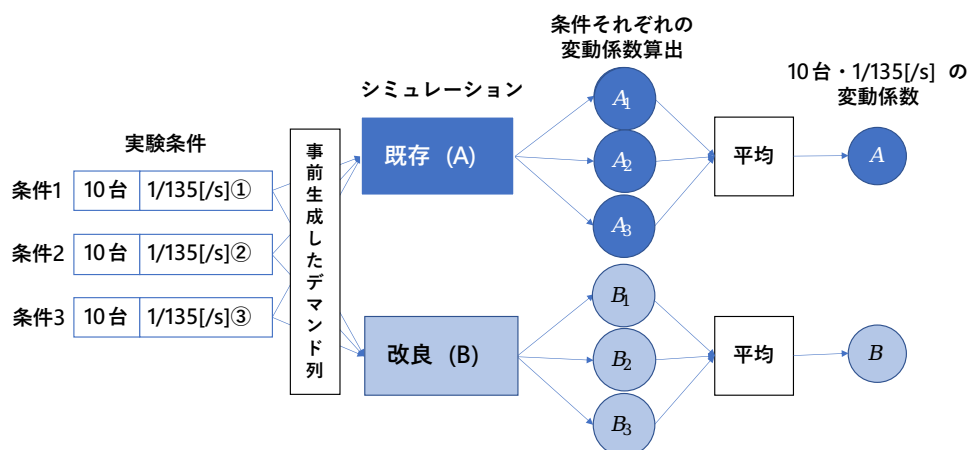


図 6.1: 車両台数 10 台・デマンド発生頻度 $1/135[\text{回}/\text{s}]$ の変動係数導出の流れ

6.1 車両待ち時間の変動係数

待ち時間の分散を評価する値として、変動係数を用いる。変動係数は標準偏差を平均値で割った数値を表す。二つのアルゴリズムの待ち時間を比較する際に、待ち時間の平均値が大きいアルゴリズムの方が、標準偏差も大きくなる可能性がある。そのため、標準偏差のみでは純粋な散布度の比較はできず、二つのアルゴリズムの待ち時間全体の変動係数の比較によって、待ち時間の最適化が行われているかを評価する。ここで、指数分布によるデマンド群によって得られた、逐次最適挿入法と改良アルゴリズムの車両待ち時間の変動係数を、車両台数・デマンド発生頻度に分け以下に示す。

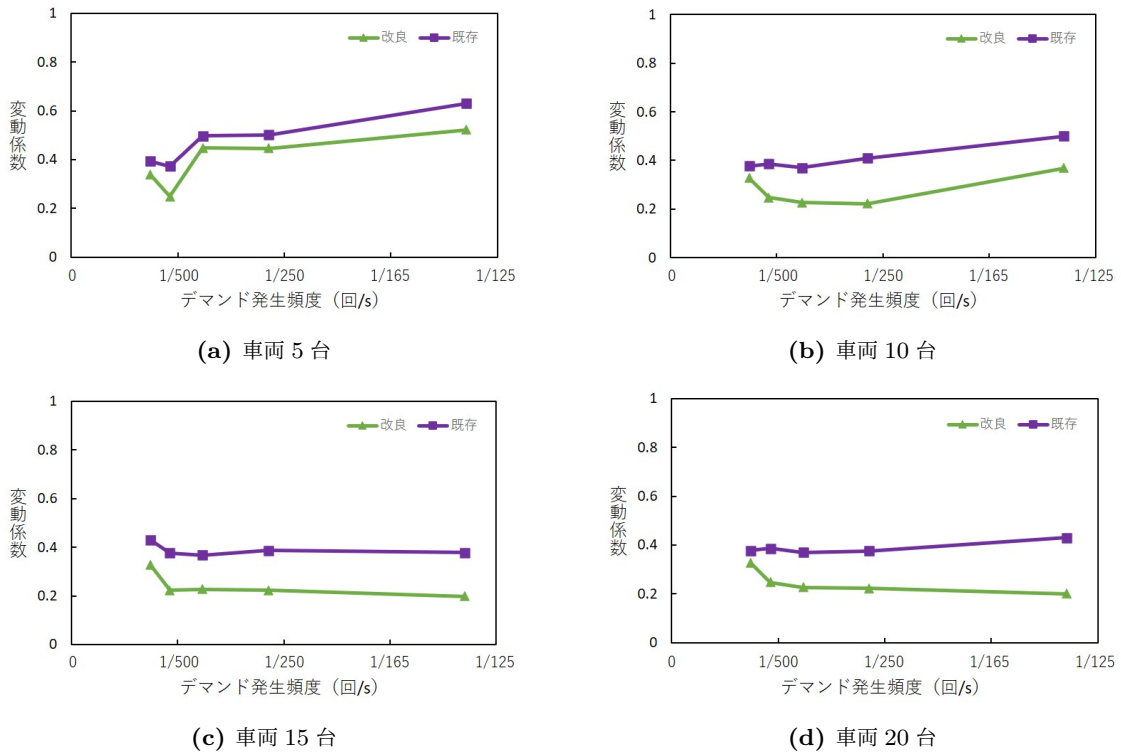


図 6.2: 車両台数ごとの変動係数 (指数分布)

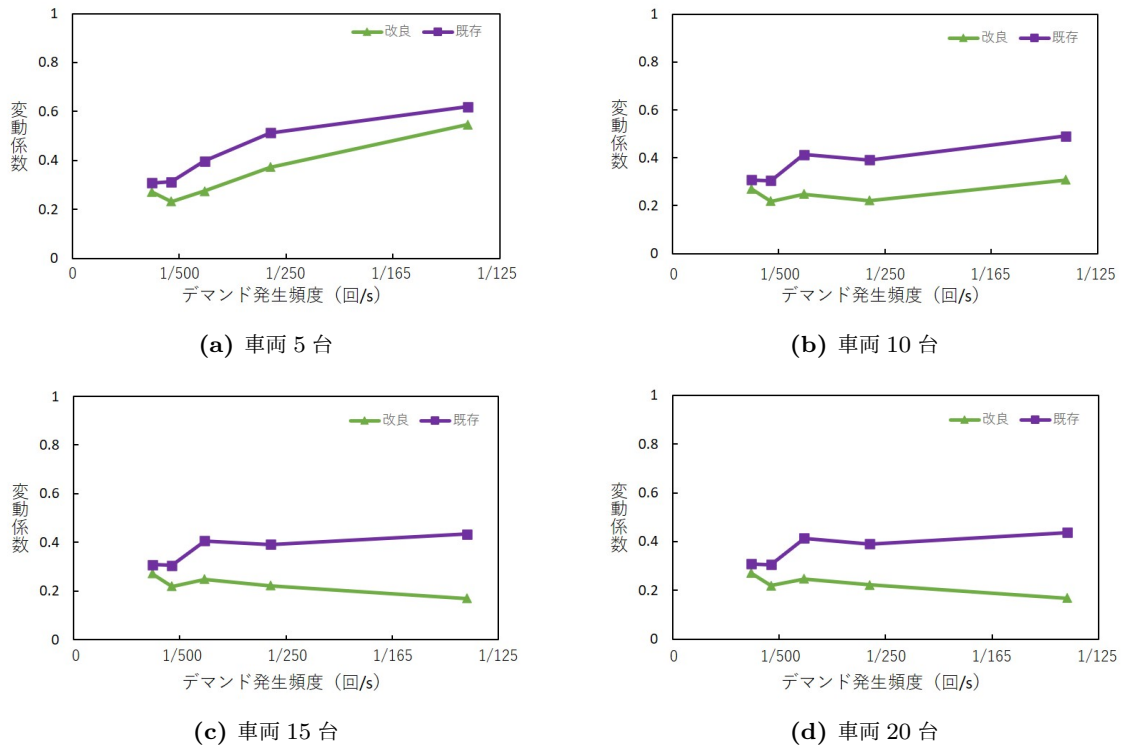


図 6.3: 車両台数ごとの変動係数 (ポアソン分布)

図 6.2 と図 6.3 から、以下のことが示唆された。

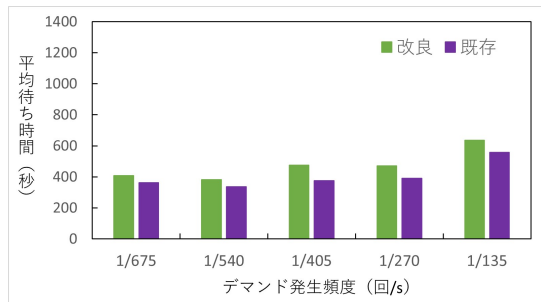
- 車両台数・デマンド発生頻度の全ての条件において、改良アルゴリズムの変動係数が逐次最適挿入法の変動係数よりも小さいことが確認できる。
- 図 6.2a, 図 6.3a を見ると、車両台数が 5 台の時、変動係数は 0.3~0.4 付近で推移し、逐次最適挿入法の変動係数と大きな差がないことが確認できる。
- 図 6.2b, 図 6.3b から、車両台数が 10 台になると変動係数は 0.2 から 0.3 付近で推移するようになることが確認できる。
- 図 6.2c, 図 6.3c, 図 6.2d, 図 6.3d を見ると、15 台と 20 台の時は、デマンド発生頻度が 1/500[回/s] よりも低いとき以外は、0.2 付近を保っている。
- すべての車両台数に共通していることは、デマンド発生頻度が 1/500[回/s] よりも低いとき、逐次最適挿入法の変動係数よりも大きく変動係数が低下しない、またはほぼ同じ数値を示す点である。

以上から、改良アルゴリズムが変動係数を安定して低下させるには、一定台数以上の車両の導入と、デマンド発生頻度がある程度高いことが条件となることが示唆される。

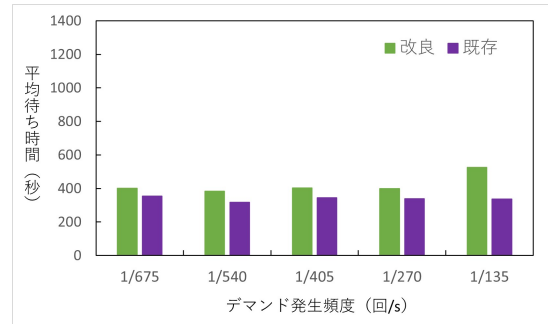
6.2 車両待ち時間の平均値・標準偏差

車両待ち時間の平均値と、秒数としてのばらつきである標準偏差を以下に示す。平均値・標準偏差の単位は秒である。配車コストの計算方法を変えたことにより、改良アルゴリズムの待ち時間が増加することが推測されている。

図表 6.4: 車両台数ごとの平均待ち時間と標準偏差（指数分布）



(a) 車両 5 台の平均待ち時間



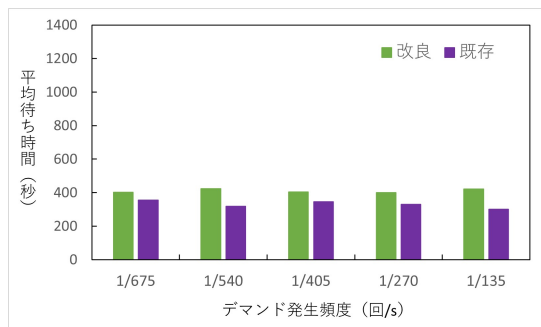
(b) 車両 10 台の平均待ち時間

(c) 車両 5 台の平均待ち時間における標準偏差

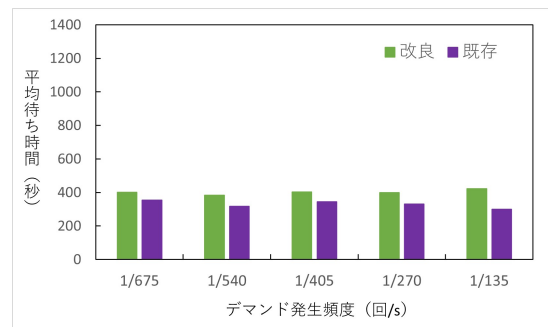
	1/675	1/540	1/405	1/270	1/135
既存	142.31	126.21	189.15	195.62	351.64
改良	139.32	94.24	217.97	212.91	332.80

(d) 車両 10 台の平均待ち時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	133.44	122.75	127.26	139.57	167.80
改良	132.53	94.71	91.99	89.03	195.48



(e) 車両 15 台の平均待ち時間



(f) 車両 20 台の平均待ち時間

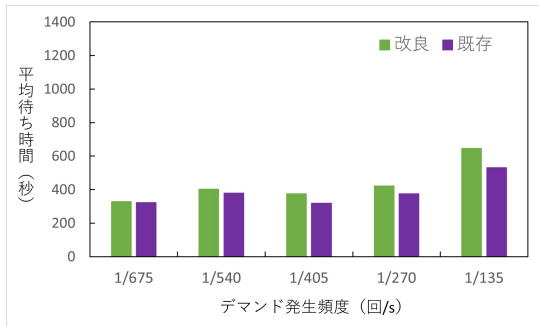
(g) 車両 15 台の平均待ち時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	133.44	122.75	126.15	124.83	129.43
改良	132.51	94.09	91.96	89.06	83.94

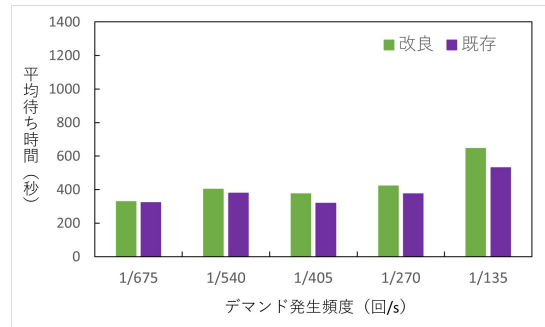
(h) 車両 20 台の平均待ち時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	133.42	122.74	127.27	124.82	129.27
改良	132.49	94.66	91.96	89.03	84.96

図表 6.5: 車両台数ごとの平均待ち時間と標準偏差 (ポアソン分布)



(a) 車両 5 台の平均待ち時間



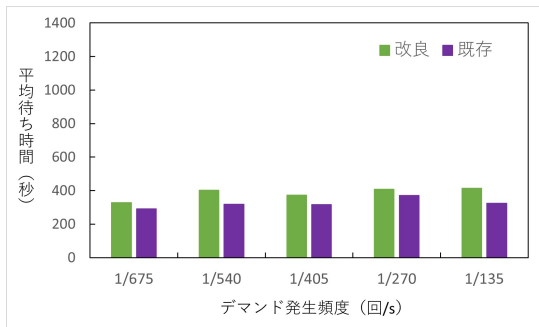
(b) 車両 10 台の平均待ち時間

(c) 車両 5 台の平均待ち時間における標準偏差

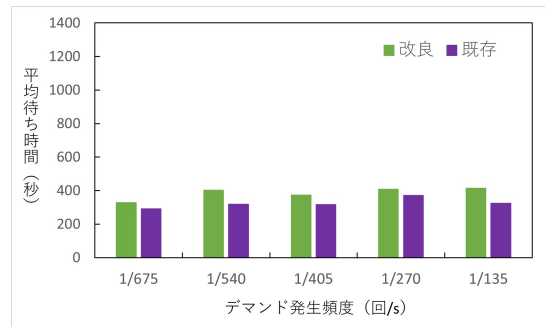
	1/675	1/540	1/405	1/270	1/135
既存	99.42	117.11	126.14	195.94	329.12
改良	88.28	93.07	102.90	158.99	351.59

(d) 車両 10 台の平均待ち時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	99.39	111.91	130.16	124.31	160.62
改良	88.28	88.35	92.51	90.58	146.62



(e) 車両 15 台の平均待ち時間



(f) 車両 20 台の平均待ち時間

(g) 車両 15 台の平均待ち時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	99.34	111.97	128.24	124.44	126.21
改良	88.34	88.39	92.50	90.64	69.95

(h) 車両 20 台の平均待ち時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	99.37	111.92	130.17	124.30	127.24
改良	88.26	88.34	92.51	90.62	69.47

図表 6.4 と図表 6.5 から、時間の平均値が逐次最適挿入法の平均値よりも約 50~100 秒ほど増加していることが確認できる。これは、逐次最適挿入法の配車コストの計算方法を変え、分散を最適化する代わりに、平均値を最適化するという特徴を失ったためであると考えられる。また、車両台数・デマンド発生頻度の違いによって、逐次最適挿入法と改良アルゴリズムの標準偏差の差が異なっていることが確認できる。

- 車両台数が 5 台のとき、表 6.4c から、デマンド発生頻度が 1/540[回/s] であれば改良アルゴ

リズムの標準偏差は低下しているが、それ以外のデマンド発生頻度では標準偏差が増加していることが確認できる。表 6.5c では、デマンド発生頻度が $1/405$ [回/s] 以下であれば改良アルゴリズムの標準偏差は 11~24 秒程度低下しているが、それ以上のデマンド発生頻度では増加する。

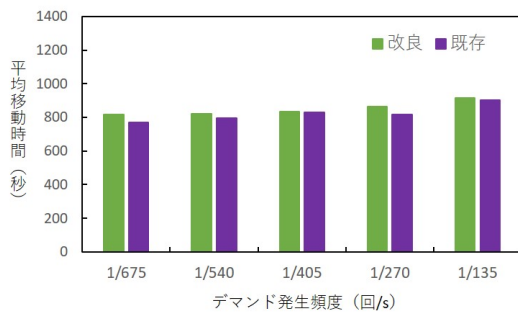
- 表 6.4d では、車両台数が 10 台のときは、デマンド発生頻度が $1/675$ [回/s] と $1/135$ [回/s] のとき、標準偏差はほぼ変わらない、もしくは増加している。一方、表 6.5d では、標準偏差が 2~57 秒程度低下している。
- 表 6.4g, 表 6.5g, 表 6.4h, 表 6.5h では、全てのデマンド発生頻度で標準偏差は 1~50 秒程度低下している。15 台のとき、表 6.4g のデマンド発生頻度が $1/675$ [回/s] のときのみ、標準偏差はほぼ変わらない。それ以外の条件では、標準偏差は約 10~46 秒ほど低下している。20 台のときも、15 台とほぼ同じ推移を示している。これは、デマンド処理に対応するための車両が 15 台で足りていたことで、同じ配車結果になったためであると考えられる。

これらの結果から、改良アルゴリズムが標準偏差を安定して低下させるには、車両台数の一定以上の導入と、デマンド発生頻度が一定以上高いことの 2 つの条件が必須であることが示唆される。

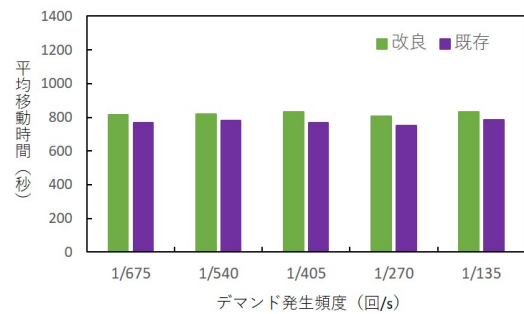
6.3 車両移動時間の平均値・標準偏差

逐次最適挿入法と改良アルゴリズムの車両移動時間を、車両台数・デマンド発生頻度に分け以下に示す。車両移動時間は、デマンドを割り当てられた車両がデマンド出発地点に到着した時刻から、デマンド目的地点に到着した時刻の差とした。

図表 6.6: 車両台数ごとの平均移動時間と標準偏差 (指数分布)



(a) 車両 5 台の平均移動時間



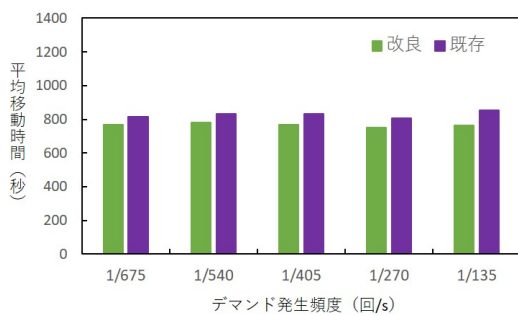
(b) 車両 10 台の平均移動時間

(c) 車両 5 台の平均移動時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	71.74	63.89	90.93	88.90	150.80
改良	65.24	46.96	98.01	99.62	149.46

(d) 車両 10 台の平均移動時間における標準偏差

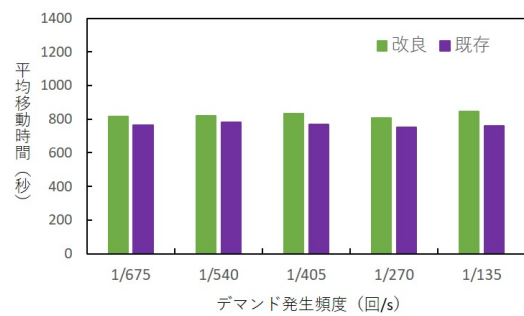
	1/675	1/540	1/405	1/270	1/135
既存	68.01	62.85	66.11	69.31	75.61
改良	61.96	46.58	44.20	44.67	89.82



(e) 車両 15 台の平均移動時間

(g) 車両 15 台の平均移動時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	66.34	63.90	65.16	62.85	68.00
改良	61.95	48.08	44.20	44.67	40.07

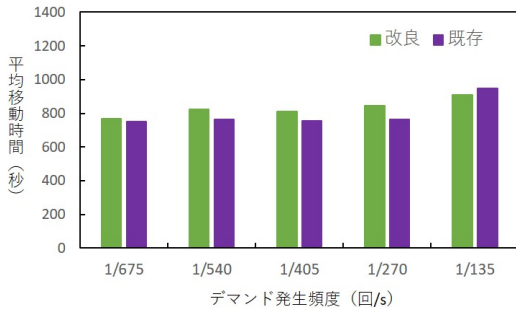


(f) 車両 20 台の平均移動時間

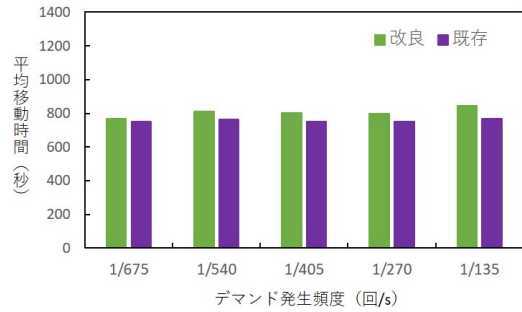
(h) 車両 20 台の平均移動時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	68.01	63.45	66.07	65.34	66.29
改良	61.95	46.58	44.20	44.67	40.76

図表 6.7: 車両台数ごとの平均移動時間と標準偏差 (ポアソン分布)



(a) 車両 5 台の平均移動時間



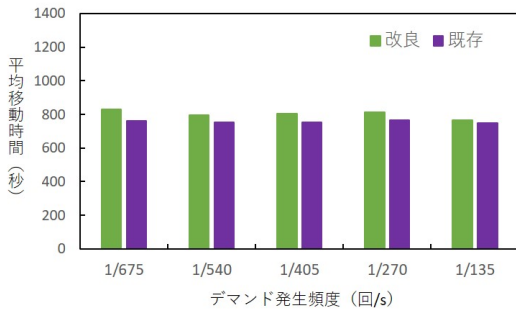
(b) 車両 10 台の平均移動時間

(c) 車両 5 台の平均移動時間における標準偏差

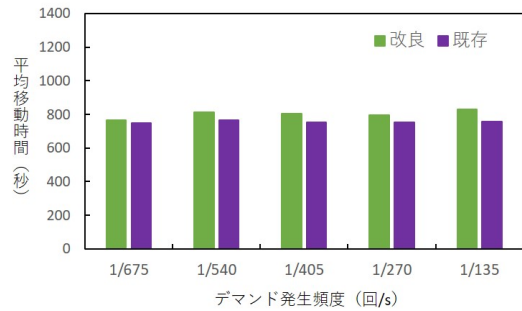
	1/675	1/540	1/405	1/270	1/135
既存	122.59	117.11	112.76	114.41	139.94
改良	121.52	106.66	103.64	105.27	154.28

(d) 車両 10 台の平均移動時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	78.96	89.11	93.81	87.58	94.11
改良	78.90	71.55	79.45	80.79	92.83



(e) 車両 15 台の平均移動時間



(f) 車両 20 台の平均移動時間

(g) 車両 15 台の平均移動時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	88.58	61.58	115.26	50.27	50.40
改良	46.54	42.33	101.76	42.23	48.05

(h) 車両 20 台の平均移動時間における標準偏差

	1/675	1/540	1/405	1/270	1/135
既存	50.89	50.44	67.06	61.76	64.61
改良	48.36	42.31	45.04	42.40	39.09

図表 6.6, 図表 6.7 から, 移動時間の平均値が逐次最適挿入法の平均値よりも約 50~100 秒ほど増加していることが確認できる. また, 車両台数・デマンド発生頻度の違いによって, 逐次最適挿入法と改良アルゴリズムの標準偏差の差が異なっていることが確認できる.

- 車両台数が 5 台のとき, 表 6.6c では, デマンド発生頻度が 1/405[回/s] 以上であれば, 既存アルゴリズムと比べ, 改良アルゴリズムの標準偏差は大きくなるが, それ以外のデマンド発

生頻度では標準偏差が低下していることが確認できる。表 6.7c では、デマンド発生頻度が $1/135$ [回/s] 以外のとき、改良アルゴリズムの標準偏差が 1~10 秒程度小さくなっていることが分かる。

- 車両台数が 10 台のとき、表 6.6d では、デマンド発生頻度が $1/135$ [回/s] のとき、改良アルゴリズムの標準偏差が増加しているが、それ以外のデマンド発生頻度では 6~25 秒程度低下している。表 6.7d では、全てのデマンド発生頻度で標準偏差が 0.6~14 秒程度低下している。
- 車両台数が 15 台、20 台のとき、表 6.6g, 表 6.7g, 表 6.6h, 表 6.7h から、全てのデマンド発生頻度で標準偏差が低下していることが確認できる。標準偏差は 2~42 秒ほど低下している。

これらの結果から、改良アルゴリズムが待ち時間の分散を最小化することによって、車両移動時間に対し 2 つの影響を及ぼすことが示唆された。1 つは、車両移動時間が増加である。2 つ目は、車両台数の一定以上の導入によって、車両移動時間の標準偏差が低下することである。

6.4 考察

シミュレーションにおける実験結果から、5km 四方の道路網における改良アルゴリズムの配車結果について、以下の特徴が示唆された。

- 1). 図 6.2, 図 6.3 から、車両台数が 15 台以上導入され、かつデマンド発生頻度が $1/405$ [回/s] 以上である場合、改良アルゴリズムは逐次最適挿入法よりも変動係数が低くなる配車を安定して行うことを確認した。
- 2). 表 6.4c · d · g · h, 表 6.5c · d · g · h から、車両台数が 15 台以上導入され、かつデマンド発生頻度が $1/405$ [回/s] 以上である場合、改良アルゴリズムは逐次最適挿入法よりも標準偏差が低くなる配車を安定して行うことを確認した。
- 3). 図 6.4a · b · e · f, 図 6.5a · b · e · f, 図 6.6a · b · e · f, 図 6.7a · b · e · f から、改良アルゴリズムは、待ち時間の分散を最小化する代わりに、待ち時間 · 移動時間の平均値を上昇させている。
- 4). 表 6.6c · d · g · h, 表 6.7c · d · g · h から、改良アルゴリズムにより分散をしたとき、車両台数が一定以上導入されていれば、車両移動時間の標準偏差が低下することを確認した。

1 と 2 から、改良アルゴリズムは車両台数の一定以上の導入によって、逐次最適挿入法よりも散布度の低い配車を安定して行えることが示唆される。車両台数が少ない場合は変動係数と標準偏差の差がない、もしくは大きくなる例があり、散布度の低さは安定しない。3 より、改良アルゴリズムは車両待ち時間 · 移動時間の平均値を逐次最適挿入法より大きくする性質を持つことが示唆される。4 より、改良アルゴリズムにより分散を最適化したことによって、車両移動時間のばらつきも小さくなることが示唆される。

よって、改良アルゴリズムを実際にサービスとして適用したとき、十分に利点を発揮するためには、車両台数の一定以上の配備、デマンド需要が十分に存在する地域への導入が条件となることが考えられる。移動時間のばらつきが小さくなるという副次的な効果が示されたが、この効果は意図したものではないため、なぜ効果が生じたかを調査する必要がある。また、逐次最適挿入法よりも待ち時間 · 移動時間の平均値が長くなることから、実用する際は乗客へあらかじめ待ち時間が長くなることの提示などが必要になることが挙げられる。

第7章

結言

7.1 結論

本研究では、SAVSにおける配車アルゴリズムである逐次最適挿入法が、分散を最適化していないことで乗客間で待ち時間のばらつきが発生する恐れがあることを指摘し、配車アルゴリズムを改良し分散を最適化するための手法を提案した。提案手法は、逐次最適挿入法の配車コストを新デマンドの待ち時間と過去に配車したデマンドの待ち時間全体との分散とし、これが最小となる車輛・挿入順となる配車を行うことで、待ち時間全体の分散が最小となることを目指した。また、シミュレーションを行うにあたり、デマンド発生頻度や発生の形態を、実社会に即したものに設定した。デマンド発生頻度は関連研究 [17] の時間帯別公共交通利用者のデータを参考にし、利用者が少ない状況から多い状況までを網羅的に検証できるようにデマンド発生頻度の設定を行った。また、指数分布とポアソン分布による確率分布によりランダムなデマンドの発生期間・回数を設定することで、実社会におけるランダム性を一部再現した。

シミュレーションの結果から、改良アルゴリズムの車輛待ち時間の分散が最適化されているかを示した。実験室結果として、車輛台数が15台以上導入され、かつデマンド発生頻度が $1/405$ [/s]以上である場合、改良アルゴリズムは逐次最適挿入法よりも変動係数が低くなる配車を安定して行うことを確認した。よって車両待ち時間の分散を安定して最適化するためには、一定以上の車両台数の導入と、デマンド需要が十分に存在することが条件となることが考えられる。従って、それらの条件をクリアした環境下であれば、改良アルゴリズムは逐次最適挿入法よりも待ち時間のばらつきを小さくできる有効性を発揮できることが示唆された。これにより、乗客個々人の待ち時間のばらつきを抑え、より不満・不公平が生じない移動サービスの提供を行えると考えられる。

7.2 今後の課題

今後の課題として、まず実世界の道路環境においても、改良アルゴリズムは同じような振舞いを見せるかどうかを検証する必要がある。シミュレーションにおいて取得した値は、 5km 四方の簡易的な道路網におけるデータである。本研究は改良アルゴリズムの基本的な特徴を示したため、函

館市の道路環境のように具体的な条件下においてどのような振舞いを見せるかを確認する必要がある。先述したように函館市の面積は約 15km^2 であり、本研究における道路網との面積比は約 1 : 9 である。本研究では一般車両や信号機などは考慮していないため、単純にデマンド発生頻度や車両台数を 9 倍することで同じような値が取得できるとは限らない。実世界の複雑な道路網が運行に及ぼす影響も現時点では計り知れないため、検証が必要である。

次に、車両台数が少ない場合でも分散を小さくできるよう、改良を進める必要がある。本研究は、改良アルゴリズムが分散を安定して小さくできる条件として、車両を一定台数以上の導入が必要であることを述べた。車両台数が少ないときは、逐次最適挿入法と比べ待ち時間の平均値が大きく、変動係数に大きな差がない。そのため、配車アルゴリズムとしての役割を安定して保つことができないという側面が存在する。サービス価値の安定的な提供のためには、車両台数が少ない場合でも分散を小さくする配車が行えるよう、さらなる改良を進めていく必要がある。

謝辞

本研究における方針やその内容に関してご指導と助言をいただきました，公立ほこだて未来大学システム情報科学部複雑系知能学科の平田圭二教授に心から感謝申し上げます．そして交通シミュレーションの構築にあたり，公立ほこだて未来大学大学院修士課程学生岩田聖氏と鈴木恵二教授に多くの支援を受けましたことを，深くお礼申し上げます．また，論文構成や内容に関して多くの助言をくださった，研究室の皆様へお礼を申し上げます．

参考文献

- [1] 国土交通省, ” 国土交通省の MaaS 推進に関する取組について”, [Online] <https://www.mlit.go.jp/maritime/content/001320589.pdf>, (2020/12/20 Access).
- [2] Sonja Heikkila : Mobility as a Service – A Proposal for Action for the Public Administration Case Helsinki, Master’s Thesis of Aalto University, 2014.
- [3] Whim, ”All transport in one app”, [Online] <https://whimapp.com/>, (2020/12/20 Access).
- [4] 中島秀之, 小柴等, 佐野渉二, 落合純一, 白石陽, 平田圭二, 野田五十樹, 松原仁. : フルデマンド型公共交通配車システムの実装と評価, 情報処理学会論文誌, Vol. 57, No.4, pp.1290-1302, 2016.
- [5] 平田圭二, 鈴木恵二, 野田五十樹, 落合純一, 金森亮, 松館渉, 中島秀之, 佐野渉二, 白石陽, 松原仁. : 完全自動リアルタイムフルデマンド交通システム SAVS 向けプラットフォームの設計と実装, 情報処理学会研究報告, Vol. 2017-ITS-68, No.1, pp.1-6, 2017.
- [6] 未来シェア, ” 令和 2 年度「地域・企業共生型ビジネス導入・創業促進事業補助金」に係る補助事業の一環として行われる「福祉 Mover」へ技術協力します”, [Online] <http://www.miraishare.co.jp/202007fukushimover/>, (2020/12/21 Access).
- [7] 岩手県紫波町, ” デマンド乗合バス「しわまる号」”, [Online] <http://www.town.shiwa.iwate.jp/soshiki/4/2/99/shiwademando/index.html>, (2020/12/21 Access).
- [8] ウィリアム・J・クック著, 松浦俊輔訳. : 驚きの数学巡回セールスマン問題, 青土社, 2013.
- [9] Cordeau, J.F. and Laporte, G. : The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms, 4OR, A Quarterly Journal of Operations Research 1, pp.89-101, 2003.
- [10] Luo, Y and Schonfeld, P. : A rejected-reinsertion heuristic for the static Dial-A-Ride Problem, Transportation Research Part B, Vol.41, No.7, pp.736–755, 2007.
- [11] Campbell, A.M. and Savelsbergh, M. : Efficient insertion heuristics for vehicle routing and scheduling problems, Transportation science, Vol.38, No.3, pp.369-378, 2004.
- [12] 内村圭一, 若林竜太, 境周平. : Dial-a-Ride 問題におけるリアルタイム乗客輸送, システム制御情報学会論文誌, Vol.15, No.8, pp.413-421, 2002.
- [13] 野田五十樹, 篠田孝祐, 太田正幸, 中島秀之. : シミュレーションによるデマンドバス利便性の

- 評価, 情報処理学会論文誌, Vol.49, No.1, pp.242-252, 2008.
- [14] Krajzewicz, D. Erdmann, J. Behrisch, M. and Bieker, L. : 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.
- [15] 小柴等, 野田五十樹, 山下倫央, 中島秀之: 実環境を考慮したバスシミュレータ SAVSQUID による実運用に向けたデマンドバスの評価, コンピュータソフトウェア, Vol.31, No.3, pp.141-155, 2014.
- [16] 岩田聖, 鈴木恵二, : 広域地方に向けた SAVS シミュレーション分析, 人工知能学会全国大会 論文集, Vol.JSAI2019, pp.1-2, 2019.
- [17] 平田圭二, 佐野渉二, 小柴等, 野田五十樹, 金森亮, 中島秀之. : Smart Access Vehicle サービス実践への取り組み サービス共創最適化のためのフレームワーク, サービス学会第 2 回国内大会講演論文集, pp305-309, 2014.