

オンデマンド交通システムのための階層的車両配置最適化

Nested Rebalancing Optimization for Mobility on Demand

西 智樹^{*1}

Tomoki Nishi

小出智士^{*1}

Satoshi Koide

大滝啓介^{*1}

Keisuke Otaki

大社綾乃^{*1}

Ayano Okoso

^{*1}豊田中央研究所

Toyota Central R&D Labs., Inc.

Mobility-on-Demand (MoD) services are promising application for Mobility as a Service. Imbalance between locations of vehicles and customer requests worsens service quality such as waiting time of customers for MoD services. Previous work has developed the method to optimize the number of vehicles at each regions at each time using Mixed Integer Programming. However the computational complexity of the method exponentially increases in proportion to the number of regions. We develop hierarchical method to optimize the number of vehicles. We evaluate the method to a real-world taxi trip dataset in numerical experiment. The result shows that the method with three layers improve the computational time compared to one with single a layer.

1. はじめに

オンデマンド交通は客からの依頼に応じて配車する交通システムであり、移動をより効率的にする有効な手段の一つである。オンデマンド交通は、1) 客からの依頼をどの車両に割り当てるか、2) 割り当てられた客をどの経路で送迎するか、3) 客を送迎していない空車を次の依頼のためにどこに移動または待機させるかの、3つのタスクに分割することができる。3つ目のタスクは、車両のリバランシング問題と呼ばれ、現時刻の車両位置の分布と将来の需要位置の分布とに大きな開きがある場合に、客の待ち時間を減らすために効果的であることが知られている [Spieser 16].

Iglesias らは各車両の配置と経路を最適化するのではなく、予め分割されたエリアに必要な車両台数の最適化問題と捉え直し、整数線形計画問題として定式化した [Iglesias 18]. この定式化では車両台数が増えても最適化にかかる時間が一定であるため、都会のような車両台数が多い場合に有効である。しかしながら、分割されたエリアの数が増えると計算時間が急速に増加するという問題があるため、現状での適用範囲は局所的な地域内での最適化に限定される。

そこで本研究では、エリアを階層的に分割し、粗いエリアから細かなエリアに必要な車両台数を最適化することで計算量を削減し、エリア数の増加に対応可能な手法を提案する。

2. 関連研究

リバランシング最適化問題の定式化は様々提案されている。Pavone らは流体モデルを用いることで最適リバランシング問題を線形計画問題として定式化できることを示した [Pavone 11]. また Zhang と Pavone はモデル予測制御を用いた分散時間モデルを提案した [Zhang 16]. Iglesias らは別の定式化として、時拡張ネットワークを用いた整数計画問題として定式化した [Iglesias 18]. この定式化では、計算量が車両の台数にはよらないため都会のような車両台数が多い時には特に有効である。しかしながら、この定式化では、予め分割したエリア数に対して指数関数的に計算量が増加するという問題があった。そ

れに対し、提案手法は階層的に最適化を行うことでエリアの数が増えた場合でも現実的な時間で最適化できる。

最適リバランシング問題は、配車リクエストの車両への割当を最適化する研究 [Bei 18, Dickerson 18, Pelzer 15] と関連している。割当最適化問題と本研究が取り扱う最適リバランシング問題とが大きく異なる点は、我々の定式化では、個別の車両とリクエストの割当は行わず、各エリア各時刻に必要な車両台数を最適化する点である。そのため車両台数やリクエスト数に対し、割当最適化問題は計算量が増加するのに対し、Iglesias らの手法や提案法では計算量は一定である。どの車両がどの配車リクエストを処理するかに対しては、提案法は最適化後、各エリア内で割当を行う必要があるが、一般に各エリアの車両台数やリクエスト数はエリア全体に比べて十分小さいため計算時間の面では問題ないと予想される。

3. 階層的リバランシング法

基本的な考え方は、分割粒度の粗い上位層で大きなエリア間の移動を伴う移動を最適化し、上位層では最適化されなかった粗い粒度のエリア内での移動に関して、より細かな粒度の下位層で最適化する。これは例えば県全域でリバランシングを最適化する場合、従来法は県全域を最初から交差点単位で最適化するのに対し、提案法は上位層で市区町村の間の移動を、下位層で各市区町村毎の移動を最適化することに対応する。

Iglesias らが提案したリバランシング問題の定式化は総車両台数が時間によらず一定であると仮定していた。しかしながら提案法では、下位層の各エリアの総車両台数は、上位層で最適化された各時刻の各粗いエリアの車両台数によって決まるため、時々刻々変化する可能性がある。そこで本章では、まず総車両台数が時々刻々変化する場合でも対応できるよう拡張した最適リバランシング問題の定式化を示し、階層的リバランシング法について述べる。

3.1 総車両台数時変最適リバランシング問題

最適リバランシング問題とは、需要を全て満たすという制約条件の下で、リバランシングにかかるコストを最小化するための各時刻各エリアの車両台数を決定する問題である。総車両台数が時々刻々変化する場合に対応できるよう拡張した定式化は下記の通りである。

連絡先: 西智樹, 豊田中央研究所, 愛知県長久手市横道 41-1,
nishi@mosk.tytlabs.co.jp

$$\min_{\mathbf{x}^p, \mathbf{x}^r, \mathbf{x}^a, \mathbf{x}^d} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} c_{ij} x_{ijt}^r, \text{ subject to,} \quad (1a)$$

$$x_{ijt}^p = \lambda_{ijt}, \quad i, j \in \mathcal{N}, t \in \mathcal{T}, \quad (1b)$$

$$\sum_{j \in \mathcal{N}} \left(x_{ijt}^r + x_{ijt}^p - x_{jit-\tau_{ji}}^r - x_{jit-\tau_{ji}}^p \right) = s_{it}, \quad \forall i \in \mathcal{N}, t \in \mathcal{T}, \quad (1c)$$

$$s_{it} := \begin{cases} x_{it}^a & t = 1 \\ x_{it}^a - x_{it}^d & t > 1 \end{cases}, \quad \forall i \in \mathcal{N}, t \in \mathcal{T},$$

$$\sum_{i \in \mathcal{N}} \left(s_{it} + \sum_{j \in \mathcal{N}} \sum_{\Delta t=1}^{\tau_{ji}} (x_{jit-\Delta t}^r + x_{jit-\Delta t}^p) \right) = V_t, \quad \forall t \in \mathcal{T}, \quad (1d)$$

$$\sum_{i \in \mathcal{N}} x_{it}^a = \begin{cases} V_t - V_{t-1} & V_t \leq V_{t-1} \\ 0 & \text{otherwise} \end{cases}, \quad \forall t \in \mathcal{T}, \quad (1e)$$

$$\sum_{i \in \mathcal{N}} x_{it}^d = \begin{cases} 0 & V_t \leq V_{t-1} \\ V_{t-1} - V_t & \text{otherwise} \end{cases}, \quad \forall t \in \mathcal{T}, \quad (1f)$$

$$\mathbf{x}^r := \{x_{ijt}^r\}, \mathbf{x}^p := \{x_{ijt}^p\}, \mathbf{x}^a := \{x_{ijt}^a\}, \mathbf{x}^d := \{x_{ijt}^d\}, \quad \forall i, j \in \mathcal{N}, t \in \mathcal{T}, \quad (1g)$$

ここで、 $\mathcal{N} := \{1, \dots, N\}$ 及び $\mathcal{T} := \{1, \dots, T\}$ はそれぞれ N 個のエリアの集合、 T 時間ステップの集合を示している。上付添え字 r, p はそれぞれ、リバランシングに関する変数および顧客を目的地に運ぶ移動に関する変数を表している。 x_{ijt} は時刻 t にエリア i からエリア j に移動する車両台数、 c_{ij} はエリア i からエリア j に移動するためにかかるコスト、 s_{it} は時刻 t にエリア i に留まっている車両台数を表している。また τ_{ij} はエリア i からエリア j に移動するのにかかる時間ステップ数を、 λ_{ijt} は時刻 t にエリア i からエリア j に移動したい顧客の数を、 V_t は時刻 t の総車両台数を表している。 x_{it}^a は時刻 t にエリア集合 $\mathcal{N}$ の外からエリア i に流入する台数を、 x_{it}^d はエリア i から $\mathcal{N}$ 外に流出する台数を示している。

リバランシング問題の目的関数 (1a) は時刻 0 から T までのリバランシングにかかるコストの合計であり、それを最小化するように各時刻の移動を最適化することが目的である。制約条件 (1b) は各時刻で客の依頼には必ず応えることを、制約条件 (1c) は車両台数の保存則であり、リバランシングまたは送迎によりエリアに流入する数と流出する数が釣り合うことを表している。制約条件 (1d) は時刻 t の $\mathcal{N}$ に存在する総車両台数が V_t であることを、制約条件 (1e) 及び (1f) は、時刻 t に $\mathcal{N}$ 外から流入出する車両台数の釣合いを示している。

3.2 階層的リバランシング最適化アルゴリズム

従来法に比べ大規模なエリアに関して高速に最適化が可能である階層的リバランシング法について述べる。基本的な考え方は、分割粒度の粗い上位層で大きなエリア間の移動を伴う移動を最適化し、その最適化結果に基づきより細かな粒度の下位層でより小さなエリア間の移動に関して最適化する。ある層のエリア l に関して 2 つのオペレータ $\text{Children}(l)$, $\text{Parent}(l)$ を定義する。 $\text{Children}(l)$ はエリア l に含まれる一つ下位層のより詳細なエリアの集合を返し、 $\text{Parent}(l)$ はエリア l が含まれる一つ上位層のより広いエリアを返す。

l 層目のエリア集合 $\mathcal{N}_l$ に含まれるエリア間の旅行時間、コスト、リクエスト数はそれぞれ $\tau_l := \{\tau_{ijt}\}_{i,j \in \mathcal{N}_l, t \in \mathcal{T}}$, $\mathbf{c}_l := \{c_{ijt}\}_{i,j \in \mathcal{N}_l, t \in \mathcal{T}}$ and $\lambda_l := \{\lambda_{ijt}\}_{i,j \in \mathcal{N}_l, t \in \mathcal{T}}$ で表す。また

$\mathcal{N}_l$ の車両の分配方法を $\mathbf{x}_l := (\mathbf{x}_l^p, \mathbf{x}_l^r, \mathbf{x}_l^a, \mathbf{x}_l^d)$ とする。ただし $\mathbf{x}_l^p := \{x_{ijt}^p\}_{i,j \in \mathcal{N}_l, t \in \mathcal{T}}$, $\mathbf{x}_l^r := \{x_{ijt}^r\}_{i,j \in \mathcal{N}_l, t \in \mathcal{T}}$, $\mathbf{x}_l^a := \{x_{it}^a\}_{i \in \mathcal{N}_l, t \in \mathcal{T}}$ 及び $\mathbf{x}_l^d := \{x_{it}^d\}_{i \in \mathcal{N}_l, t \in \mathcal{T}}$ 。エリア l の各時刻 $t \in \mathcal{T}$ の車両台数を $\mathbf{V}_l := [V_{lt}]_{t \in \mathcal{T}}$ で表す。この値は下記に示す式によりエリア l の一つ上位層でのリバランシング最適化結果に基づき算出され、エリア l の車両台数を最適化する際に利用される。

$$V_{lt} = s_{lt} + \sum_{j \in \text{Parent}(\mathcal{N}_l)} \left(x_{jlt-\tau_{jl}}^r + x_{jlt-\tau_{jl}}^p \right), t \in \mathcal{T}, \quad (2)$$

$$s_{lt} = \begin{cases} x_{lt}^a & t = 1 \\ x_{lt}^a - x_{lt}^d & t > 1 \end{cases}, t \in \mathcal{T}.$$

エリアを K 層に階層化した際の階層的リバランシング法 NERO_K を Algorithm 1 に示す。利用するには各エリア各時刻の車両台数は $\text{NERO}_K(\mathcal{N}_1, 1)$ を呼び出すことにより再帰的に求めることができる。ここで、 $\text{Demand}(\mathcal{N}_l)$ を $\mathcal{N}_l$ に含まれるエリア集合のリクエストの数を返すオペレータ、 $\text{Rebalancing}(\mathcal{N}_l, \tau_l, \mathbf{c}_l, \mathbf{V}_l, \lambda_l)$ をエリア l のリバランシング最適化結果 $(\mathbf{x}_l^r, \mathbf{x}_l^p, \mathbf{x}_l^a, \mathbf{x}_l^d)$ を返すオペレータ、 $\mathbf{V}_{max}$ を各時刻で利用可能な総車両台数とする。また $\mathcal{N}_1$ は最上位層のエリア集合である。

Algorithm 1 $\text{NERO}_K(\mathcal{N}, k)$

入力: エリアの集合 $\mathcal{N}$, 層のインデックス k

出力: リバランシングポリシーの集合 $\mathbf{x}$

```

1:  $\mathbf{x} \leftarrow \emptyset$ 
2: for  $l$  in  $\mathcal{N}$  do
3:    $\mathcal{N}_l \leftarrow \text{Children}(l)$ 
4:   if  $k = 1$  then  $\mathbf{V}_l \leftarrow \mathbf{V}_{max}$ 
5:   else  $\mathbf{V}_l \leftarrow \text{Computed using Eq. (2)}$ 
6:    $\lambda_l \leftarrow \text{Demand}(\mathcal{N}_l)$ 
7:    $(\mathbf{x}_l^r, \mathbf{x}_l^p, \mathbf{x}_l^a, \mathbf{x}_l^d) \leftarrow \text{Rebalancing}(\mathcal{N}_l, \tau_l, \mathbf{c}_l, \mathbf{V}_l, \lambda_l)$ 
8:    $\mathbf{x} \leftarrow \mathbf{x} \cup \{(\mathbf{x}_l^r, \mathbf{x}_l^p, \mathbf{x}_l^a, \mathbf{x}_l^d)\}$ 
9:   if  $k \leq K$  then  $\mathbf{x} \leftarrow \mathbf{x} \cup \text{NERO}_K(\mathcal{N}_l, k + 1)$ 
10: return  $\mathbf{x}$ 

```

4. 数値実験

本章では、タクシーの実データを用いて提案したリバランシング最適化法の有効性を検証する。提案法は階層的に最適化することで計算時間を短縮することができる。しかしながら上位層の粗い粒度のエリア間の移動時間は下位層の細かなエリア間の移動時間に比べ長く見積もる必要があるため、階層的に最適化する提案手法は従来法に比べ無駄な待機時間が発生する。そこで数値実験では提案手法の計算時間と、階層化することで発生する無駄時間について評価を行う。

4.1 実験設定

顧客のリクエストデータとして、オープンデータとして公開されているニューヨーク市マンハッタン島のタクシーのトリップデータ^{*1}を用いた。トリップデータに含まれる情報の内、乗車位置を出発位置、降車位置を目的地、乗車時刻を顧客が配車をリクエストした時刻とした。本実験では、リクエストデータ

^{*1} Brian, D. and Dan, W.: New York City Taxi Trip Data (2010-2013), <https://databank.illinois.edu/datasets/IDB-9610843> アクセス日: 2019/1/29

として 2013 年 5 月 3 日の 24 時間にマンハッタン島内で乗降車したデータを抽出し、1 時間毎に最適化した結果により評価を行った。本実験では、利用可能な総車両台数として 2013 年 5 月 3 日の各時間帯に少なくとも 1 人は客を乗せたタクシーの台数の 110%を用いた。これは、実際に客を乗せるために利用された台数に追加して、10%がリバランシングのために利用できると仮定した設定である。また最適化する際、マンハッタン島を均一な 3 種類メッシュ (500m, 1km, 2km) で分割し、各メッシュサイズでの隣接エリアまでの移動時間 Δt_{adj} はそれぞれ 3 分、6 分、12 分とした。各メッシュサイズで分割した際のエリアの数及び移動時間を表 1 に示す。

表 1: 各メッシュで分割時のエリア数及び隣接エリアまでの移動時間

	500-m mesh	1 km	2 km
N	261	79	26
Δt_{adj}	3 min.	6	12

提案手法を評価するために、階層数を 1 から 3 層まで変化した時の計算時間を評価した。階層数が 1 の場合が従来法 [Iglesias 18] におおよそ対応する*2。階層数の場合の各層のエリアのメッシュサイズの組合せを表 2 に示す。整数計画問題を最適化するソルバとして Gurobi-8.5*3 を用い、Intel Core-i7 CPU X980@3.33GHz 及び 46GB のメモリを有するコンピュータにより最適化計算を行った。

表 2: 比較手法

手法	エリアのメッシュサイズ
NERO(1 階層)	(500m)
NERO(2 階層)	(1km, 500m)
NERO(3 階層)	(2km, 1km, 500m)

4.2 実験結果

提案法の階層数と計算時間の関係を図 1 に示す。この図から提案法は 2 層、3 層と深くするに従い計算時間を短縮することができ、3 階層にすることで階層化しない場合に比べ計算時間を約 100 分の 1 にすることができた。

また、提案法の階層数と車両 1 台当りのリバランシングに費やす平均時間を図 2 に示す。この図から階層数を増やすにつれ、1 台当りのリバランシングに費やす移動時間が増加しており、3 階層の場合には階層化しない場合に比べ、リバランシングに費やす時間が約 3 分増加していることが分かった。

5. 結論

本研究では、より効率的な Mobility-on-Demand サービスの実現を目指し、サービスのエリアの数が増加により柔軟に対応できる手法を提案した。我々は、まず総車両台数が時々刻々変化する場合のリバランシング問題を整数計画問題として定式化した。次に、サービスエリアを粗い粒度で分割したエリアに対して最適化した後、その結果に基づきより詳細な粒度のエリアに対して最適化を行う階層的なリバランシング法を提案し

*2 総車両台数に関する制約 (1d) が追加されている点が異なり、制約 (1e), (1e) は 1 階層の場合常に 0 となるため無視される。

*3 Gurobi Optimization, LLC, Gurobi Optimizer Reference Manual, 2018, <http://www.gurobi.com> アクセス日: 2019/1/29

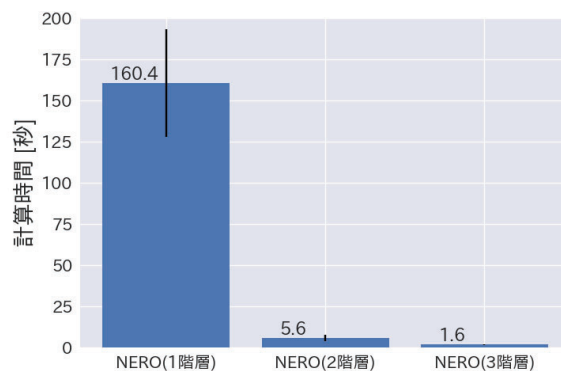


図 1: 最適化にかかる計算時間

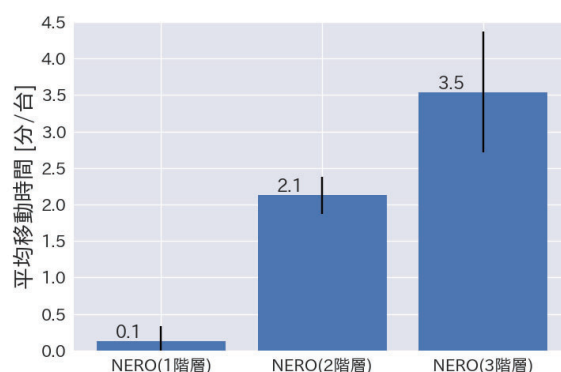


図 2: 一台当りのリバランシング費やした平均時間

た。また、提案手法の有効性を検証するために、ニューヨーク市マンハッタン島のタクシートリップデータを用い、階層数と計算時間及び、車両一台当りのリバランシングに費やす時間について評価した。その結果、提案法は階層数を増やすことで計算時間を指数関数的に削減することが出来ることを確認した。しかしながら、階層数を増やすことでリバランシングに費やす時間が増加することも分かった。今後は、より広いサービスエリアに対して提案法の有効性を検証していく予定である。

参考文献

- [Bei 18] Bei, X. and Zhang, S.: Algorithms for Trip-Vehicle Assignment in Ride-Sharing, in *AAAI Conference on Artificial Intelligence* (2018)
- [Dickerson 18] Dickerson, J. P., Sankararaman, K. A., Srinivasan, A., and Xu, P.: Allocation Problems in Ride-Sharing Platforms: Online Matching with Offline Reusable Resources, in *AAAI Conference on Artificial Intelligence* (2018)
- [Iglesias 18] Iglesias, R., Rossi, F., Wang, K., Hallac, D., Leskovec, J., and Pavone, M.: Data-Driven Model Predictive Control of Autonomous Mobility-on-Demand Systems, in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7 (2018)

- [Pavone 11] Pavone, M., Smith, S. L., Frazzoli, E., and Rus, D.: Load Balancing for Mobility-on-Demand Systems, in *Robotics: Science and Systems VII (RSS)* (2011)
- [Pelzer 15] Pelzer, D., Xiao, J., Zehe, D., Lees, M. H., Knoll, A. C., and Aydt, H.: A Partition-Based Match Making Algorithm for Dynamic Ridesharing, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 5, pp. 2587–2598 (2015)
- [Spieser 16] Spieser, K., Samaranayake, S., Gruel, W., and Frazzoli, E.: Shared-vehicle Mobility-on-Demand Systems: A Fleet Operator’s Guide to Rebalancing Empty Vehicles, *TRB 2016 Annual Meeting*, pp. 0–16 (2016)
- [Zhang 16] Zhang, R. and Pavone, M.: Control of Robotic Mobility-On-Demand Systems: a Queueing-Theoretical Perspective, *The International Journal of Robotics Research*, Vol. 35, No. 1-3, pp. 186–203 (2016)