https://ieeexplore.ieee.org/document/9838069

https://doi.org/10.23919/ECC55457.2022.9838069

Idle-vehicle Rebalancing Coverage Control for Ride-sourcing systems*

Pengbo Zhu[1], Isik Ilber Sirmatel[2], Giancarlo Ferrari Trecate[3], Nikolas Geroliminis[1]

Abstract— **Ride-sourcing system can provide passengers with fast and efficient service with a fleet of vehicles, while asymmetry between origin and destination distributions of trips, nonuniform passenger's demand for rides in different districts creates imbalances in the spatial distribution of these vehicles. Thus proactively relocating idle vehicles to the high-demand regions, also known as vehicle rebalancing is an emerging problem that can have a significant improvement for the efficiency of urban transportation. We formulate this problem as a coverage problem for coordination and deployment of multiple mobile agents in city scenarios, which vehicles can benefit from by allocating them according to the different demand densities of different city districts. A Voronoi-based control algorithm is proposed by leveraging the local information of each vehicle. The effectiveness of the proposed method is validated by a simulator modeled on a real road map from Shenzhen, China. Compared to baseline, our proposed method is able to serve more trips with less passenger waiting time.**

I. Introduction

Nowadays, the development of wireless communication and the widespread use of smartphones have popularized ride-sourcing services which revolutionizes the mobility in urban areas. Such as Uber, Lyft and DiDi, Transportation Network Companies(TNCs) provides an internet-based platform for passenger demand and vehicle supply in real time. They deploys a group of vehicles within city areas, serving people's requests for in-time and door-to-door rides.

However, the imbalance of demand and supply violates the system efficiency. It can be caused by the asymmetry between the trip origin and destination distribution and the different demands for rides in different districts. As a consequence of the unideal spatial distribution of ride-sourcing vehilces, there is a long passenger waiting time and a high cancellation rate. Thus the benefit of improving the guidance or repositioning idle/empty vehicles is expected to achieve efficiency and sustainability. And it is of great interest to overcome this problem from a control point of view that deploys coordinated control algorithms to rebalance vehicles to districts with current or future high demand [1].

Although studies in ride-sourcing systems on price policy, vehicle routing [2], fleet sizing [1] an vehicle-splitting[3] problems have been conducted by many authors, this spatial vehicle rebalancing problem is still insufficiently explored. On the other hand, most papers addresses the need for car-sharing and bike-sharing systems [4], which has a large rebalancing time interval(e.g., sharing-bikes are relocated by mini-vans during the midnight once per day), so far only a few in the ride-sourcing systems [3][5]. In [6], assuming n vehicles are allocated at n stochastic median locations to serve orders surrounding its Voronoi cell, a rebalancing policy is introduced for light-load scenarios. [7] introduces a fluid discrete-time approximation model and an optimization approach. A model predictive control scheme is presented to address constraints on vehicle availability in [8] and [9]. Also reinforcement learning algorithms has been applied in [10]. In [11], empty autonomous vehicles are simply sent to underserved areas. However, most of these works formulate the system with a limited number of virtual stations and a small fleet size, which is hard to be applied in large-scale and door-to-door services, while in this paper, we can provide node-level rebalancing position instead of the rather rougher region-level ones.

In the past several decades, coverage control algorithm has played an important role in multi-agent systems due to its ability to coordinate and control simultaneously [12]. The objective of coverage control is to optimize a selected criterion, e.g. minimize the energy cost [13] or maximize the covered area [14], [15], [16]. The system is described in a predefined environment, where mobile agents, such as UAVs, vehicles, and robots, move around and react autonomously using local information collected from their neighbors. Most of these works are based on the Voronoi partition to determine an optimal configuration of multiple agents over a bounded area. Centroidal Voronoi configurations [17], the generators of which coincide with the center of mass (i.e., centroid) of each Voronoi cell, are widely used in location optimization problems. Lloyd's algorithm [18] provides a tractable approach to compute the centroidal Voronoi configuration and steers agents towards the centroids of their own cells. However, the aforementioned works focus on continuous environments (i.e., geographical spaces), where the mobile agents can move in any direction. Considering a practical setting, in this paper we extend these previous results by applying a coverage control algorithm to a discrete map, where the vehicles are moving along real urban roads and can only change their directions at their next intersections.

To the best of our knowledge, this work constitutes the first attempt at developing fleet balancing control systems via application of the coverage control method. The following parts of this paper are organized as follows: Firstly,

*Research supported by NCCR Automation, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 51NF40_180545)

[1] École Polytechnique Fédérale de Lausanne, Urban Transport Systems Laboratory, 1015 Lausanne, Switzerland, pengbo.zhu@epfl.ch, nikolas.geroliminis@epfl.ch

[2] Trakya University, Department of Electrical and Electronics Engineering, Control Section, Türkiye, iilbersirmatel@trakya.edu.tr

[3] École Polytechnique Fédérale de Lausanne, Dependable Control and Decision Group, 1015 Lausanne, Switzerland, giancarlo.ferraritrecate@epfl.ch

problem formulation for coverage control in continuous environment are presented. The proposed method is first examined in a simple simulator in continuous space with constant speed for all vehicles, and then a simulator built on a real mega-city map with discrete space is introduced. We study the assumptions and approximations in the discrete space operating environment for evaluating our proposed method in a practical setting. Various simulation scenarios employing different imbalance levels between the trip origin and destination distribution are conducted. Passenger waiting times and request completion rates are used for comparing the performance of the proposed method with a baseline without rebalancing. Finally, conclusions and potential future steps are discussed.

II. Problem statement

The coverage control problem is well-defined by a large number of existing studies in robotic fields. The following problem statement and control law formulation adapts these works to a mobile vehicle system which is largely based on reference [13] and [14], and the corresponding practical meaning of variables existing in real transport networks are explained.

A. Preliminaries

Assume that the region of interest $\Omega \subset \Re^2$ is a convex space. A probability density function $\phi : \Omega \to [0, 1)$ is a continous integrable function which reflects the possibility of occurence of event on each point in Ω in general localization optimization problem. Specifically, here ϕ is utilized to describe how much demand for rides is on each position in the urban scenario. A number of n mobile agents or moving vehicles are deployed whose position vectors are $X = \{x_1, x_2, ..., x_n\}, x_i \subset \Omega$, and the movements of vehicles are constrained inside Ω. Each agent i is assumed to cover a circular area with a limited radius r. We define the covered area as $S = \{S_1, S_2, \ldots, S_n\}$, where

$$S_i(x_i, r) = \{q \in \Omega : \|q - x_i\| \leq r\}, \tag{1}$$

where $\|\cdot\|$ is Euclidean norm, as the blue circular area shown in Fig. 1c.

Distributed control algorithms have attracted much attentions in recent research due to their computation efficiency which only require local information. They can be implemented on each agent to compute its own control actions by communicating with its spatial neighbors. Considering distributed coverage problems for a set of agents can be addressed with help of Voronoi or Voronoi-inspired partitioning scheme, the Voronoi diagram is used here to completely partition the environment Ω as

$$V_i(x_i) = \{q \in \Omega : \|x_i - q\| \leq \|x_j - q\|, \forall j \in I_n, j \neq i\}, \quad i \in I_n \tag{2}$$

as the orange area shown in Fig. 1c, with the property that

$$\bigcup_{i \in I_n} V_i = \Omega, \qquad int\{V_i\} \cap int\{V_j\} = \emptyset, \quad \forall i \neq j \tag{3}$$

where $int\{V_i\}$ denotes the interior of V_i. Voronoi diagram can give a tessellation that the responsibility area for an agent (Voronoi cell) is the area in Ω including all points that are closer to it than any other agent.

As the duality of Voronoi diagram and Voronoi-weighted Delaunay graphs, we can compute Voronoi cell V_i of agent i only with its Delaunay neighbors N_i, which is defined as

$$N_i = \{j \in I_n, j \neq i : V_i \cap V_j \neq \emptyset\}, i \in I_n. \tag{4}$$

A performance function $f : [0, \infty) \to \Re$ is a differentiable function which degrades with the distance between a agent and a given position, i.e., $\|x_i - q\|$. Here we choose $f(\cdot) = -\|\cdot\|^2$. The goal is to maximize the covered area by all agents while also taking the density function ϕ into account. Thus the coverage objective function $H{:}\Omega \to \Re$ can be formulated as follows,

$$H(X, W) = - \sum_{i=1,2,\ldots,n} \int_{q \in W_i} \|x_i - q\|^2 \phi(q) dq \tag{5}$$

where $W_i = V_i \cap S_i$, as the gray area shown in Fig. 1c.

B. Distributed Control Law Formulation

$Proposition$: The local maximum of H can be obtained when X_i are located at centroids of their respective Voronoi cells, i.e., the position of each agent x_i simultaneously satisfies two conditions: It is the the seed/generator of its Voronoi cell, at the same time, it is located at its center of mass(i.e., centroid). This critical partition P and points configuration X for H is named as *Centroidal Voronoi Configuration* [17].

Similarly, for a region $W_i \subset \Re^n$, if we consider the probability density function ϕ as a mass density function, then the mass M_{W_i}, first moment L_{W_i}, centroid C_{W_i}, and polar moment of inertia J_{W_i,x_i} can be obtained as follows respectively,

$$\begin{aligned} M_{W_i} &= \int_{W_i} \phi(q) dq, \quad L_{W_i} = \int_{W_i} q\phi(q) dq, \\ C_{W_i} &= \frac{M_{W_i}}{L_{W_i}}, \\ J_{W_i,x_i} &= \int_{W_i} \|x_i - q\|^2 \phi(q) dq. \end{aligned} \tag{6}$$

According to parallel axis theorem, we can rewrite the polar moment of inertia as

$$J_{W_i,x_i} = J_{W_i,C_{W_i}} + M_{W_i} \|x_i - C_{W_i}\|^2, \tag{7}$$

where $J_{W_i,C_{W_i}}$ is the polar moment of inertia of the region W_i about its centroid C_{W_i}.

Considering Eq.5 again, the objective function and its partial derivative with respect to x_i can be expressed as:

$$\begin{aligned} H(X, W) &= -\sum\nolimits_{i=1}^{n} J_{W_i,x_i} \\ &= -\sum\nolimits_{i=1}^{n} J_{W_i,C_{W_i}} - \sum\nolimits_{i=1}^{n} M_{W_i} \|x_i - C_{W_i}\|^2, \\ \frac{\partial H(X, W)}{\partial x_i} &= -2M_{W_i}(x_i - C_{W_i}). \end{aligned} \tag{8}$$

Obviously, solving $\partial H(X,W)/\partial x_i = 0$ leads to the conclusion that the optimal configuration involves positioning all agents at the centroids of their respective Voronoi cells, i.e. $x_i = C_{W_i}$. This optimal partition of environment and critical agent positions yield the so called *Centroidal Voronoi Configuration*.

By allocating n agents, the control algorithm should maximize the coverage objective function H. To achieve this goal, the control law proposed here makes agent position x_i follow a gradient descent flow. Let the first order derivative of agent displacement as control input be as follows

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = u_i. \tag{9}$$

By LaSalle's principle, it can be proved that when we set the local control law as

$$u_i = -k_p(x_i - C_{W_i}). \tag{10}$$

will steer the agent team to converge to the centroidal Voronoi configuration, where k_p is a positive control gain. We use a rasterized area to calculate the centroid [19]. Furthermore, when agent's movement obeys the control law in Eq.10, the objective function H can be guaranteed to increase due to

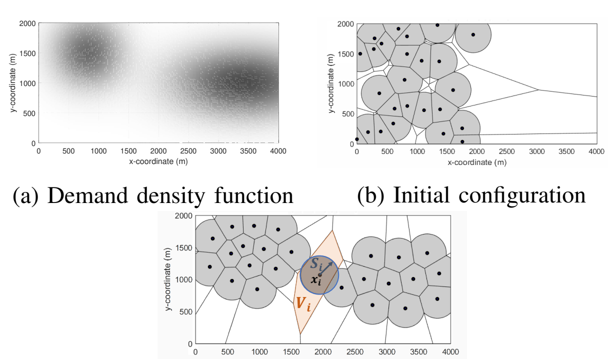
$$\frac{\mathrm{d}H}{\mathrm{d}t} = \frac{\partial H(X,W)}{\partial x_i}\frac{\mathrm{d}x_i}{\mathrm{d}t} \tag{11a}$$

$$= 2k_p \sum\nolimits_{i=1}^{n} M_{W_i}\|x_i - C_{W_i}\|^2 > 0. \tag{11b}$$

Note that this control algorithm shows adaptability when dealing with a dynamic system, e.g., when the empty vehicle fleet size is changing with time due to departure or arrival of vehicles, the control law can always allocate agents to the optimal configuration for the current situation.

III. Simulation Study

A. Case Study I: Continuous map case



(a) Demand density function (b) Initial configuration

(c) Final configuration

Fig. 1: Continuous case study

In this simulation, mobile agents can move freely inside a rectangle-shape continuous map with maximum speed v_{max}. The discrete-time dynamics for each agent is

$$x_i(k+1) = x_i(k) + Tu_i(k). \tag{12}$$

Under the control law in Eq. 10, we add the speed limitation to the agents. Let the agent move toward the centroid of its Voronoi cell at its maximum speed v_{max}, then the control input is

$$u_i(k) = -v_{max} \cdot \frac{(x_i(k) - C_{W_i}(k))}{\|(x_i(k) - C_{W_i}(k)\|}, \tag{13}$$

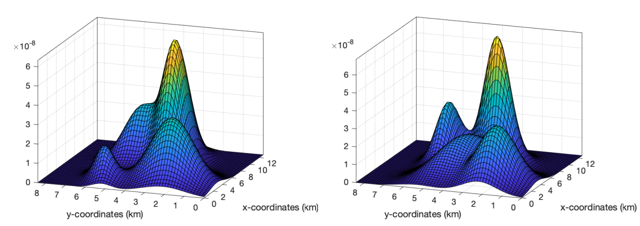
where T is the sampling time.

A fleet of 25 vehicles are employed in this simulation with random initial positions as shown in Fig. 1b. The demand density function $\phi(q)$ follows a bimodal distribution, and its values are illustrated by color gradient in Fig. 1a, where the high demand areas are indicated with dark gray. Fig. 1b and Fig. 1c show the initial and final fleet configurations, respectively. As can be seen from the comparison between Fig. 1b and Fig. 1c, the fleet positions converge from their initial random positions to the centroidal voronoi configuration with the proposed control law.

B. Case Study II: Discrete space case with Shenzhen network

1) Introduction for simulation architecture: Our proposed control method is examined on a discrete space simulator based on the urban road network of Shenzhen, China. The network consists of 1858 intersections and 2013 road links. Given a undirected graph $G = (V, E)$ to represent the city map, where E is the set of real roads and V is the set of vertices, i.e. intersections for urban roads, and please note here V is not Voronoi cell in Section II. More detailed information can be found in [20].

2) Data preprocessing and generation: Collected in 24 hours from Luohu District in Shenzhen, the historical data contains 199,819 trips consisting of the following: Origins, destinations, and the time when passengers send out a request for a ride. The demand level can be evaluated by the rate of requests originating from each intersection. With sufficient historical trip data, we can use the Gaussian mixture model approach to estimate the origin distribution, i.e. the continuous demand density function $\phi(\cdot)$ required by our control algorithm. Similar to the destination, the estimate distributions are shown in Fig. 2



(a) origin distribution. (b) destination distribution.

Fig. 2: Estimated Gaussian mixture models for trip origin and destination distributions.

To be able to generate simulation scenarios emphasizing the difference between trip origin and destination distributions, we introduce an imbalance parameter γ. Assume that the global maximum of the origin distribution is $p_{o,max}$, let

$$p'_o(q) = p_{o,max} - p_o(q), \qquad q \in V, \tag{14}$$

By normalizing over all points, we create an artificial distribution p_o' which has the maximum difference from the origin distribution. Here, $p_o(q)$ is the probability on q for original origin distribution.

Then we can generate the destination distribution as follows,

$$p_d'(q) = \gamma \cdot p_d(q) + (1-\gamma)\cdot p_o'(q), \qquad q \in V, \tag{15}$$

where $p_d'(q)$ and $p_d(q)$ indicate the probability of generated and original destination distributions respectively.

When $\gamma = 1$, the generated destination distribution is equal to the original destination distribution. The smaller the γ is, the more discrepancy is introduced between the origin and generated destination distributions. When $\gamma = 0$, the generated destination distribution has a shape that is maximally different than the origin one.

3) Experimental Setup: The passenger requests are sampled from the original origin distribution p_o, and generated artificial destination distributions p_d' with different γ values. The time pattern of the requests follows a low-high-low demand file where each period lasts 1 hour and about 2400 trip requests are introduced in a 3-hour simulation. Fig. 3 shows a snapshot of the simulator. All vehicles move along a discrete geographical space representing real urban roads and can only stop or change their directions at next intersections. The blue dots stand for idle vehicles, and the gray circular disk is depicts the r-limited area covered (from the perspective of the coverage control objective function) by each vehicle. These empty vehicles always move following their centroids at each time step. During our simulations, it shows that the system performances are improved with the increase value of r at first but will converge with a big enough value around $1000m$ in the end. Considering the computation load, a bigger value of r actually requires for a larger scale of the demand density information, so we set $r = 1000m$ here.

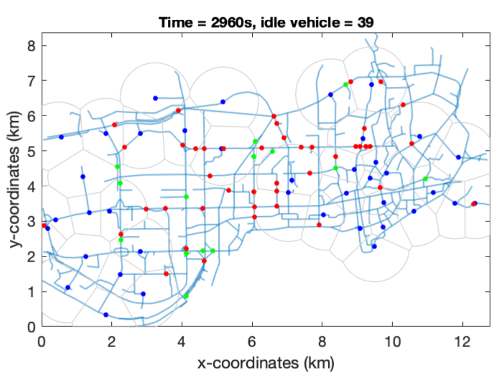


Fig. 3: A snapshot of the simulator. The idle/empty, passenger-assigned, and passenger-carrying vehicles are dots in blue, green and red respectively. A demo video is available on youtube : `https://youtu.be/1TJapI8qtwI`

Each passenger trip request has a maximum waiting time tolerance w_{tol}. When one passenger i sends out a request at $t_0(i)$, the system will search for available idle vehicles and calculate the estimated pick-up time $t_p^e(i)$ for the nearest vehicle according to current moving speed. If $t_p^e(i) - t_o(i) < w_{tol}$, this vehicle will be assigned to the passenger(represented as a green dot on Fig. 3). It then travels toward the origin position of the passenger to pick her/him up; after picking up the passenger (represented as a red dot on Fig. 3), it will travel to the destination. The traveling path is computed by Floyd-Warshall algorithm [21] which gives the shortest path between any two nodes in a graph. The passenger-assigned (i.e., green) and passenger-carrying (i.e., red) vehicles have no contribution towards the coverage objective, while the proposed control method will continuously balance the idle vehicles (i.e., blue; the remainder of the vehicle fleet) in order to guarantee more vehicles gravitate towards the high-demand areas. Therefore, there can be a quick response for current and future requests. Once a passenger arrives at her/his destination, the vehicle becomes idle again and joins the group of vehicles operated by the coverage control algorithm.

Considering that the vehicle might move in the network with various speeds at different times, we denote the time when passenger i is actually picked up as $t_p^r(i)$, and the total number of successfully completed orders as N_1. The average waiting time can be computed as:

$$t_w = \frac{\sum_{i=1}^{N_1} t_p^r(i) - t_o(i)}{N_1}, \tag{16}$$

However, if $t_p^e(i) - t_o(i) > w_{tol}$, then the passenger keeps waiting for being matched for 1 min. At $(t_0(i)+1)$, if there are still no vehicles available that can respond to this passenger, this order is deemed as cancelled. In this case, we assume that the passenger will choose to drive her/his private vehicle to travel, which will increase the accumulation m in the roads leading to congestion[20].

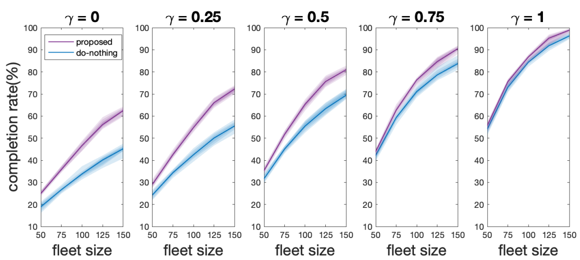
$$v(m) = \begin{cases} 36e^{(\frac{-29m}{72000})}, & m \le 4320, \\ 6.31 - 2.33(m-4320), & 4320 < m \le 7200, \\ 0, & m > 7200. \end{cases} \tag{17}$$

Considering the MFD obtained in [22] which expresses a function between the network speed $v(km/h)$ and accumulation of private and ride-sourcing vehicles $m(veh)$ in Eq. (17), increasing the number of private vehicles leads to a lower average speed. We can calculate the average system time t_{sys} as

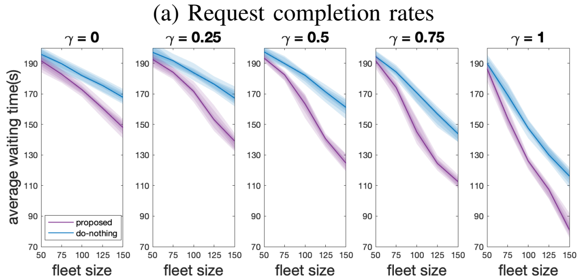
$$t_{sys} = \frac{\sum_{i=1}^{N_1}(t_p^r(i) - t_o(i)) + N_2 \cdot \alpha \cdot w_{tol}}{N}, \tag{18}$$

where N_2 is the number of canceled orders, N is the total number of requests during the simulation time (with $N = N_1 + N_2$), α is a weight parameter representing the time value penalty for cancelled orders. The percentage of successfully completed requests (i.e., request completion rate) can be found from $100\% \times N_1/N$. In the simulations and analysis, we choose $\alpha = 1.5$ and $w_{tol} = 5$ minutes.

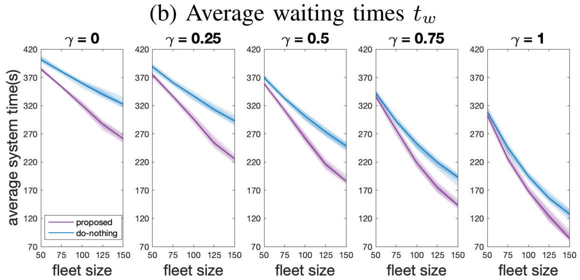
4) Simulation results and analysis: In order to illustrate the influence of trip origin and destination imbalances, we create artificial scenarios with different levels of imbalance parameter γ. Our method is compared with the baseline method, where vehicles stay at the destination of their last order until they are dispatched to their next passengers (which we denote here as the *'do-nothing policy'*). Our proposed method (shown in purple) is compared with the baseline method (shown in blue) using a set of 30 randomly generated simulation scenarios, and the results are shown in Section III-B.4, evaluating the two methods on request completion rates, average waiting times and system times, with varying values of γ and fleet size. The solid lines presents the mean value for the 30 runs, while percentiles are shown in shades for varying degrees $(25\%, 50\%, 75\%,$ and 90%, respectively).



(a) Request completion rates



(b) Average waiting times t_w



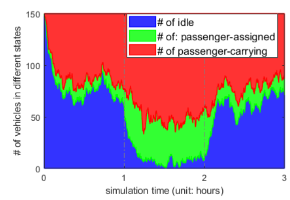
(c) Average system times t_{sys}

Fig. 4: Comparison for various γ and fleet size values.

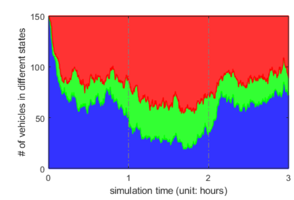
For all simulations using various values of the origin destination demand imbalance parameter γ, our method can improve system performance by yielding lower waiting and system times, and is able to serve more trips. This verifies that the coverage control scheme can efficiently balance the spatial distribution of idle vehicles by allocating more vehicles around high-demand areas and dynamically rebalance their positions after dropping off passengers.

When $\gamma = 1$, both methods do well due to the origin and destination distributions being similar, requiring little rebalancing where the do-nothing approach is a good enough choice. While for the request completion rate for $\gamma = 1$, no significant differences occur between the two methods (i.e., only an improvement of around 3%), our method shows substantially decreased waiting and average system times, especially with a large fleet size. With lower γ values the performances of both methods decline as expected. With an increased imbalance, there will be more orders with origins and destinations further away from each other, causing the coverage control algorithm to require more time to steer the vehicles from their last destinations towards the high-demand areas. However, the results show that our proposed method yields better results than the baseline.

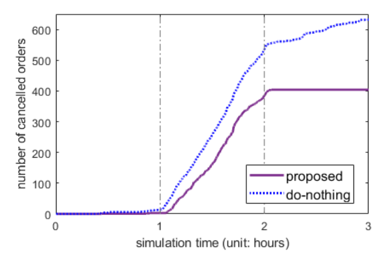
From the results with varying fleet size values, it can be seen that the performance metrics improve for both methods with increased fleet size as expected. Furthermore, our proposed method can achieve similar or even better performance with 25 or fewer vehicles. For example, in Fig. 4b, for $\gamma = 0.5$ and a fleet size of 100, our method can yield an average waiting time of around 160 s compared to the baseline with 180 s, indicating an improvement of about 11%; to achieve a t_w around 160 s, the baseline requires a fleet size about 150, which is 50 vehicles more than our proposed method.



(a) proposed method



(b) do-nothing policy



(c) No. of accumulated canceled orders

Fig. 5: Comparison of different states of vehicles and number of canceled orders

Furthermore, without loss of generality, one simulation experiment under origin-destination trip demand imbalances $\gamma = 0.5$ with a fleet size of 150, 2400 requests in 3 hours is presented here to demonstrate the improvement of our proposed method compared with the do-nothing policy. The results in Fig. 5 depict the trajectories of vehicle numbers with various operating (i.e., idle, passenger-assigned, and passenger-carrying) states together with the number of cancelled orders. The solid lines present our proposed method while results of the do-nothing policy are shown by dotted lines. The vehicle state trajectories in Fig. 5a indicate that the proposed method is able to operate the fleet more efficiently as a larger amount of vehicles are actively serving passengers for most of the time. On the other hand, the do-nothing policy operates with a larger amount of idle vehicles, which manifests itself as a greater amount of cancelled orders, as can be seen from Fig. 5c. During the hour 1-2, a steeper upward tilt to the curve for the do-nothing policy can be observed. And after the high-demand period, the number of canceled orders for the do-nothing policy still keeps increasing, while our proposed one can manage to respond to new requests almost without any more cancellations. From Table I we can see that the proposed method can increase completed requests, while decreasing average waiting and system times, showing substantial improvement in system operation. Overall, the results indicate that the ability of coverage control in keeping the vehicles in advantageous locations for demand responsiveness, thus succeeding to have more vehicles matched to passengers in a timely manner, translates to superior performance over the do-nothing policy, suggesting strong potential for practice in ride-sourcing system management.

TABLE I: Performance metrics of two methods

	proposed method	do-nothing policy	improvement
completion rate (%)	82.89	73.19	13.25 % ↑
average waiting time (s)	132.46	173.88	23.82 % ↓
average system time (s)	186.80	247.91	24.65 % ↓

IV. Conclusion

In this paper, we proposed the application of the coverage control method (from the robotics and motion planning literature) to rebalance vehicle fleets for ride-sourcing systems. For countering spatiotemporal imbalances in the origins and destinations of trip demands, our proposed method can dynamically rebalance spatial distribution of idle vehicles to serve more trips with less waiting time. Our coverage control algorithm is tested on both continuous and discrete space simulators using real road network geometry and trip data from the city of Shenzhen and its performance is compared with a do-nothing policy. The results currently rely on a prior knowledge of a static demand density function. Thus in future work, a method to estimate time-varying demands from real-time data can be investigated. Furthermore, as the ride-splitting service can answer more requests with smaller fleet size, research on pooling problems is a research priority.

References

[1] A. Wallar, M. Van Der Zee, J. Alonso-Mora, and D. Rus, "Vehicle rebalancing for mobility-on-demand systems with ride-sharing," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4539–4546.

[2] J. Alonso-Mora, A. Wallar, and D. Rus, "Predictive routing for autonomous mobility-on-demand systems with ride-sharing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3583–3590.

[3] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences*, vol. 114, no. 3, pp. 462–467, 2017. [Online]. Available: https://www.pnas.org/content/114/3/462

[4] A. Faghih-Imani, R. Hampshire, L. Marla, and N. Eluru, "An empirical analysis of bike sharing usage and rebalancing: Evidence from barcelona and seville," *Transportation Research Part A: Policy and Practice*, vol. 97, pp. 177–191, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965856416311648

[5] B. Boyacı, K. G. Zografos, and N. Geroliminis, "An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations," *Transportation Research Part B: Methodological*, vol. 95, pp. 214–237, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261515301119

[6] M. Pavone, K. Treleaven, and E. Frazzoli, "Fundamental performance limits and efficient polices for transportation-on-demand systems," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5622–5629.

[7] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, "Load balancing for mobility-on-demand systems," in *Robotics: Science and Systems*, 2011.

[8] F. Miao, S. Lin, S. Munir, J. A. Stankovic, H. Huang, D. Zhang, T. He, and G. J. Pappas, "Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach," in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, ser. ICCPS '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 100–109. [Online]. Available: https://doi.org/10.1145/2735960.2735961

[9] R. Zhang, F. Rossi, and M. Pavone, "Model predictive control of autonomous mobility-on-demand systems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1382–1389.

[10] J. Wen, J. Zhao, and P. Jaillet, "Rebalancing shared mobility-on-demand systems: A reinforcement learning approach," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 220–225.

[11] A. Carron, F. Seccamonte, C. Ruch, E. Frazzoli, and M. N. Zeilinger, "Scalable model predictive control for autonomous mobility-on-demand systems," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 635–644, 2021.

[12] S. Martinez, J. Cortes, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75–88, 2007.

[13] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[14] S. Papatheodorou, Y. Stergiopoulos, and A. Tzes, "Distributed area coverage control with imprecise robot localization," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, 2016, pp. 214–219.

[15] S. Papatheodorou and A. Tzes, *Theoretical and Experimental Collaborative Area Coverage Schemes Using Mobile Agents*, 11 2018.

[16] S. Papatheodorou, A. Tzes, K. Giannousakis, and Y. Stergiopoulos, "Distributed area coverage control with imprecise robot localization: Simulation and experimental studies," *International Journal of Advanced Robotic Systems*, vol. 15, no. 5, p. 1729881418797494, 2018. [Online]. Available: https://doi.org/10.1177/1729881418797494

[17] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, pp. 637–676, 1999.

[18] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[19] N. P. Rougier, "A density-driven method for the placement of biological cells over two-dimensional manifolds," *Frontiers in Neuroinformatics*, vol. 12, 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fninf.2018.00012

[20] C. V. Beojone and N. Geroliminis, "On the inefficiency of ride-sourcing services towards urban congestion," *Transportation Research Part C: Emerging Technologies*, vol. 124, p. 102890, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X20307907

[21] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, no. 6, p. 345, June 1962. [Online]. Available: https://doi.org/10.1145/367766.368168

[22] Y. Ji, J. Luo, and N. Geroliminis, "Empirical observations of congestion propagation and dynamic partitioning with probe data for large-scale systems," *Transportation Research Record*, vol. 2422, no. 1, pp. 1–11, 2014. [Online]. Available: https://doi.org/10.3141/2422-01