Transportation Research Part C 131 (2021) 103336

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Decentralised cooperative cruising of autonomous ride-sourcing fleets

Linji Chen, Amir Hosein Valadkhani, Mohsen Ramezani *

The University of Sydney, School of Civil Engineering, Australia

ARTICLE INFO

Keywords:
Mobility on-demand
Matching
E-hailing
Distributed search
Communication reliability
Anticipatory planning
Multi-agent systems

ABSTRACT

As transportation network companies and automobile manufacturers continue to invest in the development of self-driving vehicles, it can be expected that autonomous taxi (a-taxi) fleets will become a major component of on-demand transport services in the foreseeable future. The majority of existing automated fleet management systems focus on central dispatch strategies that rely on real-time information and communication. This paper proposes a novel *decentralised cooperative cruising* method for offline operation of a-taxi fleets, which serves as a contingency plan during a full communication shutdown. The proposed method acts as an emergency plan for the system to continue serving passengers with the objective of maximising the total number of served passengers by the fleet. The method uses historical trip data to estimate PageRank centralities of roads as a proxy of long-term likelihood of finding waiting passengers over a series of trips. The proposed method uses this metric to (i) compute weighted shortest paths for vacant a-taxi cruising route planning, and (ii) partition the network into homogeneous regions for effective cruising destination choice (mission planning). The movements of vacant a-taxis between regions are modelled as a Markov chain such that a transition probability matrix is computed to achieve the optimal spatial distribution of vacant a-taxis to maximise the total expected pick-ups by the fleet, estimated based on bilateral meeting functions. Compared to benchmark strategies which select destinations randomly and cruise along the shortest travel time path, the proposed method shows significant improvements in service performances for different fleet sizes.

1. Introduction

On-demand shared mobility services such as ride-sourcing systems have experienced explosive growth in recent years and have emerged as an essential mode for daily commute. Meanwhile, investments have flowed into the autonomous vehicle industry (e.g. by Google, Uber, Tesla, Argo, and Cruise among others) to commercialise self-driving vehicles. With a merger between the two, it is expected that on-demand mobility services will deploy self-driving technology in the near future, and autonomous taxi (a-taxi) fleets will become a pillar of on-demand transportation services.

The majority of ride-sourcing fleet management methods in literature require real-time (i) location information of vehicles/a-taxis (moving agents) and waiting passengers (stationery resources), and (ii) communication between a central dispatching unit and a-taxis. These make centralised methods prone to communication disruption. In this paper, we propose a *decentralised* method to determine cruising destinations and cruising paths for fully automated taxi fleets when the communication between a-taxis and passengers via the

* Corresponding author.
E-mail address: mohsen.ramezani@sydney.edu.au (M. Ramezani).

https://doi.org/10.1016/j.trc.2021.103336
Received 24 December 2020; Received in revised form 30 May 2021; Accepted 4 August 2021
0968-090X/

central dispatch unit is lost. Furthermore, the method is designed to be *cooperative* such that a-taxis cruise by anticipating the movements of other a-taxis to collectively increase the total number of served passengers, rather than competing with each other. The designed method enables the a-taxi fleet to continue operating during a communication shutdown by utilising historical data of passengers pick-up and drop-off locations.

Designing cruising and matching strategies for fleets of moving agents (e.g. taxis, ride-sourcing vehicles, a-taxis, and in general the supply) to meet/find stationary resources (e.g. waiting passengers, and in general the demand) have been studied in the literature considering different problem settings. For static type of dial-a-ride problems (i.e. the future information of supply and demand is known in advance with less emphasis on real-time changes), the optimal cruising path for picking up multiple passengers is often studied as a pick-and-delivery problem (Psaraftis et al., 2016; Berbeglia et al., 2010; Naccache et al., 2018; Cordeau, 2006; Colorni and Righini, 2001). For dynamic types of the problem with real-time spatial and temporal changes in supply and demand (e.g. taxi systems), a plethora of papers tackle different operational aspects of these systems, mostly with centralised considerations. For example, Wong et al. (2014, 2015) suggested a cruising path for taxis by finding the cumulative probability of successful passenger pick-ups along the routes. Ramezani and Nourinejad (2018) studied the use of centralised macroscopic approaches for transferring (repositioning) of taxis in large networks. A centralised dispatching system by shifting the priority from customers to taxis is studied in Maciejewski et al. (2016). A combination of centralised and decentralised dispatching systems has been proposed to decrease the required workload of dispatching in Duan et al. (2020). More specifically, various centralised dispatching methods for autonomous fleets to find passengers have been developed, for instance see (Liang et al., 2020; Hörl et al., 2019; Vosooghi et al., 2019; Hyland and Mahmassani, 2018; Ma et al., 2017) among others. Interested readers may also refer to review articles (Pillac et al., 2013; Narayanan et al., 2020; Hyland and Mahmassani, 2017; Mourad et al., 2019) that introduce detailed characteristics of on-demand shared mobility systems and shared autonomous fleets.

To date, few studies have been dedicated to address the effect of incomplete, uncertain, or complete shutdown of information and communication channels on the operation of ride-sourcing fleets. This requires investigating decentralised or distributed cruising methods. For example, Guo and Wolfson (2018) utilised a gravitational model for decentralised route planning of fleet vehicles. Ayala et al. (2018) fused historical data with online information modification to temporarily modify the attractiveness of locations in which there are no resources (e.g. waiting passengers) to reduce sensitivity to errors and noises in real-time communication. Hu et al. (2019) and Buchin et al. (2019) investigated the dispatch problem when taxi drivers make decentralised cruising decisions without communicating among themselves. Hu et al. (2019) used k-means method to spatially cluster Manhattan into 150 regions. Each region is ranked by a weight affected by the number of pick-ups and drop-offs in the region. Taxi drivers then choose their destinations based on the ranking. Buchin et al. (2019) mainly uses a Lévy flight strategy to limit the choice set of intersections as cruise destinations before performing a weighted random selection. Both methods have a risk of creating local supply surplus and accumulating excessive vacant taxis over time.

The issue of communication loss can also be tackled with a different approach; by designing algorithms that are resilient to communication loss or disruptions. For instance, resilient control strategies have been developed to address cyber-attacks and disruptions in perimeter control applications (Haddad and Mirkin, 2020; Mercader and Haddad, 2021). The proposed method can be classified as an offline anticipatory multi-agent method that aims to search for static (impatient) targets with unknown-locations in a given environment. Multi-agent coordination is often studied in the field of robotics. For example, Zedadra et al. (2016) investigated foraging problems which require agents to roam in an unknown environment to search for resources (i.e. food) and transport them back to depots (i.e. nests). The algorithm relies on indirect communication by leaving trail marks (i.e. pheromones). The limitation of real-time communication and information are also common in research on unmanned aerial vehicles (UAVs) and autonomous underwater vehicles (AUVs) due to their complex operational environments. Although the methods cannot be directly appplyied to transportation networks, concepts such as fleet cooperation and mission planning are relevant to our paper. Hadi et al. (2021) reviews recent literature of cooperative control methods for AUVs, and Eaton et al. (2016) reviews articles for UAVs.

The proposed cruising method has two main components: regional profitability estimation and dynamic probabilistic decision-making. The regional profitability is determined by the normalised PageRank (PR) value of each road. The roads PR values reflect the profitability of each road in terms of servicing a sequence of passengers. This paper utilises a breadth-first search (BFS) algorithm to partition the network into a number of homogeneous regions with respect to the obtained normalised PR values. The partitioning algorithm guarantees the connectivity within each region. By aggregating the normalised road PR values in a region, the overall regional profitability can be obtained. The a-taxi cruising within each region is designed to be a random search as all the roads in each region are expected to have similar level of long-term profitability.

Each a-taxi independently chooses to either stay in their region and continue random search or leave for one of the neighbouring regions. To determine the optimal inter-regional movement for vacant a-taxis, first, the method estimates the optimal number of vacant a-taxis in each region to maximise the expected number of passenger pick-ups in the entire network. We adopt a Cobb-Douglas meeting function with two inputs (i.e. the number of vacant a-taxis and the estimated number of waiting passengers) to estimate the number of pick-ups in each region. Second, the evolution of the number of empty a-taxis in each region is modelled as a Markovian process to determine the transition probability matrix that distributes vacant a-taxis among the regions to reach the obtained optimal number of vacant a-taxis in each region. Technologically, one can envision the proposed method to be implemented by an offline on-board module installed in each a-taxi.

The remainder of this article is organised as follows. Section 2 defines the problem and lists relevant assumptions. Section 3.1 presents the outline of the method. In Section 3.2, we introduce the road PR value and its application in the proposed *decentralised cooperative cruise* strategy. Section 3.3 proposes the partitioning algorithm that is developed based on BFS algorithm. The proposed cruising method for finding the optimal inter-regional movements of a-taxis is elaborated in Sections 3.4 and 3.5. The advantages of

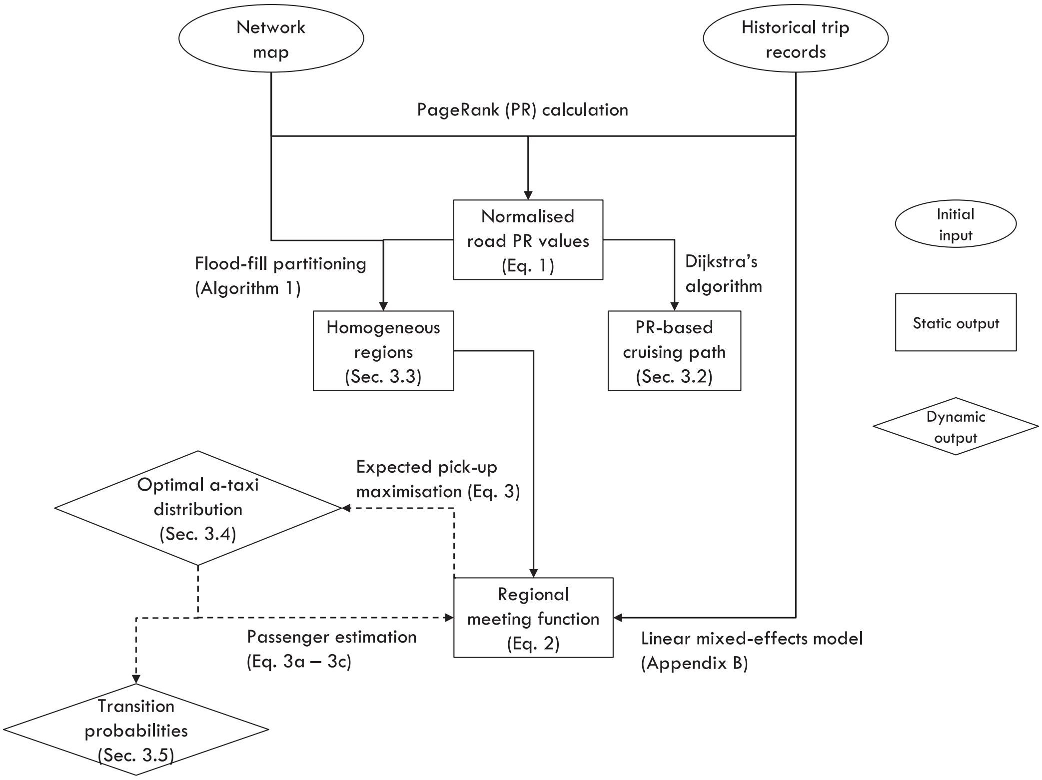


Fig. 1. Framework of the *decentralised cooperative cruising* method. Initial inputs are the network map and historical passenger origins/destinations. Static outputs are computed at the beginning of communication loss and remain fixed afterwards. Dashed arrows indicate how the dynamic outputs are updated at fixed time intervals. Vacant a-taxis use the corresponding matrix of transition probabilities to choose their destination regions (mission planning) and reach there via the PR-based shortest paths (route planning).

using the proposed method are investigated by implementing the algorithm on NYC Taxi & Limousine Commission (TLC) datasets. The datasets and simulation benchmarks are presented in Section 4. The results are discussed in Section 4.4 and the paper is concluded in Section 5.

2. Problem definition

Let us assume a fully automated ride-sourcing system that consists of autonomous vehicles (a-taxis) only. The principal operations of the system is governed by a central platform that collects real-time information of origin and destination of travel requests and the current location and status of a-taxis (occupied or vacant). The central platform also assigns vacant a-taxis to unserved waiting passengers. This paper focuses on designing an offline cruising method for such a system when there is a disruption in real-time communication. We assume the disruption results in no communication among the a-taxis, between the a-taxis and the platform, between the passengers and the platform, and between the passengers and the a-taxis. This cruising method would be decentralised because the central platform has no role in dispatching the vacant a-taxis. Instead, a-taxis would control how to search for waiting passengers on their own. We design this decentralised cruising method to be cooperative as well. That is, the primary goal is to enable the system to continue operating during communication shutdown with the main objective of increasing the total number of pick-ups by the fleet (not individual a-taxis). This objective is expected to reduce the search time of a-taxis and waiting time of passengers.

Before the communication shutdown and loss of communication with the central dispatch platform, a-taxis receive route planning decisions from the centralised dispatching system that, for example, uses bipartite matching to maximise passenger pick-ups while minimising the total cruising distance, e.g. see Zhan et al. (2016). During communication disruption, an on-board module implements the proposed decentralised coordinated method as a common optimisation algorithm. Each a-taxi individually selects a destination (mission planning) and search for waiting passengers along its cruising paths (route planning) to maximise the total expected pick-ups in the network by the whole fleet. The proposed method utilises historical passenger pick-up and drop-off times and locations to partition the network into homogeneous regions in terms of the expected profitability.

In this paper, we assume that

(i) There is no real-time communication between a-taxis and the central dispatch unit. This means an a-taxi cannot communicate with waiting passengers or other a-taxis during the communication shutdown.

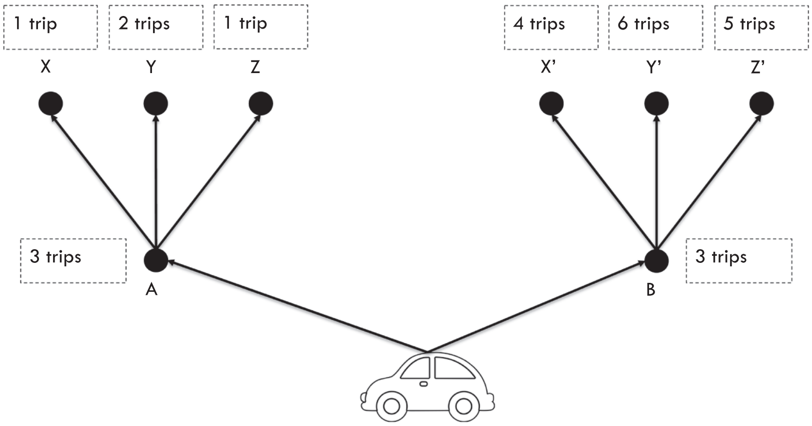


Fig. 2. There are 3 trip requests on both roads A and B, with destinations $[X, Y, Z]$ and $[X', Y', Z']$ respectively. Road B is more profitable (with a higher PR value) for the fleet operator in the long term because there is a greater probability to find the next passenger on the subsequent destination roads.

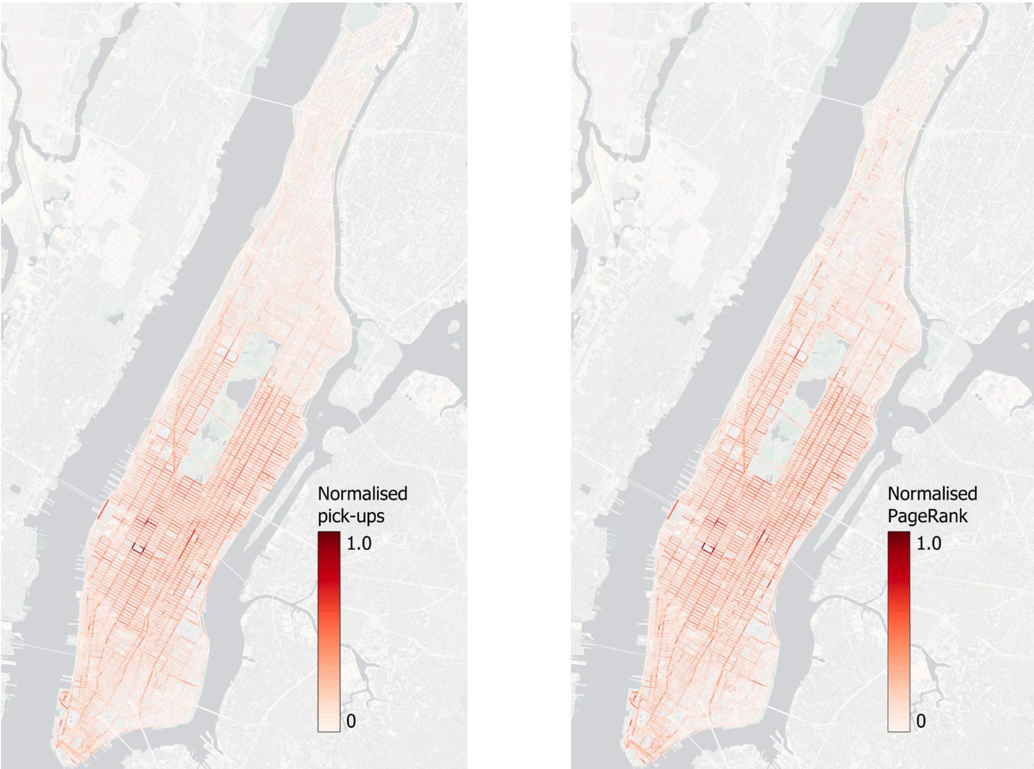
(ii) The cruising behaviour of a-taxis is decided by an on-board module that implement the proposed *decentralised cooperative* method. Their cruising decisions depend on both time and location.
(iii) A-taxis have the last known positions of waiting passengers before the communication shutdown. This serves as the initial condition of communication shutdown period.
(iv) During the communication loss, the total number of a-taxis (i.e. fleet size) does not change.
(v) Occupied a-taxis provide trip services to passengers by delivering them to their destinations via the shortest *travel time* route.
(vi) Static map information and historical origin and destination data of trips are available to the proposed method.
(vii) Vacant a-taxis are allowed to provide street-hailing services, meaning they can pick up passengers on their cruising route. They can detect and pick up a waiting passenger within a 20-s travel time radius.
(viii) The dynamic effect of congestion, and other time-varying road conditions such as speeds and traffic signals are not considered.
(ix) Stationary passengers do not wait indefinitely (Wang et al., 2020). In this paper, they choose another travel mode after a fixed patience time of 10 min. An order expiration negatively impacts the overall service quality.

3. Method

3.1. Method outline

The proposed *decentralised cooperative cruising* method functions without real-time information of passengers and a-taxis. As inputs, the method requires the network map and historical trip records (origins and destinations). These inputs are used to estimate a metric, PageRank (PR) value, that indicates each road's long-term likelihood that a-taxis can find passengers on. The PR value serves two purposes. First, it is used in Dijkstra's shortest path algorithm to scale road travel times so that a-taxis tend to travel along the more profitable routes (route planning, see Section 3.2). Second, we use it to partition the network graph into regions of roughly the same PR values so that a cruising destination can be selected at a regional level (mission planning, see Section 3.3). Next, historical training data are used to model the aggregated dynamics of pick-ups in regions represented by parsimonious meeting functions (Section 3.4). These calibrated functions are used to estimate the desired number of vacant a-taxis in each region in each time period. During each interval, vacant a-taxis cruise between regions to achieve the desired numbers. A probability matrix is computed to guide a-taxis inter-regional cruising (Section 3.5). The method considers the vacant a-taxi distribution in regions as the network state, and estimates a transition matrix to reach the desired state as the steady state of a Markov chain. Fig. 1 outlines the proposed method.

The proposed method is *cooperative*. This is realised by letting every vacant a-taxi follow the same zonal transition probability matrix to optimise the unified objective of maximising the expected passenger pick-ups by the fleet (see Eq. 3). Each a-taxi does not seek to maximise individual profits, but to improve the overall number of trip services in the network. The method outline in Fig. 1 shows a set of static outputs to characterise the network conditions that remain fixed during the communication shutdown. The dynamic outputs are calculated at discrete time intervals (e.g. every 5 min) to adjust mission planning decisions, considering potential variations in the passenger demand and location of vacant a-taxis. For practical implementations, this method can be executed by on-board computer modules installed in each a-taxi. The module can be automatically activated once communication with the central management unit is disrupted.



(a) (b)

Fig. 3. Comparison between (a) normalised road pick-up counts and (b) normalised PR values. The intensity of red colour represents the magnitude of values. Given the non-linear distribution, raw PR values and trip counts are raised to a power of 0.35 before being scaled between 0 and 1 to illustrate extreme values.

3.2. PageRank-based cruising

We propose a novel approach to compute efficient cruising paths which maximise the probability of finding a series of passengers using the PageRank (PR) algorithm (Page et al., 1999). Unlike a typical fastest/shortest path with a myopic and greedy approach, the proposed cruising method considers the probability of finding subsequent passengers after each drop-off. The long-term profitability of a road for the fleet operator (in terms of more potential pick-ups over a series of trips) is reflected by the number of passengers and whether their destinations are also profitable. The example in Fig. 2 explains why PR is a more appropriate measure of road profitability than trip pick-up counts.

The PR algorithm was initially designed to analyse browsing activities on the Internet. It is one of the fundamental algorithms to rank popular and influential websites in recommendation systems. In a directed graph, the PR centrality is a node property representing its importance, as a variant of the eigenvector centrality. The PR calculation is based on the concept of Markov chains. The state space of node PR values, which sums to 1, is iteratively updated by passing down source node values to child nodes until the state space converges. The source node value is shared equally among its out-flowing/downstream nodes. Usually a damping/teleportation factor less than 1 is used to account for users who access a node via means not represented in the graph. In a transportation network, the damping/teleportation factor can be set as 1 because teleportation is not possible.

Road PR values are estimated with historical trip records to quantify the long-term profitability of roads by considering trips as a series of pick-ups and drop-offs. Since the profitability of a road is determined by how many potential trips it leads to, the PR value of a road is shared among all roads where passengers would travel to this destination based on the ratio of trip volumes. This process is iterated until the PR values converge (e.g. with a tolerance of 10^{-6}). After convergence of the PR calculation, the final PR value of Road i is normalised by its travel time as,

$$PR(i) = \frac{1}{tt(i)} \sum_{j \in D(i)} \frac{w_{ij} PR(j)}{L(j)} \tag{1}$$

where $tt(i)$ is the fixed travel time to traverse Road i; $D(i)$ is the set of passenger trip destinations from Road i; $L(j)$ is the total number of trips from all roads in the network where passengers travel to Road j as their trip destination, and w_{ij} is the total number of trips from Road i to j during the study period. In other words, $\sum_i w_{ij} = L(j) \forall i \in$ all roads. Once converged, the PR value is normalised by $tt(i)$ to mitigate bias against shorter roads.

Results of road PR values using historical trip origin-destination pairs are compared with trip pick-up counts in Fig. 3. The distribution of both values are non-linear, with extremely high values concentrated at certain locations due to high volumes of trip records at those roads. Their values show a strong linear correlation with Pearson's coefficient of 0.971 and Spearman's coefficient of 0.979. In addition, the correlation coefficients are 0.518 and 0.794 respectively between normalised drop-off counts and PR values. As shown in the figure, PR values are able to identify areas with high passenger trip demand.

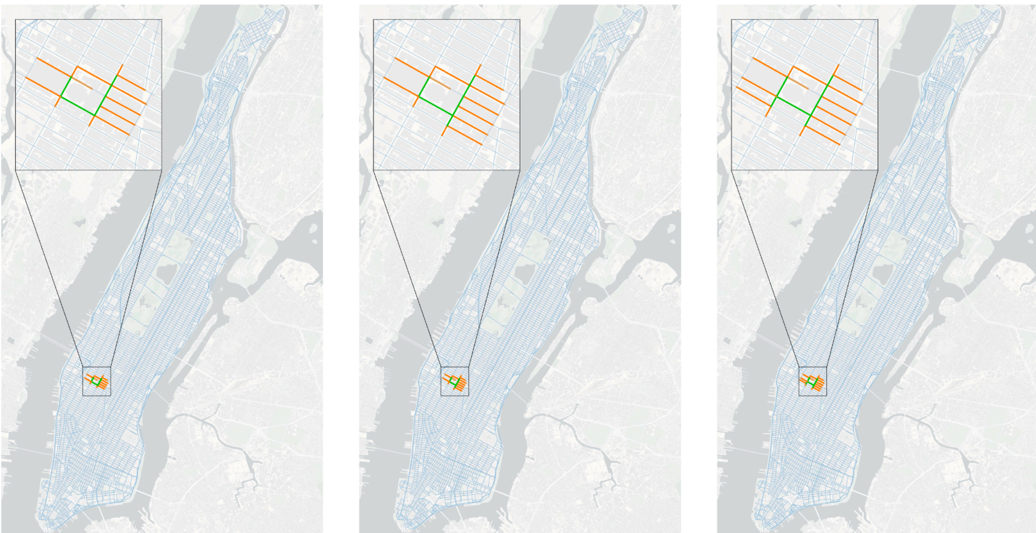
Dijkstra's shortest path algorithm has been known as an efficient method in route planning problems (Dijkstra, 1959; Zwick, 2001). To compute the most efficient cruising path, we apply Dijkstra's algorithm with the inverse of normalised PR (i.e. $1/PR(i)$) as the weight of each road. The algorithm identifies a (directed) sequence of roads between a source road and a destination road such that the total path weight is minimised. In the proposed cruising method, a-taxis plan their cruising routes by following these PR-based shortest paths between their current locations and cruising destinations.

3.3. Network partitioning

The second (offline) step of the method is to partition the city network by grouping roads with similar normalised PR values to enable the efficient selection of cruising destinations for mission planning. Hence, a-taxis could roam randomly within each region with a similar expected long-term profitability. This process of identifying relatively homogeneous regions is beneficial by: (1) aggregating historical data to analyse systematic patterns rather than noisy fluctuations; (2) providing a cruise direction rather than a specific location to avoid inefficiencies in overcrowding; and (3) enabling an elegant mathematical model to minimise the passenger order expiration ratio.

The proposed partitioning method guarantees to divide the network into non-overlapping connected regions without specifying the exact number of regions or the location of region centroids. Regions are formed in iterations such that one region is generated at a time. Regions are connected, meaning each region can be accessed via at least one neighbouring region. We use a variant of breadth-first search (BFS) to cluster neighbouring roads into a region. Unlike a basic BFS, our algorithm sequentially expands each region and the search depth is relative to the current region (a group of multiple roads) instead of a single road (i.e.the root node in a graph). This partitioning approach is similar to a flood-fill algorithm which has been applied in robotics to avoid obstacles, and in transportation problems to calculate the shortest paths with distance maps (Anvari et al., 2015). A BFS approach is relatively flexible to impose boundary constraints while ensuring node connectivity within regions.

The partitioning algorithm clusters unlabelled roads starting from the one with the highest PR value. A region expands by including a connected neighbouring road directly upstream or downstream to the current region. Neighbouring roads are arranged in a candidate queue (a data structure similar to an ordered list) such that the best road candidate is evaluated first in each iteration. The selection criterion of road addition is based on the change in regional homogeneity, represented by the standard deviation of road PR values within each region. In each iteration, the road that increases regional homogeneity the most (or increases the heterogeneity the least) among the road candidates is added to the current region. Once a road is added, its connected neighbours are also added to the



(a) (b) (c)

Fig. 4. Roads are sequentially added to a region (in green). Neighbouring candidates (in orange) are directly connected to the region. The selected candidate minimises the standard deviation in PR values. The region expands in one of the neighbouring directions, analogous to how a landscape is flooded gradually. The expansion terminates when the increase in intra-regional heterogeneity exceeds the dynamic threshold θ_R. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

candidate queue for the next iteration of expansion.

A dynamic threshold θ_R is defined depending on the number of identified regions (R), as a termination condition to identify region boundaries. The expansion of a region ends if the addition of a road increases regional heterogeneity more than the threshold. The initial cut-off threshold θ_0 is negative such that regional homogeneity is improved with the inclusion of each road. Since the distribution of PR values usually follows a power law distribution (Avrachenkov and Lebedev, 2006; Chen et al., 2014), it is expected that the improvement in regional homogeneity would decrease as more roads are clustered. Therefore, θ_R is halved for each new region to relax the termination condition to avoid splitting peripheral regions which have similarly low PR values. After the termination of regional expansion, a new region is initiated from the road with the highest PR value in the remainder of the unclustered network (not necessarily connected with existing regions). The process is repeated until all roads are clustered.

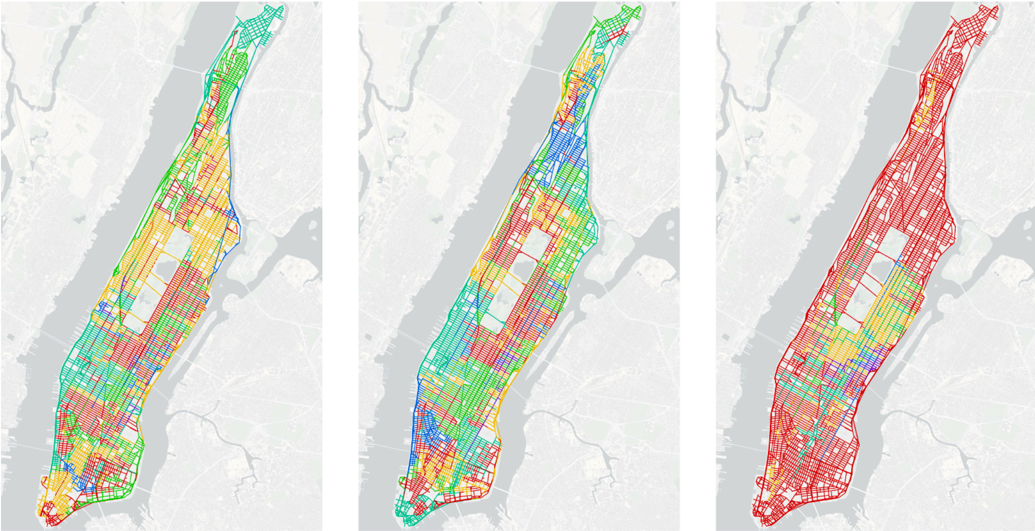
While the threshold θ_R specifies the tolerance for heterogeneity in PR values, sizes can be controlled by limiting the minimum and maximum number of roads in each region. The lower limit $k_{\min}$ determines the minimum number of roads to attain an acceptable level of accuracy in regional PR means and variances. The upper limit $k_{\max}$ prevents over-sized regions. If a region has less than $k_{\min}$ roads, it will be merged into a neighbouring region with the closest mean PR value, which may result in sizes larger than $k_{\max}$.

Fig. 4 is an example to show how neighbouring roads are clustered sequentially. Pseudo-code of the proposed partitioning method is stated in Algorithm 1.

Algorithm 1. Proposed Network Partitioning Algorithm

```
Initialisation:
    SortedRoads ← Sort(Roads)                          // Descending road PR
    R ← 0                                              // Initial region
    Q ← PriorityQueue                                  // Candidate queue
    θ_0 ← NegativeNumber                               // Initial threshold

Main Partition(SortedRoad):
    for r_0 ∈ SortedRoads do
        if Region(r_0) is NULL then
            Region(r_0) ← R                            // First road in a new region
            k ← 1                                      // Number of roads in the current region
            μ_k ← PR(r_0)                              // Initial mean PR
            σ_k^2 ← 0                                  // Initially zero variance

            while Q ≠ ∅ do
                r_i ← Pop(Q)                           // The next best candidate
                flag ← IncludeRoad(k, μ_k, σ_k^2, PR(r_i), θ_R)
                if Region(r_i) is NULL and flag is TRUE then
                    Region(r_i) ← R
                    k, μ_k, σ_k^2 ← UpdateStats(k, μ_k, σ_k^2, PR(r_i))
                    for r_j ∈ FindNeighbours(r_i) do
                        Q ← Q ∪ j                      // Consider neighbours

                if k > k_max then
                    Q ← ∅                              // Terminate expansion

            if k < k_min then
                Merge(R)                               // Merge with neighbouring region

            R ← R + 1                                  // Clear queue, new region
            θ_R ← 2^(−R) θ_0                           // New termination threshold

Function UpdateStats(k, μ_k, σ_k^2, X):
    X_{k+1} ← X
    μ_{k+1} ← (k μ_k + X_{k+1}) / (k+1)                // Welford's method, Appendix A
    σ_{k+1}^2 ← (k σ_k^2 + (X_{k+1} − μ_k)(X_{k+1} − μ_{k+1})) / (k+1)
    return k + 1, μ_{k+1}, σ_{k+1}^2

Function IncludeRoad(k, μ_k, σ_k^2, X, θ_R):
    k + 1, μ_{k+1}, σ_{k+1}^2 ← UpdateStats(k, μ_k, σ_k^2, X)
    return σ_{k+1} − σ_k ≤ θ_R                         // Selection criterion
```

Some examples of partitioning outcomes are presented in Fig. 5 where neighbouring regions are plotted in different colours. Figure (b) which contains minimum 50 and maximum 500 roads in each region is used in numerical experiments, given its suitable region sizes and level of homogeneity in road PR values.



(a) [10, 500] 146 regions (b) [50, 500] 39 regions (c) [50, ∞] 25 regions

Fig. 5. Different region sizes with [$k_{\min}$,$k_{\max}$] roads. Larger limits generally result in more regions. Also shown in Appendix C, increasing region sizes lead to more intra-region heterogeneity. However, smaller regions would compromise the accuracy of meeting function estimations (Section 3.4). Thus, the 39-region partitioning outcome is selected.

3.4. Optimal spatial distribution of vacant A-taxis

The primary objectives of the *decentralised cooperative cruising* method is to pick up as many passengers as possible without real-time information or communication. The theoretical approach of the cruising strategy is based on bilateral meeting/matching functions. Schroeter (1983) introduced such a function to analyse matching behaviours between traditional taxis and passengers for a radio dispatch system and the airport taxi stand operation. It is also widely applied in the labour market to model search frictions between the demand side and supply side (in this context, waiting passengers and vacant a-taxis) (Mortensen and Pissarides, 1994; Petrongolo and Pissarides, 2001). A meeting function gives reasonably accurate estimates of the passenger-vehicle dynamics for regions with homogeneous traffic states (Yang et al., 2010; Yang and Yang, 2011; Ramezani and Nourinejad, 2018; Nourinejad and Ramezani, 2020). This paper uses a Cobb-Douglas meeting function shown in eq. (2).Function inputs are updated at discrete time intervals (τ) to capture effects of applying the proposed method over time.

$$M_I(\tau) = \alpha_I R_I(\tau)^{\gamma_1} V_I(\tau)^{\gamma_2} \tag{2}$$

where $M_I(\tau)$,$R_I(\tau)$, and $V_I(\tau)$ are the expected number of pick-ups, the average number of waiting passengers, and the average number of vacant a-taxis respectively in Region I during time interval τ. Static parameters α_I,γ_1, and γ_2 describe pick-up patterns in each region, where α_I is a unique constant for Region I. The elasticity coefficients, γ_1 and γ_2, are fixed and common for all regions. These static parameters are obtained by fitting a linear mixed model, explained in detail in Appendix B.

By utilising the concept of meeting function, the total expected pick-ups in the network (with N regions as an outcome of Section 3.3) can be maximised as in eq. (3):

$$V_I^*(\tau) = \underset{V_I(\tau)}{\operatorname{argmax}} \sum_{I=1}^{N} \widehat{M_I}(\tau) \tag{3}$$

$$\text{s.t.}\, R_I(\tau) = \widehat{R_I}(\tau-1) + \Delta R_I^{\mathrm{h}}(\tau) - \widehat{M_I}(\tau-1) \quad \forall I \in \{1,\cdots,N\} \tag{3a}$$

$$\widehat{R_I}(\tau) \geqslant 0 \qquad \forall I \in \left\{1,\cdots,N\right\} \tag{3b}$$

$$\widehat{R_I}(0) \text{ is known} \qquad \forall I \in \left\{1,\cdots,N\right\} \tag{3c}$$

$$\sum_{I=1}^{N} V_I(\tau) = \beta F. \tag{3d}$$

The exact number of waiting passengers during simulation cannot be directly monitored by the system without real-time communication. However, it can be estimated by a mass conservation relation as in constraint (3a). The expected number of waiting passengers in Region I at each interval τ is raised by the historical number of new passenger trip requests $\Delta R_I^{h}(\tau)$, and reduced by the number of expected pick-ups during the previous interval $\widehat{M_I}(\tau-1)$. Although the underlying assumption of achieving all expected pick-ups may not be realised and would underestimate the resultant $\widehat{R_I}(\tau)$, it is partially offset by overlooking the passenger cancellation which overestimates passenger counts. In addition, the estimated number of waiting passengers in Region I must satisfy the non-negative constraint in (3b). Nevertheless, a mass conservation model still has the shortcoming of accumulating errors over time.

The initial conditions are specified in (3c) and (3d). We interpret the initial waiting passenger distribution $R_I(0)$ as the last known position of trip requests before communication shutdown. We also assume the vacant a-taxi fleet size as a ratio (β) of the total fleet size (F) to estimate network conditions during the period of communication loss. Ratio β is assumed to be time-invariant. Based on simulation results shown in Fig. 6a, the number of vacant a-taxis varies smoothly and tends to be stable. Thus, we assume a fixed $\beta = 1/3$ for simplification. Numerical experiments have indicated insensitivity of this ratio and yield similar results even if the actual number of vacant a-taxis is applied.

Eq. (3) is computed for every interval τ during the period of communication loss. The resultant $V_I^*(\tau)$ values represent the desired number of vacant a-taxis in each region such that the total number of pick-ups in the network is maximised. *Decentralised cooperative cruise* gives mission planning recommendations in the form of a probability matrix such that each vacant a-taxi can stochastically move between regions to achieve the desired numbers V_I^*.

3.5. Steady-state transition probabilities for decentralised cruising

The *regional cooperative cruise* of a-taxis can be considered as a discrete-time Markov chain with a state vector representing the number of vacant a-taxis in each region and a transition matrix $\boldsymbol{P}(\tau)$ representing their movements between regions during each update interval (τ). In a Markov chain, an initial state vector converges to a steady-state distribution $\boldsymbol{\pi}^\tau$ after applying $\boldsymbol{P}(\tau)$ repeatedly such that $\boldsymbol{\pi}^\tau \boldsymbol{P}(\tau) = \boldsymbol{\pi}^\tau$.

The vector $\boldsymbol{\pi}^\tau$ represents the steady-state outcome of a-taxi cruising. It can be related with the optimal number of vacant a-taxis in each region (V_I^* in eq. (3)) to find an appropriate $\boldsymbol{P}(\tau)$ such that the realisation of $\boldsymbol{P}(\tau)$ guides the vacant a-taxis to their desired regional destinations. There are N elements in $\boldsymbol{\pi}^\tau$, representing N regions, so the probability matrix $\boldsymbol{P}(\tau)$ has a dimension of $N \times N$.

The transition matrix $\boldsymbol{P}(\tau)$ can be solved by eq. (4) which minimises the 2-norm (squared root of the sum of squared errors) in convergence. In theory, the resultant $\boldsymbol{P}(\tau)$ converges any initial vector to the steady state $\boldsymbol{\pi}^\tau$ after several iterations. The $\boldsymbol{P}(\tau)$ is used as a probability matrix for cruising a-taxis to decide to either stay within their current region or leave for a neighbouring region.

$$\underset{\boldsymbol{P}(\tau)}{\operatorname{argmin}} \quad \| \boldsymbol{\pi}^\tau \boldsymbol{P}(\tau) - \boldsymbol{\pi}^\tau \| \tag{4}$$

$$\text{s.t.}\, \boldsymbol{\pi}^\tau = \left[V_1^*(\tau), V_2^*(\tau), \cdots, V_N^*(\tau)\right] \tag{4a}$$

$$\boldsymbol{P}(\tau) \cdot \boldsymbol{I}_{N\times 1} = \boldsymbol{I}_{N\times 1} \tag{4b}$$

$$0 \leqslant \boldsymbol{P}(\tau)_{I,J} \leqslant \mathbf{A}_{I,J} \qquad \forall I, J \in \{1, \cdots, N\}. \tag{4c}$$

Condition (4a) defines the steady-state distribution $\boldsymbol{\pi}^\tau$. Matrix $\boldsymbol{P}(\tau)$ is a right stochastic matrix, meaning its row entries sum to 1 (4b). Element $\boldsymbol{P}(\tau)_{I,J}$ denotes the probability of an a-taxi in Region I choosing Region J as its cruising destination. We design the cruising method to only recommend connected neighbouring regions to vacant a-taxis. To this end, an adjacency matrix $\boldsymbol{A}$ is defined by the network partitioning outcome to limit transition probabilities (4c). Element $\boldsymbol{A}_{I,J}$ is 1 if Regions I and J are neighbours to each other, and 0 otherwise.

Nonetheless, the effectiveness of transition matrix $\boldsymbol{P}(\tau)$ is limited. Specifically, not all vacant a-taxis can arrive at their target road within the interval τ, meaning the state space may not converge to the optimal state $\boldsymbol{\pi}^\tau$ before a new transition matrix is computed. The choice of τ is complex depending on the dynamic network traffic conditions, size of regions, a-taxi travel speeds, and discretisation of continuous a-taxi movements. In this paper, we use $\tau = 5$ min and a-taxis do not update their destinations during cruising even though there might be a new $\boldsymbol{P}(\tau)$ matrix for optimal mission planning.

After an a-taxi selects a destination region based on the probability matrix, a specific road is chosen randomly inside the region if the a-taxi chooses to stay within the current region because the intra-regional road PR values are similar. If an a-taxi chooses to leave for one of the neighbouring regions, the method scales each of their selection probabilities by the travel time to a random road in these regions. Depending on the current location of the cruising a-taxi, this means a closer neighbouring region is more likely to be selected as the destination than a distant one such that a-taxis may arrive at their target regions sooner. Algorithm 2 explains the secondary destination choice process which determines the destination road to compute the cruising path. Eventually, vacant a-taxis follow the PR-based shortest paths (see Section 3.2) to cruise and search for waiting passengers en route.

Algorithm 2. Destination Road Choice

```
Main RoadChoice(vehicle, P(τ)):
    I ← Region(vehicle)                              // Current region
    J ← Random(Set(regions), P(τ)[I, :])             // Destination region

    if J == I then
        j ← Random(Set(roads ∈ I))                   // Stay in current region
    else
        AR ← FindAdjacent(I)                         // Set of adjacent regions
        for K ∈ AR do
            k ← Random(Set(roads ∈ K))               // A random road
            tt(i, k) ← TravelTimeBetween(i, k)
            P(τ)[I, K] ← P(τ)[I, K] / tt(i, k)       // Normalise probabilities
        end

        J ← Random(AR, P(τ)[I, :] \ {I})             // Re-select region
        j ← Random(Set(roads ∈ J))                   // Leave for destination
    end
end
```

4. Numerical experiments

The effectiveness of the proposed method is tested in an event-based simulation. The simulation arranges a-taxis' and passengers' activities in an event queue by their trigger times. A-taxis and passengers interact in the Manhattan transportation network which is represented by a directed graph. New York City yellow taxi trip records in 2016 (Taxi and Limousined Limousine Commission, 2016) are used, both as a historical dataset for our method and a replication of morning peak passenger trip requests during the communication disruption (Hamedmoghadam et al., 2019). The historical dataset consists of 20 weekdays in May 2016 and the test dataset contains 5 randomly selected weekdays in June 2016. These trip records provide historical trip details such as passenger trip origin/destination and their pick-up time. There are a total of $1,059,948$ trip records in the historical dataset and $260,191$ entries in the test dataset after data cleaning and filtering.

The morning peak hours from $7-10$ am (3 h) are considered, with the first hour being a warm-up period during which the *central matching* system manages a-taxi movements. Between $8-10$ am, it is assumed that there is no communication between the central dispatching unit and the a-taxi fleet. In addition, an a-taxi cannot receive real-time information of other a-taxis or passenger orders. Each a-taxi follows the cruising recommendation from the offline on-board module that implements the proposed decentralised cooperative cruising method such that the fleet productivity is maximised during the communication shutdown.

4.1. Simulator

This paper uses COMSET simulator which is programmed to specifically evaluate the efficiency of taxi search strategies without real-time communication (van Barlingen et al., 2019; ACM, 2019). In COMSET, the Manhattan road network is constructed as a directed graph using OpenStreetMap data. Intersections are constructed as nodes and roads as edges. The graph is processed such that intersections in close proximity are grouped as one node. All dead ends are also removed to make the graph strongly connected which prevents a-taxis from being trapped. As a result, the graph contains 4360 intersections and 9542 roads connecting these intersections. The COMSET simulator simplifies the effects of congestion and road conditions by scaling all vehicle speeds as some fixed proportions of the road speed limits. Speeds are calibrated such that the simulated shortest trip travel time is similar to the historical trip duration.

In the simulator, all events are associated with a location and a trigger time. Passengers remain stationary with two possible states — waiting or expired. Passengers wait for vacant a-taxis before they cancel their order upon reaching a fixed 10-min patience time. Once picked up, passengers are transported to their destinations via the shortest *travel time* path. The corresponding a-taxi becomes occupied and stops searching. Without accessing real-time passenger locations, a pick-up is successful if an a-taxi is within 20-s travel time from a passenger (about $50-60$ m detection range on most roads). It considers the travel direction so that a-taxis must be able to reach the passenger location within 20 s. Vacant a-taxis cruise to their planned destination via the sequence of roads advised by the proposed cruising method. A new destination is planned if no passengers are found along the cruising route.

4.2. Experiment setup and benchmark strategies

At the beginning of the simulation, a-taxis with a fixed fleet size are randomly placed on the network graph. A warm start is applied for one hour which uses a central bipartite matching algorithm at 10-s intervals. To be precise, the minimum-weight maximum bipartite matching considers the shortest travel times between all possible waiting passengers and vacant a-taxis as weights. The solution gives one-to-one pairs of passenger-vehicle assignments until one of the set is exhausted, while minimising the total travel time for all pick-ups. The computation of a centralised dispatch requires accurate vehicle and passenger locations.

Besides the warm start, we also include *central matching* to show near-optimal values for comparison, even though it has the unfair advantage of real-time information and communication. The other benchmark strategy is *random destination* which points vacant a-taxis to a random intersection in the network. A-taxis then follow the shortest path based on either travel time (*tt*) or PageRank (PR) to reach the cruising destination. The *random destination* method with PR-based shortest paths has the same route planning logic as the proposed method but without mission planning considerations.

These benchmarks represent different levels of information availability. *Random destination* with *tt*-based shortest paths does not use any real-time or historical data. The PR-based shortest cruising paths utilise historical trip records to calculate road PR values. The proposed *decentralised cooperative cruise* also uses historical data for road PR values and regional meeting functions. *Central matching* needs access to real-time information.

4.3. Simulation outputs

Simulation results include 4 performance measures to evaluate the *decentralised cooperative cruise* method. The mean a-taxi cruise time denotes how long an a-taxi remains vacant on average during the 2 h. The mean passenger wait time reflects how long a passenger has waited on average until pick-up or order expiration. The order expiration percentage measures how many passengers are not picked up by any a-taxi within 10 min. The total number of pick-ups is the number of successful passenger pick-ups by the entire a-taxi fleet in 2 h.

4.4. Results

This section evaluates simulation results of the proposed cruising method and compares them with the benchmark strategies introduced in Section 4.2. *Central matching* is also included to show near-optimal values. However, it should be noted that the centralised strategy takes full advantage of real-time locations of a-taxis and passengers while the other strategies do not use such information. Although the proposed strategy cannot achieve the same level of passenger pick-up rate as *central matching*, it still shows significant improvements over *random destination* benchmarks.

To evaluate the effectiveness of following a PR-based shortest path for cruising, the *random destination* strategy is applied with both the travel time (*tt*) and PR-based shortest paths. Based on the mean results of 5 test days in Table 1, the PR-based shortest cruising path yields better performances than *tt*-based paths by prioritising roads with a higher likelihood to find passengers upon each drop-off. With 6000 a-taxis, an average of 1343 more trips can be served in 2 h, and an average of 35 s waiting time reduction per trip can be achieved. Service performance improvements of applying the proposed *decentralised cooperative cruise* against the benchmark

Table 1
Mean simulation outputs over 5 test days of the proposed and benchmark cruising strategies. Fleet sizes ranging from 3000 to 7000 are tested. Percentage improvements in the parentheses are compared against benchmark results using *random destination* with *tt*-based shortest paths.

Fleet size	Mean A-taxi cruise time (s)	Mean passenger wait time (s)	Order Expiration	Total pick-ups
Random destination with *tt*-based shortest paths				
3000	2450.2	406.6	53.4%	15715.2
4000	2901.2	358.6	45.2%	18705.4
5000	3321.8	319.8	38.8%	20973.8
6000	3796.2	288.0	34.4%	22643.4
7000	4380.4	263.2	31.2%	23798.0
Random destination with PR-based shortest paths				
3000	1837.8 (−25.0%)	362.2 (−10.9%)	46.6% (−6.8%)	18288.4 (+16.4%)
4000	2524.4 (−13.0%)	313.8 (−12.5%)	39.2% (−6.0%)	20968.0 (+12.1%)
5000	3111.2 (−6.3%)	278.6 (−12.9%)	34.4% (−4.4%)	22772.4 (+8.6%)
6000	3752.8 (−1.1%)	253.0 (−12.2%)	31.0% (−3.4%)	23986.4 (+5.9%)
7000	4391.2 (−0.2%)	236.2 (−10.3%)	28.8% (−2.4%)	24786.2 (+4.2%)
Decentralised cooperative cruise				
3000	2133.0 (−12.9%)	362.2 (−10.9%)	46.0% (−7.4%)	18502.0 (+17.7%)
4000	2681.2 (−7.6%)	297.2 (−17.1%)	35.8% (−9.4%)	22174.8 (+18.5%)
5000	3231.2 (−2.7%)	249.0 (−22.1%)	29.4% (−9.4%)	24583.2 (+17.2%)
6000	3750.2 (−1.2%)	214.4 (−25.6%)	25.4% (−9.0%)	26140.2 (+15.4%)
7000	4223.4 (−3.6%)	189.4 (−28.0%)	22.4% (−8.8%)	27204.2 (+14.3%)
Central matching				
3000	915.4 (−62.6%)	371.4 (−8.7%)	38.8% (−14.6%)	21849.2 (+39.0%)
4000	1257.6 (−56.7%)	314.4 (−12.3%)	23.4% (−21.8%)	27830.2 (+48.8%)
5000	1656.2 (−50.1%)	273.6 (−14.4%)	10.2% (−28.6%)	32762.8 (+56.2%)
6000	2287.4 (−39.7%)	194.4 (−32.5%)	2.0% (−32.4%)	35464.8 (+56.6%)
7000	3167.6 (−27.7%)	150.0 (−43.0%)	0.6% (−30.6%)	35804.8 (+50.5%)

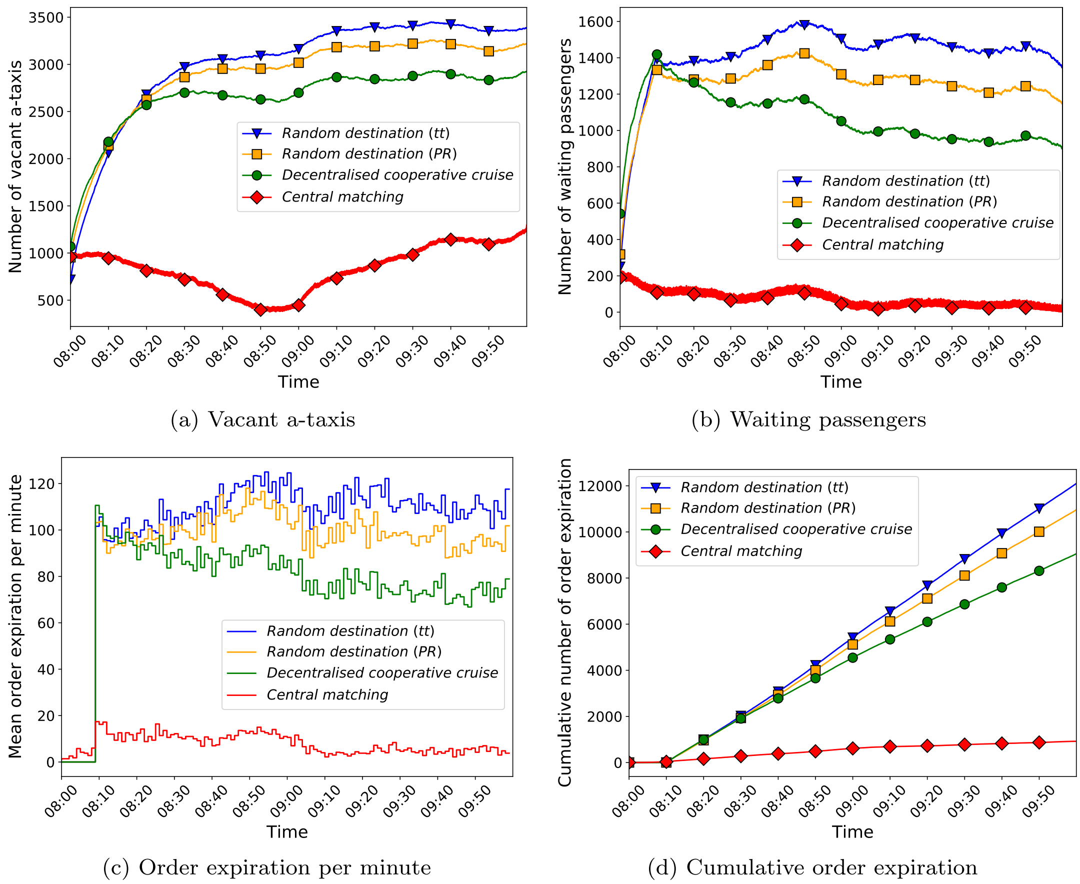


Fig. 6. Time evolution of 5-day average performance measures for 6000 a-taxis. As expected, decentralised strategies (without real-time information) result in more vacant a-taxis and waiting passengers than *central matching* (with real-time information). *Decentralised cooperative cruise* shows improvements over *random destination* benchmarks with approximately 240–360 fewer vacant a-taxis, about 190–350 fewer waiting passengers on average. There are also 16–25 fewer order expiration per minute (or 1900–3000 fewer total expiration in 2 h).

strategy are shown in the parentheses in Table 1. All four measures demonstrates improvements over different fleet sizes.

As the total fleet size increases from 3000 to 7000, trip service performances improve in general. Larger a-taxi fleets serve more trips at the expense of longer average a-taxi cruising time. The average passenger waiting time is nearly halved (47.7% reduction) and order expiration can be reduced by 23.6% as the fleet size increases from 3000 to 7000. With 3000 a-taxis, the proposed method has similar performances as the *PR-based random destination* strategy. However, as the fleet size increases, the proposed method becomes significantly more efficient, especially with respect to the number of total passenger pick-ups (up to 14.3%) and mean passenger waiting time (up to 28%).

Central matching benchmark results are included to indicate the near-optimal performance. The effectiveness of the proposed method can be analysed by evaluating the improvement in performance measures from a simple *random destination* strategy towards the near-optimal *central matching* scenario that uses real-time information. As the fleet size increases, a centralised system outperforms decentralised strategies to a greater extent. For instance, the proposed method is 45.4% as efficient as *central matching* in passenger pick-up numbers with 3000 a-taxis, but only 28.4% as efficient with 7000 a-taxis. Regarding the mean passenger waiting time, the decentralised methods may even result in shorter waiting times with a small fleet size because a-taxis tend to prioritise passengers along profitable routes. With 7000 a-taxis, the proposed method can reduce the mean passenger waiting time from 263.2 seconds to 189.4 seconds, which is still 65.2% as efficient as *central matching* which has a mean waiting time of 150 s. Note that the calculation of mean waiting time includes cancelled trip requests as 600 s each, leading to even lower mean waiting times for the proposes method if only served passengers are considered. This observation suggests that service frequency on roads with higher PR values increases at the expense of lower coverage in the less popular areas, when we compare *decentralised cooperative cruising* against *central matching*.

Simulation results for 6000 a-taxis are analysed to compare the efficiency of decentralised methods with the theoretical scenario without communication loss (i.e. *central matching*). The 5-day mean simulation outputs are plotted as time-series in Fig. 6 from $8:00-10:00$ after the 1-h warm start which implements *central matching*. Fig. 6a and b show the time evolution of total numbers of vacant a-taxis and waiting passengers in the network respectively from $8:00-10:00$. The proposed strategy performs better than *random destination* benchmarks with fewer vacant a-taxis and waiting passengers for most of the simulation duration. The number of

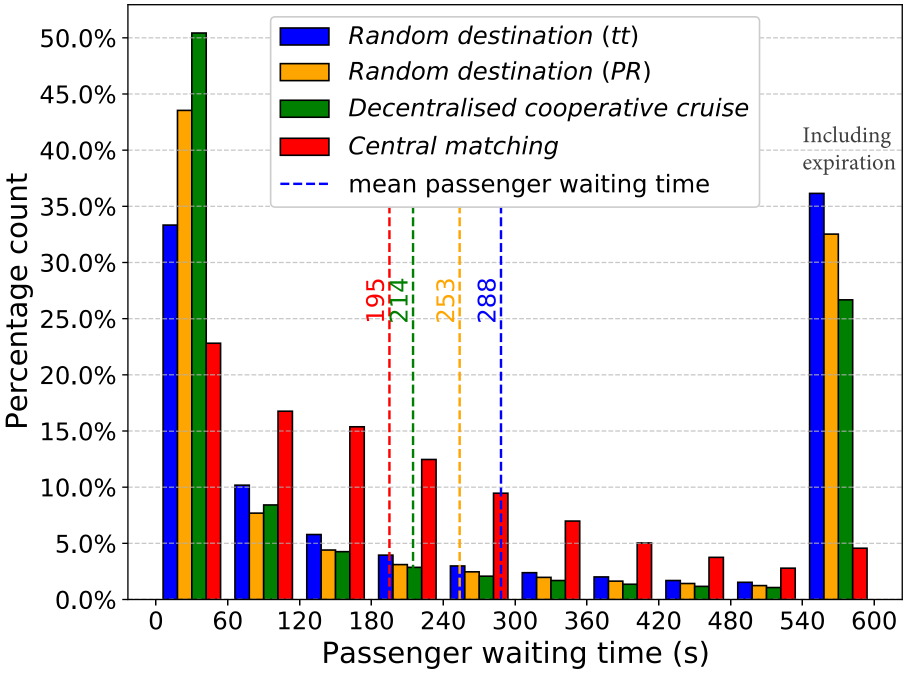


Fig. 7. Distribution of passenger waiting times over 5 test days with 6000 a-taxis. Passengers wait longer with *random destination* than *decentralised cooperative cruise* in general. Since a trip request expires in 10 min, the last bin contains a significant proportion of trip counts. Mean passenger waiting times are plotted as dashed vertical lines.

order expiration per minute and the cumulative number are plotted in Fig. 6c and d respectively. The rate of increase shows how the proposed *decentralised cooperative cruise* reduces order expiration over time.

Fig. 7 is a histogram of the percentage distribution of passenger waiting times on all 5 test days. An order expiration is equivalent to a fixed 600-s waiting time, leading to high percentage counts in the last bin. *Decentralised cooperative cruise* outperforms both *random destination* benchmarks with shorter passenger waiting times in general, evident from the right-skewed distribution. Most passengers (more than 50%) are picked up within 1 min, suggesting how the proposed cruising method can successfully identify new trip requests at the "hot spots". *Central matching* distribution is more evenly distributed, with the highest percentage counts in all bins except the two ends, because central assignments are not affected by a-taxis' detection range. It allows a-taxis to find passengers without being in a close proximity, thus showing different pick-up patterns.

5. Summary and future research

This paper has proposed a *decentralised cooperative cruising* strategy for fully automated taxi (a-taxi) fleets when the communication between the a-taxis and passengers via a central dispatch unit is lost. Unlike centralised fleet management systems, the proposed contingency strategy is *decentralised* such that a-taxis make cruising decisions individually based on recommendations of the offline method. The proposed decentralised cruising method is designed as a *multi-agent cooperative* method with a unified goal to maximise the expected total number of passenger pick-ups in the network by the fleet, rather than to maximise individual gains. The method utilises historical trip records to mathematically derive the desirable numbers of vacant a-taxis in different regions of the network and recommends the best travel paths between cruising origins and destinations.

Historical trip origin-destination data are used to calculate road PageRank centralities and partition the network to make efficient cruising decisions (route and mission planning) in the absence of real-time information. The proposed method maximises the total expected passenger pick-ups in the network which can be estimated from the macroscopic pick-up patterns in each region. A Cobb-Douglas regional meeting function is derived by fitting simulation outputs using historical datasets with a linear mixed-effects model. Optimal numbers of vacant a-taxis in each region can be estimated based on estimated number of waiting passengers. By considering a-taxi movements between regions as a Markov chain, a transition probability matrix is calculated such that numbers of vacant a-taxis converge to the optimal values. The proposed method mathematically determines the recommended cruising destinations and travel routes for a-taxis when there is no communication channel among the central dispatch unit, a-taxis, and passengers.

The effectiveness of the proposed method is compared against benchmark strategies in an event-based simulator, using real-world

Manhattan trip records on 5 different test days. The proposed method has shown significant improvements over the *random destination* benchmarks, especially in the passenger service aspects for a large fleet size. While the performance of PR-based random destination diminishes as the fleet size increases, the regional cruising component in the proposed method is able to retain these benefits and further reduces the mean passenger wait time by up to 28%.

There are some potential future research directions. The proposed method can be improved by incorporating demand forecasting (Ke et al., 2019). We can investigate the use of mixed fleet compositions with different cruising purposes. A part of the fleet can be exploitative, while the other part can be exploratory and cruise to serve more remote trips for service coverage. Electric vehicles can also be considered as an alternative vehicle type since a-taxis are likely to use renewable energy sources (Jing et al., 2017; Jing et al., 2018; Bongiovanni et al., 2019). Different levels of communication loss can be considered. For example, passenger locations might be unavailable while communication with the central dispatch unit still functions such that a-taxis can still coordinate effectively. Designing robust centralised dispatching methods resilient to partial communication disruptions also needs further research.

CRediT authorship contribution statement

Linji Chen: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing – original draft, Writing – review & editing. **Amir Hosein Valadkhani:** Methodology, Investigation, Writing – original draft. **Mohsen Ramezani:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was partially funded by the Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA) DE210100602.

Appendix A. Welford's method for computing variance

Welford's method (Welford, 1962) elaborated in eq. (A.1) is an online algorithm for calculating mean and variance of *n* samples. In our case, the standard deviation in normalised PR indicates intra-regional homogeneity. We apply Welford's method to track changes in regional homogeneity which determines whether a road among all candidates is qualified to be added into a region.

$$\bar{x}_n = \frac{(n-1)\bar{x}_{n-1} + x_n}{n} \tag{A.1}$$

$$\sigma_n^2 = \frac{(n-1)\sigma_{n-1}^2 + \left(x_n - \bar{x}_{n-1}\right)\left(x_n - \bar{x}_n\right)}{n}.$$

Appendix B. Meeting function estimation

The meeting function parameters are calculated by fitting a linear mixed-effects model where coefficients γ_1 and γ_2 are fixed effects applicable to all regions and α_I is a random effect which varies among different regions. The α_I values represent intrinsic properties of each region such as their size and internal connectivity.

Observations of variables $M_I(\tau), R_I(\tau)$, and $V_I(\tau)$ are collected and aggregated at 5-min intervals from the *random destination* benchmark simulation (see Section 4.2) for 20 days in the historical dataset. Observations are grouped by their pick-up regions. A linear form can be obtained by taking the natural logarithm of eq. (2),

$$\ln M_I(\tau) = \ln \alpha_I + \gamma_1 \ln R_I(\tau) + \gamma_2 \ln V_I(\tau).$$

The two fixed-effect coefficients γ_1,γ_2, and region-based random effect coefficient α_I are fitted using an expectation maximisation algorithm and solved by maximum likelihood estimation, with normally distributed errors (Seabold and Perktold, 2010; Lindstrom and Bates, 1988). The resultant coefficients adhere to the non-negativity constraint.

With 39 regions in the network, fixed coefficients γ_1 and γ_2 are 0.354 and 0.148 respectively. The value of α_I ranges from 0.60 to 10.48, with a mean of 3.54. Fig. B.8 demonstrates the goodness-of-fit of the regional meeting function estimates per 5-min interval. The calibrated meeting function can accurately estimate the number of pick-ups for the given $R_I(\tau)$ and $V_I(\tau)$, with $R^2 = 0.959$, $\text{MAE} = 4.36$, and $\text{RMSE} = 7.43$.

The value of α_I is often modelled as a dependent variable of a characteristic variable (Φ_I) which represents some spatial property of the region. For example, in Yang et al. (2010):

$$\alpha_I = u(\Phi_I)^w \tag{B.1}$$

$$\ln \alpha_I = \ln u + w \ln \Phi_I \tag{B.2}$$

where u and w are positive constants. In this case, the value of α_I shows a strong linear correlation with the total regional PR, with a Pearson's correlation coefficient of 0.849. Fitting eq. (B.2) by an ordinary least squres (OLS) regression yields $\ln u = 3.14 (u = 23.09)$ and $w = 0.358$ with $R^2 = 0.721$.

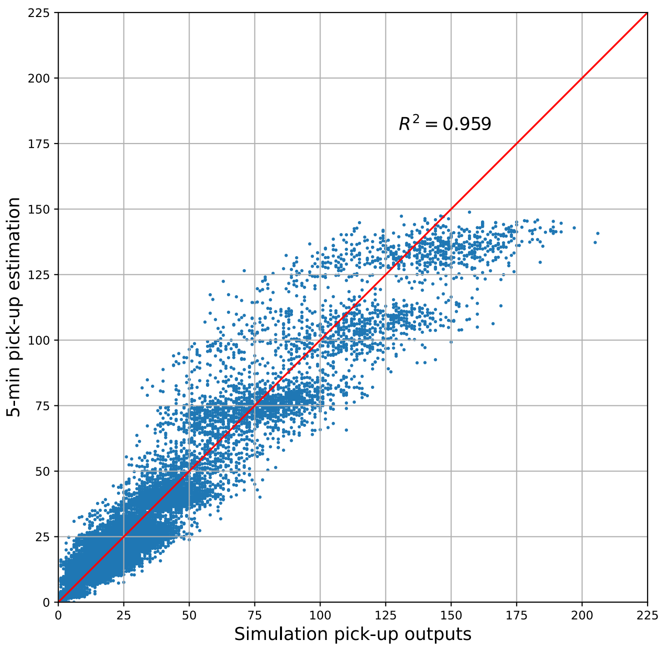


Fig. B.8. 5-min pick-ups estimated by the calibrated meeting functions vs. *random destination* simulation results, using the 20-day historical dataset.

Appendix C. Clustering analysis

To quantify regional homogeneity and compare it between different partitioning outputs, we use a metric known as the Silhouette Index (SI), or mean road silhouette score which is a common measure of clustering quality. It takes a value between -1 (poorly clustered) and 1 (perfectly clustered). SI of Region I is calculated as

$$\mathrm{SI}_I = \left(\sum_{i=1}^{k_I} \frac{b(i) - a(i)}{\max(a(i), b(i))} \right) \bigg/ k_I \tag{C.1}$$

$$a(i) = \frac{|\mathrm{PR}(i) - \mathrm{PR}(j)|}{k_I - 1} \tag{C.2}$$

$$b(i) = \frac{|\mathrm{PR}(i) - \mathrm{PR}(j')|}{k_J} \tag{C.3}$$

where $a(i)$ is the intra-regional mean absolute difference in road PR values for road $j \neq i$ in the same region; $b(i)$ is the inter-regional mean absolute difference in road PR values for road j' in the closest neighbour; k_I is the number of roads in Region I.

Some examples of SI results are listed in Table C.2. Regional homogeneity generally improves with more, smaller regions. However, it also means fewer samples are available for the calibration of regional meeting functions, compromising the accuracy of macroscopic pick-up dynamics. Thus, $k_{\min}$ is preferably larger than $10-20$ to be able to represent the local attractiveness with its PR value. The value of $k_{\max}$ is dependent on the total number of roads in the network. It should be set such that inter-regional cruising does not require too much time for the method to be effective. Our numerical experiment uses the 39-region output with $k_{\min} = 50$ and $k_{\max} = 500$.

Table C.2
Descriptive statistics of SI results for some network partitioning outcomes. Different groups of $[k_{\min}, k_{\max}]$ result in different number of regions. The minimum (*Min*) and maximum (*Max*) SI values reflect its range. The standard deviation (*SD*) indicates the dispersion or scatter in regional SI values.

$[k_{\min}, k_{\max}]$	*No. regions*	*Mean*	*Median*	*Min*	*Max*	*SD*
[10, 500]	146	0.3761	0.4073	−0.6400	0.9994	0.3997
[10, ∞]	104	0.2948	0.2860	−0.5721	0.9721	0.3291
[50, 500]	39	0.2022	0.2146	−0.6753	0.9708	0.4418
[50, ∞]	25	0.1743	0.0764	−0.5653	0.8774	0.3716
[70, 500]	33	0.1209	0.1534	−0.5970	0.9682	0.4119
[70, ∞]	23	0.1794	0.2970	−0.4976	0.8265	0.3467

References

ACM (Ed.), 2019. 8th ACM SIGSPATIAL GIS Cup, ACM SIGSPATIAL.
Anvari, B., Bell, M.G., Sivakumar, A., Ochieng, W.Y., 2015. Modelling shared space users via rule-based social force model. Transp. Res. Part C: Emerg. Technol. 51, 83–103.
Avrachenkov, K., Lebedev, D., 2006. PageRank of Scale Free Growing Networks. Research Report RR-5858. INRIA.
Ayala, D., Wolfson, O., Dasgupta, B., Lin, J., Xu, B., 2018. Spatio-Temporal Matching for Urban Transportation Applications. ACM Trans. Spatial Algorithms Syst. 3, 11:1–11:39.
van Barlingen, R., Ferreira, J., Klimovic, T., Schols, J., de Vries, W., Xu, B., 2019. Comset-giscup.
Berbeglia, G., Cordeau, J.F., Laporte, G., 2010. Dynamic pickup and delivery problems. Eur. J. Oper. Res. 202, 8–15.
Bongiovanni, C., Kaspi, M., Geroliminis, N., 2019. The electric autonomous dial-a-ride problem. Transp. Res. Part B: Methodol. 122, 436–456.
Buchin, K., Kostitsyna, I., Custers, B., Struijs, M., 2019. A Sampling-based Strategy for Distributing Taxis in a Road Network for Occupancy Maximization (GIS Cup). In: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Association for Computing Machinery, New York, NY, USA, pp. 616–619.
Chen, N., Litvak, N., Olvera-Cravioto, M., 2014. Pagerank in scale-free random graphs. arXiv:1408.3610.
Colorni, A., Righini, G., 2001. Modeling and Optimizing Dynamic Dial-a-Ride Problems. Int. Trans. Oper. Res. 8, 155–166.
Cordeau, J.F., 2006. A Branch-and-Cut Algorithm for the Dial-a-Ride Problem. Oper. Res. 54, 573–586.
Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271.
Duan, L., Wei, Y., Zhang, J., Xia, Y., 2020. Centralized and decentralized autonomous dispatching strategy for dynamic autonomous taxi operation in hybrid request mode. Transp. Res. Part C: Emerg. Technol. 111, 397–420.
Eaton, C.M., Chong, E.K.P., Maciejewski, A.A., 2016. Multiple-Scenario Unmanned Aerial System Control: A Systems Engineering Approach and Review of Existing Control Methods. Aerospace 3, 1.
Guo, Q., Wolfson, O., 2018. Probabilistic spatio-temporal resource search. GeoInformatica 22, 75–103.
Haddad, J., Mirkin, B., 2020. Resilient perimeter control of macroscopic fundamental diagram networks under cyberattacks. Transp. Res. Part B: Methodol. 132, 44–59.
Hadi, B., Khosravi, A., Sarhadi, P., 2021. A Review of the Path Planning and Formation Control for Multiple Autonomous Underwater Vehicles. J. Intell. Robot. Syst. 101, 67.
Hamedmoghadam, H., Ramezani, M., Saberi, M., 2019. Revealing latent characteristics of mobility networks with coarse-graining. Sci. Rep. 9, 7545.
Hörl, S., Ruch, C., Becker, F., Frazzoli, E., Axhausen, K.W., 2019. Fleet operational policies for automated mobility: A simulation assessment for Zurich. Transp. Res. Part C: Emerg. Technol. 102, 20–31.
Hu, Q., Ming, L., Tong, C., Zheng, B., 2019. An Effective Partitioning Approach for Competitive Spatial-Temporal Searching (GIS Cup). In: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Association for Computing Machinery, New York, NY, USA, pp. 620–623.
Hyland, M., Mahmassani, H.S., 2018. Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests. Transp. Res. Part C: Emerg. Technol. 92, 278–297.
Hyland, M.F., Mahmassani, H.S., 2017. Taxonomy of Shared Autonomous Vehicle Fleet Management Problems to Inform Future Transportation Mobility. Transp. Res. Rec. 2653, 26–34.
Jing, W., An, K., Ramezani, M., Kim, I., 2017. Location Design of Electric Vehicle Charging Facilities: A Path-Distance Constrained Stochastic User Equilibrium Approach. J. Adv. Transp. 2017, e4252946.
Jing, W., Ramezani, M., An, K., Kim, I., 2018. Congestion patterns of electric vehicles with limited battery capacity. PLoS One 13, e0194354.
Ke, J., Yang, H., Zheng, H., Chen, X., Jia, Y., Gong, P., Ye, J., 2019. Hexagon-Based Convolutional Neural Network for Supply-Demand Forecasting of Ride-Sourcing Services. IEEE Trans. Intell. Transp. Syst. 20, 4160–4173.
Liang, X., Correia, G.H.d.A., An, K., van Arem, B., 2020. Automated taxis' dial-a-ride problem with ride-sharing considering congestion-based dynamic travel times. Transp. Res. Part C: Emerg. Technol. 112, 260–281.
Lindstrom, M.J., Bates, D.M., 1988. Newton-raphson and em algorithms for linear mixed-effects models for repeated-measures data. J. Am. Stat. Assoc. 83, 1014–1022.
Ma, J., Li, X., Zhou, F., Hao, W., 2017. Designing optimal autonomous vehicle sharing and reservation systems: A linear programming approach. Transp. Res. Part C: Emerg. Technol. 84, 124–141.
Maciejewski, M., Bischoff, J., Nagel, K., 2016. An Assignment-Based Approach to Efficient Real-Time City-Scale Taxi Dispatching. IEEE Intell. Syst. 31, 68–77.
Mercader, P., Haddad, J., 2021. Resilient multivariable perimeter control of urban road networks under cyberattacks. Control Eng. Pract. 109, 104718.
Mortensen, D.T., Pissarides, C.A., 1994. Job Creation and Job Destruction in the Theory of Unemployment. Rev. Econ. Stud. 61, 397–415.
Mourad, A., Puchinger, J., Chu, C., 2019. A survey of models and algorithms for optimizing shared mobility. Transp. Res. Part B: Methodol. 123, 323–346.
Naccache, S., Côté, J.F., Coelho, L.C., 2018. The multi-pickup and delivery problem with time windows. Eur. J. Oper. Res. 269, 353–362.
Narayanan, S., Chaniotakis, E., Antoniou, C., 2020. Shared autonomous vehicle services: A comprehensive review. Transp. Res. Part C: Emerg. Technol. 111, 255–293.
Nourinejad, M., Ramezani, M., 2020. Ride-Sourcing modeling and pricing in non-equilibrium two-sided markets. Transp. Res. Part B: Methodol. 132, 340–357.
Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66. Stanford InfoLab.

Petrongolo, B., Pissarides, C.A., 2001. Looking into the Black Box: A Survey of the Matching Function. J. Econ. Lit. 39, 390–431.
Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L., 2013. A review of dynamic vehicle routing problems. Eur. J. Oper. Res. 225, 1–11.
Psaraftis, H.N., Wen, M., Kontovas, C.A., 2016. Dynamic vehicle routing problems: Three decades and counting. Networks 67, 3–31.
Ramezani, M., Nourinejad, M., 2018. Dynamic modeling and control of taxi services in large-scale urban networks: A macroscopic approach. Transp. Res. Part C: Emerg. Technol. 94, 203–219.
Schroeter, J.R., 1983. A Model of Taxi Service under Fare Structure and Fleet Size Regulation. Bell J. Econ. 14, 81–96.
Seabold, S., Perktold, J., 2010. statsmodels: Econometric and statistical modeling with python. In: 9th Python in Science Conference, pp. 92–96.
Taxi and Limousine Commission, 2016. Yellow taxi trip records.
Vosooghi, R., Puchinger, J., Jankovic, M., Vouillon, A., 2019. Shared autonomous vehicle simulation and service design. Transp. Res. Part C: Emerg. Technol. 107, 15–33.
Wang, X., Liu, W., Yang, H., Wang, D., Ye, J., 2020. Customer behavioural modelling of order cancellation in coupled ride-sourcing and taxi markets. Transp. Res. Part B: Methodol. 132, 358–378.
Welford, B.P., 1962. Note on a Method for Calculating Corrected Sums of Squares and Products. Technometrics 4, 419–420.
Wong, R.C.P., Szeto, W.Y., Wong, S.C., 2014. A cell-based logit-opportunity taxi customer-search model. Transp. Res. Part C: Emerg. Technol. 48, 84–96.
Wong, R.C.P., Szeto, W.Y., Wong, S.C., 2015. A two-stage approach to modeling vacant taxi movements. Transp. Res. Part C: Emerg. Technol. 59, 147–163.
Yang, H., Leung, C.W.Y., Wong, S.C., Bell, M.G.H., 2010. Equilibria of bilateral taxi-customer searching and meeting on networks. Transp. Res. Part B: Methodol. 44, 1067–1083.
Yang, H., Yang, T., 2011. Equilibrium properties of taxi markets with search frictions. Transp. Res. Part B: Methodol. 45, 696–713.
Zedadra, O., Seridi, H., Jouandeau, N., Fortino, G., 2016. A Cooperative Switching Algorithm for Multi-Agent Foraging. Eng. Appl. Artif. Intell. 50, 302–319.
Zhan, X., Qian, X., Ukkusuri, S.V., 2016. A Graph-Based Approach to Measuring the Efficiency of an Urban Taxi Service System. IEEE Trans. Intell. Transp. Syst. 17, 2479–2489.
Zwick, U., 2001. Exact and approximate distances in graphs — a survey. In: auf der Heide, F.M. (Ed.), Algorithms — ESA 2001. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 33–48.