

最適化ミニ研究

神奈川県央地域でのデマンド型交通の対象地区の選定

2026/4/30 永井

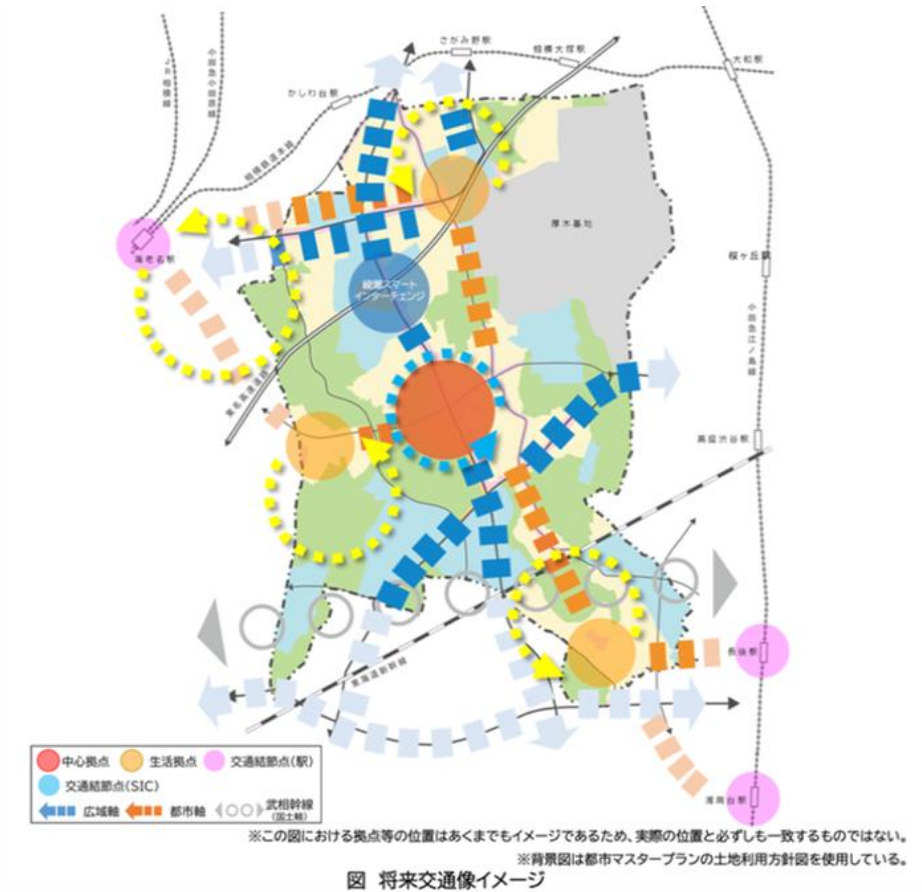
目次

- 課題設定
- 検討条件の設定
- 定式化方法の比較
- 再定式化
 - 車両台数の決定変数化、相乗リ効果の定式化
- 高コスト車両導入の感度分析
 - 介護対応高機能車の導入検討
- まとめ
- 参考；LLMの活用

課題設定

- 首都圏でも通勤・通学者の減少, 運転手不足が加速
- 神奈川県内では路線バスの減便や廃止が発生しており, 公共交通の確保・サービス維持が課題
- AIや自動運転技術を活用したオンデマンド交通に期待
- 県央地域は従来より鉄道網が希薄で自動車の分担率が高く, 課題は深刻

→ 県央地域 (綾瀬/海老名/大和/座間市) へのオンデマンド交通の導入にあたって, 対象ゾーンの検討を実施

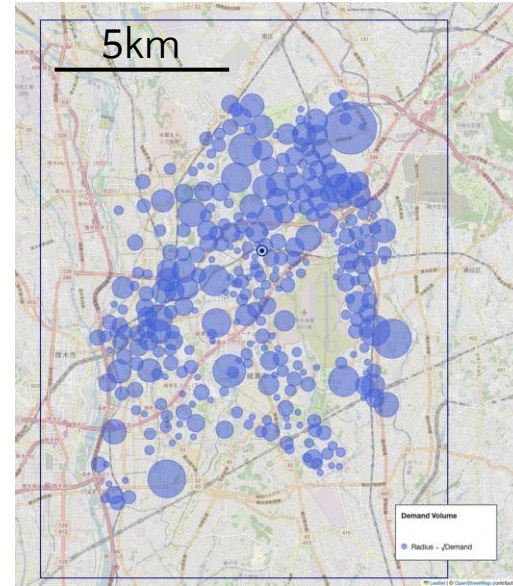


綾瀬市内には鉄道駅が1つもない

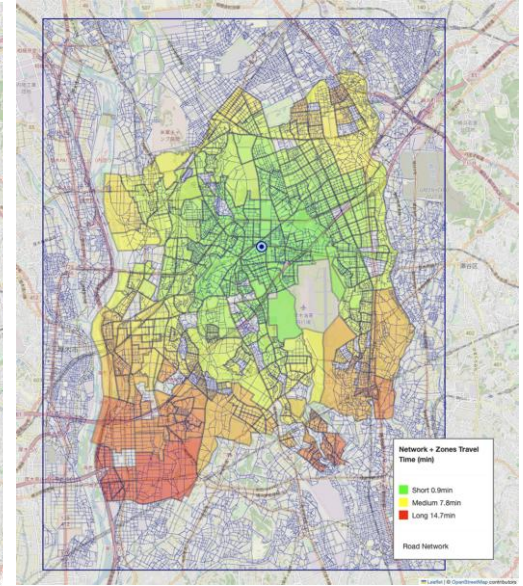
かながわ交通計画 (2022.3)
綾瀬市総合都市開発計画 (2026.3)

検討条件の設定, 前処理

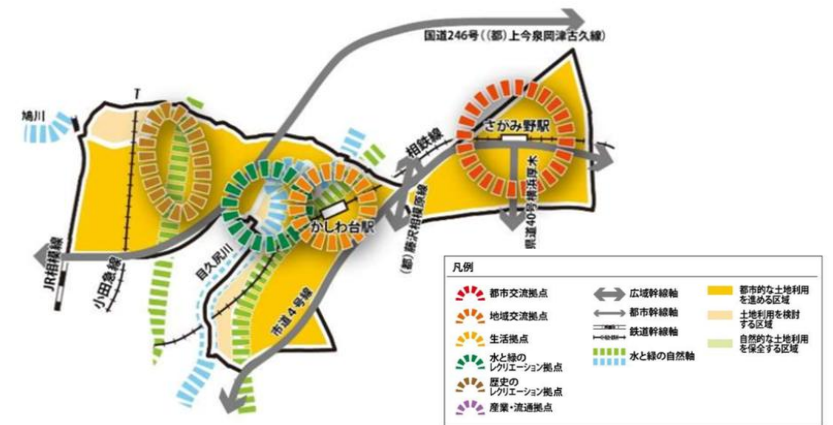
- 使用したデータ
 - 2019年東京都市圏パーソントリップ調査
- 対象4市内相互発着のトリップのうち, 下記を抽出し, 地区コード単位で割り付け
 - 朝7-10時台
 - 自転車、バイク、乗用車、タクシー、路線バス
 - 抽出後件数 1,723件, ゾーン数 319
- 道路はOpenStreetMapから抽出し, 各地区コードの需要重心最寄りのノードに代表点を設定
- ハブから各代表点への自由流到達時間を計算
- 海老名市都市マスタープランで「都市交流拠点」に位置付けられており, 対象範囲のおおよそ中心に位置するさがみ野駅の南口をハブに設定



対象の需要



ハブからの到達時間
(自由流)



海老名市都市マスタープラン (2023年7月)

定式化方法の比較 - 方法

➤ ゾーン毎に, サービス対象に含めるかどうかを選択する

➤ 半径 (時間距離) 優先

到達時間が一定以内であれば対象に含める

$T=2$ [分] から1分刻みの各条件で計算し, 最適半径を決定

➤ 需要優先

需要と運行コストのバランスを取る0-1計画問題

scipy.optimize.milpで最適化 (→分枝カット法)

➤ カバー率優先

需要のうち, 何%をカバーするかを制約に加える

scipy.optimize.milpで最適化 (→分枝カット法)

➤ 全方針共通の制約

容量制約

$$\text{subject to } \sum_i \tilde{q}_i s_i x_i \leq \eta KH$$

➤ その他定義, 設定

ゾーン*i*の需要[件]

q_i

受容率

$\alpha = 0.1$

車両の台数[台]

$K = 12$

ゾーン*i*の有効需要[件]

$\tilde{q}_i = \alpha q_i$

目標カバー率

$\gamma = 0.5$

車両の最大稼働率

$\eta = 85$

ゾーン*i*のハブからの距離[km]

d_i

サービスのバッファ時間[分/件]

$b = 8$

サービス運営時間[分]

$H = 240$

ゾーン*i*の所要時間[分/件]

$s_i = 2 \frac{d_i}{V} \times 60 + b$

料金(固定部)[円/件]

$m_1 = 250$

サービス時間と需要の換算係数[分]

$\lambda = 0.003$ (半径優先), 0.04 (需要優先)

ゾーン*i*の料金[円/件]

$c_i = m_1 + m_2 d_i$

料金(距離比例部)[円/(件・km)]

$m_2 = 120$

料金と需要の換算係数[円]

$\mu = 0.0002$

$V = 18$

車両の平均速度[km/h]

ハブから1件毎に往復した場合にかかる時間

$$x_i \in 0, 1, \quad y_i(T) = \begin{cases} 1 & (t_i \leq T) \\ 0 & (t_i > T) \end{cases}$$

$$\text{maximize } F(T) = \sum_i \{ \tilde{q}_i y_i(T) - \lambda s_i y_i(T) \}$$

$$\text{maximize } \sum_i (\tilde{q}_i - \lambda s_i - \mu c_i) x_i$$

$$\text{minimize } \sum_i s_i x_i$$

$$\text{subject to } \sum_i \tilde{q}_i x_i \geq \gamma \sum_i \tilde{q}_i$$

定式化方法の比較 – 結果

半径 (時間距離) 優先

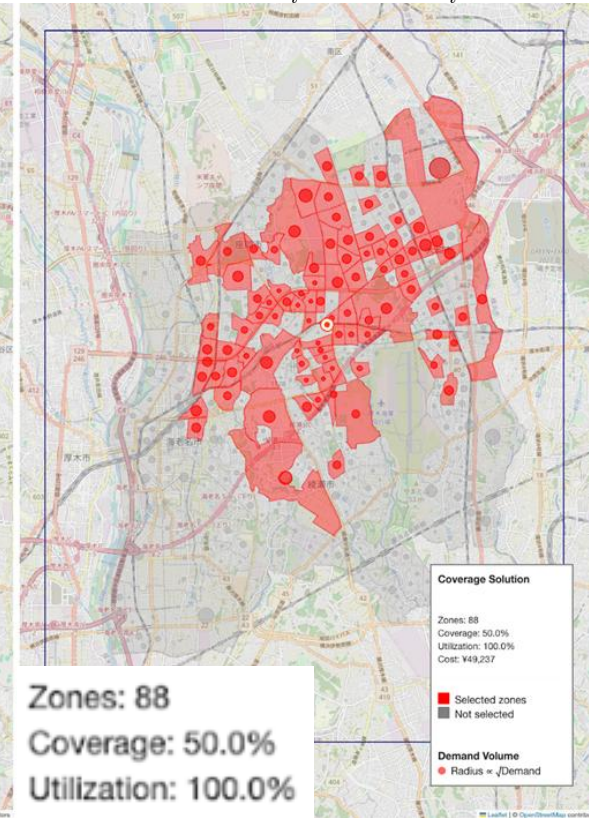
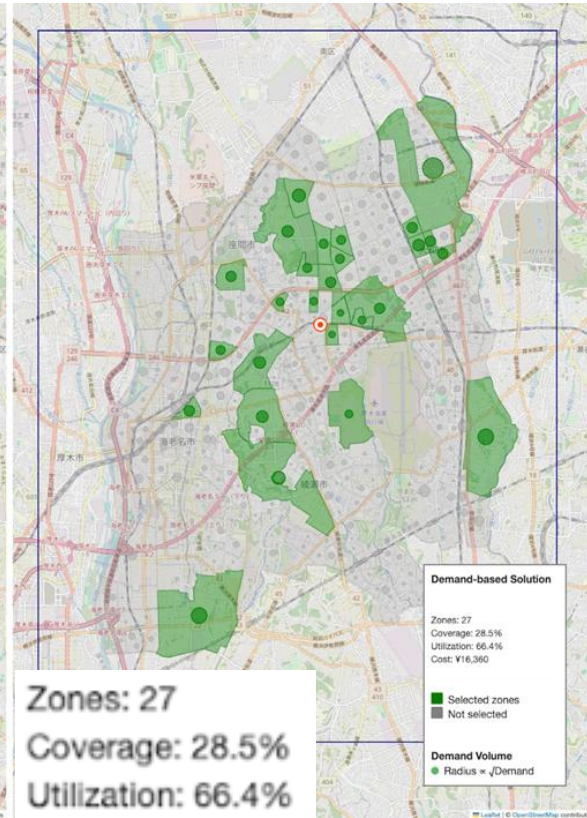
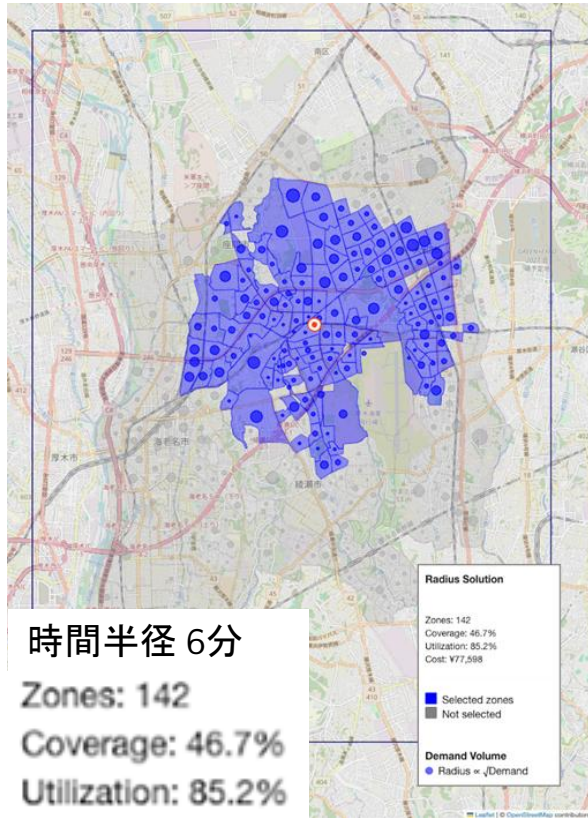
$$\text{maximize } F(T) = \sum_i \{\tilde{q}_i y_i(T) - \lambda s_i y_i(T)\}$$

需要優先

$$\text{maximize } \sum_i (\tilde{q}_i - \lambda s_i - \mu c_i) x_i$$

カバー率優先

$$\begin{aligned} &\text{minimize } \sum_i s_i x_i \\ &\text{subject to } \sum_i \tilde{q}_i x_i \geq \gamma \sum_i \tilde{q}_i \end{aligned}$$



- 需要・カバー率優先では、遠くても需要の大きいゾーンを対象に含められている
- 需要カバー率/容量使用率は、半径優先>カバー率優先>需要優先
→ 需要が分散していて、対象ゾーンを遠方に広げると効率が悪い

需要優先とカバー率制約の組み合わせ

- 実務上は, 目標カバー率を先に定め, 運用コストを最小化しつつ効用を最大化したいのではないか?
- また, 各需要を1件1往復で拾う設定も, 非現実的
- 目的関数, 制約を再設定

$$\begin{aligned}
 & \text{maximize} && \sum_i \tilde{q}_i x_i - \nu k \\
 & \text{subject to} && \sum_i \tilde{q}_i s'_i x_i - \eta H k \leq 0 \\
 & && \sum_i \tilde{q}_i x_i \geq \gamma \sum_i \tilde{q}_i \\
 & && x_i \in 0, 1 \\
 & && \underline{1 \leq k \leq 30, \quad k \in \mathbb{Z}}
 \end{aligned}$$

車両の台数[台]

車両台数と需要の換算係数[件/台]

相乗りを考慮した所要時間[分/件]

相乗りによる所要時間減少係数

ゾーン*i*の有効需要密度[件/km²]

k

$\nu = 20$

$\underline{s'_i = (1 - \varepsilon \sqrt{p_i}) s_i}$

$\varepsilon = \frac{0.7}{\sqrt{20}}$

p_i

車両1台分のコストと, 需要20件が等価

密度最大のゾーンで所要時間が7割減になるような設定

$$\begin{aligned}
 \mathbf{x} &= (x_1 \quad x_2 \quad \dots \quad x_n \quad k)^\top \\
 \mathbf{c} &= (\tilde{q}_1 \quad \tilde{q}_2 \quad \dots \quad \tilde{q}_n \quad -\nu)^\top \\
 \mathbf{A} &= \begin{pmatrix} \tilde{q}_1 s'_1 & \tilde{q}_2 s'_2 & \dots & \tilde{q}_n s'_n & -\eta H \\ -\tilde{q}_1 & -\tilde{q}_2 & \dots & -\tilde{q}_n & 0 \end{pmatrix} \\
 \mathbf{b} &= (0 \quad -\gamma \sum_i \tilde{q}_i)^\top
 \end{aligned}$$

とすれば,

$$\begin{aligned}
 & \text{maximize} && \mathbf{c}^\top \mathbf{x} \\
 & \text{subject to} && \mathbf{A}^\top \mathbf{x} \leq \mathbf{b}
 \end{aligned}$$

と書けるので, 整数計画問題として解ける

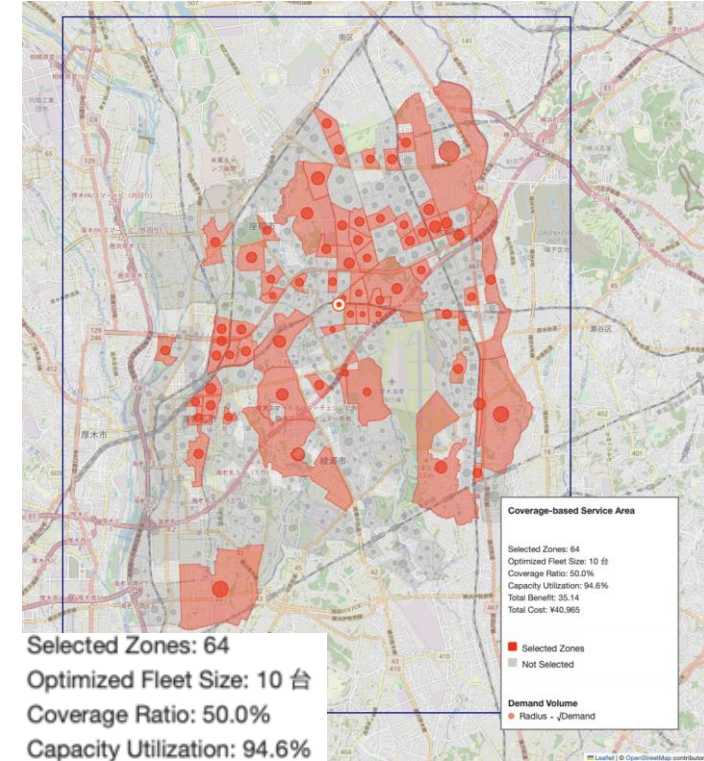
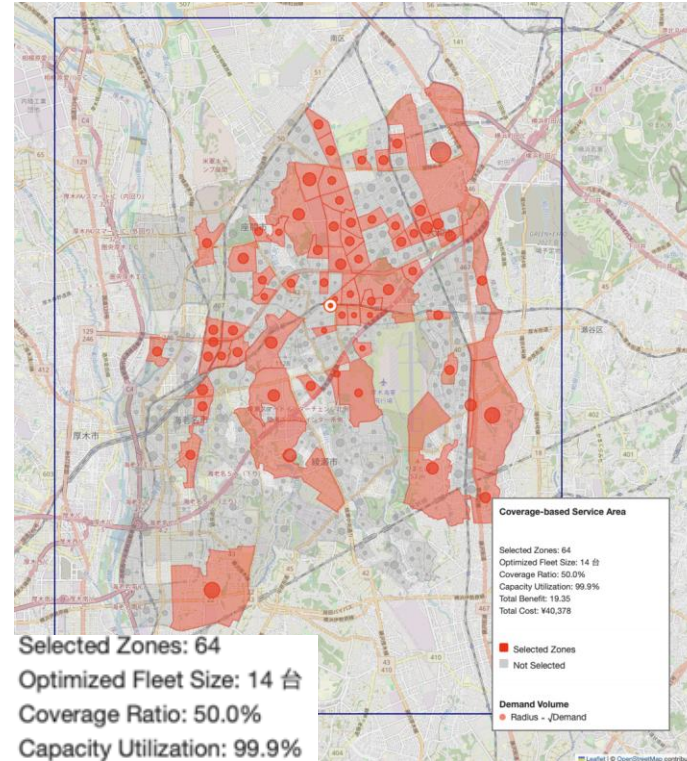
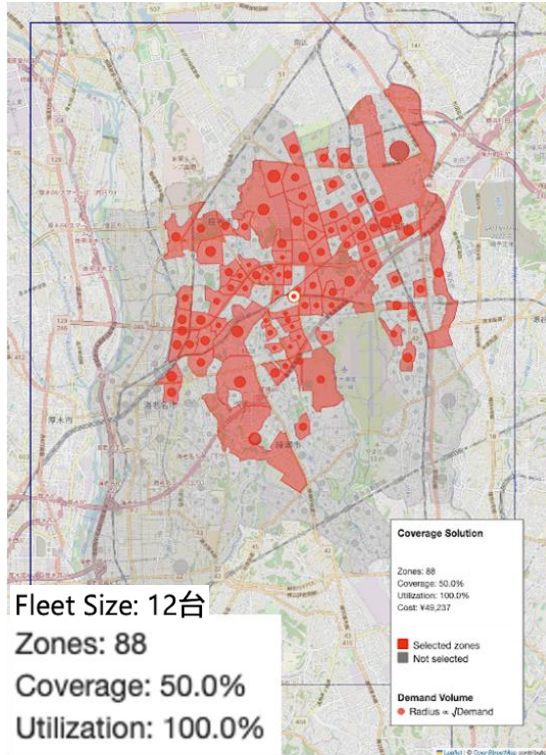
需要優先とカバー率制約の組み合わせ - 結果

再定式化後

(カバー率優先)

相乗リ係数なし

相乗リ係数あり



- 車両台数を決定変数に含めることで、台数を増やして遠方の需要を拾う解を見つけられた
- 相乗リ効果を取り入れることで、車両台数をさらに減らすことができる

車両使用変更の感度分析 – 設定

- 綾瀬市ではデマンド型交通の実証運行が始まっているが、対象者は自力乗降可能な人に限られている

対象者

1,以下の条件に全て該当し、事前に登録を行った方

- ・ 市内在住65歳以上
- ・ 運転免許を持っていない
- ・ 自ら車両に乗降できる

綾瀬市AIデマンド型交通「あやモビ」

https://www.city.ayase.kanagawa.jp/soshiki/toshiseibika/doro_kotsu/5/22638.html

- 乗降に補助が必要な人や時間がかかる人を対象にしようとした場合、
- 車両仕様を変えず、時間をかけて乗降させる (乗降地での介助などを想定)
 - 高機能な車両を導入し、その他の人と同等の時間で乗降させる
- のどちらが効果的かを検討してみる
- また、高機能車両を導入する場合、通常仕様車に比べてどの程度のコスト差であれば許容しうるか？



高機能車両の例

<https://www.kurumaerabi.com/magazine/articles/281/>

車両使用変更の感度分析 – 定式化

- 各ゾーンについて、乗降に時間がかかりそうな利用者による需要の割合を算出し、需要に重みづけ

□ 対象者は広めに、60歳以上 or 外出に関する身体的な困難さ ≥ 2 (外出に杖が必要)

1,723件中、474件が該当

- 通常仕様車の場合は、乗降困難者率に応じてバッファ時間を増加させる

$$\begin{aligned} & \text{maximize} && \sum_i w_i \tilde{q}_i x_i - \nu' k \\ & \text{subject to} && \sum_i \tilde{q}_i s_i'' x_i - \eta H k \leq 0 \\ & && \sum_i \tilde{q}_i x_i \geq \gamma \sum_i \tilde{q}_i \\ & && x_i \in 0, 1 \\ & && 1 \leq k \leq 30, \quad k \in \mathbb{Z} \end{aligned}$$

ゾーン*i*の需要重み

$$w_i = 1 + \theta r_i$$

$r_i = 1$ のとき重み $(1 + \theta)$ 倍

\tilde{q}_i のうち乗降困難者率

$$r_i$$

乗降困難者の重み係数

$$\theta$$

車両台数とコストの換算係数[件/台]

$$\nu' = \nu \quad (\text{通常仕様車}), \quad \beta \nu \quad (\text{高機能車})$$

高機能車の車両コスト係数

$$\beta$$

所要時間[分/件]

$$s_i'' = \begin{cases} (1 - \varepsilon \sqrt{p_i}) (2 \frac{d_i}{V} \times 60 + \underbrace{(1 + (\zeta - 1)r_i)b}_{\text{通常仕様車}}) & (\text{通常仕様車}) \\ s_i' = (1 - \varepsilon \sqrt{p_i}) (2 \frac{d_i}{V} \times 60 + b) & (\text{高機能車}) \end{cases}$$

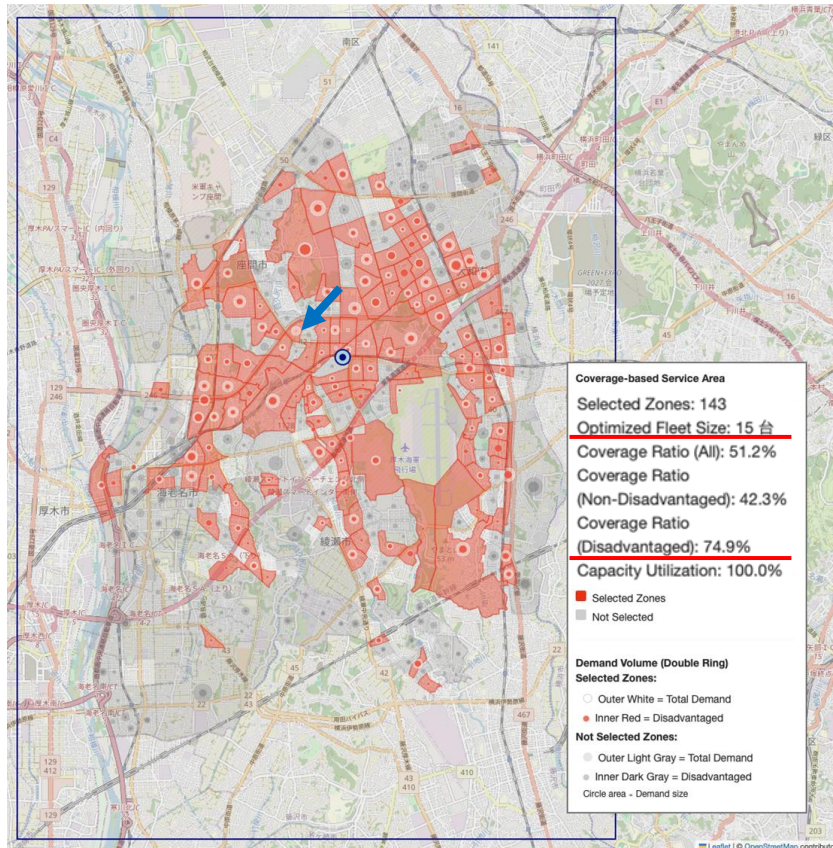
乗降困難によるバッファ時間増加係数

$$\zeta = 8$$

$r_i = 1$ のときバッファ時間が8倍 (64分)

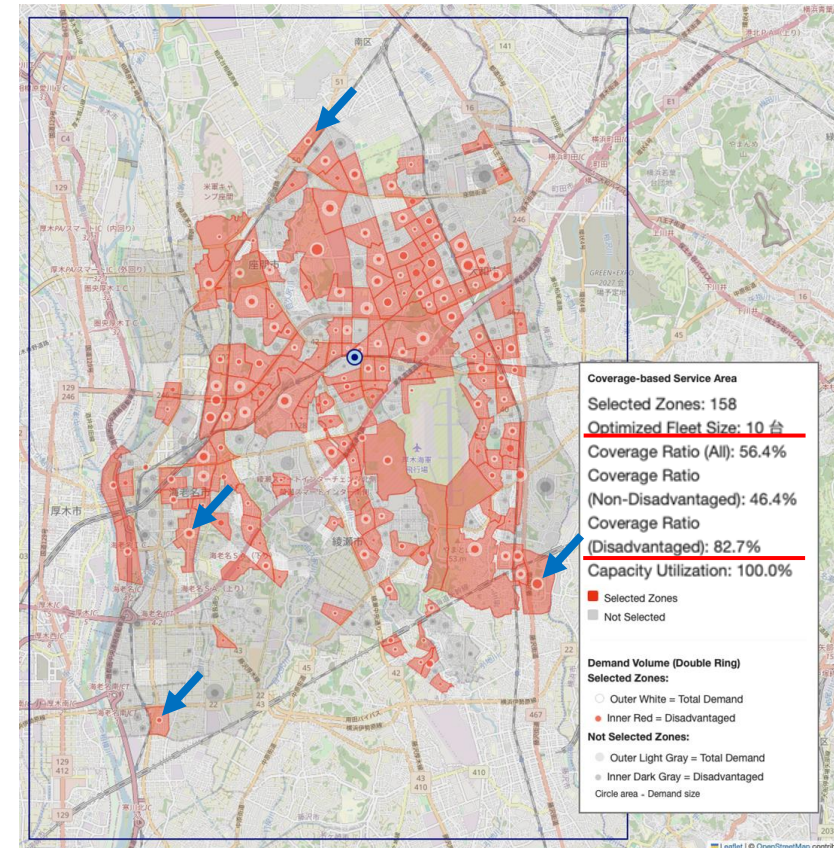
車両使用変更の感度分析 - 結果①

通常仕様車



高機能車 ($\beta=1.2$)

$\vartheta=10$



- 高機能車を投入すると, 遠方だが乗降困難者の需要が大きいゾーンを対象に含めやすくなる
- (当然, 通常仕様車の所要時間延長幅によるが) 高機能車に置き換えた方が台数も少なくて済む
- 対照ゾーンに含まれる通常利用者の需要カバー率も増加する
- 一方で, 乗降困難者率の低いゾーンは対象から外れてしまうものもある

車両使用変更の感度分析 – 結果②

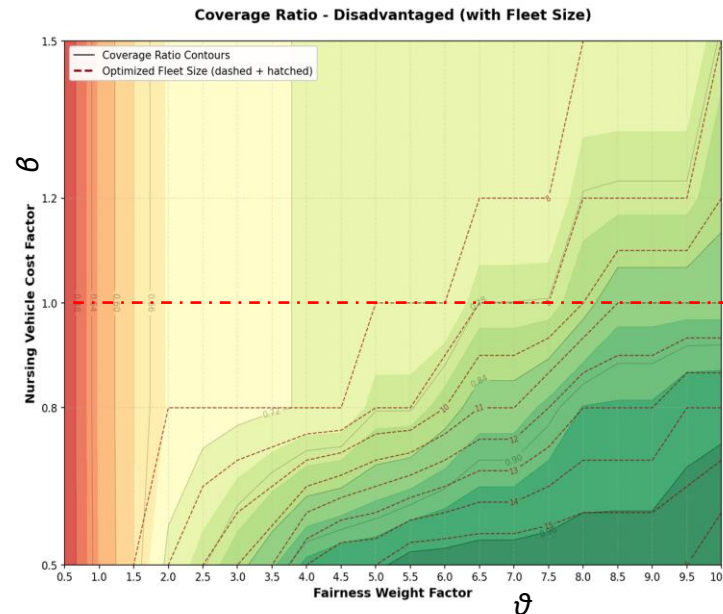
- 乗降困難者のウェイト $\vartheta = \{0.5, 1, 1.5, \dots, 10\}$, 高機能車のコスト係数 $\beta = \{0.5, 0.8, 1, 1.2, 1.5\}$ について, 最適化計算を実行し, 乗降困難者のカバー率を確認

通常仕様車

カバー率 (困難)

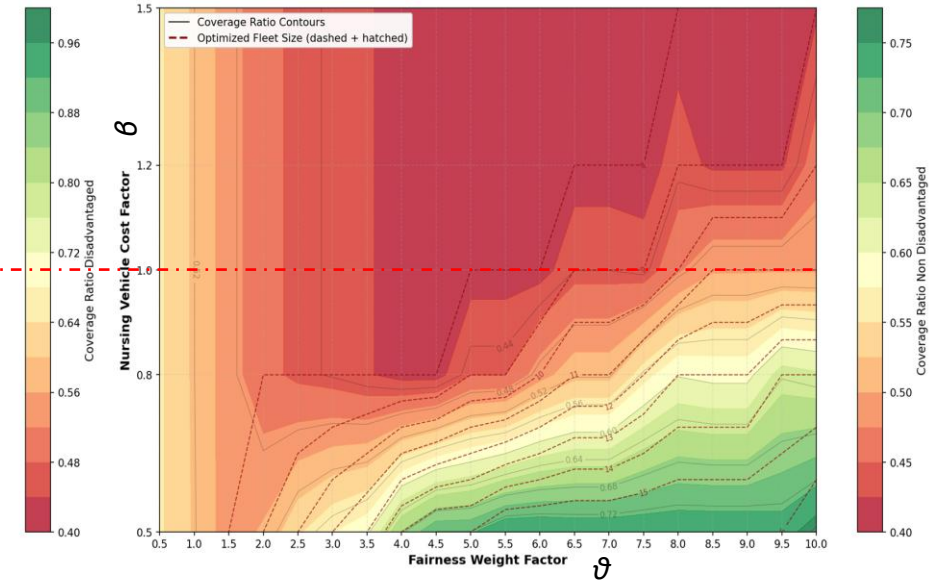
台数

カバー率 (通常)



高機能車

Coverage Ratio - Non-Disadvantaged (with Fleet Size)



- 当然, 乗降困難者を重視するほど, また, 高機能車コストが下がるほど, 乗降困難者のカバー率は上がる
- 通常仕様車の場合は, 乗降困難者の重視に伴って通常利用者のカバー率が下がるが, 高機能車の場合は伴って上がる
→ 通常仕様車だと乗降困難者の需要を拾うと通常利用者を切り捨ててしまうが, 高機能車だと両立できる
- 「車が高いが乗降困難者を重視したい」場合, $\vartheta = 1$ のときより通常利用者が損をする領域がある
→ 不公平感に繋がりがねず, 避けるべき領域? (高機能車を無理に導入しない方がいい)

今回の場合, $\vartheta \leq 6$ では通常仕様車より高い高機能車は導入しづらく, $\vartheta \geq 8$ 程度から, $\beta = 1.2$ の車両の導入が現実的になる

- 含む or 含まない (0 or 1) や、複数の選択肢のある要素を決定変数にできる整数計画問題は、説明力が高く、かつ一般的なソルバーでも解きやすく強力
- 整数計画問題に限らないが、目的関数や制約条件の設定、パラメータの設定によって結果は大きく変わりうるので、よく考えて定式化するのが重要
- 今回は対象ゾーンの選択に留まってしまったが、より実際に近い検討をするには、経路最適化 (VRP) も取り入れるべき
- ハブをどこに置くか、何か所置くか、も検討対象になり得る
- EV化を考えると、連続走行時間などを制約にすることが考えられ、結果が変わり得る

◆ VS code 拡張機能の GitHub Copilot, Gemini Code Assist を利用

➤ GitHub Copilot

- チャット, インライン提案ともに優秀 (精度, 速度)
- 残念ながら, 4/20以降, 学生アカウントの無制限使用への新規登録が停止されているよう

<https://github.com/orgs/community/discussions/192963>

- Free 版は利用制限 (月単位) までそれほど余裕がなかった

➤ Gemini Code Assist

- Copilot に劣る印象だが, チャットもインライン提案も十分使える
- 混雑時 (?) の応答時間が遅め
- ECCS アカウントは Code Assist に対応していない?
- Google ドライブの有料プランに付帯していたので, それを利用

- 無制限に利用できないプランでは, インライン補完のような軽い使い方と, ゼロからコードを書かせるような重い使い方を意識的に使い分けることで節約できそう

8. What exactly changed for Education users?

These updates apply across Copilot plans, including Education users.

- The main changes are:

- New signups for Copilot Student, Copilot Pro, and Copilot Pro+ are paused
- Opus-family models are being removed from Copilot Pro.
- New rate limits are being introduced.
- More transparent rate-limit progress is being added.

- What is not changing:

- Code completion remains free.
- Existing users will not be removed.