2019 年度　芝浦工業大学

学　士　論　文

題目：『ユーザー再配置を考慮したワンウェイ型
カーシェアリングの再配置最適化』

Car sharing optimization for one-way system based on user relocation

学　科　機械制御システム学科

学籍番号　BQ16048

ふりがな　せき　こうたろう

氏　名　関　倖太郎

指導教員　長谷川　浩志

1. 研究背景

近年，シェアリングエコノミーという新たな形のサービスが世界的に人がりつつある．シェアリングエコノミーとは，個人が保有する遊休資産を他人に有償で貸し出すといったサービスの総称である．これにより，典型的には個人が保有する遊休資産の活用による収入，借主は所有することなく利用ができるというメリットがある[1]．シェアリングエコノミーはシリコンバレーを起点にグローバルに成長してきた．図 1.1 に PwC によるシェアリングエコノミーの市場規模に関する予測を示す[1]．PwC によると，2013 年に約 150 億ドルの市場規模が 2025 年には約 3,350 億ドル規模に成長する見込みである[1]．

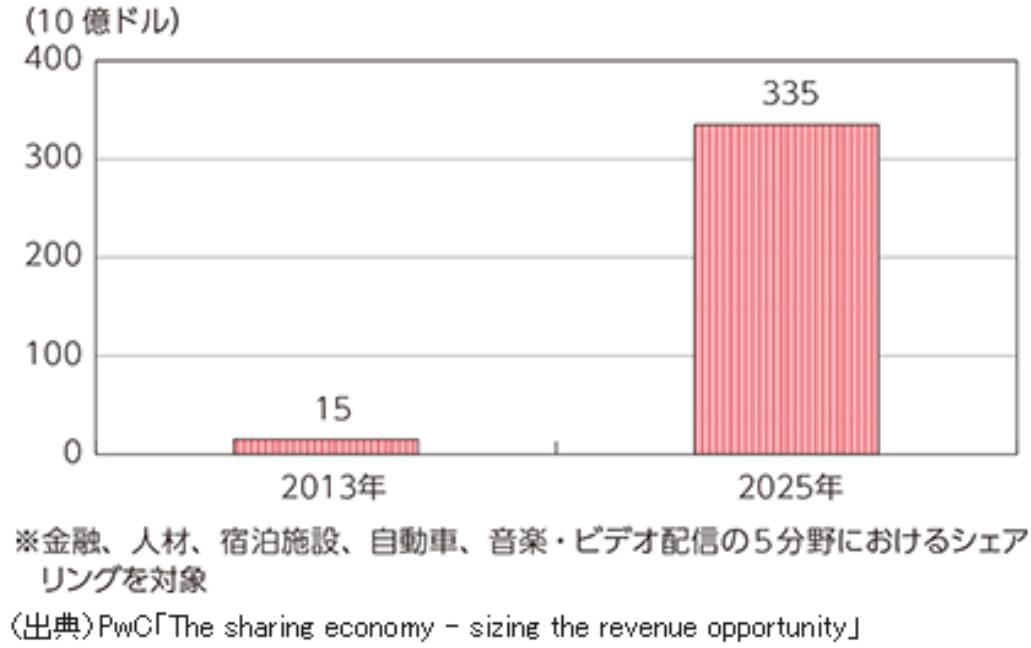


図 2.1 シェアリングエコノミーの市場規模

サービスの内容は多岐にわたり，日本国外ではすでにこれら多くのサービスが展開されている．

例えば，Airbnb[2]は空いているもしくは一時的に使用していない家や空きスペースを必要な人に有償で貸し出すことができるサービスである．図 1.2 は Airbnb のサービス提供イメージである[3]．

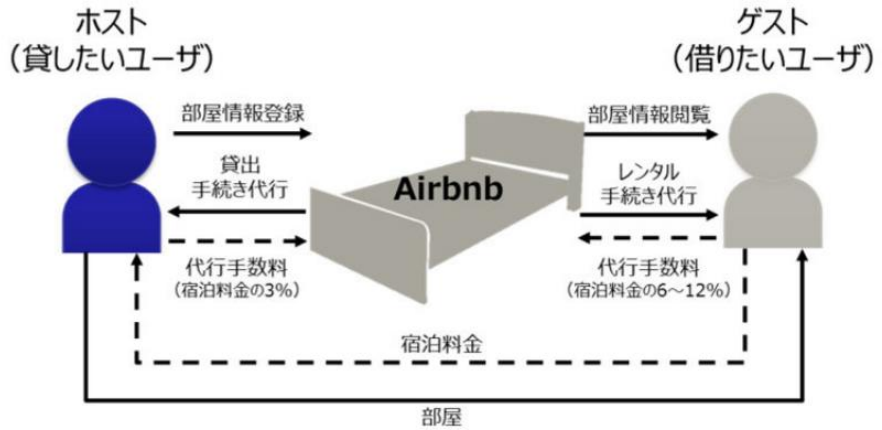


図 2.1.2 Airbnb のサービスイメージ

2008 年に創業した同サービスは現在世界 192 か国 33000 都市でサービスを提供し，一日あたりの利用者数は 400 万人を超えた[1][4].

こうしたサービスはモビリティ業界でも注目を集めており，自転車を借りることのできるサービスとして 2007 年より velib'[5]という自転車シェアリングサービスがフランスでは提供されている. velib'は 300m 間隔でステーションが設置されており 23600 台以上の自転車が導入されている. パリ市民の移動手段として導入されたが，観光客にとっても手軽な移動手段として利用することができ，その両面でも流目されている.

OpenStreet 株式会社は Hello cycling というサービスを提供している[6]. Hello cycling はシェアリングサービスを相互利用できる交通インフラである. シェアサイクルサービスを利用する際に利用者はサービスごとにユーザー登録をする必要があったが，シェアサイクル事業を手掛ける会社が急増している現在，これが手間になっていた. Hello cycling はこれを統合し，異なる企業のステーションや自転車を利用できるようにした. 関東では株式会社コインパークによるシェアペダル[7]，小山市役所による，らくーる[8]など様々な機関，企業が参入している.

自動車に関しても同様のサービスが提供されている. 相乗りによって車両の座席を貸し出すライドシェアリングと呼ばれているものと車両そのものを貸し出すカーシェアリングである. ライドシェアリングは移動したい人近くにいたドライバと相乗りするといったものであり，2009 年にスタートした Uber[9]の場合，タクシー会社だけでなく個人のドライバとも連携している[1].

それぞれサイクルシェアリングやカーシェアリング，ライドシェアリングなどのサービスが普及してきた理由は様々である. しかし MaaS という考え方の登場がこれらのサービスの普及を後押しする共通の理由として考えられる.

電車やバス，飛行機など複数の交通手段を乗り継いで移動する際，それらを跨いだ移動ルートは検索可能となったが，予約や運賃の支払いは，各事業者に対して個別に行う必要がある.

このような仕組みを，手元のスマートフォン等から検索～予約～支払を一度に行えるように改めてユーザーの利便性を大幅に高め，また移動の効率化により都市部での交通渋滞や環境問題，地方での交通弱者対策などの問題の解決に役立てようとする考え方の上に立っているサービスが MaaS である[10].

こうした観点からするとカーシェアリングにおいては，サービスの利用により利用者自身の経済的負担が軽減されることはもちろん，MaaS や 5G の導入と併せてより柔軟な移動手段を選択できるようになり，公共交通機関の利用が増加し環境負荷の軽減にもつながる. また，社会全体で考えると，自動車保有台数の軽減により交通量および二酸化炭素の排出量，駐車スペースの削減を図ることができる[11].

近年，維持費など金銭面の影響を受け，日本では東京など一部の地域で乗用車を所有する人が減少しており，以上に述べた理由などによってカーシェアリングの急速な普及が進んでいると考えられる.

図 1.3 は一般財団法人自動車検査登録情報協会が公開している 2009 年から 2019 年までの東京都における乗用車登録台数を示している[12].

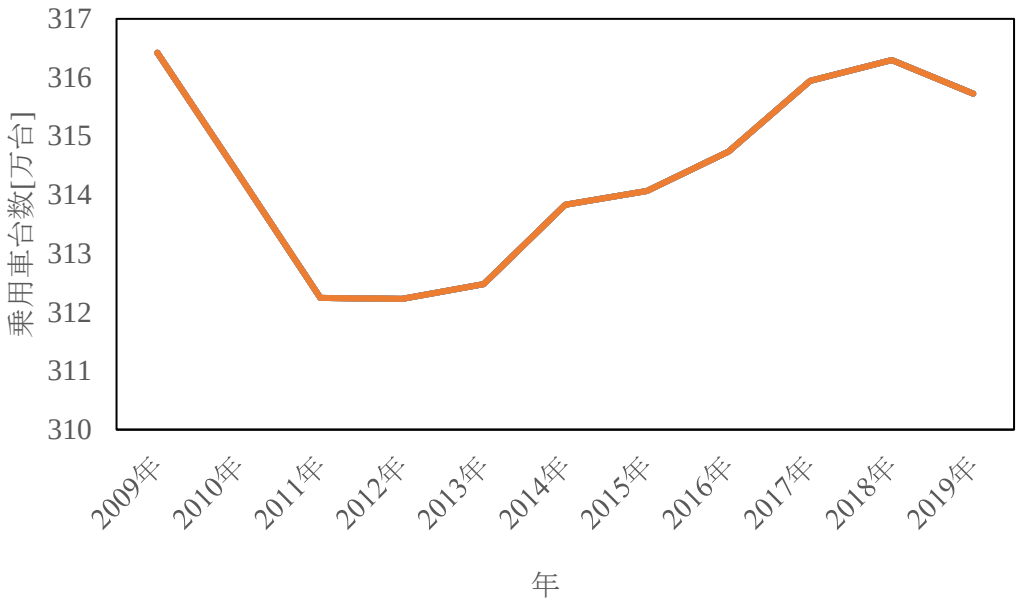


図 2.1.3 東京都における乗用車の保有台数[12]

年ごとのばらつきはあるものの，東京都における乗用車の保有台数はおおむね横ばいであることがわかる．一方で図 1.4 は東京都が公開している 2009 年から 2019 年までの東京都における人口の推移である[13]．

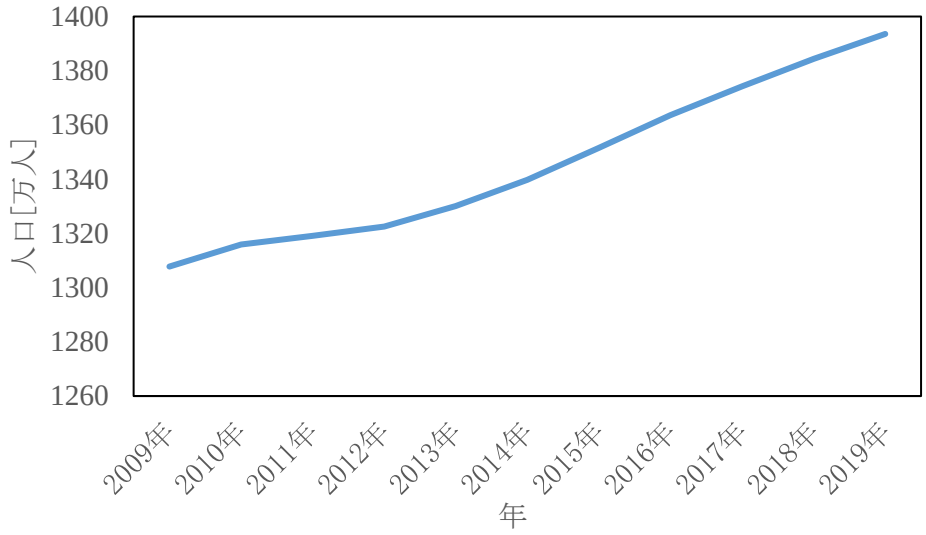


図 2.1.4 東京都における人口の推移[13]

東京都における人口はゆるやかな増加傾向にあり，10 年間で約 90 万人増加している．これらから東京都における一人あたりの乗用車の保有台数は以下図 1.5 のようになった．

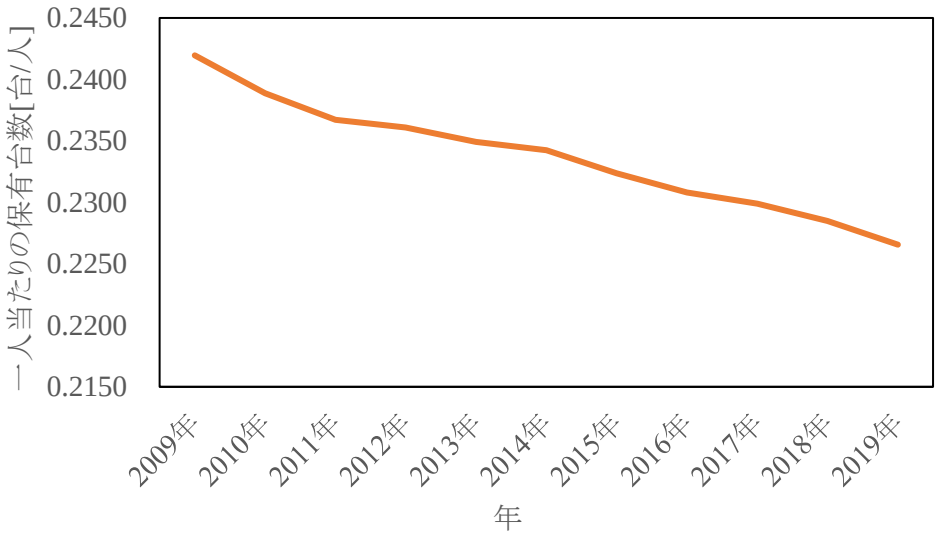


図 2.1.5 東京都における一人当たりの乗用車保有台数の推移

また，図 1.6 に公益財団法人交通エコロジー・モビリティ財団によるカーシェアリングにおける車両台数と会員数の推移を示す[14].

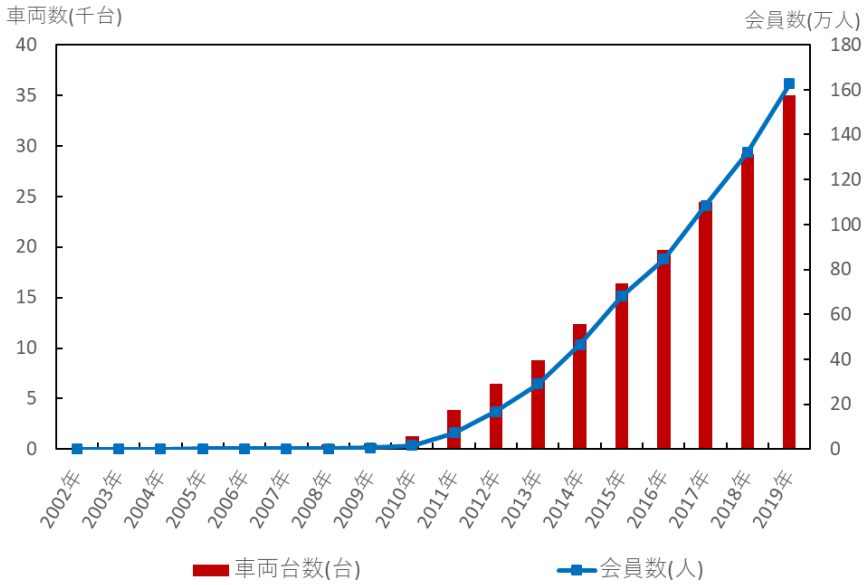


図 2.1.6 カーシェアリングにおける車両台数と会員数の推移[14]

これよると，2010 年以降を皮切りに，カーシェアリングの会員数，車両数ともに年々増加傾向にあり，2019 年にはすでに会員数が 150 万人にもおよんでいる.

以上のデータより日本では東京など一部の地域で乗用車を所有する人が減少しており，それに伴ってカーシェアリングの急速な普及が進んでいると考えられる.

カーシェアリングサービスは，貸し出し場所と返却場所が同一であるラウンド型と貸し出し場所と返却場所が異なるワンウェイ型に大別される．このワンウェイ型にも，ステーションベースとフリーフ

ローティングの二つのサービスがある. ステーションベースでは専用のステーションを必要とするのに対して, フリーフローティングではステーションを必要としないため, サービス提供地域内であれば駐車可能な路側帯や公共駐車場に返却できる[15].

日本においては法規制などにより, フリーフローティングでのサービス実現は困難であると考えられている. このことから日本ではラウンド型が主流となっているが, ステーションベースでのカーシェアリングも注目されており実証実験を行った例が存在する. 最近ではタイムズ 24 とトヨタ自動車が共同で Ha:mo という超小型電気自動車を用いたワンウェイ型ステーションベースでの実証実験を豊田市や東京, 沖縄など一部の地域で行った[16].

しかし, ワンウェイ型では車両の偏在が生じてしまい, 利用したいが車両が不足している, もしくは車両を返却したいが駐車スペースが存在しないといった要求拒否が生じてしまう. そしてこの問題を解消するためには車両の再配置が必要であるが, コストがかかってしまう. そのため車両を効率的に再配置するためのアルゴリズムや新たなシステムが必要である.

2. 類似研究

本章では，本研究の参考にした研究，類似している研究についてまとめ，紹介する．

2.1. 再配車によらない電気自動車の共同利用システムの効率化に関する研究[17]

中山らは，電気自動車 (以下 EV) を用いて運用時間中に再配置を行わない前提で，遺伝的アルゴリズムを用いたシステムの効率化を目指した[17]．

2.1.1. 研究手法

この研究では京都パブリックカーシステムを事例として取り上げた．京都パブリックカーは平成 12 年よりサービスを開始した，超小型 EV の共同利用サービスである．サービス提供当初，配置ステーション数は 7，EV の配車数は 35 台であった．

運用時間中に再配置を行わずに車両の偏りを防ぐ手法として会員属性の分散による需要の均一化や予約の選定が提案された．予約の選定には，予約の際に利用者が申告する出発駐車場，返却駐車場，出発予定時刻，返却予定時刻，予定走行距離などが利用された．

このようなシステムのために，システムの利用人数とシステム側のコストのトレードオフを考慮する必要があった．この研究では 1 日当たり 1EV 当たりの利用回数を最適化問題の目的関数とし，これを最大化した．制約条件には利用者の満足度を担保するため，全予約申し込み数に対する実際の予約成立回数の割合が 50%以上になることなどを設定した．

最適化計算を行うにあたって，時間を離散化した各時点での各 EV の状態量を逐一再現していく，ピリオディックスキャン方式に基づくシミュレーションアプローチを採用した．利用者がいつどこからどこへと向かうのかといった需要データに関してはパーソントリップ調査結果を用いて予測を行った．

最適化パラメータとしては配置する総 EV 数や EV の初期配置，会員数，会員構成比などを用いた．また，シミュレーション結果と後の実証実験での実データの比較なども行っていた．

2.1.2. 結果

シミュレーションからわかったことを以下に示す．まず実際の利用データと需要データのミスマッチである．パーソントリップ調査の結果から算出した予測データと実際の需要には差があり，これは EV などを利用する人々が単に EV へと普段の移動手段を変更するわけではなく，システム利用に適する形でトリップが発生したためだと考えられた． EV の配置台数は概ね 0.5 前後となり，駐車スペースに対しておよそ半分強となる場合が最適解となっていた．また，EV の偏在をなくすように予約を制限すべきかについては，いずれのケースでも予約制限をするべきではないことが結論づけられた．

2.2. ワンウェイ方式カーシェアリングにおける潜在的利用者を活用した予約受託率最大化手法[18]

奈良先端科学技術大学院大学の千住琴音はワンウェイ方式カーシェアリングサービスにおける車両偏在問題の解決のため依頼トリップという概念を導入した[18]．仮想的なサービスの利用者に対して依頼を行い，移動したい場所へと車両を移動させる手法である．

2.2.1. 研究手法

この類似研究では依頼トリップを導入することで，車両偏在問題を解決すると述べた．しかし，システム実現のためには以下に挙げる二つの課題に取り組む必要があった．

1. 最小の依頼で車両偏在問題に対する最大の効果 (トリップ成立数の最大化) を得るための依頼トリップ区間の導出方法．
2. その依頼トリップを受託する可能性の高いユーザーの抽出．

これらに対して，

1. シミュレーションにより仮想的なワンウェイ方式カーシェアリングサービスを構築し，車両の適切な移動方法について検討．
2. 実際のワンウェイ方式カーシェアリングサービス事業者から得られる実データを分析し，実利用者の行動特徴を明らかにするとともに，その結果に基づいて依頼戦略を検討．

というアプローチをとった．この提案手法の流れを図 2.2.1 に示す．

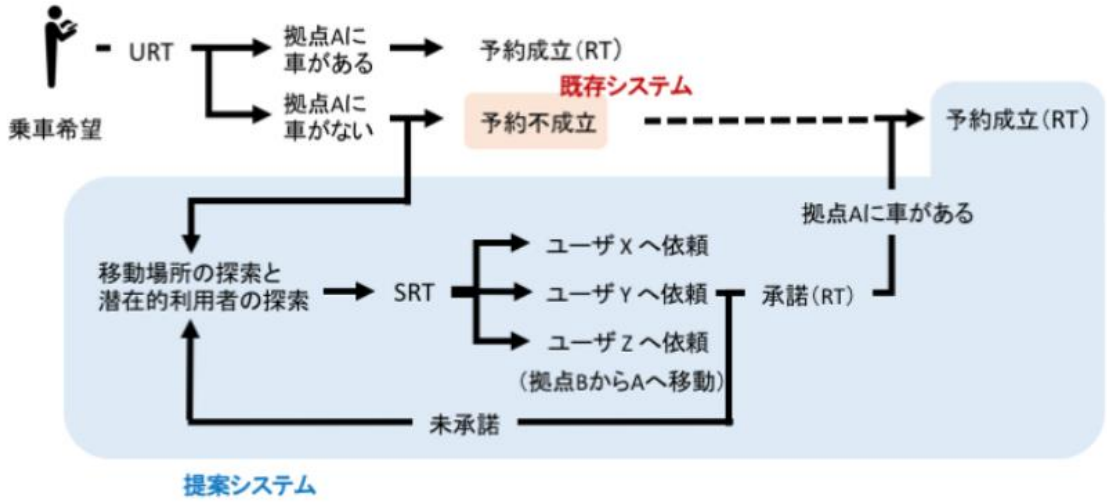


図 2.2.1 提案手法のフロー

利用者による乗車希望予約を，利用者の要求トリップ（User Request Trip：URT），確定した予約を確定トリップ（Reserved Trip：RT）として定義した．URT は，利用者の出発地，目的地，出発時刻，許容時間からなるものである．許容時間とは，出発時刻の変更可能な時間範囲を指す．この手法においては，予約不成立となった場合，車両偏在の状況から最適な車移動のトリップを，過去の利用情報から潜在的利用者の探索を行う．システムは探索した複数の潜在的利用者に，定められた時間や出発地，返却地に基づく車活用を依頼する．これをシステムによる依頼トリップ（System Request Trip：SRT）と定義した．

この研究ではこれらを最適化問題として定式化を行った．また，シミュレーションでは，要求トリップ数の最大化を目指し，提案手法による効果を検討した．同様に，ある確率で依頼トリップが断られることの影響についても検討した．

シミュレーションは要求トリップ数が 30 件あったある 1 日に関して，1 日あたり 5 台の運営によるカーシェアリングサービスを想定した．拠点は 6 つとし，各拠点間の移動距離はいずれも 30 分以内であると想定している．また，依頼トリップの受託率は 0.2，0.4，0.6，0.8，1.0 と値を変化させて受託

された要求トリップ数をみる.

このシミュレーションでは総当たりでルートを考えたため，車台数や要求トリップ数を増加させると膨大な計算時間を必要とした. しかし，これらの手法の場合，計算時間の観点から実運用への適用は困難である. 総当たり手法を採用した場合，計算時間が指数関数的に増加することは自明であり，拡張性の観点からも不適切である. これをうけて，千住は準最適解を考慮した.

実際のワンウェイ型カーシェアリングサービスの利用者データはパーク 24 株式会社の乗り捨てカーシェアリングサービス，Times Car Plus × Ha:mo の運用データを用いた.

2.2.2. 結果

千住は前節で紹介した条件のもと 100 回のシミュレーションを行った.

まず，受諾された要求トリップ件数を確認すると，潜在的利用者の受諾率増加とともに受諾された要求トリップ件数は大きくなる. 例えば，受諾率が 0 のとき，受諾された要求トリップ件数は 15.09 件だが，受諾率 0.4 であれば受諾された要求トリップ件数は 20.32 件となる. つまり潜在利用者 5 人のうち 2 人が協力してくれれば，本来車に乗れなかった 5 人が乗れるようになるのである. 同様に，受諾された依頼トリップ件数や依頼トリップ受諾割合 0 のときの要求トリップ件数との差も，潜在的利用者の受諾率増加とともに大きくなる. しかし，依頼トリップ 1 件に対する要求トリップ増加件数は受諾率 40%と 60%のときが最も大きくなり，その後はほとんど変化しないことがわかった.

千住の研究では再配置問題に対して依頼トリップを導入することでこれを解決しようと試みた. どのユーザーに対して再配置を依頼するかどうかはサービスの実運用データを分析して決定した. また，利用者の再配置受託率が及ぼす結果について考察を行い，受託率が 40%～60%で要求トリップが最大化されることをシミュレーションにより確かめた.

2.3. Car relocation for carsharing service: Comparison of CPLEX and Greedy Search [19]

1 章で述べたような車両の偏在によってステーションの空きがなくなり車両を返却できない，もしくは利用可能な車両がステーションに存在しないことによるサービスの利用不可といったリスクを回避することが再配車の主たる目的である. これに対し，UTBM の Rabih Zakaria らはこれを整数線形計画問題として定式化して解決しようと試みた. 同様に貪欲法によるシミュレーションも行った[19].

2.3.1. 研究手法

Rabih Zakaria らはこの問題を整数線形計画モデルとして定式化し，IBM 社が提供するソルバー，CPLEX での最適化に加えて独自の貪欲法によるシミュレーションを行った. シミュレーションを通して解決できる要求拒否の数と従業員の数，シミュレーションの実行時間などを比較，考察した.

このとき，利用するステーションなどのデータはパリでのワンウェイ方式カーシェアリングサービスの実運用データを用いた. 利用者がいつ，どこからどこへ向かうかといった需要に関するデータはパリの人々の移動データを基に予測，作成した. 移動時間は 2.1 節で述べた類似研究とは異なり，すべてのステーション間の移動が 1 分以内に完了するものとしていた.

2.3.2. 結果

CPLEX と貪欲法を比較すると，貪欲法を用いることで複雑な条件下であってもこの問題をより速く解決することが結論づけられた. 図 2.3.1 は貪欲法と CPLEX，それぞれによる解法における再配置

を行う従業員の数，シミュレーションの実行時間の関係を示したものである．

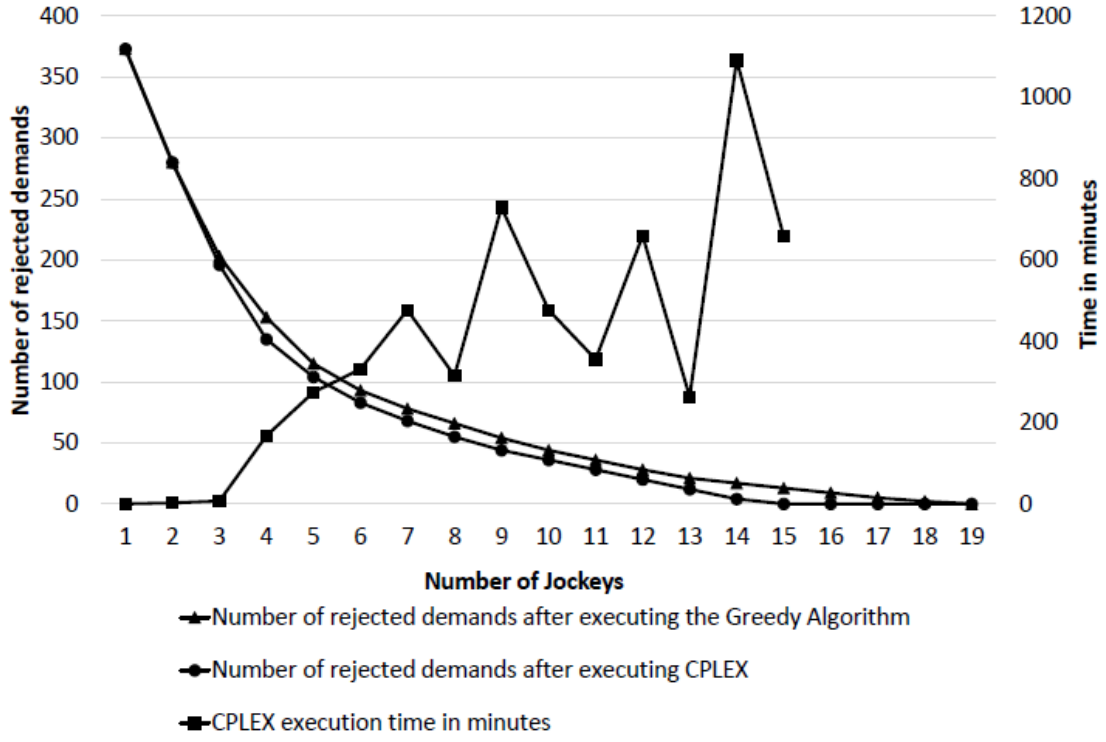


図 2.3.1 ILP VS our Greedy Algorithm results(50 stations, 10 places per station and 6 trips per car)

Rabih Zakaria らによると CPLEX による解法の場合，図 2.3.1 に示すように従業員の数などに指数関数てきに比例し，複雑な条件の場合，27 時間経っても解を得ることはできなかった．一方で貪欲法の場合，ほとんどの条件下で，1 秒程度で解を得ることができた．また，従業員の数を増やせばそれだけ要求拒否を削減することができるが，ある一定の数以上の場合要求拒否の数を著しく削減することはできなくなった．

Rabih Zakaria らは再配置問題を整数線形計画問題として定式化を行い，これを解いた．アルゴリズムとしては貪欲法がより早くシミュレーションを行え，要求拒否も削減できることがわかった．しかし，整数線形計画問題として数理モデル化している側面が大きく現実との距離感は遠い．例として移動時間を 1 分としている点や再配置にかかるコストが挙げられる．

3. 本研究の目的

1 章の研究背景ではワンウェイ方式カーシェアリングにおいて車両の偏在が発生し，これを再配置する必要があることを述べた．また，この再配置にかかるコストが日本でのワンウェイ方式カーシェアリング普及の妨げになっている．

これに対して国内外問わず再配置問題をより効率的に解決しようという動きがみられた．これらの研究の多くは再配置問題を最適化問題として扱うもので，整数線形計画問題などで定式化されることが多かった．その中でも 2.3 節では，シミュレーションを貪欲法というアルゴリズムを用いて実行し，従来の解法よりも早く実行できることが示された．しかし，2 章で説明したような類似研究の場合多くがこれを最適化問題として定式化しており，移動時間など複雑な条件は考慮されていなかった．2.1 節で述べたような利用者を用いた再配置に関する研究も存在する．2.1 節では利用者に要求トリップを依頼し，利用者が一定の確率で依頼トリップを受託する場合にどのように要求拒否数を削減できるかを考察していた．また，日本で提供されたサービスの実データを用いたシミュレーションなども見られたが，本研究では現状，実データを得られていない．

これらを踏まえて，本研究の目的は利用者による再配置を考慮することで再配置におけるコストを削減することである．実際に存在するカーシェアリングサービスのステーションデータなどを用いて，仮想的なワンウェイ方式カーシェアリングサービスのシミュレーションを行う．シミュレーションを行うことによって，仮設がどのように結果に影響するのかを考察する．

また，先行研究をうけて本研究では， (1) 需要データの作成，(2) 移動時間の考慮， (3) 価値関数の定義， (4)利用者による再配置の導入，を行う．

(1) では実際のカーシェアリングサービスの運用データが入手できない状態であるため，独自に需要データの作成をする必要がある．

(2) 2.2 節や 2.3 節で説明した類似研究ではいずれも移動時間を簡単のため考慮していなかったが，実用化を考慮して本研究ではこれを考慮することにした．

(3) は，移動時間を新たに考慮することを踏まえて，新たな価値関数を導入する．

(4) では，利用者による再配置を導入する．2.2 節で述べた類似研究同様，確率により利用者が再配置を行うかどうかを決定するが，利用者がどの程度の確率で再配置を行うかわからない状態でシステムはどのようにふるまうべきなのかといったことについて検討する．

4. 研究手法

本章では本研究の流れおよび使用するシステム構成と手法について説明する.

4.1. 再配置の流れ

1 章の研究背景で述べた要求拒否と再配置の詳細な流れについて説明する. 図 4.1.1 は要求拒否と再配置の流れについての図である.

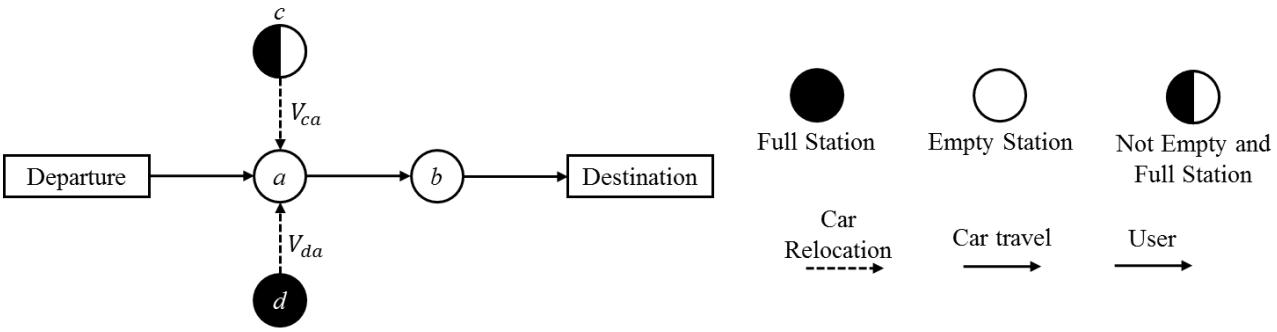


図 4.1.1 要求拒否と再配置の流れ

図 4.1.1 において利用者はステーションaから車両を利用し, ステーションbで車両を返却して目的地へ向かう. このときステーションaは空車の状態なので利用できず, さらに近くのステーションdでは満車のため他の利用者が車両を返却することもできない. そのため, ステーションdもしくは利用可能な車両が存在するステーションcから車両をステーションaへ移動させてこれを解決する. このとき, どちらのステーションから車両を移動させるかを何らかの価値関数V_{ij}にて判断する必要がある. この価値関数の定義については 4.5.3 節で詳しく説明する. $V_{da} > V_{ca}$であるとすればステーションdからステーションaへ再配置を行うことで利用者は目的地へ最短ルートでたどり着くことができる.

要求拒否は上記のように二種類に分けることができる. 一つ目はステーションが満車で車両が返却できない状態である. これについて Rejected Demand because station is Full を省略して RSF と呼ぶ. 二つ目はステーションが空車で車両を利用できない状態である. これについても同様に Rejected Demand because station is Empty を省略して RSE と呼ぶ[19].

4.2. システム構成

システムは大きく分けて四つのプロセスに分けることができる.

1. パラメータの記述
2. データの取得, 整形
3. シミュレーション
4. 結果の出力

まずシミュレーションで用いるパラメータを定義, 次にパラメータに応じたステーションの位置や移動時間などのデータをナビタイムジャパン社が提供する NAVITIME API[20]から取得し, データ整形を行う. パラメータとデータからシミュレーションを行い, 結果を出力するという流れになっている.

このサイクルの中で, 結果からパラメータ, 導入したアルゴリズムやシステムの有効性や妥当性を考察する. 各プロセスでの使用手法について次節で説明する.

4.3. パラメータについて

パラメータを変更することで実験条件を変更するため，シミュレーションの一つ目のプロセスではパラメータを記述する．ただし，現状では日本での実証実験やサービスの導入例が少ない．したがってユーザー数や時間当たりの利用者の利用頻度については実際のデータが入手できておらず暫定の固定値として扱っている．また，これらの暫定的なパラメータに関しては将来実証実験の結果などを即時反映できるよう設定項目に追加した．設定するパラメータについて表 4.3.1 に示す．

表 4.3.1 シミュレーションで用いたパラメータとその説明

パラメータ	意味
T	シミュレーション内での試行時間
N	シミュレーションで考慮するステーション数
SELECT_RATIO	考慮するシミュレーションの間隔を表す
RANDOM_MODE	需要行列の生成に用いる確率分布
MU	正規分布における平均
SIGMA	正規分布における分散
LAMBDA	ポアソン分布における平均および標準偏差
RELOCATION	再配置をするかどうか決定するブール値
TRAVEL_TIME	移動時間を考慮するかどうかを決定するブール値
N_e	考慮する従業員の数
USER_RELOCATION	利用者による再配置を考慮するかどうかを決定するブール値

4.4. データの取得，整形

二つ目のプロセスではパラメータに応じてデータを取得し，必要に応じて整形している．

4.4.1. ステーションデータ

まず，ステーションに関するデータについて説明する．ステーションのデータは図 4.1.1 に示したようにある地点から近い順番に任意の数だけデータを取得する．現段階では十分な数だけステーションが存在するのはある一定の地域に限られるからである．また，ステーションが密集している地域である場合，単純に中心地から近い順番にデータを取得したときに非常に距離が近いステーション群となってしまうため，4.3 節で述べた SELECT_RATIO という整数値をとるパラメータを設けている．図 4.4.2 と図 4.4.3 にその様子を示す．

図 4.4.1 ステーションデータの取得範囲[21]

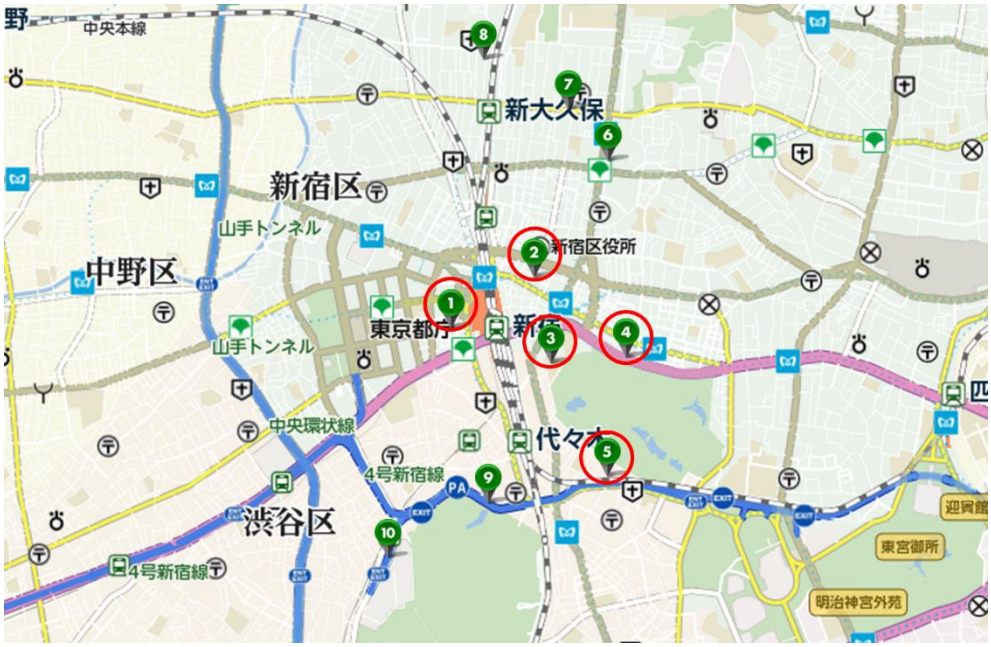


図 4.4.2 SELECT_RATIO=1 の場合のステーション選択の様子

図 4.4.2 はパラメータが 1 の時のステーション選択の様子であるが，このときは中心地から近いステーションから順に選択し，シミュレーションに用いる．

図 4.4.3 はパラメータが 3 の場合である．この場合，中心地から 1 番目に近いステーション，4 番目に近いステーション，7 番目に近いステーション，10 番目に近いステーションといったように 2 つずつ飛ばし飛ばしでデータを取得する．

これらのデータからは，ステーション数を N とすると，各ステーション間の距離を格納する$N \times N$行列 (以下 S_distances と表記) ，各ステーション間の移動時間を格納する$N \times N$行列 (以下 S_traveltimes と表記) ，各ステーションのカーシェアリング用車両収容台数を格納する$1 \times N$行列 (以下 S_vhecles と表記) を生成している．

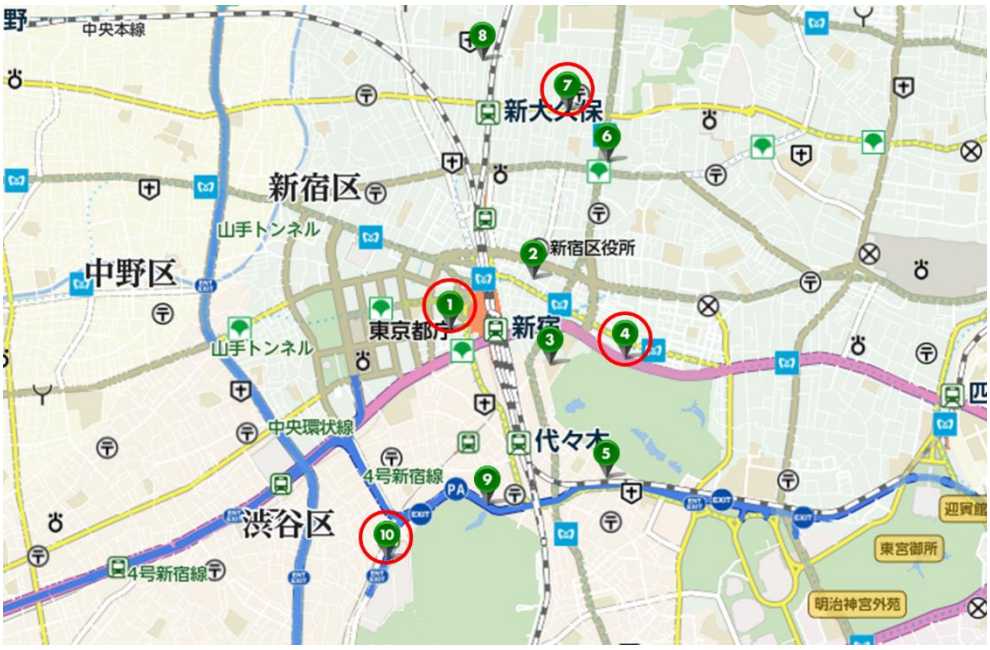


図 4.4.3 SELECT_RATIO=3 の場合のステーション選択の様子

4.4.2. 需要行列

シミュレーションを行う際に利用者がいつどのステーションからどのステーションへ向かおうとしているかというデータが必要になる. 2 章先行研究で述べたように, 先行研究ではパリでの実運用で得られた乗客の利用データから需要予測データを作成し, これを用いた. しかし, 本研究では日本での予測データが作成困難であること日本での実運用を考慮するため, 需要データを乱数より生成することとした.

シミュレーション上で扱うため, 需要データは時間, 出発するステーション, 目的のステーション, これら 3 つの情報を次元にもつ 3 次元行列として定義し, この$T \times N \times N$行列を需要行列と呼ぶことにする. ただし, T はシミュレーションの試行時間, N は考慮するステーションの数である. 需要行列 D の時刻 t における要素 D_t を式(1)に示す.

$$D_t = \begin{pmatrix} 0 & \cdots & d_{t0n} \\ \vdots & \ddots & \vdots \\ d_{tn0} & \cdots & 0 \end{pmatrix} \tag{1}$$

D_t における要素d_{tij}は時刻 t においてステーション i にからステーションj に向かう需要の数を示す.

利用者がいない場合もしくはインデックスが $i = j$ となる場合には値 0 をとる.

需要データの生成には乱数を用いるが, 本研究では正規分布とポアソン分布による乱数, 二通りの分布にしたがう場合を扱った. 図 4.4.4 は正規分布とポアソン分布の確率密度を比較したものである. 図 4.4.4 を見ると, 条件を揃えればその確率分布はほとんど同様だということがわかる. しか

し，パラメータの設定方法を考慮するため，二通りの確率分布を扱うことにする．

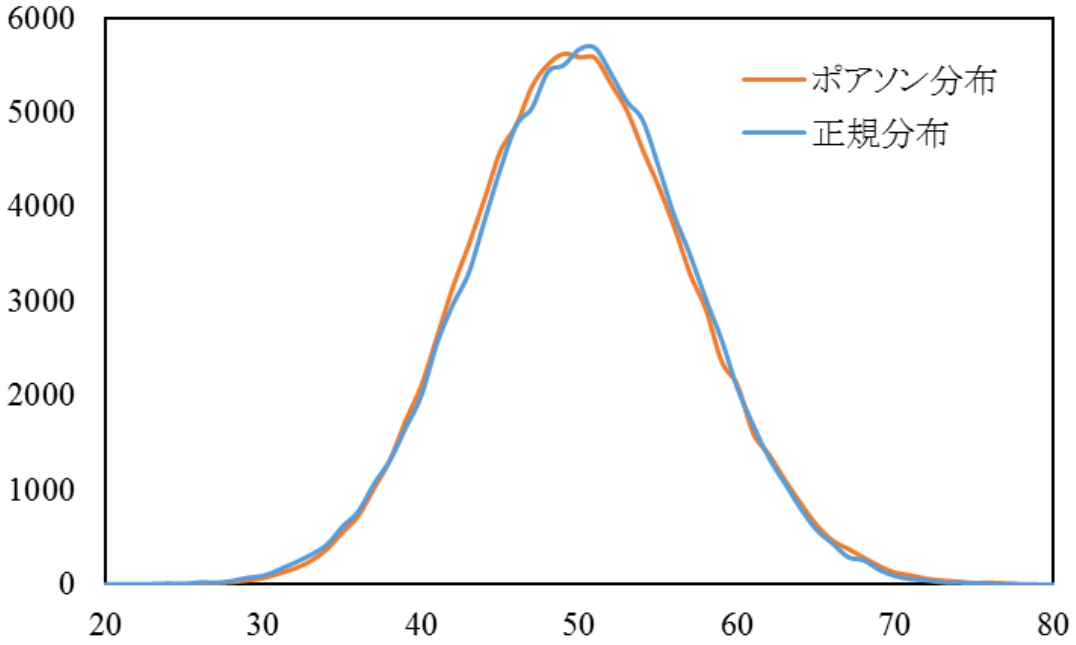


図 4.4.4 正規分布とポアソン分布の確率分布の比較

正規分布とは確率密度関数が式(2)で与えられる連続変数の分布である[22]．

$$f_{(x)} = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2}$$

ただし，μ は平均，σ^2 は分散を表す．正規分布にしたがう乱数は整数値ではなく，負の値も取り得る．そこで本研究では平均，分散を調整して乱数を四捨五入，また，値が負の値の場合は 0 とした．また，単位時間あたりの需要数の予測値をdとしたときの平均μの求め方について述べる．需要行列は$T \times N \times N$の行列であり，要素の数はTN^2である．したがって乱数を生成する際のパラメータμは一時間当たりの需要数として考慮すると式(3)のようにして計算する．

$$\mu = \frac{d}{60N^2} \tag{3}$$

ポアソン分布とは単位時間あたりに平均 λ 回起こる現象が，ある期間に X 回起こる確率にしたがう分布のことである．ポアソン分布の確率密度関数は式(3)で与えられる[23][24][25]．

$$f_{(x\,|\,\lambda)} = \frac{\lambda^x e^{-\lambda}}{x!} \ (x = 0, 1, 2, \dots) \tag{4}$$

ポアソン分布の場合，乱数は値として 0 および自然数をとる．したがって正規分布のように乱数

を加工する必要はない. ポアソン分布では平均, 分散ともにλである. したがって式(3)と同様にして計算する. λの計算式を式(5)に示す.

$$\lambda = \frac{d}{60N^2} \tag{5}$$

シミュレーションを行う際は，λ を 4.3 節で述べたように暫定の固定値として扱う.

4.5. シミュレーション

4.1 節から 4.4 節までを踏まえて実際に行うシミュレーションの手法について以下で説明する.

4.5.1. 需要の処理

以下の図 4.5.1 は 4.2 節と 4.3 節で述べた処理後のシミュレーションの需要の処理フローについて説明した図である. ただし, この処理は再配置の処理を含んでいない.

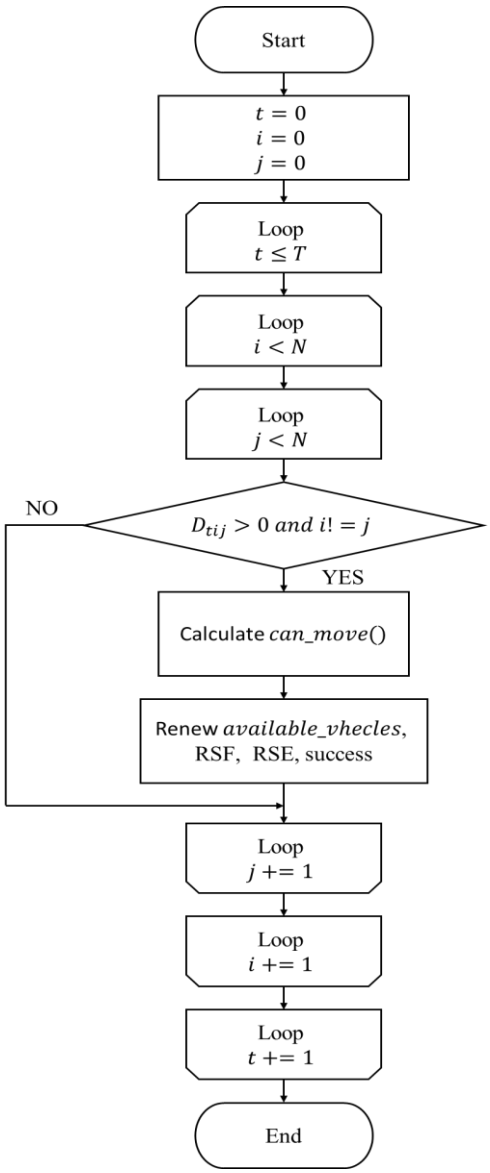


図 4.5.1 需要の処理

ここでシミュレーションの試行時間を T , 考慮するステーション数を N , 時刻ごとステーションごとの利用可能車両台数を格納している$N \times T$行列を $available\_vhecles$, 需要行列をD, $RSF, RSE, success$はそれぞれの数を示している. 需要行列のすべての要素について調べ, もし需要があれば移動可能かどうかを調べる関数can_moveを実行し, その結果を$RSF, RSE, success$に格納している. このとき同時に$available_vhecles$の更新も行っている.

4.5.2. 貪欲法

本章以降では再配置に関する処理について説明する. この再配置の選択に関しては貪欲法というアルゴリズムを利用しており, 本節ではまず貪欲法について説明する.

貪欲法は問題の要素を複数の部分問題に分割してそれぞれを独立に評価を行う. 本研究においては, ある瞬間(時刻)にどのステーションからどのステーションへと再配置を行うかという問題がこの部分問題にあたる. すなわち貪欲法とはこの部分問題に対して評価値の高い順に選択を行うことで解を得るというアルゴリズムのことである.

動的計画法と異なる点としては, 状態を常に一つしか保たないということが挙げられる. したがって一度選択した要素を再考することは無い. 本研究のような選択肢が膨大である最適化問題においては最適解を得ることは困難である. このことから貪欲法といったアルゴリズムを用いることで, できるだけ精度の良い近似解を求めるために貪欲法を利用する.

以下に整数計画問題の代表例であるナップサック問題への貪欲法の適用例を示す. ナップサック問題とは最大容量がWで与えられているナップサックに対して, 品物の集合$I = \{1,2,\cdots,N\}$, $i \in I$において重みw_i, 価値v_iである各品物を可能な限り詰込んで, 価値を最大化するというものである. 貪欲法ではこのすべての品物$i \in I$についてw_i/v_iを計算し, この値が高い順番に可能な限り品物を入れていく. このように, 貪欲法ではナップサック問題でいうw_i/v_iのような選択基準となる評価値が必要である. 貪欲法を本研究に適用する際の評価値について以下で説明する.

4.5.3. 価値関数

4.5.2 節で述べた評価値を本研究では価値関数V_{ij}と呼ぶことにする. これはステーションiからステーションjへ移動する再配置の価値を示している. まず, 2 章で述べた先行研究で用いられていた価値関数を式(6)および式(7)に示す[5].

$$cost = \frac{1}{E - G + 1} + \Delta \tag{6}$$

$$\Delta = \frac{1}{t_G + 1} \times G - \frac{1}{t_E + 1} \times E \tag{7}$$

ここで E は再配置によって取り除かれる要求拒否の数, G は再配置によって新たに生成される要求拒否の数を示す. t_Gはその要求拒否が生成されるまでの時間, t_Eはその要求拒否が取り除かれるまでの時間のことである.

式(6)からもわかる通り, 価値関数は再配置の価値をコストとして表現しており, このコストが低い選択肢を選ぶ必要がある. しかし, 本研究ではこの選択肢の評価値としてのコストと実際の再配置が要するコストとを区別するため, 前者を価値関数, 後者はそのままコストと呼ぶことにする.

本研究で用いる価値関数V_{ij}を式(8)に示す.

$$V_{ij} = \frac{1}{E - G + 1} + \Delta + w_t t_{ij} \tag{8}$$

式(7)に関しては本研究でも同様に用いる. 式(8)において新たに導入した変数w_t, t_{ij}はそれぞれ移動時間の重みと任意のステーションi, j間の重みである. 本研究では現実の移動時間を考慮するため新たに移動時間の項を追加した.

4.5.4. 再配置の分類

再配置の具体的な手法について説明する. 4.2 節で要求拒否は二種類存在することを説明した. これらに対して再配置は一つの要求拒否を解決できる再配置が二種類, 二つの要求拒否を同時に解決できる再配置が二種類, 合計三種類存在する. これらについて以下で順番に述べる.

一つ目は RSE を解決する再配置である. その様子を図 4.5.2 に示す.

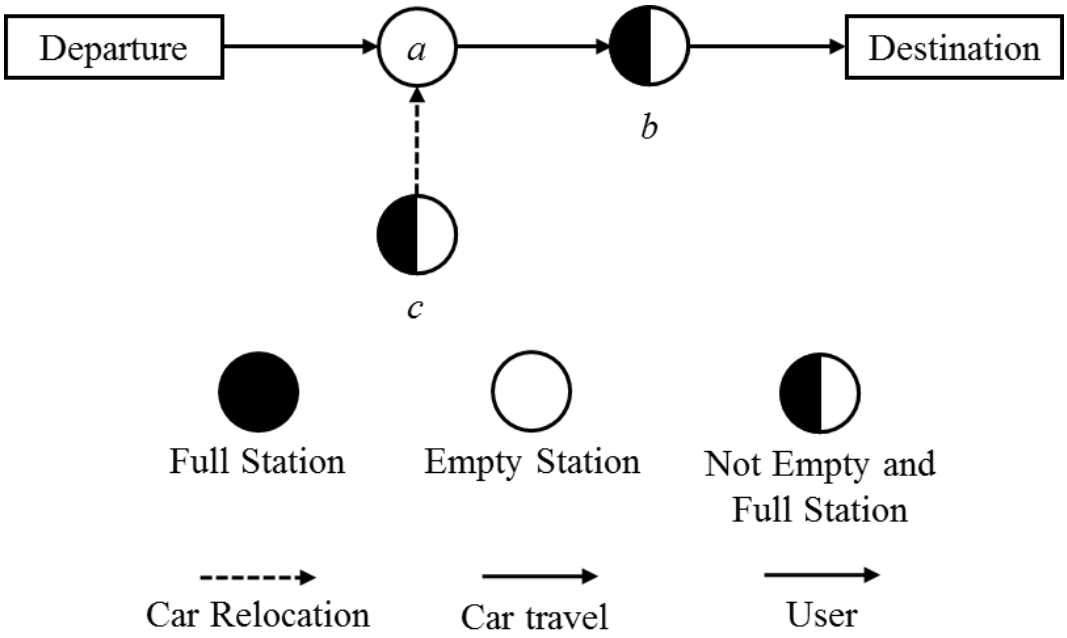


図 4.5.2 RSE を解決する再配置の様子

利用者はステーションaで車両を利用し, ステーションbへ返却して目的地へと向かう. この際, 図 4.5.2 のようにステーションaが空車である場合, つまり RSE である場合ステーションcから余っている車両をステーションaへ再配置することで RSE を解決することができる.

二つ目は RSF を解決する再配置である. その様子を図 4.5.3 に示す. この場合も利用者はステーションaから車両を利用し, ステーションbへ返却するが, 図 4.5.3 のようにステーションbが満車の場合, つまり RSF である場合車両を返却することができない. これに対して近隣で空きがあるステーションcにステーションbの車両を移動させることで RSF を解決する.

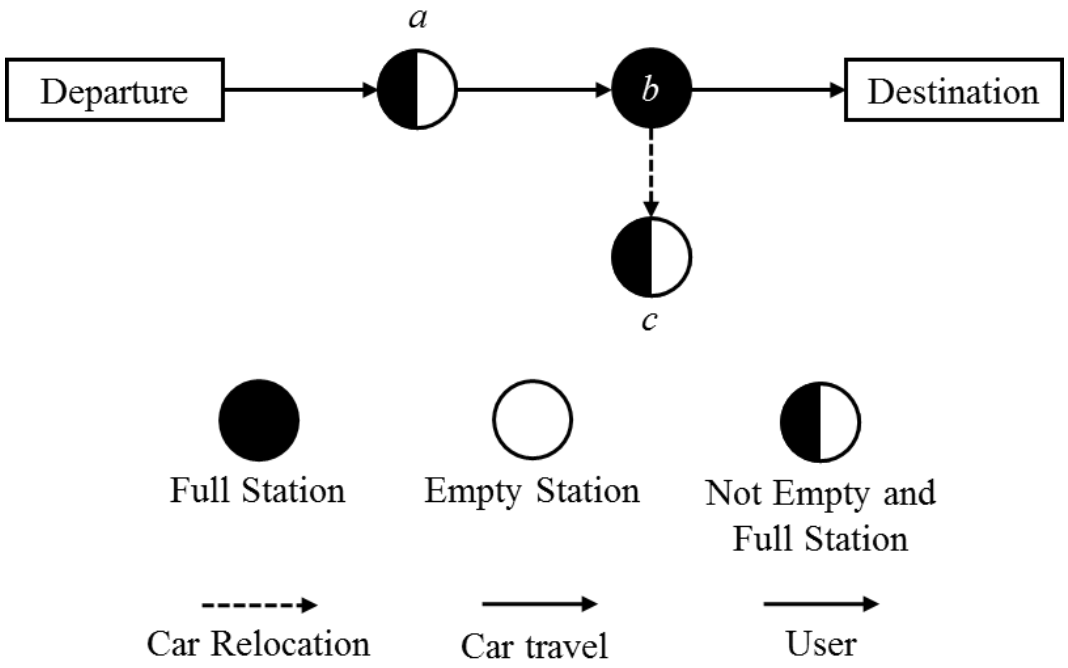


図 4.5.3　RSF を解決する再配置の様子

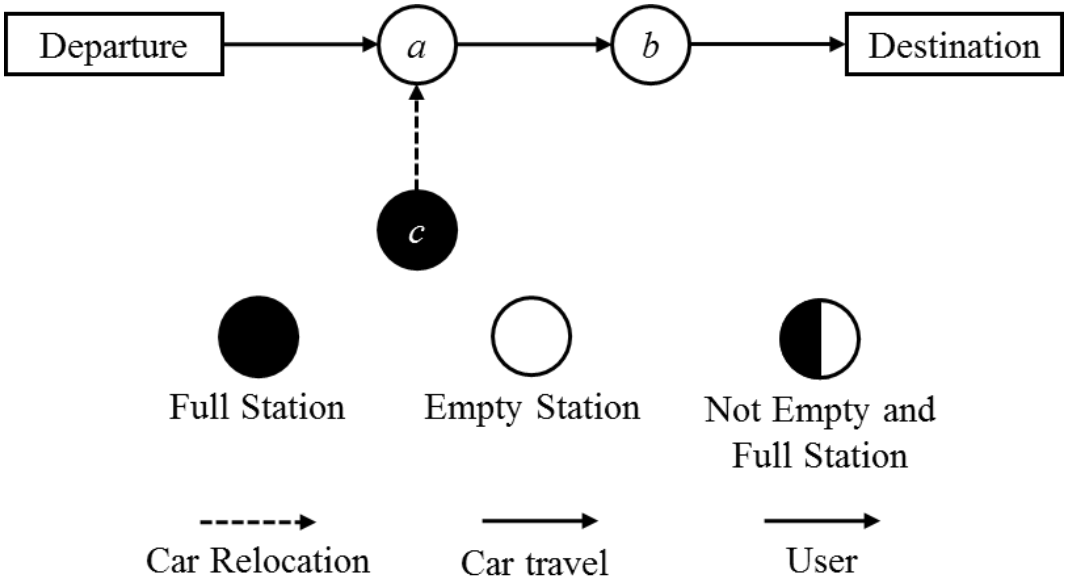


図 4.5.4　RSF と RSE を同時に解決する再配置の様子

図 4.5.4 は RSF と RSE を同時に解決する再配置の様子を示したものである．車両を利用しようとしているステーションaが空車で，同時に近隣のステーションcは満車の状態となっている．この場合，ステーションcからステーションaに車両を再配置することで二つの要求拒否を同時に解決することができる．

4.5.5. 再配置の処理

4.5.4 節で述べた各再配置について具体的な処理方法を以下に示す. 図 4.5.5 は三通りの再配置についてそれぞれを探索する処理のフローチャートである.

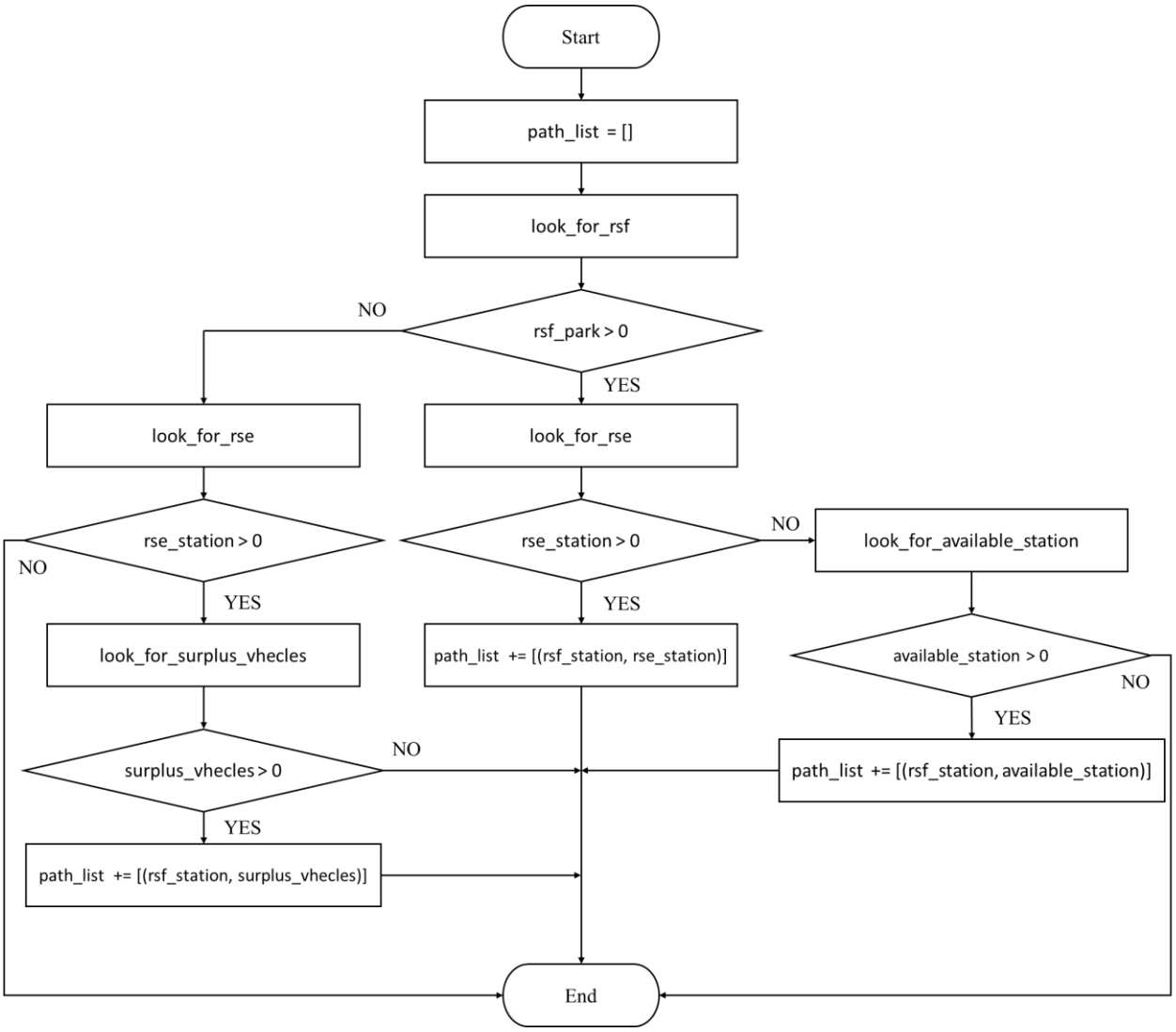


図 4.5.5 再配置探索のフローチャート

図 4.5.5 において$path_list$は再配置可能な経路を格納しておく配列, $look\_for\_rsf$, $look\_for\_rse$, $look\_for\_available\_park$, $look_for_surplus_vhecles$はそれぞれ, RSF となるステーション, RSE となるステーション, 駐車可能なステーション, 車両が余っているステーションを探索する処理である. そしてそれらの探索で得られたステーションを格納しておく変数が$rsf_station$, $rse\_station$, $available\_station$, $surplus_vhecles$となっている.

それぞれの条件に当てはまるステーションを探索後, RSF であるステーションから RSE であるステーション, RSF であるステーションから余分に空きがあるステーション, 余分に車両を有するステーションから RSE であるステーション, これらいずれかの組み合わせの経路を再配置可能な経路として格納している. $look_for_rsf$などの具体的な処理内容については以下に示す.

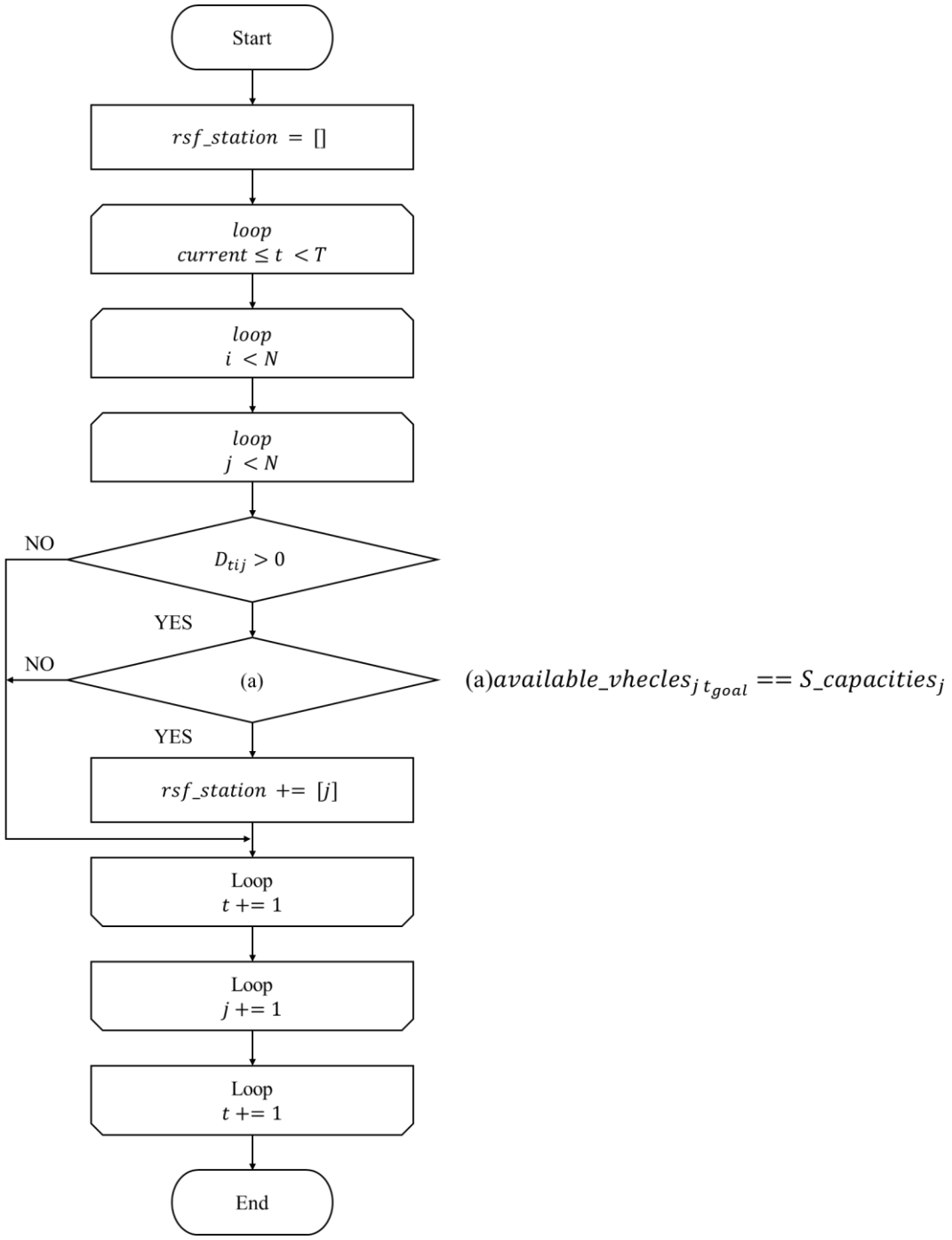


図 4.5.6 look_for_rsf の処理内容

図 4.5.6 は$look_for_rsf$の具体的な処理内容を示している． ただし, $available_vhecles_{it}$とは時刻tにおけるステーションiの利用可能な車両台数, t_{goal}はステーションiから出発してステーションjに到着するまでの時間, $S_\text{capacities}_i$はステーションiにおける最大収容可能台数を示している. つまり, 現在時刻$current$以降のすべてのステーションの組み合わせにおいて需要の有無を確認し, もしある場合には収容可能かどうかを調べる. そして収容できない場合には RSF とみなして

$rsf_station$に格納する. 一連の処理を通してこの処理群では任意の時刻における RSF となるステーションの配列を返す.

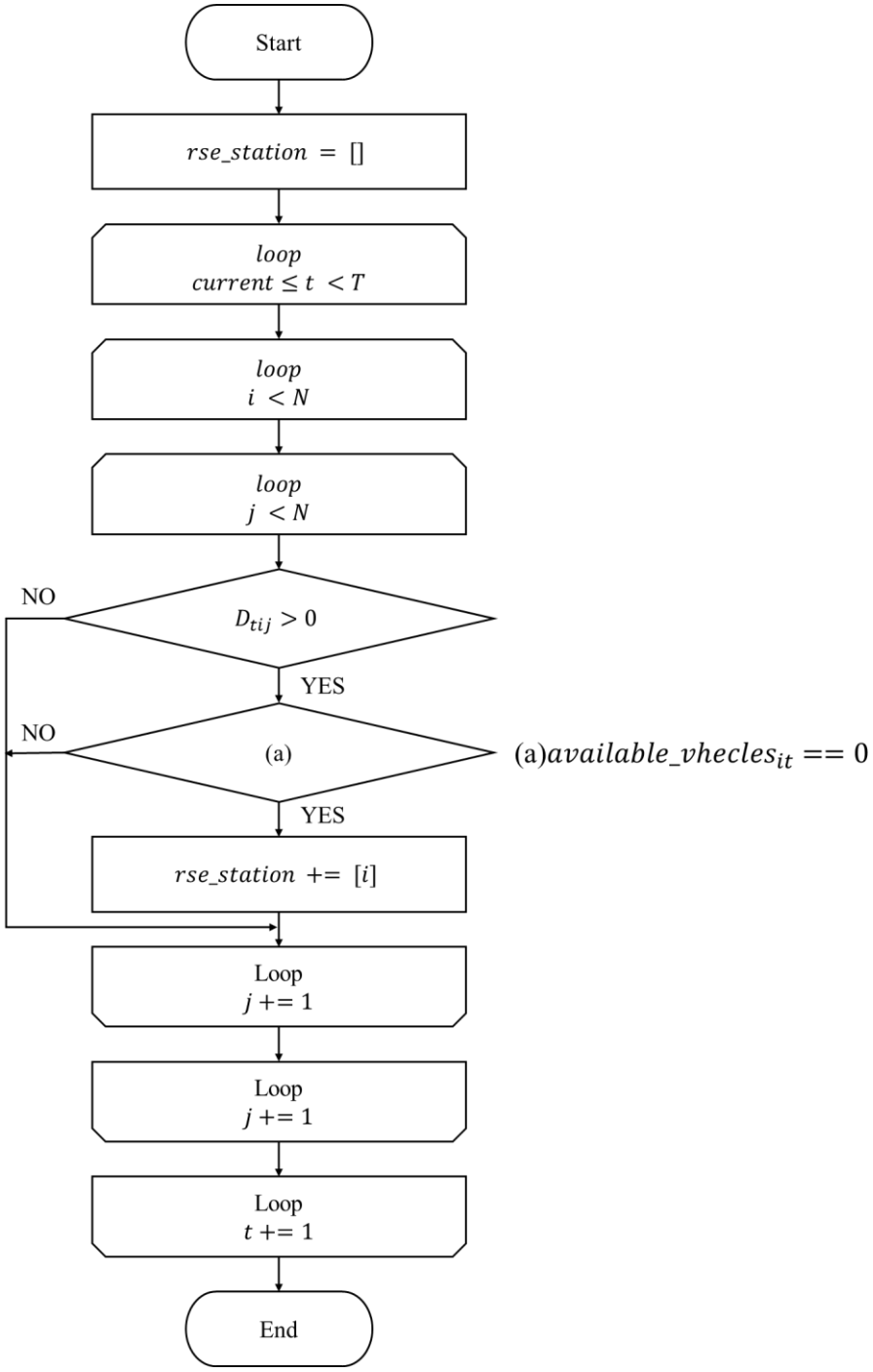


図 4.5.7 単体 RSE における look_for_rse の処理内容

$look_for_rse$は状況によって二通りの処理を行う. 図 4.5.5 において$look_for_rse$の処理は二か所で行われており, それぞれ RSF が存在する場合と存在しない場合である. これらについて RSF が存在する状態で探索する RSE を RSF-RSE, 存在しない状態では単体 RSE と呼ぶことにする.

図 4.5.7 に単体 RSE における$look_for_rse$の処理内容を示す. 図 4.5.7 において$rsf_station$は RSF となるステーションを格納しておく配列, $current$は現在時刻を表す. 現在時刻以降のすべてのステーションの組み合わせについて調べ, 需要が存在し尚且つ出発地のステーションiに車両が存在しない場合に単体 RSE とみなして$rse_station$に格納する. 一連の処理を通してこの処理群では任意の時刻における RSE となるステーションの配列を返す.

RSF-RSE における look_for_rse の処理内容を図 4.5.8 に示す. ただし, $rse_station$は RSE となるステーションを格納する配列, rsfは$look_for_rsf$で得られた RSF となるステーション, t_{rsf}は対象の RSF が発生する時刻, ステーションiは RSE となり得るステーション, t_{goal}は RSF となるステーションからステーションiへの移動が完了するまでの時間である. 図 4.5.8 中の条件分岐では図中の式(a)~(d)をすべて満たすときに真となる. 式(a)ではまず時刻t_{end}にて需要が存在するかを調べる. 式(b)では出発元の RSF となるステーションとステーションiが同一でないことを確認している. 式(c)は時刻t_{end}における需要がステーションiに存在する車両台数を上回っている場合に真となる. 式(d)は時刻t_{start}において RSF となるステーションに車両が存在するかどうかを調べている. 式(d)は$t_{start} \neq t_{rsf}$であるために必要となる.

つまり, この処理では現在時刻から次の RSE が発生するまでの間に, 次の RSF となるステーションからたどり着くことができるステーションを探索している. ステーションiがこの条件を満たせば, $rse_station$に格納し返り値とする.

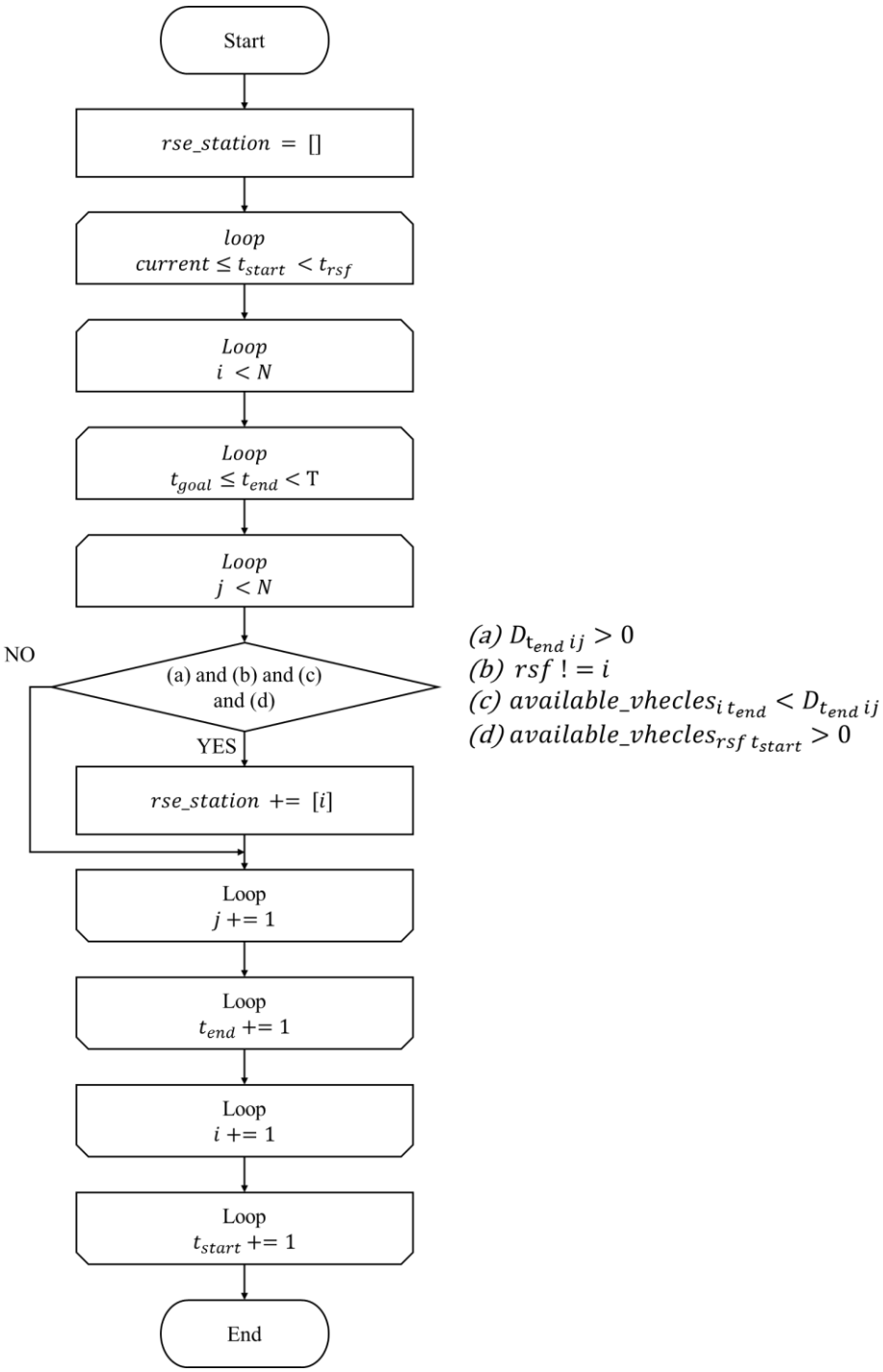
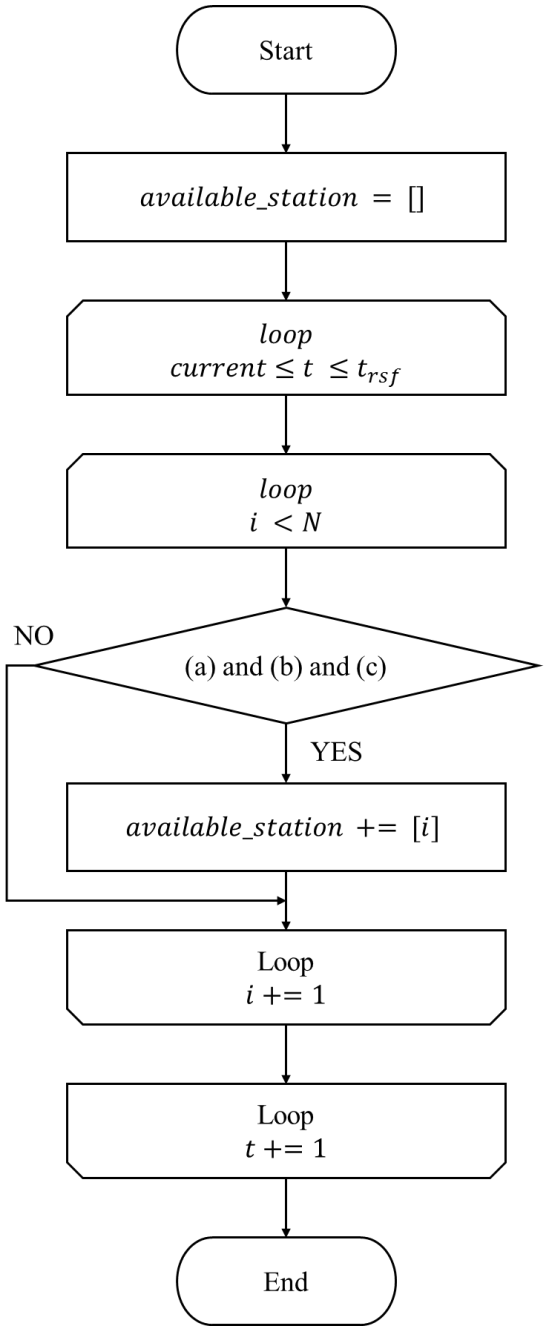


図 4.5.8 RSF-RSE における look_for_rse の処理内容



(a) $i\ !=rsf$
(b) $available\_vhecles_{i\ t+t_{goal}} < S\_capacities_i$
(c) $available\_vhecles_{rsf\ t} > 0$

図 4.5.9 look_for_available_station の処理内容

図 4.5.9 に$look_for_available_station$の処理内容を示す. ただし, $available_station$は空きがあり駐車が可能なステーションを格納する配列, t_{rsf}は次の RSF が生じる時刻, t_{goal}は RSF となるステーションからステーションiへと移動が完了する時刻を表す. この処理では現在時刻から次の RSF が生じるまでに, RSF となるステーションから移動可能なステーションが存在するかを調べるものである. 図中の式(a)では移動可能なステーションが RSF となるステーションと同一でないことを調べている. 式(b)では時刻t_{gola}において移動先のステーションiに空きがあるかどうかを調べている. 式(c)は RSF となるステーションが任意の時刻において利用可能な車両があるかどうかを調べている. これら式(a)~(c)をすべて真で満たすときステーションiを$available_station$へと格納し, これを返り値とする.

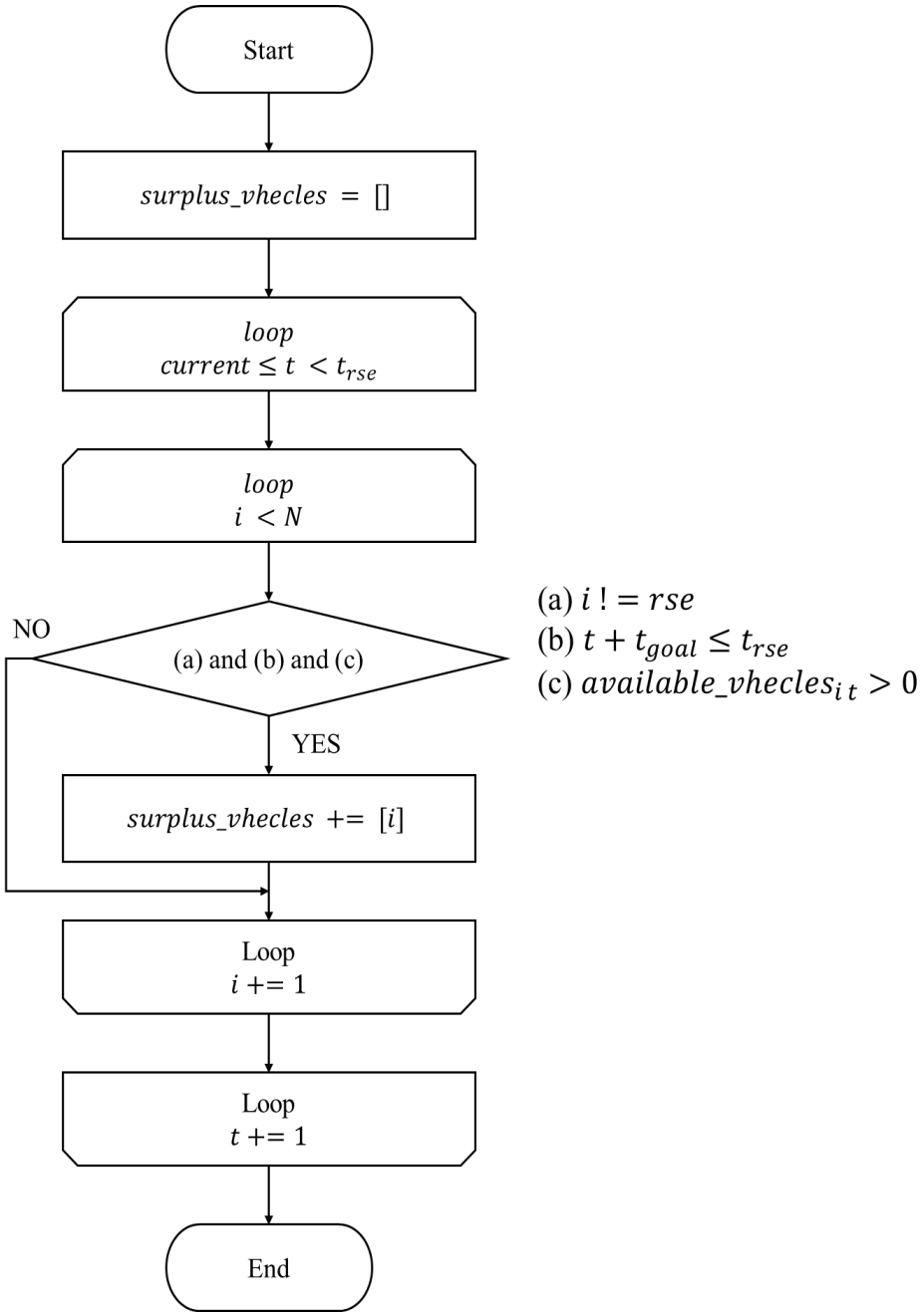


図 4.5.10 look_for_surplus_vhecles の処理内容

図 4.5.10 に$look_for_surplus_vhecles$の処理内容を示す. ここで$surplus_vhelces$は車両を余分に保持しており他のステーションへと供給可能なステーションを格納する配列, t_{rse}は次の RSE が生じる時刻, t_{goal}は車両を供給可能なステーションiから RSE となるステーションへの移動時間を表す.

この処理では現在時刻から次の RSE が生じるまでの間に RSE となるステーションへと車両を供給できるステーションを探索しており, 図 4.5.10 中にある式(a)は RSE となるステーションと供給可能なステーションが同一でないこと, 式(b)では車両を供給完了する時刻が, RSE が生じる時刻を

超えていないこと，式(c)は任意の時刻においてステーションiが他のステーションへと車両供給可能かどうかをそれぞれ保証している．条件に合うステーションiが存在すれば$surplus_vhecles$へと格納し，これを返り値としている．

4.5.6. 利用者による再配置導入

本研究では利用者による再配置を導入することで移動コストの削減を図る．利用者uが再配置を行うかどうかは利用者ごとにランダムに生成した確率Puで定義する．この確率を用いた新たな価値関数$W(u)$と，ある閾値を比較して再配置を利用者に依頼するか，従業員が行うかを決定する．$W(u)$を式(9)に示す．

$$W_{(u)} = \frac{P_u}{V_{ij}} \tag{9}$$

再配置可能な経路が複数存在する場合はそれぞれを式(8)で定義した価値関数にて比較を行う．

また，利用者が再配置をするかどうかはシステム側には正確には定義できない．したがってシミュレーションを行う中で推定値を用いる．すべてのP_uを1として初期化する．その後再配置を依頼した数をnとして，以下の式(10)のように更新する．

$$P_u = \frac{nP_u + 1}{n + 1} \tag{10}$$

また，利用者による再配置に対して何らかの報酬を支払う場合，これをP_uに反映させる．これを基にP_uの報酬を考慮した更新式(11)に示す．

$$P_u = P_u + 0.1R_{ij} \tag{11}$$

このときR_{ij}は再配置に応じて得られる報酬である．R_{ij}は再配置をするかどうかによって異なる．再配置をした場合は以下式(12)のようになる．

$$R_{ij} = -1 \tag{12}$$

再配置を行わなかった場合負の報酬を与える．利用者が実際に損をするわけではないが，再配置の確率を追加で減少させるために負の値となっている．一方で再配置を行った場合のR_{ij}は式(13)のようになる．

$$R_{ij} = 0.1t_{ij} \tag{13}$$

移動時間が長い再配置ほど利用者にとって負荷が大きいと考えたため移動時間に比例する関数とした.

5. シミュレーション結果

本章では様々に条件を変更して行ったシミュレーション結果について結果を示す.

5.1. 需要数とステーション数の関係

4.4.2 節では需要行列を乱数により生成することを示した. 本節では乱数のパラメータを変更し, 考慮するステーション数に対する需要拒否, 受託の割合などを検証する.

5.1.1. 分布による影響

利用する乱数は正規分布にしたがう場合とポアソン分布にしたがう場合があるが, それぞれの分布の違いによる影響を検証する. 表 5.1.1 にそれぞれの分布における乱数のパラメータ, 表 5.1.2 にそれぞれの場合での需要の合計数を示す.

表 5.1.1 乱数のパラメータ

Parameter	Value
d	500
N	0.0735
μ, λ	0.0833
σ^2	0.2887

表 5.1.2 需要総数

	正規分布	ポアソン分布
需要総数	1764	2028
平均	0.0845	0.0735

パラメータの値は同様に設定しているが, 正規分布の場合負の値を 0 にしているため需要の総数は低くなっている. それぞれの需要の分布を図 5.1.1 と 5.1.2 に示す.

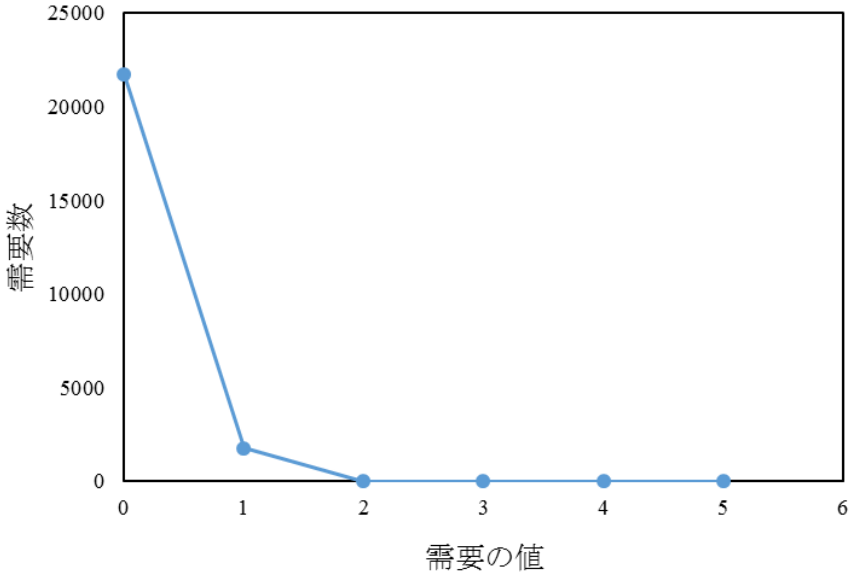


図 5.1.1 正規分布における需要の分布

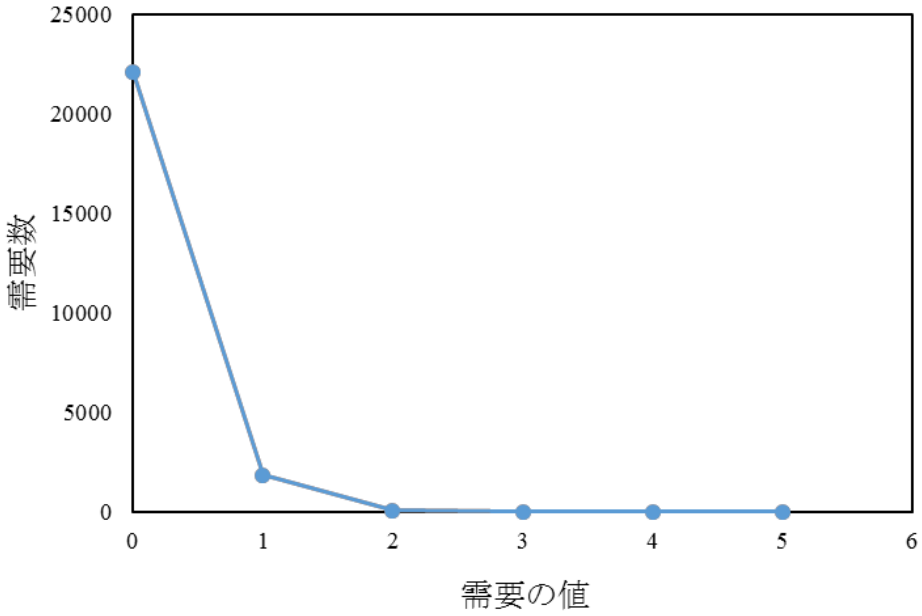


図 5.1.2　ポアソン分布における需要の分布

図 5.1.1 と図 5.1.2 を比較すると大きな差異は見られないことがわかる．したがってパラメータ設定が容易であることから本研究では以降需要行列はポアソン分布を用いて生成する．

5.1.2. 需要数と要求受託率の関係

本節では需要の総数と，すべて需要数に対する要求受託の割合（要求受託率と呼ぶ），要求拒否の割合（要求拒否率と呼ぶ）の関係について述べる．需要の総数を変化させるため，需要行列生成のパラメータλのみを変更してシミュレーションを行う．その他の条件については変更していない．λは 0 から 0.1 まで 0.001 ずつ増加させ，それぞれについてシミュレーションを行った．図 5.1.3 にλごとの要求受託率および要求拒否率を示す．

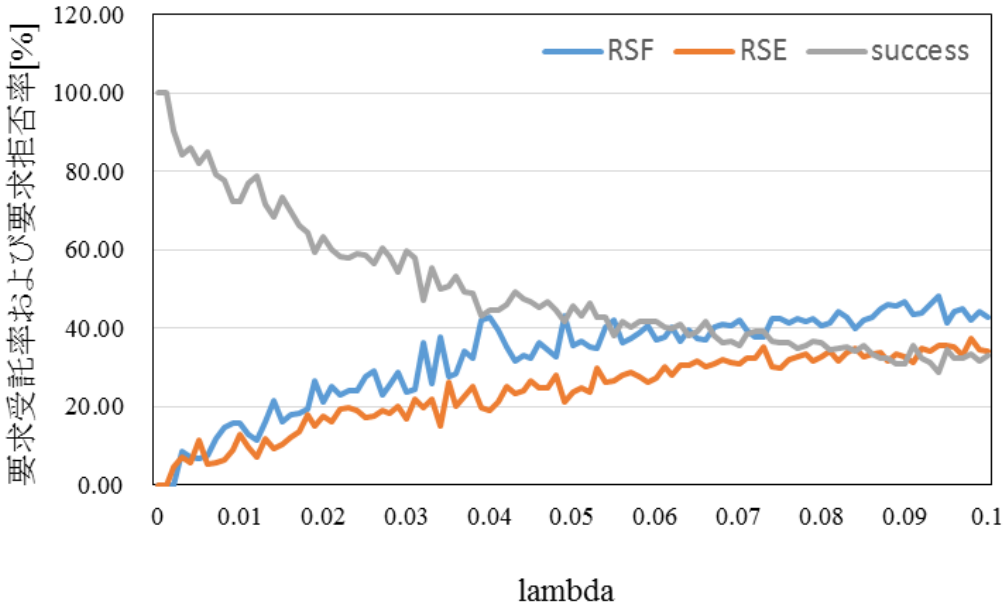


図 5.1.3　0.1 までの区間における**λ**と要求受託率および要求拒否率の関係

図 5.1.3 を見ると，λが 0.001 付近のごくわずかな区間でのみ要求受託率が 100%に近く，逆にλが 0.05 より増加しても要求受託率は大きな変化を見せない．より詳しく調べるため，同様の条件でλを 0 から 0.01 の区間で 0.0001 ずつ増加させてシミュレーションを行った．その結果を図 5.1.4 に示す．

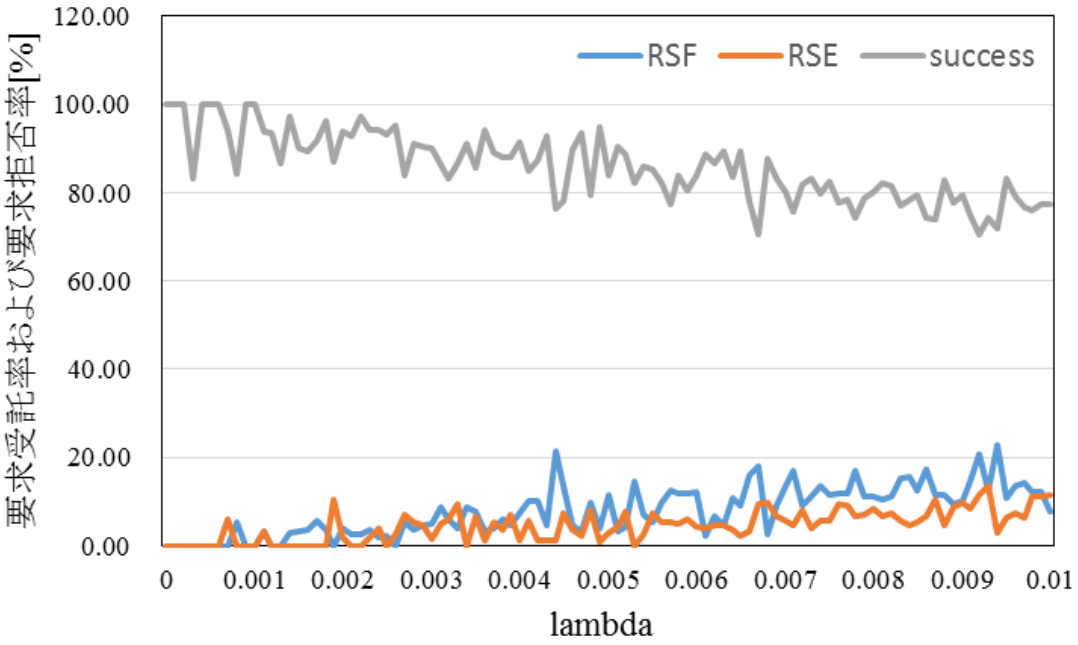


図 5.1.4 0.01 までの区間における**λ**と要求受託率および要求拒否率の関係

図 5.1.4 を見るとグラフはなめらかな曲線にならず，値にかなりばらつきがあることが分かる．また，λの値から一時間あたり，N個のステーション当たりの需要数は，$\lambda = 0.001$のとき$d = 6$，$\lambda = 0.01$のときは$d = 60$となる．図 5.1.3 および図 5.1.4 のシミュレーションではいずれも考慮するステーション数は 10 としている．10 か所のステーションで，一時間当たりの需要数が 60 というのは，毎分いずれかのステーションが利用されるということである．これらのことより現実的に考えて，λの値は 0.01 以下の値，本研究では 0.003 で固定する．

5.2. 再配置の有無

本節ではまず移動時間を考慮せずに再配置を行う場合と，再配置を行わない場合を比較して再配置の有効性について調べる．そのため表 5.2.1 の条件でシミュレーションを行った．

表 5.2.1 再配置の有効性を確認するための実験条件

Case	A	B
T	4h	4h
N	10	10
Random mode	Poisson	Poisson
Lambda	0.003	0.003
Travel time	FALSE	FALSE
Value function	-	Previous
Relocation	FALSE	TRUE
User relocation	FALSE	FALSE

また，ケースごとの要求受託率および要求拒否率を図 5.2.1 に示す．

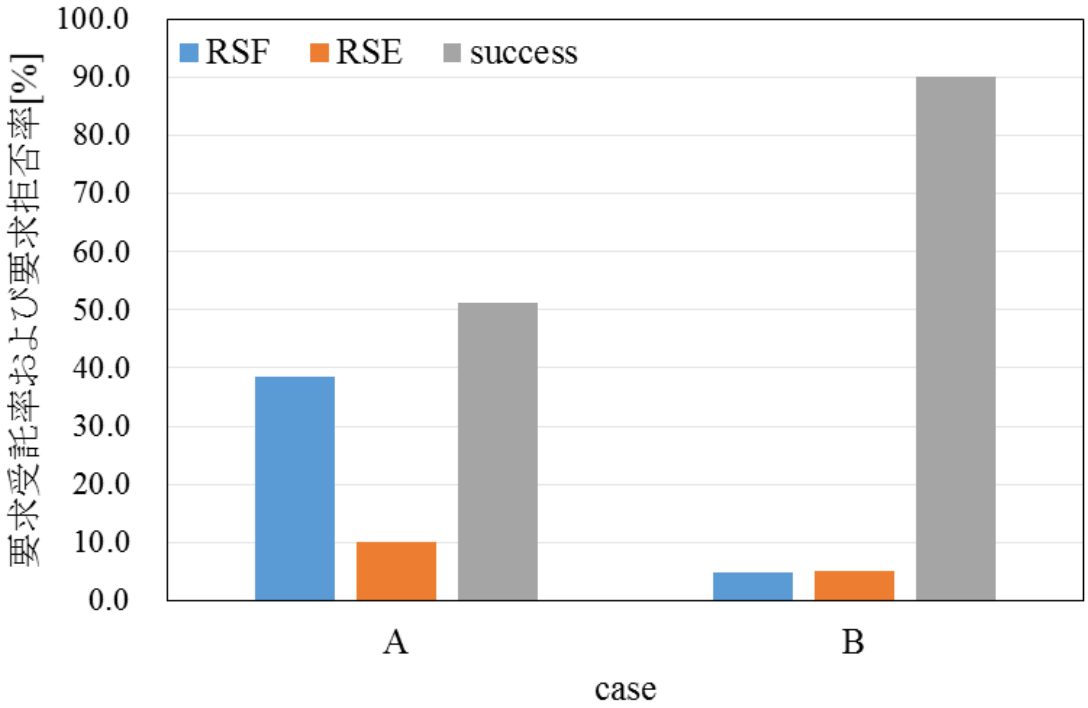


図 5.2.1 条件 A と B における要求受託率および要求拒否率

表 5.2.1 で，条件 A は再配置を行わない場合，条件 B は再配置を行う場合である．いずれの場合も再配置を行うかどうか以外の条件は同様である．

図 5.2.1 を見ると条件 A の場合約 50%の要求は受託されているが，残りの約 50%は要求拒否となってしまっている．このうち約 40%は RSF，約 10%は RSE となっている．40%の RSF は，各ステーションの収容可能台数が初期状態からほとんどないために起こっており，車両が不足するよりも先

に満車のステーションが出てきてしまうことがわかる.

5.3. 移動時間の影響

5.2 節では再配置そのものの有効性を調べた. 本節では移動時間の考慮が再配置にどのような影響を及ぼすかについて示す. 表 5.3.1 の条件でシミュレーションを行った.

表 5.3.1 移動時間の影響をみるためのシミュレーション条件

Case	A	B	C
T	4h	4h	4h
N	10	10	10
Random mode	Poisson	Poisson	Poisson
Lambda	0.003	0.003	0.003
Travel time	FALSE	FALSE	TRUE
Value function	-	Previous	Previous
Relocation	FALSE	TRUE	TRUE
User relocation	FALSE	FALSE	FALSE

再配置をしなかった場合, 移動時間を考慮せず再配置した場合, 移動時間を考慮して再配置した場合の三つのケースについて比較を行う.

表 5.3.1 において条件 C は移動時間を考慮したものである. ただし, 価値関数は先行研究のものを用いている. 条件 B では移動時間を考慮しない場合で, 条件 C と同様価値関数は先行研究のものを使用している. このように, 条件 B は移動時間と価値関数など先行研究と同じ条件としている. 表 5.3.1 の条件で行ったシミュレーション結果を図 5.3.1 に示す.

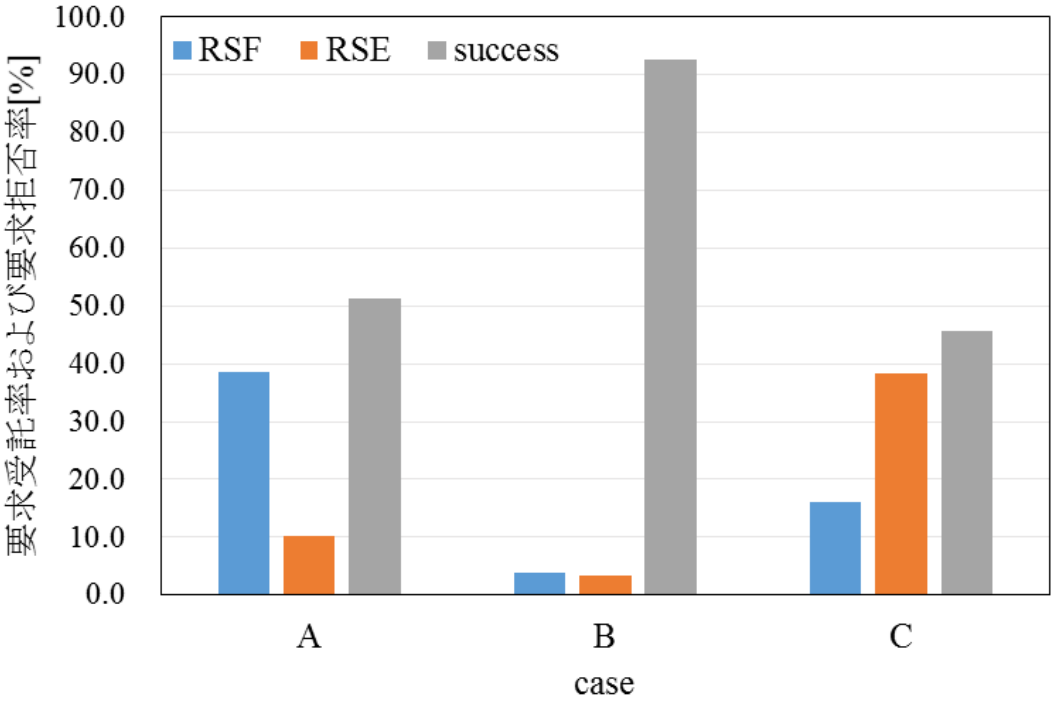


図 5.3.1 条件 A と B, C における要求受託率および要求拒否率

図 5.3.1 において再配置しない場合には要求受託率が約 50%となっているが，移動時間を考慮して再配置を行った場合，要求受託率は 50%に届かない．要求拒否率の内訳も条件 A と C では大きく異なる．また，移動時間を考慮しない先行研究と同じ条件下 (条件 B) の場合，要求受託率は先行研究と同様に 90%ほどになっていることがわかる．

図 5.3.1 において条件 A と条件 C を見ると要求拒否の中でもその内訳が大きく異なることが分かる．条件 A では RSF が約 40%を占めるのに対して条件 C では RSE が約 40%を占めている．再配置をしていない場合と比較しているにも関わらず，上級受託率は上がっていないのである．RSF に対する再配置は移動時間を考慮した際でも比較的容易に行えることが理由として挙げられる．RSF に対する再配置のうち，空いているステーションへと移動させる場合では，需要が発生するまでに移動をはじめ，その後他の空いているステーションに到着する時間は任意となるためである．一方で移動時間を考慮した場合の RSE に対する再配置では，他の車両が余っているステーションから車両が不足しているステーションへと需要が発生するまでに移動を完了させなければならない．仮に車両が余っているステーションが存在するとしても，間に合わないと判断されればこれは要求拒否とみなされてしまう．この移動時間に起因する RSE をうまく扱うための新たな仕組みが必要であったことがわかる．

5.4. 価値関数の影響

本節では価値関数の影響について述べる．4.5.3 節で述べたように，本研究で使用する価値関数は移動時間を考慮するために式(8)を用いる．5.3 節で用いた先行研究の価値関数を使用する条件 C と本研究で導入した新しい価値関数を用いる条件 D でのシミュレーション結果を比較する．シミュレーション条件を表 5.4.1 に示す．

表 5.4.1 価値関数の影響をみるためのシミュレーション条件

Case	C	D
T	4h	4h
N	10	10
Random mode	Poisson	Poisson
Lambda	0.003	0.003
Travel time	TRUE	TRUE
Value function	Previous	New
Relocation	TRUE	TRUE
User relocation	FALSE	FALSE

表 5.4.1 の条件下でのシミュレーション結果を図 5.4.1 に示す．

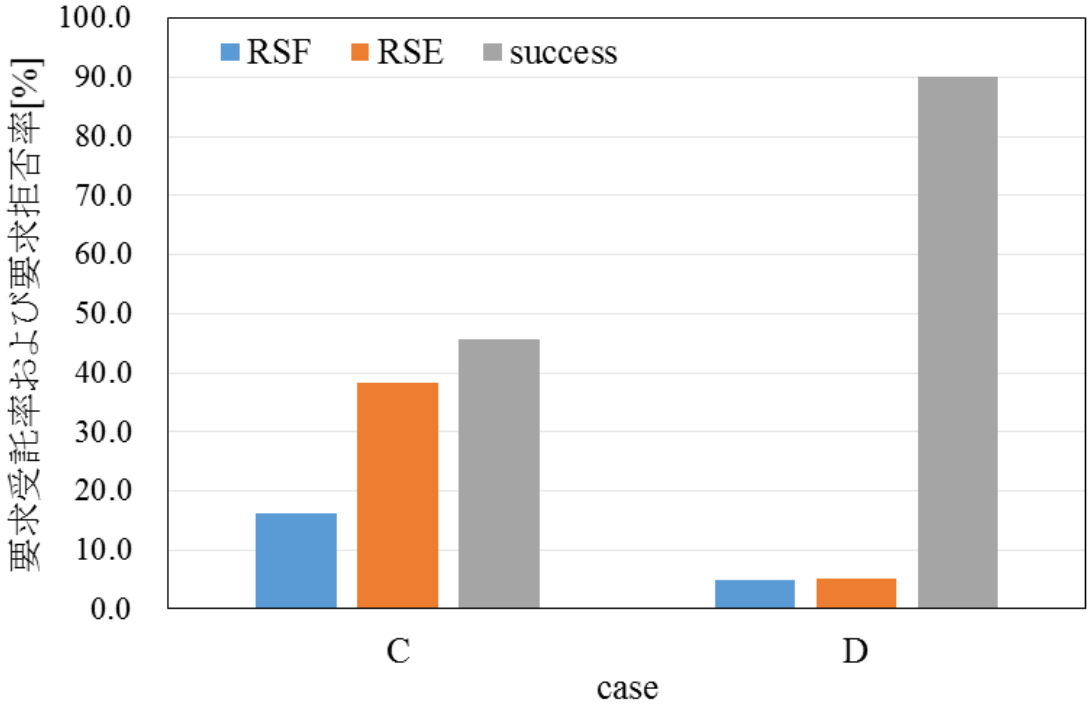


図 5.4.1 条件 C と D における要求受託率および要求拒否率

図 5.4.1 を見ると，条件 C と条件 D はいずれも移動条件を考慮しているが条件 D の場合は要求拒否率が 10%程度に抑えられており，価値関数を変更するのみで要求拒否を約 20%に減少することができていることがわかる．

6.2 節で述べた通り，新しく更新した価値関数では主に RSE の再配置を行う際に需要の発生までに間に合うかどうかが重視されているためだと考えられる．

逆に価値関数を導入したことによるデメリットも存在する．価値関数自体を導入する場合，4.4.5 節で述べたような処理をする必要がある．移動時間を考慮するのみでもこの処理は複雑さを増し，結果としてシミュレーション時間が大幅に増加する．これらの処理自体は全探索と呼ばれるもので，すべての場合におけるシミュレーションを行う，効率の悪いものである．シミュレーションの実行時間に関する実験，考察は 5.6 節でより詳しく行う．

5.5. 利用者による再配置の影響

本節では利用者による再配置の影響について述べる．4.5.6 節では，利用者ごとの確率に応じて利用者が再配置をするかどうかを決定すると述べた．よって利用者が再配置を行わない場合には要求拒否が生じてしまう．したがって利用者と従業員，両方の再配置を考慮し，これと利用者のみで再配置を行った場合を比較する．シミュレーション条件を表 5.5.1 に示す．

表 5.5.1 利用者による再配置の影響をみるためのシミュレーション条件

Case	D	E	F
T	4h	4h	4h
N	10	10	10
Random mode	Poisson	Poisson	Poisson
Lambda	0.003	0.003	0.003
Travel time	TRUE	TRUE	TRUE
Value function	New	New	New
Relocation	TRUE	TRUE	TRUE
User relocation	FALSE	TRUE	TRUE
Employee relocation	TRUE	TRUE	FALSE

条件 E と F, いずれも利用者による再配置を導入している. 条件 E では従業員の再配置も同時に行う. 表 5.5.1 の条件下でのシミュレーション結果を図 5.5.1 に示す.

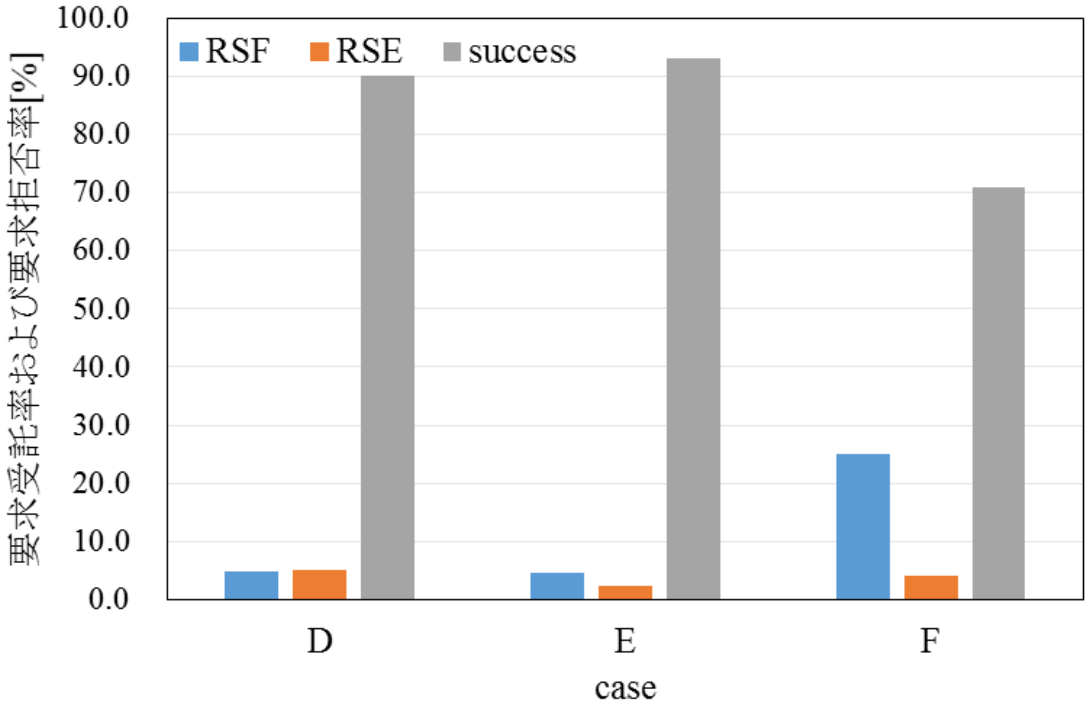


図 5.5.1 条件 D および E, F における要求受託率と要求拒否率

図 5.5.1 の条件 F (利用者のみの再配置) における結果を見ると, 単純に確率により利用者が再配置を行わない場合, それだけ要求拒否が生じてしまうことがわかる. 一度利用者に再配置を委託した場合, その後従業員は再配置を行うことができない, という条件を加えたためである. これをうけて提案したのが 4.5.6 節で述べた従業員と利用者を併用し, あらかじめ利用者か従業員, どちらが再配置を行うかどうかを新たな価値関数を用いて選択する手法である.

図 5.5.1 における条件 D (従業員のみの再配置) と条件 E (従業員と利用者, 両方の再配置) を比較すると, 条件 E における要求受託率の方が高いが, ほとんど同様の結果がでていることがわかる.

また, 再配置に要したコストについても述べる. 再配置の具体的な移動コストは燃費や経路, 交通状況など様々な条件を考慮する必要があり, たとえ考慮したとしても具体的な移動コストを求めることは困難である. そこで, 今回は移動コストの目安として移動時間を考慮する. 図 5.5.2 に条件 D および E, F における従業員の再配置に要した総移動距離を示す.

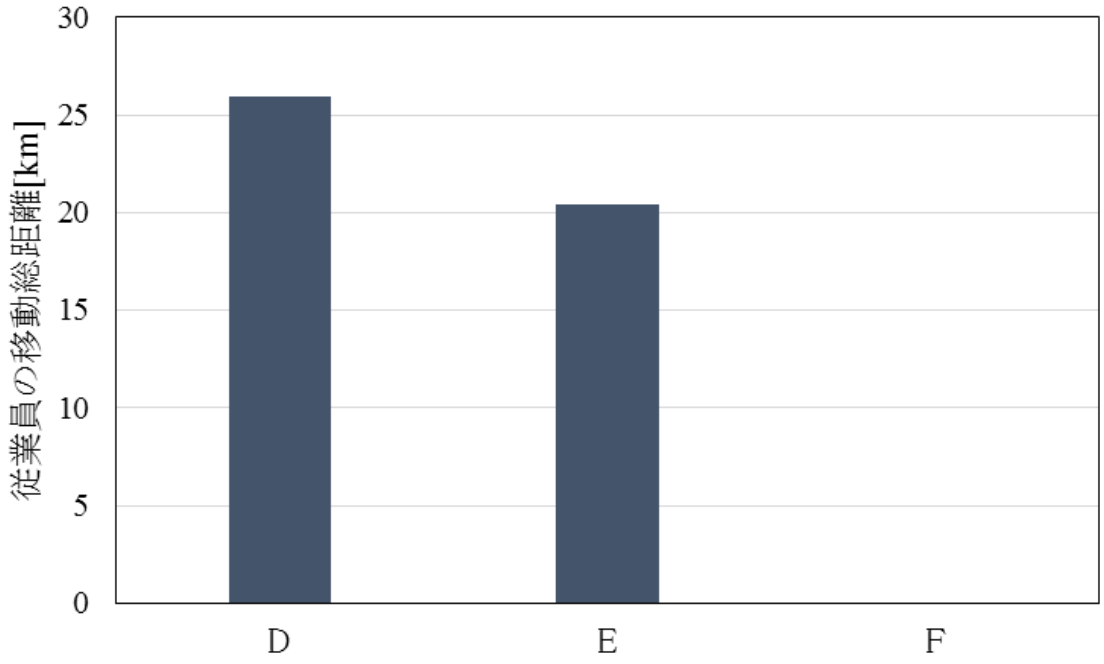


図 5.5.2 条件 D および E, F における再配置に要した総移動距離

図 5.5.2 をみると, 条件 E の方が条件 D よりも総移動距離は約 20%程度削減できていることがわかる. また, 条件 F は利用者のみの再配置としているため従業員の移動距離は 0 となっている. 条件 E において, 従業員の総移動距離のみを考慮すれば利用者のみ, もしくは従業員のみの再配置の場合と比較して, 要求受託率を維持したままコストが削減できていると考えられる.

しかし, 実際には利用者に報酬を支払わなければ利用者による再配置を実現することはできない. したがって移動距離に対してどの程度の報酬を利用者に支払うべきなのか, どの程度の報酬を支払えばどの程度の確率で利用者は再配置を行うのかといった点について, 利用者の実データの分析, 数理モデル化を行う必要がある. また, コストに関しても再配置のコストと利用者に支払う報酬のコストを両方考慮した最適化が必要となる.

5.6. シミュレーションの実行時間

本節ではシミュレーション時間の実行時間について述べる. 先行研究でのシミュレーションは貪欲法により高速に行うことが可能だと 2 章で述べた. だが, 実際には価値関数の計算や移動時間を考慮したことにより実行時間が大幅に増加していることが考えられる. 表 5.6.1 はシミュレーションを行

った PC のシステム情報である. なお, シミュレーションの使用言語は Python3.7.6 である.

表 5.6.1 実行 PC のシステム情報

項目	情報
CPU	Intel(R) Atom(TM) x5-Z8300 CPU @1.44GHz
RAM	2.00GB
OS	Windows 10 Home

表 5.6.2 に実行時間を計測したシミュレーション条件を示す.

表 5.6.2 シミュレーション時間を計測した条件

Case	A	B	C	E	F
T	4h	4h	4h	4h	4h
N	10	10	10	10	10
Random mode	Poisson	Poisson	Poisson	Poisson	Poisson
Lambda	0.003	0.003	0.003	0.003	0.003
Travel time	FALSE	FALSE	TRUE	TRUE	TRUE
Value function	-	Previous	Previous	New	New
Relocation	FALSE	TRUE	TRUE	TRUE	TRUE
User relocation	FALSE	FALSE	FALSE	FALSE	TRUE

表 5.6.2 の条件下で計測した実行時間を図 5.6.1 に示す.

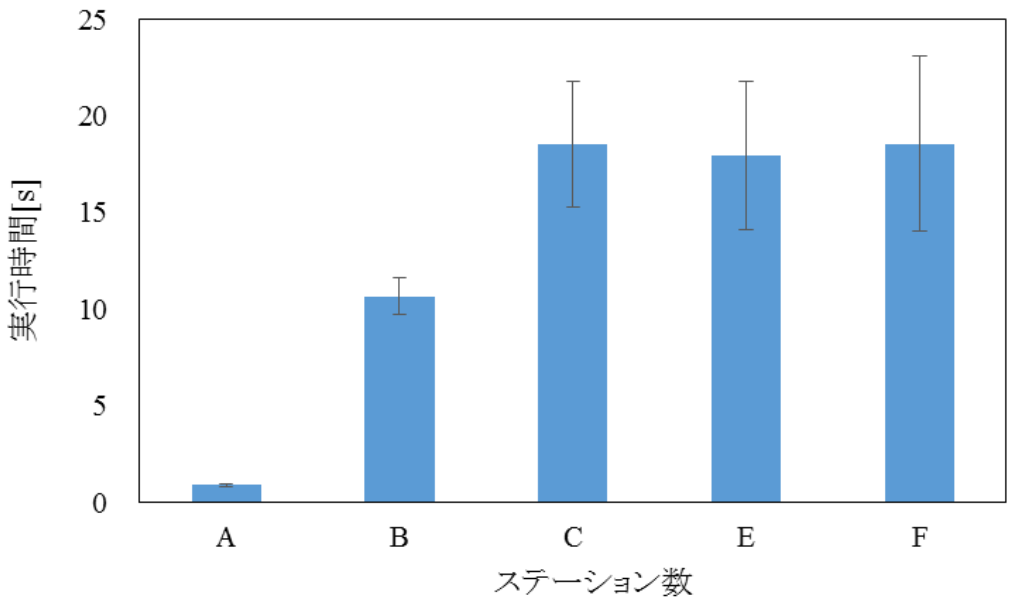


図 5.6.1 シミュレーション時間の計測結果

実行時間を計測するにあたって，それぞれの条件で 10 回の試行を行い，その平均値を用いている．また，エラーバーはそれぞれの標準偏差を示している．

先行研究とは実行環境が異なるが，条件 5.2.1 では先行研究と大きな差はなく，平均約 1 秒程度で実行することができている．逆に，条件 5.2.2 のように再配置を導入した場合には平均 10 秒を超え，移動時間を考慮する条件 5.3.1 や 5.4.1 では更に平均 15 秒を超えていることがわかる．6.3 節でも述べたが，これは再配置を行う際に，RSF または RSE の探索に全探索を用いているためだと考えられる．また，移動時間を考慮するのみで特に RSE を探索する処理に時間がかかることも挙げられる．一方で条件 5.4.1 と 5.5.1 を比較しても大きな差はないことがわかる．利用者が再配置をするかどうかの処理は単純なものとなっているためである．

また，シミュレーションの実行時間は考慮するシミュレーション数によっても大きく変化する．図 5.6.2 と図 5.6.3 に条件 5.4.1 を基に行ったシミュレーションにおける，考慮するステーション数と実行時間の関係を示す．

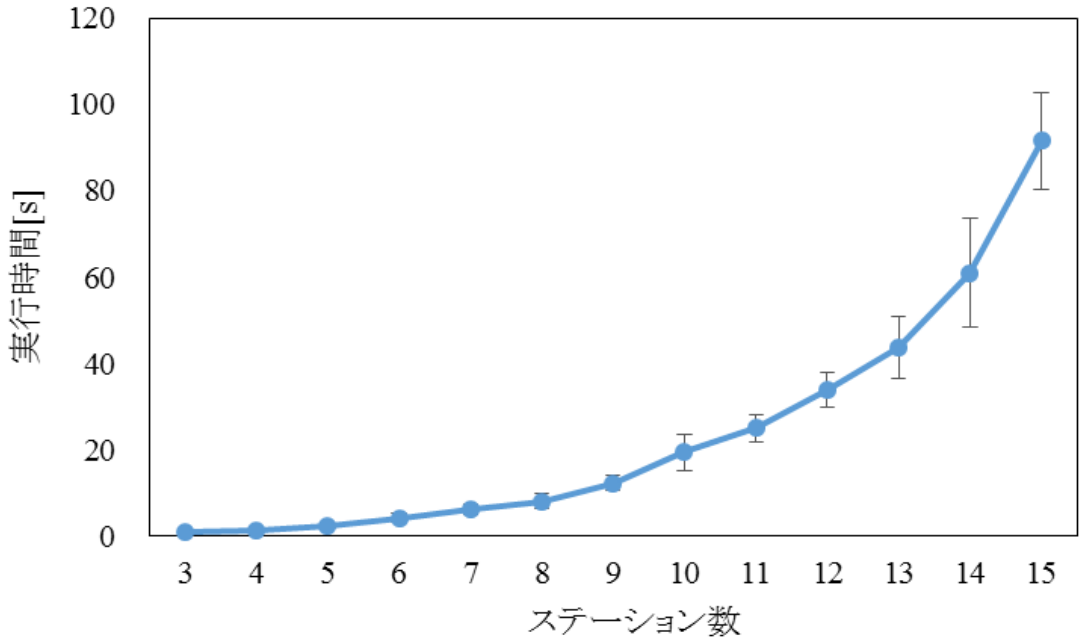


図 5.6.1 ステーション数とシミュレーション実行時間の関係 **(3 ≤ N ≤ 15)**

図 5.6.2 はステーション数 3 から 15 までの計測結果である．グラフからもわかる通り実行時間はステーション数に関して指数関数的に増大していることがわかる．

シミュレーション内では再配置のため様々な処理をしていることを 4.5.5 節では述べた．例えば各需要があるかどうかはすべてのステーション間について探索を行うため，この処理だけでもT × N × N回の試行が行われる．その他の処理についても同様に考えると計算オーダは理論的には$O(TN^2)$であることがわかる．

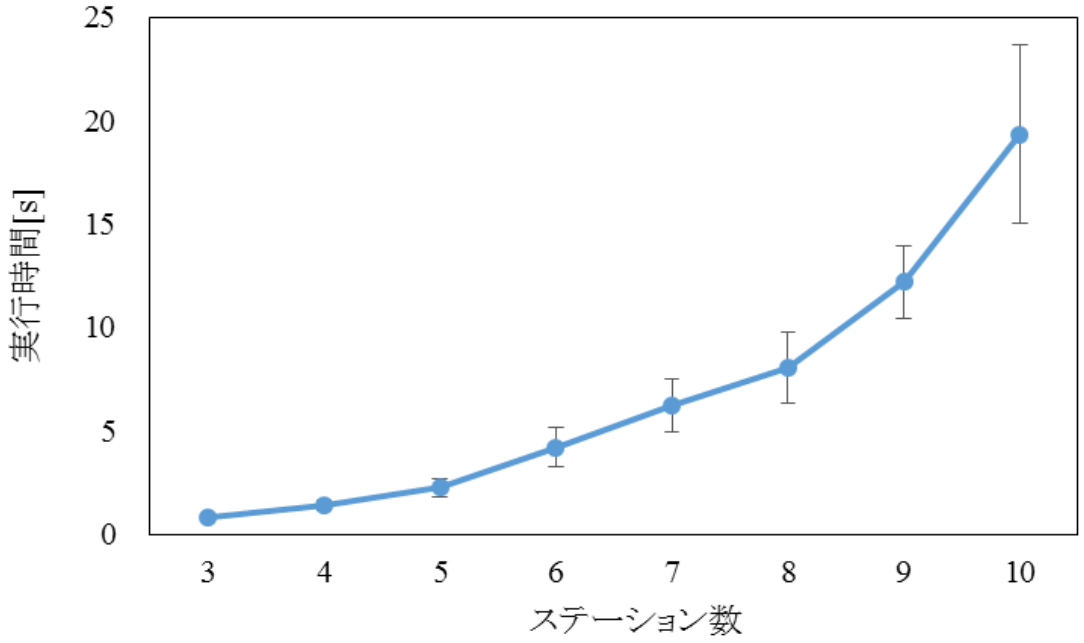


図 5.6.2 ステーション数とシミュレーション実行時間の関係 (**3 ≤ N ≤ 10**)

また，図 5.6.3 は特にステーション数 10 までの結果を拡大したものである．これら二つの図からは共通してエラーバーが大きくなっていることが分かる．図 5.6.2 と図 5.6.3 のエラーバーも共通して標準偏差を表している．図 5.6.1 でも同様であるが，同じ条件下であっても毎回実行時間には大きなばらつきがあることが確認できた．同じ条件下，つまり需要の総数やステーション数が同じであっても，探索しなければならない再配置の経路，価値関数など処理内容が大きく変化し，結果として実行時間に差が出ると考えられる．

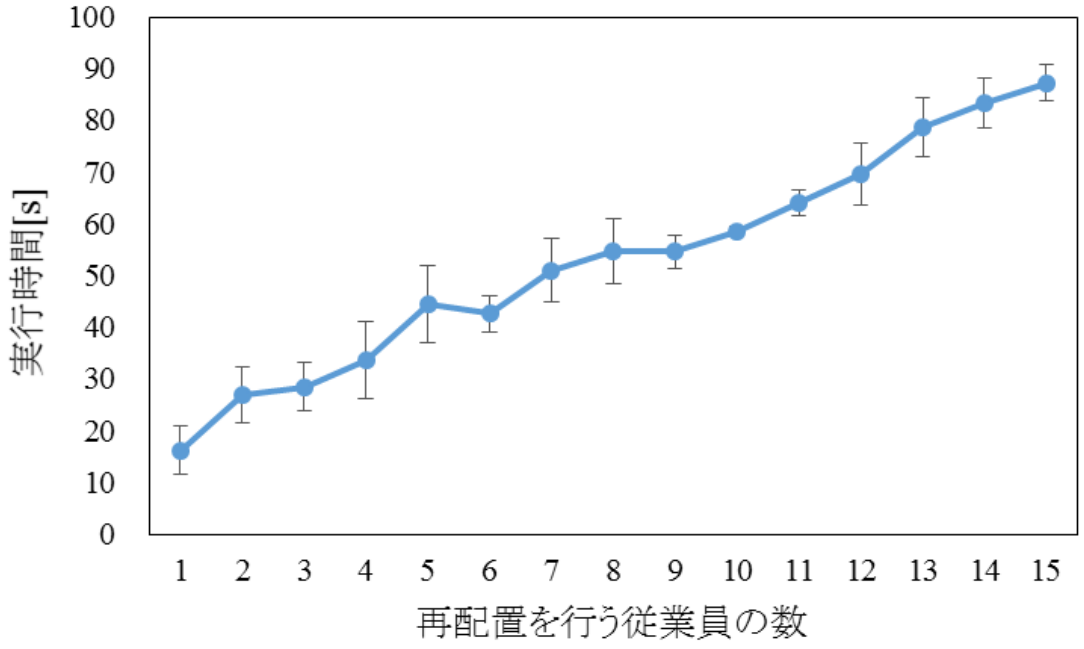


図 5.6.4 再配置を行う従業員の数とシミュレーション実行時間の関係

同様に，再配置を行う従業員の数を変更したときにどのような影響が出るのかについても検証した．図 5.6.4 は従業員の人数ごとのシミュレーションの実行時間を示したものである．

図 5.6.4 を見ると再配置をする従業員の人数を増加させるとシミュレーションの実行時間もそれに伴って増加していることがわかる．しかし，図 5.6.2 などのステーション数を増加させたときと比較すると比較的緩やかな勾配になっていて，指数関数的には増大していない．再配置を行う従業員の数はシミュレーションを行う場合，使用者の数ともとらえることができる．実用化を考えると，利用者の人数は今後更に増加することから従業員の人数を増加させても実行時間が変化しないようシステムを改める必要がある．

6. 結言

本研究では先行研究に加えて再配置問題の日本への適用，現実データの利用，移動時間の考慮，利用者による再配置の導入を行った．新たに移動時間を考慮することで要求拒否数は大幅に増加することがわかったが，価値関数を変更することで依然として要求受託率を 90%程度に維持することができた．しかし，利用者による再配置の導入では，従業員の総移動距離は約 20%程度削減できた．このとき要求受託率はほとんど変化せず 90%を維持することができた．

現状，三つの課題が挙げられる．

一つ目はデータについてである．本研究ではデータを入手できなかったためポアソン分布によるランダムなデータを利用した．2 章の類似研究ではパーソントリップ調査の結果を用いる方法について触れたが，現実の利用者とは異なる結果を示した．したがって今後は実証実験などの実データをパーソントリップ調査で拡張した新たな需要データを用いたシミュレーションを行う必要がある．

二つ目は利用者の数理モデル化である．利用者に依頼を行ったとき，どの程度の確率で再配置を行うのか，または利用者にどの程度の報酬を支払えばよいのか，といった利用者の行動について数理モデル化してシステムに組み込む必要がある．

三つ目はアルゴリズムそのものについてである．2 章の先行研究では貪欲法を用いることで高速にシミュレーションを行うことが可能だと述べた．しかし，新たな価値関数や移動時間といった概念を導入したことにより，シミュレーション時間は大幅に増加した．総当たり的に再配置経路を探索することなく最適解もしくは準最適解を導く手法が必要である．

また，貪欲法では逐一その状態における最善の策を選択する手法であるが，移動時間を考慮する場合，システムは数分先，あるいは実用化を考えれば数時間先の状態を予測してふるまう必要がある．

これらのことより，今後の展望としてはある程度先を見越して利得の最大化を行う強化学習の導入が考えられる．学習に時間はかかるもののこれによりコストの低い再配置の選択を期待する．

謝辞

本研究を遂行するに当たり，研究の場を与えて頂き，終始温かくご指導，ご鞭撻を承りました芝浦工業大学 長谷川浩志教授に心からの感謝の意を表します．

本研究の遂行に当たり，多大な御助言，御協力を頂きました宮木厳太朗氏に謹んで深甚なる謝意を表します．

また，研究の進捗に関する御助言，御指導，研究生活等，多方面にて支えて頂きました堤淳介，鈴木隆介，柳生耕平，角田わたる，横山翔，豊嶋葵輝，竹崎奨吾に感謝致します．

最後に，大学にて研究する機会を与えてくださり，常に支えてくれた家族に深く感謝致します．

関 倖太郎

参考文献

[1]. 総務省, シェアリングエコノミーとは, https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h27/html/nc242110.html, 最終閲覧日: 2020 年 1 月 28 日.

[2]. Airbnb, Inc., https://www.airbnb.jp/, 最終閲覧日:2020 年 1 月 28 日.

[3]. 総務省, 社会課題解決のための新たな ICT サービス・技術への人々の意識に関する調査研究(平成 27 年), https://www.soumu.go.jp/johotsusintokei/linkdata/h27_06_houkoku.pdf, 最終閲覧日:2020 年 1 月 28 日.

[4]. Airbnb, Inc., https://news.airbnb.com/ja/airbnb%E5%8F%B2%E4%B8%8A%E6%9C%80%E9%AB%98%E3%81%AE%E5%AE%BF%E6%B3%8A%E8%80%85%E6%95%B0%E3%82%92%E8%A8%98%E9%8C%B2%EF%BC%81400%E4%B8%87%E4%BA%BA%E3%81%8C%E4%B8%96%E7%95%8C%E3%81%A7%E3%81%A4%E3%81%AA/, 最終閲覧日:2020 年 1 月 28 日.

[5]. Smovengo / Syndicat Autolib' Vélib' Métropole., https://www.velib-metropole.fr/, 最終閲覧日:2020 年 1 月 28 日.

[6]. Open Street Ink., https://www.hellocycling.jp/, 最終閲覧日:2020 年 1 月 28 日.

[7]. 株式会社コインパーク, https://sharepedal.hellocycling.jp/, 最終閲覧日:2020 年 1 月 28 日.

[8]. 小山市役所, https://lacool.hellocycling.jp/, 最終閲覧日:2020 年 1 月 28 日.

[9]. Uber, Inc., https://www.uber.com/jp/ja/, 最終閲覧日:2020 年 1 月 28 日.

[10]. 総務省, 次世代の交通 MaaS, https://www.soumu.go.jp/menu_news/s-news/02tsushin02_04000045.html, 最終閲覧日:2020 年 1 月 28 日.

[11]. 公益財団法人日本自動車教育振興財, カーシェアリングと若者のクルマ利用, http://www.jaef.or.jp/6-traffi-cation/img/TC_34_t.pdf, 最終閲覧日:2020 年 1 月 28 日.

[12]. 一般財団法人自動車検査登録情報協会都道府県別・車種別自動車保有台数(軽自動車含む), https://www.airia.or.jp/publish/statistics/number.html, 最終閲覧日:2020 年 1 月 13 日.

[13]. 東京都総務局統計部-東京都の統計, https://www.toukei.metro.tokyo.lg.jp/jsuikei/js-index.htm, 最終閲覧日:2020 年 1 月 13 日.

[14]. 公益財団法人交通エコロジー・モビリティ財団, http://www.ecomo.or.jp/environment/carshare/carshare_graph2019.3.html, 最終閲覧日:2020 年 1 月 13 日.

[15]. 中村謙太, “ワンウェイ型カーシェアリングシステムの導入可能性と最適ステーション配置”, 土木学会, 2017.

[16]. TIMES MOBILITY CO., LTD., https://share.timescar.jp/tcph/, 最終閲覧日:2020 年 1 月 28 日.

[17]. 中山昌一朗, 山本俊行, 北村隆一, 再配置によらない電気自動車の共同利用システムの効率化に関する研究, 土木計画学研究・論文集 19 巻, 2002.

[18]. 千住琴音, ワンウェイ方式カーシェアリングにおける潜在的利用者を活用した予約受託率最大化手法, 奈良先端科学技術大学院大学, 2018.

[19]. Rabih Zakaria, Laurent Moalic, Mohammad Dib, Alexandre Caminada, “Car relocation for carsharing service: Comparison of CPLEX and Greedy Search”, IEEE, 2014.
[20]. ナビタイムジャパン社, NAVITIME API | ナビタイム API/SDK サービス｜ナビタイムジャパン, https://api-sdk.navitime.co.jp/api/, 最終閲覧日:2020 年 1 月 22 日.
[21]. ナビタイムジャパン社, ナビタイム地図検索サービス, https://www.navitime.co.jp/maps/poi?lon=139.766889&lat=35.68103, 最終閲覧日:2020 年 1 月 22 日.
[22]. 日本規格協会, JIS Z 8101-1:1999 統計−用語と記号−第 1 部:確率及び一般統計用語, http://kikakurui.com/z8/Z8101-1-1999-01.html, 最終閲覧日:2020 年 1 月 22 日.
[23]. 中田寿夫, 内藤貫太, 学術図書出版社, 確率・統計, 2017, pp.77-78.
[24]. 中妻照雄, 朝倉書店, Python によるベイズ統計学入門, 2019, pp.60-61.
[25]. 岡本正芳, コロナ社, 工学系のための確率・統計-確率論の基礎から確率シミュレーションへ-, 2013, pp.29-30.