A Predictive Chance Constraint Rebalancing Approach to Mobility-on-Demand Services

Communications in Transportation Research 3 (2023) 100097

Contents lists available at ScienceDirect

Communications in Transportation Research

journal homepage: www.journals.elsevier.com/communications-in-transportation-research

Full Length Article

A predictive chance constraint rebalancing approach to mobility-on-demand services

Sten Elling Tingstad Jacobsen [a,b,*], Anders Lindman [b], Balázs Kulcsár [a]

[a] Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, 41258, Sweden
[b] Volvo Cars AB, Gothenburg, 41878, Sweden

ARTICLE INFO

Keywords:
Mobility-on-Demand
Travel demand uncertainty
Fleet optimization
Gaussian process regression
Chance constraint optimization
Energy efficiency

ABSTRACT

This paper considers the problem of supply-demand imbalances in Mobility-on-Demand (MoD) services. These imbalances occur due to uneven stochastic travel demand and can be mitigated by proactively rebalancing empty vehicles to areas where the demand is high. To achieve this, we propose a method that takes into account uncertainties of predicted travel demand while minimizing pick-up time and rebalance mileage for autonomous MoD ride-hailing. More precisely, first travel demand is predicted using Gaussian Process Regression (GPR) which provides uncertainty bounds on the prediction. We then formulate a stochastic model predictive control (MPC) for the autonomous ride-hailing service and integrate the demand predictions with uncertainty bounds. In order to guarantee constraint satisfaction in the optimization under estimated stochastic demand prediction, we employ a probabilistic constraining method with user-defined confidence interval, using Chance Constrained MPC (CCMPC). The benefits of the proposed method are twofold. First, travel demand uncertainty prediction from data can naturally be embedded into the MoD optimization framework, allowing us to keep the imbalance at each station below a certain threshold with a user-defined probability. Second, CCMPC can be relaxed into a Mixed-Integer-Linear-Program (MILP) and the MILP can be solved as a corresponding Linear-Program, which always admits an integral solution. Our transportation simulations show that by tuning the confidence bound on the chance constraint, close to optimal oracle performance can be achieved, with a median customer wait time reduction of 4% compared to using only the mean prediction of the GPR.

1. Introduction

The fast growing urbanization in the world puts major challenges on urban transportation (Raposo, 2019). In Europe, the urbanization is expected to grow from 74% in 2018 to 84% in 2050 (United Nations Publications, 2019). Traditionally, urban transportation is improved by infrastructure investments in road expansions and public transportation. However, with recent development of new technologies within automation, connectivity, electrification and shared services, there is a potential for new transport solutions to satisfy the increasing demand. Transportation modes that have gained huge interest and market share are mobility-on-demand (MoD) services, e.g., mobility-as-a-service providers and car-rental pools (Zardini et al., 2022). These types of services are more flexible than public transportation and can in fact complement it (Salazar et al., 2018). Moreover, the combination of MoD and autonomous vehicles (AVs), Autonomous Mobility-on-Demand (AMoD), has been at the center of research for over a decade (Zardini et al., 2022). AMoD is predicted to become one of the major means of transportation in cities (Litman, 2020). However, a major concern with MoD and AMoD services is that they have a tendency to become imbalanced, i.e., have a mismatch between vehicles and requests in different parts of the service area. The imbalance is due to unevenly distributed stochastic spatial and temporal travel patterns, which give rise to a poor quality of service (George, 2012). To handle this issue, the service providers may match requests with vehicles centrally and proactively send vehicles to areas with predicted high imbalances. A vital part of the AMoD system is to predict the stochastic travel pattern in order to match demand (Zardini et al., 2022). Predictions come with uncertainty in travel patterns and can have a large influence on the AMoD performance. In this paper, we propose a method for efficient AMoD fleet control with probabilistic guarantees on the imbalance. The method is applied and tested for an autonomous ride-hailing service but can be applied to any MoD or AMoD system.

There are different methodologies for modeling and predicting travel demand patterns. Previous work can be divided into two categories:

* Corresponding author. Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, 41258, Sweden.
E-mail addresses: stingsta@volvocars.com (S.E. Tingstad Jacobsen), kulcsar@chalmers.se (B. Kulcsár).

https://doi.org/10.1016/j.commtr.2023.100097
Received 18 January 2023; Received in revised form 3 May 2023; Accepted 5 May 2023
Available online 19 July 2023

Nomenclature

$1-\varepsilon$	Confidence level
D	Destination of trip
$F_{\lambda_{ij}(t)}(z)$	Cumutlativ distribution function of $\lambda_{ij}(t)$
N	Number of partitioned station
O	Origin of trip
T	Number of time intervals for the time horizon
Δt_{GPR}	Update interval for Gaussian Process Regression
Δt_{MPC}	Update interval for Model Predictive Control Algorithm
$\mathbb{P}_{ij}(t)$	Probability distribution of $\lambda_{ij}(t)$
$\mathcal{T}$	Discrete set of time intervals
μ	Mean prediction of Gaussian Process Regression
σ	Standard deviation of mean prediction of Gaussian Process Regression
c_φ	Cost of each vehicle
k	Upper bound on the imbalance
t_0	Current time step
$x^{\text{r}}_{ij}(t)$	A decision variable for the number of vehicles to rebalance from station i to station j at time interval t
$\boldsymbol{\Delta t}$	Discrete time interval length
$\Theta \subset \mathbb{R}^2$	Operating area for vehicles
$\boldsymbol{\kappa_{ij}(t)}$	Travel time, in discrete time intervals, to drive from station i to station j at time t
$\boldsymbol{\lambda_{ij}(t)}$	Number of customers that wants to travel from station i to station j at time t
$\boldsymbol{\varphi_i(t)}$	Initial number of idle vehicles in each station
$\boldsymbol{c_\lambda}$	Cost of leaving out one customer that wants to go from station i to station j at time t
$c^{\text{r}}_{ij}(t)$	Cost of rebalancing one vehicle from station i to station j at time t
$\boldsymbol{s_{ij}(t)}$	A decision variable for the imbalance, i.e., describes how many customers to not pick-up at time t that wants to go from station i to station j
$x^{\text{c}}_{ij}(t)$	Number of vehicles to drive customers from station i to station j at time interval t

parametric and non-parametric travel demand prediction (in view of the structure of the demand probability distribution function). In case of parametric solutions, the underlying form of the demand is assumed to be known (Rasmussen and Williams, 2005). These models span from simple linear regression to fitting of distributions to data (Rasmussen and Williams, 2005). Poisson distributions are commonly used as parametric models (Braverman et al., 2019; Zardini et al., 2022) as well as Gaussian distributions (Mao et al., 2020). However, assuming such distributions are oversimplifications of demand patterns since the travel demand often follows an unknown spatio-temporal probability distribution. The complexity of the spatio-temporal travel demand patterns has lead to an increased focus on non-parametric approaches. These models do not make strong assumptions on the form of the travel demand (probability distribution), which make them more flexible to learn arbitrary patterns. Different non-parametric approaches have been used for demand prediction, such as Long Short-Term Memory neural (LSTM) networks (Iglesias et al., 2018; Tsao et al., 2018). A drawback of neural network methods is that they require large data sets to be reliable. Their strong dependence on hyperparameters and initial conditions may hinder efficient fitting (Pereira et al., 2022). Studies on using LSTM for predicting mobility movements have been shown to be efficient with up to 80% accuracy in predicting mobility patterns (Zhao et al., 2016). On the other hand, the uncertainty in the prediction should also be predicted and accounted for in the optimization, calling for explicit uncertainty parameterization. One way of modeling the uncertainty is to assume that the travel demand belongs to an uncertainty set. The uncertainty set can be constructed from data using hypothesis testing (Miao et al., 2017). To make the uncertainty set representative enough, large data sets are required and the correlation between different time intervals is neglected. Hence, one plausible solution to explicit uncertainty estimation for forecasting spatio-temporal data with uncertainties for both small and large data sets is Gaussian Process Regression (GPR) (Rasmussen and Williams, 2005). For small datasets, GPR is often superior to other prediction methods and it provides a confidence on the prediction, which is beneficial for robustness (Rasmussen and Williams, 2005). However, to the best of our knowledge, the efficiency of GPR is yet to be reported in combination with optimization of AMoD systems.

There has been extensive research on different modeling and control algorithms for MoD and AMoD systems. Earlier works focused on reactive control methods, from the Hungarian method (Kuhn, 1955) to control methods based on queueing-based models (Pavone et al., 2012; Ruch et al., 2020). More recently, the use of future demand together with model predictive control (MPC) has been proven to be highly effective (Iglesias et al., 2018; Lacombe et al., 2021; Miao et al., 2017; Tsao et al., 2018; Zhang et al., 2016). Zhang et al. (2016) proposed to use MPC to solve the dispatching and rebalancing problem. However, they did not consider any demand forecasting method and, in addition, the computational complexity increased with the number of vehicles. These issues were addressed in Iglesias et al. (2018), although the uncertainty in the demand prediction was neglected. In Tsao et al. (2018), a nominal prediction method is used via sample average approximation. A robust, minmax uncertainty handling is presented in Miao et al. (2017). Guo et al. (2021) proposed a robust optimization model that combines matching of demand and vehicles with rebalancing but demand prediction was not considered. Another robust optimization method considered for vehicle rebalancing is distributionally robust optimization model with enhanced linear decision rule (He et al., 2020). As indicated above, there is no unique way of introducing travel demand into AMoD algorithms. Methods that take the uncertainty of the demand into account have been proven to be efficient but complex. Therefore there is a need for more transparent, scalable, computationally efficient, and accurate methods.

A promising approach is to incorporate GPR and MPC and couple them stochastically via the uncertainty bound provided by GPR. One appealing solution is to solve the stochastic and uncertain MPC problem under probabilistic constraints, i.e., chance constraint (Charnes and Cooper, 1959). This methodology has proven successful for control of autonomous racing and autonomous underwater vehicles (Hewing et al., 2019). Furthermore, chance constraint optimization (CCO) has been used in many resource allocation problems (Grosso et al., 2014; Ono and Williams, 2008; Varga et al., 2020), which are similar to the control of AMoD systems. The benefit of CCO is that, via probabilistic constraining, we can adjust the solution implicitly. This is beneficial since the two objectives of controlling a AMoD fleet, service and cost, are contradicting. Generally, the better the service, the higher the cost and vice versa. The relaxation of the CCO is typically very complex unless the probability distribution is assumed to be known. The combination of GPR and chance constrained MPC has the potential to provide a powerful methodological environment.

The main contribution of this paper is to combine data driven demand prediction with model based predictive AMoD resulting in a chance constraint optimization. This is done by first, formulating a Chance

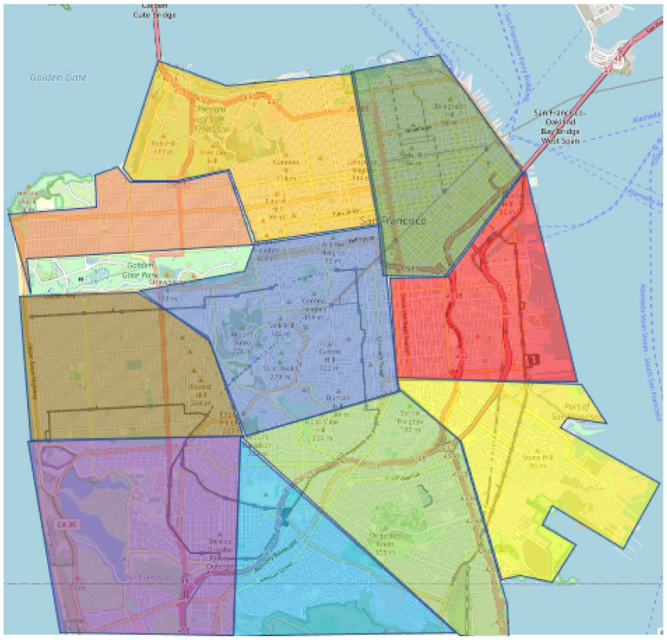

Fig. 1. A visual representation of how the city of San Francisco could be partitioned into different stations (https://planet.osm.org). Each colored area represents one station.

Constrained MPC (CCMPC), which is a probabilistic approach to solving stochastic optimization problems. Second, we propose a GPR for predicting travel demand time-series. The prediction given by the GPR contains both a mean prediction and an uncertainty bound. Hence, GPR naturally fits into the CCMPC framework. By means of separability and by knowing the form of the estimated PDF, probabilistic constraints can be reformulated into a deterministic optimization. To the best of our knowledge, no other study has focused on the combination of GPR and CCMPC in the AMoD setting. Previous studies in this area have either focused on only the demand prediction part or on AMoD control methods, which assume simplified demand modeling or demand modeling that require large datasets. Third, the proposed optimization is benchmarked in the high fidelity transport simulator AMoDeus (Ruch et al., 2018). This is important in order to get an accurate measure of different metrics, such as pick-up time and vehicle mileages. Many studies consider less accurate in-house transport simulators based on simplified road and traffic models (Iglesias et al., 2018; Miao et al., 2017; Tsao et al., 2018).

The outline of this study is the following. First, we present the model of the AMoD systems in Section 2. In Section 3, we first formulate the MPC and then the Chance Constrained MPC, which is later relaxed to a mixed integer linear program (MILP) using GPR and the separable model. The transport simulation methodology and results are presented and discussed in Section 4. Finally, Section 5 concludes the paper with a short discussion and future work.

2. AMoD modeling

In this section, we first describe the stochastic discrete-time linear model of the AMoD system, building on prior work presented in Iglesias et al. (2018) and Tsao et al. (2018). Our approach assumes that travel demand follows some unknown spatio-temporal probability distribution, and uses Gaussian Process Regression (GPR) to predict demand, which is subsequently incorporated into a chance constraint model for the system.

Specifically, the AMoD model captures the movement of both vehicles and customers, while ensuring that the number of vehicles and customers remains conserved. Our model also accounts for the mismatch between vehicles and customers in different parts of the city, and offers a way to optimize fleet operations under uncertain demand conditions. For a detailed description of all notions introduced throughout the paper, we refer to glossary in Appendix.

The bounded operation area is a two-dimensional map denoted by $\Theta \subset \mathbb{R}^2$. We assume that the map is partitioned into N regions, referred to as stations (Fig. 1), and these partitions are predetermined. The partitioned city is modeled as a complete graph, where the nodes represent the stations, and the edges represent the distances and travel time between the stations. The travel time and distances between the stations are considered exogenous, and do not vary with traffic. The AMoD model operates in discrete time with a fixed sampling interval of Δt. At each time step, new customers arrive at the stations, awaiting pick-up by vehicles. We assume that the origin O and destination D of each trip are constrained to lie within the operation area, i.e., $O, D \subseteq \Theta$. We denote the current time step as t_0.

2.1. States and decision variables

There are several non-negative integer states and decision variables in the system. The first state of the system is the number of customers that want to travel from station i to station j at time t and is denoted by $\lambda_{ij}(t)$. This is a stochastic variable and each $\lambda_{ij}(t)$ is assumed to have an unknown time-varying probability distribution, $\mathbb{P}_{ij}(t)$. Furthermore, it is assumed that the values of λ_{ij} at different time t are correlated, while spatial values at different stations are not correlated. The initial state of the customer demand, $\lambda_{ij}(t_0)$, is the number of outstanding customer that wants to go from station i to station j. The second state is the average travel time between stations, denoted by $\kappa_{ij}(t)$. Since the model is discrete, $\kappa_{ij}(t)$ is also discretized into time intervals. The third and final state is the initial position of idle vehicles in each station and is denoted, $\varphi_i(t)$. Vehicles that are traveling are assumed to be idle when they reach their destination.

There are three decision variables in this model. The main decision variable is the movement of the vehicles when they are empty, and this will be referred to as rebalancing. The number of vehicles to rebalance from station i to station j at time t is denoted $x^{\mathrm{r}}_{ij}(t)$. The second decision variable is the number of vehicles that serve travel demand traveling from station i to station j and is denoted $x^{\mathrm{c}}_{ij}(t)$. The decision variable $s_{ij}(t)$ describes the imbalance in station i for travel demand with destination j. The imbalance is the difference between customers and vehicles in each station.

2.2. Vehicle conservation

The principle of vehicle conservation stipulates that vehicles cannot disappear or appear in the model during a specific time period, denoted by $\mathcal{T} = \{1, 2, \ldots, T\}$, where T is the total number of time intervals. This conservation principle is enforced by a vehicle conservation constraint, which requires that the difference between the number of vehicles entering and departing the station is equal to the initial number of vehicles present in the station at the beginning of each time interval. Mathematically, this constraint can be expressed as follows:

$$\sum_{j \in N} x^{\mathrm{c}}_{ij}(t) + x^{\mathrm{r}}_{ij}(t) - x^{\mathrm{c}}_{ji}(t - \kappa_{ji}) - x^{\mathrm{r}}_{ji}(t - \kappa_{ji}) = \varphi_i(t), \qquad \forall i \in N, t \in \mathcal{T} \tag{1}$$

where $x^{\mathrm{c}}_{ij}(t)$ and $x^{\mathrm{r}}_{ij}(t)$ denote the number of vehicles that leave station i to station j for driving customers and rebalancing, respectively, at time t; $x^{\mathrm{c}}_{ji}(t - \kappa_{ji})$ and $x^{\mathrm{r}}_{ji}(t - \kappa_{ji})$ denote the number of vehicles that enter station i from station j for driving customers and rebalancing, respectively, at time $t - \kappa_{ji}$; and $\varphi_i(t)$ denotes the initial number of vehicles present in station i at time t. The constraint must hold for all stations i and time intervals t.

2.3. Imbalance

The imbalance is the difference between number of travel request and vehicles in each station. Ideally, the imbalance is zero at all time, i.e., there is a perfect match between the number of travel demand and vehicles:

$$\lambda_{ij}(t) - x^c_{ij}(t) = 0, \forall i,j \in N, t \in \mathcal{T} \tag{2}$$

However, if there are more customers than vehicles, constraint (2) is violated. Therefore, this constraint needs to be relaxed to ensure feasibility, which is done by introducing the slack variable $s_{ij}(t)$:

$$s_{ij}(t) = \lambda_{ij}(t) - x^c_{ij}(t) \quad \forall i,j \in N, t \in \mathcal{T} \tag{3}$$

If $s_{ij}(t) > 0$, i.e., there are more request than available vehicles, the remaining request should be served at a later time step. Hence, we carry on $s_{ij}(t)$ to the next time step if $t > t_0$,

$$\begin{aligned} s_{ij}(t+1) &= s_{ij}(t) + \lambda_{ij}(t+1) - x^c_{ij}(t+1) \\ &\forall i,j \in N, t \in \{t_0+1, \dots, T+t_0\} \end{aligned} \tag{4a}$$

$$s_{ij}(t_0) = \lambda_{ij}(t_0) - x^c_{ij}(t_0) \qquad \forall i,j \in N \tag{4b}$$

The state $x^c_{ij}(t)$ cannot be larger than the number of travel request since it represents only vehicles that drives customer, i.e., the imbalance should be greater or equal to zero,

$$s_{ij}(t) \geq 0, \quad \forall i,j \in N, t \in \mathcal{T} \tag{5}$$

The combination of constraint (5) and that $s_{ij}(t)$ is a integer decision variable, gives that

$$s_{ij}(t) \in \mathbb{N}, \quad \forall i,j \in N, t \in \mathcal{T}$$

3. Model predictive control of AMoD with probabilistic guarantees

In this section, a model predictive controller (MPC) for AMoD is proposed based on the model described in the previous section. The MPC is first written as a stochastic mixed integer linear program (sMILP). The optimization problem is a MILP since several variables are restricted to be integer values (Walukiewicz, 2013). The sMILP is then reformulated as a chance constraint optimization. Finally, the Chance Constrained MPC is relaxed into a deterministic MILP using assumptions on the probability distribution of the stochastic variable.

3.1. Model predictive control of AMoD

The above mentioned approach to AMoD is a discrete time optimization problem that is solved for T time steps into the future, also called time horizon. The solution of the sMILP is a sequence of optimal decision variables for the time horizon. Only the optimal decisions belonging to the first time step in the horizon is used whilst the rest is discarded. In the next time step, the states in the sMILP is updated and the sMILP is solved again, i.e., receding horizon control. The following optimization problem is formulated:

$$\underset{x^r_{ij}, s_{ij}}{\text{minimize}} \quad \sum_{t=t_0}^{T+t_0} \sum_{i,j=1}^{N} c^r_{ij}(t) x^r_{ij}(t) + c_\lambda(t) s_{ij}(t) \tag{6a}$$

$$\text{subject to} \quad s_{ij}(t_0) = \lambda_{ij}(t_0) - x^c_{ij}(t_0) \qquad \forall i,j \in N \tag{6b}$$

$$s_{ij}(t+1) = s_{ij}(t) + \lambda_{ij}(t+1) - x^c_{ij}(t+1), \quad \forall i,j \in N, t \in \mathcal{T} \tag{6c}$$

$$\sum_{j=1}^{N} x^c_{ij}(t) + x^r_{ij}(t) - x^c_{ji}(t-\kappa_{ji}) - x^r_{ji}(t-\kappa_{ji}) = \varphi_i(t) \qquad \forall i \in N, t \in \mathcal{T} \tag{6d}$$

$$x^r_{ij}, s_{ij}, x^c_{ij}(t) \in \mathbb{N} \qquad \forall i,j \in N, t \in \mathcal{T} \tag{6e}$$

Optimization Problem (6) is a stochastic optimal control problem. The constraints in the MPC come from the model described in the previous section, Eq. (2). The first two constraints are the imbalance in the system for $t = t_0$, Eq. (6b), and for $t > t_0$, Eq. (6c). The third constraint is the network flow conservation, Eq. (6d). It prohibits new vehicles from appearing or disappearing from the system. The last constrain enforces the decision variables to belong to the natural numbers set, which are all non-negative integers.

The objective is to offer a good service to the customers and to ensure that this is done efficiently, Eq. (6a). Hence we want to minimize the mismatch, $s_{ij}(t)$, between customers and vehicles. This mismatch can be minimized by rebalancing vehicles in-between stations. The rebalancing comes with a cost for the operator and this cost should also be minimized. There is a trade-off between the imbalance and the rebalancing cost. This trade-off can be tuned by choosing appropriate values for the imbalance cost, $c_\lambda(t)$, and the rebalancing cost, $c_{ij}(t)$. The imbalance cost should reflect the cost of making customers wait, which could be varying over time. The rebalancing cost is a combination of distance and travel time. To be able to find a good weighting of the costs, Pareto analysis is used.

3.2. Chance constrained MPC

As mentioned in the previous section, the demand prediction, $\lambda_{ij}(t)$, is assumed to follow a probability density distribution, $\mathbb{P}_{ij}(t)$. Hence, we can reformulate the imbalance constraint, Eq. (6c), to have a probability distribution fulfilled with some confidence $1-\varepsilon$, where $\varepsilon \in [0,1]$, see Eq. (7c). One of the benefits of this formulation is that we can decide the confidence based on what risk we want to take. Therefore, the following sMILP problem is proposed:

$$\underset{x^r_{ij}, s_{ij}}{\text{minimize}} \quad \sum_{t=t_0}^{T+t_0} \sum_{i,j=1}^{N} c^r_{ij}(t) x^r_{ij}(t) + c_\lambda(t) s_{ij}(t) \tag{7a}$$

$$\text{subject to} \quad \text{Eqs. (6b), (6d), and (6e)} \tag{7b}$$

$$\mathbb{P}_{ij}\left(s_{ij}(t+1) = s_{ij}(t) + \lambda_{ij}(t+1) - x^c_{ij}(t+1) \leq k\right) \geq 1-\varepsilon \quad \forall i,j \in N, t \in \mathcal{T} \tag{7c}$$

In constraint Eq. (7c), the constant k is an upper bound on the imbalance $s_{ij}(t+1)$. The Chance Constraint Optimization (CCO) problem can be difficult to solve (Van Ackooij et al., 2011). There are several methods to reformulate the chance constraints into deterministic constraints. One method is to consider that the probability distribution belongs to a set of distributions, called ambiguity sets (Van Parys et al., 2016). In this work, we use the separable model for reformulation of the chance constraints (Prékopa, 2013).

3.3. Separable model

In the imbalance constraint, Eq. (7), the uncertainty and the decision variables enter in an affine way. This is a special case of the chance constraint and is referred to as a separable chance constraint (Shapiro et al., 2021). A separable chance constraint with known probability distribution can be reformulated as a deterministic constraint. We can rewrite the separable chance constraint to a deterministic constraint by using the cumulative distribution function (CDF):

$$F_{\lambda_{ij}(t)}(z) := \mathbb{P}_{ij}\left(\lambda_{ij}(t) \leq z\right) \tag{8}$$

With the use of the CDF, Eq. (8), the chance constraint, Eq. (7c), can be written as

$$F_{\lambda_{ij}(t+1)}\left[k + x_{ij}^{c}(t+1) - s_{ij}(t)\right] \geq 1 - \varepsilon$$

Then by taking the inverse CDF we get the following constraint,

$$k + x_{ij}^{c}(t+1) - s_{ij}(t) \geq F^{-1}_{\lambda_{ij}(t+1)}(1-\varepsilon) \tag{9}$$

$F^{-1}_{\lambda_{ij}(t+1)}(1-\varepsilon)$ is also called the quantile function. Constraint (9) is deterministic if the CDF is known. In this study, the travel demand, $\lambda_{ij}(t)$, is predicted using Gaussian process regression (GPR). The GPR gives a mean prediction, μ, and a confidence bound on the prediction, σ, where the confidence is assumed to follow a Gaussian distribution. We can therefore use the cumulative distribution function for a Gaussian distribution, which is defined as

$$F(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{(z-\mu)^2}{2\sigma^2}}\mathrm{d}z \tag{10}$$

Given a mean, μ, and a standard deviation, σ, the cumulative distribution function is explicit, hence constraint (9) is also explicit. The chance constraint formulation in Eq. (7) can therefore be reformulated into the deterministic constraint (9).

The stochastic variable is bounded to be larger or equal to zero, $\lambda_{ij}(t) \geq 0$. However, the Gaussian distribution might take both positive and negative values. Hence, we use the truncated normal distribution for negative values of the quantile function. The lower truncated CDF, is given by

$$F_{\mathrm{tr}}(x;\mu,\sigma) = \begin{cases} 0 & \text{if } \ x < 0 \\ \dfrac{F(x;\mu,\sigma) - F(0;\mu,\sigma)}{1 - F(0;\mu,\sigma)} & \text{if } \ x \geq 0 \end{cases}$$

where $F(x;\mu,\sigma)$ is the standard normal CDF and $F(0;\mu,\sigma)$ is the CDF evaluated at zero. Truncating the Gaussian distribution for GPR has been proposed in several papers, including (Jensen et al., 2013) and (Swiler et al., 2020).

3.4. Gaussian Processes regression (GPR) for time-series modelling

A Gaussian Process (GP) is a non-parametric probabilistic model that can be used for making predictions. Gaussian Process Regression (GPR) is an effective tool for predicting time series with uncertainty bounds (Roberts et al., 2013). GPRs have been successfully applied to predict time-series mobility data (Gammelli et al., 2020) as well as non-negative traffic volume time series data (Xie et al., 2010).

The GPR can be explained from the function perspectives, called the function-space view (Rasmussen and Williams, 2005). Consider a black box system with input $\mathbf{t}$ and output $\lambda = f(t)$, where $f(t)$ is an unknown function. Assume that we have historic input- and output-data from this system, called the training dataset $\mathcal{D} = \{(\mathbf{t}_i, \lambda_i)|i = 1, \ldots, n\}$. There are infinitely many functions that can be fitted on the dataset. In GPRs, a probabilistic method is used to find the best function fit. This is done by assigning a multivariate probability distribution to the entire function-space. By using a probability distribution of the function space, it is possible to include confidence of the prediction.

Based on prior knowledge and a training dataset, the aim of GPR is to find the underlying multivariate distribution. Prior knowledge can be incorporated into the fitting process; for example, periodicity or smoothness properties of $f(t)$. In GPR, the underlying multivariate distribution is assumed to be a multivariate normal distribution. Hence the estimated output follows a normal distribution, $\lambda_1, \ldots, \lambda_n \sim \mathcal{N}(\mu(\mathbf{t})_{i,\ldots,n}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}_{\mathbf{i},\mathbf{j}} = \mathrm{Cov}(\lambda_i, \lambda_j) = k(t_i, t_j)$ is the covariance function, also called kernel, and $\mu(\mathbf{t})$ is the mean function. Thus, the Gaussian process is completely defined by its mean and covariance functions according to

$$f(t) \sim \mathcal{GP}(\mu(\mathbf{t}), k(\mathbf{t}, \mathbf{t}')) \tag{11}$$

An important aspect of kernels is that they are only dependent on the inputs. The covariance function can be any function that generates a positive semi-definite covariance matrix (Kocijan 2016). When selecting different kernels, prior knowledge of the data is used. If we assume a smooth function, the radial basis function kernel (RBF) can be used

$$k_{\mathrm{RBF}}(\boldsymbol{t}, \mathbf{t}') = \exp\left(-\frac{\|\boldsymbol{t} - \mathbf{t}'\|^2}{2l^2}\right) \tag{12}$$

where l is the lengthscale hyperparameter. If the data is periodic, a periodic kernel is proposed:

$$k_{\mathrm{Periodic}}(\boldsymbol{t}, \mathbf{t}') = \exp\left(-2\frac{\sin^2\left(\frac{\pi}{p}(\boldsymbol{t} - \mathbf{t}')\right)}{l^2}\right) \tag{13}$$

where p is the period and l is the lengthscale hyperparameter. The sum and multiplication of two kernels is also a kernel (Rasmussen and Williams, 2005).

When the kernels have been selected, the hyperparameters are trained on the dataset by maximizing the log-marginal likelihood (Rasmussen and Williams, 2005). The log marginal likelihood is given by

$$\log p(\boldsymbol{y}|X,\theta) = -\frac{1}{2}\boldsymbol{y}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{y} - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{n}{2}\log(2\pi) \tag{14}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix,

$$\boldsymbol{\Sigma}_{\boldsymbol{n},\boldsymbol{n}} = \begin{pmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,n} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n,1} & k_{n,2} & \cdots & k_{n,n} \end{pmatrix} \tag{15}$$

A gradient method is used to find the hyperparameters that maximize the log marginal likelihood, i.e., the partial derivatives of Eq. (14) with respect to the hyperparameters are computed:

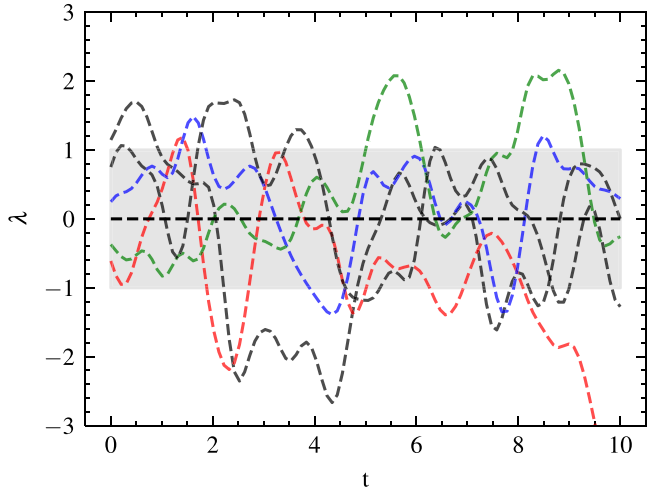

Fig. 2. Sample functions generated from the prior distribution of the locally periodic kernel. Each sample function is plotted as a dotted line in a different color, illustrating the variability of functions that can be generated from this distribution.

$$\frac{\partial}{\partial \theta_i}\log p(\mathbf{y}|X,\theta) = -\frac{1}{2}\mathbf{y}^\top \mathbf{\Sigma}^{-1}\frac{\partial \mathbf{\Sigma}}{\partial \theta_i}\mathbf{\Sigma}^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}\left(\mathbf{\Sigma}^{-1}\frac{\partial \mathbf{\Sigma}}{\partial \theta_i}\right) \tag{16}$$

The computational complexity of training the GPR is mainly due to the need of finding the matrix inversion of $\mathbf{\Sigma}$ and requires the computational complexity ($\mathcal{O}(n^3)$) (Rasmussen and Williams, 2005). From the tuned kernels and mean function, future prediction can be made using conditional probability on the posterior distribution. Given a new input t^*, the predictive distribution of the corresponding output λ^* is a Gaussian distribution with mean and variance:

$$\hat{\mu}(t^*) = \boldsymbol{k}^{*\top}\mathbf{\Sigma}^{-1}\mathbf{y} \tag{17}$$

$$\hat{\sigma}^2(t^*) = k(t^*,t^*) - \boldsymbol{k}^{*\top}\mathbf{\Sigma}^{-1}\boldsymbol{k}^* \tag{18}$$

where $\mathbf{\Sigma}$ is the covariance matrix for the training data, $\boldsymbol{k}^*$ is the vector of covariances between t^* and n training points.

In this work, we will use a locally periodic kernel which is the multiplication of the RBF and Periodic kernels. Periodic kernels assumes perfect correlation between data points that are $N \times p$ distances apart, i.e., $t - t' = N \times p$, where N is an integer. This strict periodicity assumption is not valid for most stochastic functions. While travel demand data have some periodicities they are not strict, e.g., the exact time and extend of people commuter patterns varying from day to day. Locally periodic kernels allow the shape of the periodic parts to vary over time and are therefore better suitable for travel demand prediction. Arbitrary function samples from the prior of the locally periodic kernel can be seen in Fig. 2. It can be seen that there are local periodicity in each sample but the periodicity can change over time.

3.5. Chance constraint MPC (CCMPC) with GPR

The chance constraint in optimization Problem (7) can be reformulated to an deterministic constraint using the separable model in Section 3.3. The mean and standard deviation in Eq. (10) is estimated using GPR, Eqs. (17) and (18). Hence the chance constraint represent the confidence in the prediction of the travel demand. The estimated mean and standard deviation are denoted $\hat{\mu}$ and $\hat{\sigma}$, Eqs. (17) and (18), respectively. The final optimization problem can be written as

$$\underset{x^r_{ij}, s_{ij}}{\text{minimize}} \quad \sum_{t=t_0}^{T+t_0}\sum_{i,j=1}^{N} c^r_{ij}(t)x^r_{ij}(t) + c_\lambda(t)s_{ij}(t) \tag{19a}$$

subject to

$$s_{ij}(t+1) = F^{-1}_{\lambda_{ij}(t+1)}(1-\varepsilon;\hat{\mu},\hat{\sigma}) - x^c_{ij}(t+1) + s_{ij}(t) \le k, \quad \forall i,j \in N, t \in \mathcal{T} \tag{19b}$$

(6b), (6d), and (6e) (19c)

The chance constraint in Eq. (7c) is reformulated to Eq. (19b) using the separable model in Eq. (9). The optimization problem, (19), is now an deterministic MILP. An important property with Problem (19) is that the MILP is totally unimodular and hence the corresponding linear program (LP) solution will always be integral. Therefore, the optimization Problem (19) can be solved efficiently as a LP with the simplex method. The proof that Eq. (19) is totally unimodular can be found in Appendix.

3.6. Minimal fleet size

The chance constraint Eq. (7c) guarantees that the imbalance in each station i is below a threshold k with probability $1-\varepsilon$. However, this guarantee is only valid if we have enough vehicles in the station to drive the predicted demand, $x^c_{ij}(t)$. The decision variable $x^c_{ij}(t)$ is constrained by Eq. (6d). Hence, we need to rebalance vehicles between station and ensure that the total fleet size is large enough. The minimal fleet size can be found by solving the following optimization problem.

$$\underset{x^r_{ij}, s_{ij}, \varphi_i(0)}{\text{minimize}} \quad \sum_{t=t_0}^{T+t_0}\sum_{i,j=1}^{N} c^r_{ij}(t)x^r_{ij}(t) + c_\lambda(t)s_{ij}(t) + c_\varphi\varphi_i(0) \tag{20a}$$

subject to

$$s_{ij}(t+1) = F^{-1}_{\lambda_{ij}(t+1)}(1-\varepsilon;\hat{\mu},\hat{\sigma}) - x^c_{ij}(t+1) + s_{ij}(t) \le k, \forall i,j \in N, t \in \mathcal{T} \tag{20b}$$

$$\sum_{j=1}^{N} x^c_{ij}(0) + x^r_{ij}(0) \le \varphi_i(0) \qquad \forall i \in N \tag{20c}$$

$$\sum_{j=1}^{N} x^c_{ij}(t) + x^r_{ij}(t) - x^c_{ji}(t-\kappa_{ji}) - x^r_{ji}(t-\kappa_{ji}) = 0 \qquad \forall i \in N, t \in \mathcal{T} \tag{20d}$$

(6e), (6b), and $\varphi_i(0) \in \mathbb{N}$ (20e)

The optimization problem, Eq. (20), minimizes the rebalancing, x^r_{ij}, the imbalance, $s_{ij}(t)$, and the initial number of vehicles in each station, $\varphi_i(0)$. The imbalance is constrained by the upper threshold k. When seeking the minimal fleet size required to serve all demand directly, we set $k = 0$ to ensure that there is no imbalance, $s_{ij}(t) = 0$. The optimization problem is solved for a full day.

3.7. Algorithm

The proposed algorithm for 1 time step is presented in Algorithm 1. The control actions are updated every Δt_{MPC} minute and the GPR is updated every Δt_{GP} minute. For dispatching of request and vehicles in each station, we use the Hungarian algorithm (Kuhn, 1955). It is important to note that the Hungarian can only match request and vehicles in the same station. The proposed Algorithm 1 makes sure that there are enough vehicles in each station to serve the request. Before the dispatching and rebalancing started, the city/rural area is discretized into N stations using k-means clustering on historical request location. A visual representation of the dispatching, prediction, and rebalancing can be

Fig. 3. Visual representation of the dispatching, travel demand prediction, and MPC. The map is split into N stations and there is an initial number of requests and vehicles. First, the requests are matched with vehicles in the same station. Then the travel demand is predicted, here represented by the colored bars. Based on the travel demand prediction the AMoD optimization problem is solved (Algorithm 1) and the vehicles are rebalanced accordingly.

seen in Fig. 3.

Algorithm 1. AMoD dispatching and rebalancing.

Algorithm 1: AMoD dispatching and rebalancing

Input Time t, System state $\phi_i(t)$, Historical demand $\lambda_{ij}^{\text{hist}}$, Currently waiting customer $\lambda_{ij}^{\text{wait}}$, probability $1-\epsilon$.
Output: Control action x^{r}.
if $t \mod \Delta t_{GP}$ **then**
  Train GPR
  Obtain $\{\hat{\mu}(t), \hat{\sigma}(t)\}_{t\in\mathcal{T}}$
else
  Use previous GPR
  Obtain $\{\hat{\mu}(t), \hat{\sigma}(t)\}_{t\in\mathcal{T}}$
end
if $t \mod \Delta t_{MPC}$ **then**
  $x^{\text{r}} \leftarrow$ solve (19)
end
return x^r

4. Case study

In this section, the proposed Chance Constrained MPC algorithm outlined in Algorithm 1 is tested in realistic AMoD scenarios using a high-fidelity transport simulator.

4.1. Simulation environment

The high-fidelity transport simulator AMoDeus (Ruch et al., 2018) was used for contrasting and benchmarking of the AMoD algorithm (Algorithm 1). AMoDeus is an open-source agent-based transport simulator based on Multi-Agent Transport Simulator (MATSim) (Horni et al., 2016). It was intentionally developed to simulate AMoD systems and to test new algorithms for fleet control. With AMoDeus and MATSim large scale transport simulations for one full day can be performed. The transport network is constructed using a queue based approach. Several realistic mobility scenarios for different cities are implemented in AMoDeus for benchmark testing. We chose to simulate the San Francisco scenario for this work. The San Francisco scenario is based on a taxi dataset from 2008 (Piorkowski et al., 2022). The data contains mobility traces from 500 taxi vehicles in San Francisco and contains 464,045 customer trips, which were collected between May 17, 2008 and June 10, 2008. In this study, we have chosen to simulate Thursday, May 29th, which corresponds to a total of 11,453 requests. The transport simulations were performed on a MacBook Pro with a 2.3 GHz Quad-Core Intel core i7 processor and 16 GB of RAM. We used IBM CPLEX to solve optimization problem Eqs. (19) and (20) (Cplex, 1987). Optimization

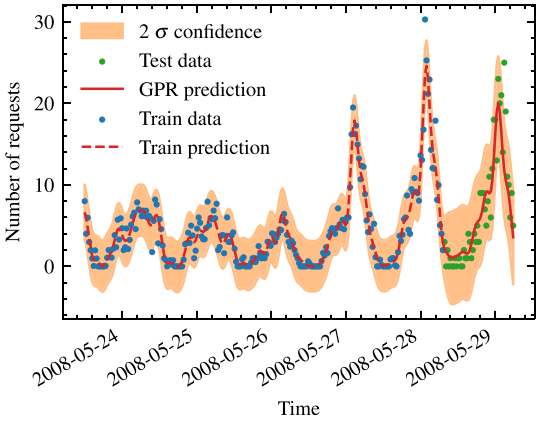

Fig. 4. Gaussian Process regression of requests going from one station to another. The blue data points indicate the training data while the green data points correspond to the request we want to predict. The red line is the prediction of incoming requests and the orange area is the corresponding 95% confidence bound.

Table 1
Metric scores for GPRs.

Metric	Mean	Standard deviation
Explained variance score	0.362	0.225
Mean squared error	2.054	5.151

Problem (19) is totally unimodular and hence an LP solver such as Primal Simplex can be used. Optimization Problem (20) is an MILP and hence a MILP solver is used, such as branch-and-bound.

4.2. Travel demand prediction

The GPR is implemented using the GPyTorch library in Python (Gardner et al., 2018). GPyTorch is a highly efficient implementation of GPR that leverages PyTorch for GPU acceleration. This enables us to train GPR models at a high speed, making real-time usage possible.

Our training data consist of the flow of requests between all stations per time interval. Therefore, a separate GPR is trained for each specific flow. In this study, we divided San Francisco into 10 stations, requiring us to train 100 GPR models. The number of GPR models scales quadratically with the number of stations ($\mathcal{O}(N^2)$). While optimal partitioning or dynamic repartitioning is beyond the scope of this paper, we found that 10 districts were sufficient based on the size of the city and data availability.

We used the previous five days' data from the day we want to predict for training. In this study, we used training data from 2008 to 05–24 to 2008-05-28. The trained mean and 95% confidence for the flow between stations 0 and 3 can be seen in Fig. 4. The mean prediction and confidence accurately describe the data, including the peak demand on day 2008-05-29, where the mean prediction is low, but the confidence interval encompasses the peak test data points. The explained variance score and mean squared error (MSE) for all trained GPR models are shown in Table 1. The low mean explained variance score with a relatively high standard deviation indicates that the selected kernel is not suitable for predicting all flows due to randomness in travel patterns. However, since we account for prediction uncertainty in Algorithm 1, we can handle this explicitly. Additionally, the mean MSE metric is quite low, indicating that our average behavior is adequate.

The use of Gaussian process regression (GPR) can lead to non-physical solutions, with λ_{ij} taking negative values, especially when the data points are close to zero, as illustrated in Fig. 4. This issue is most pronounced during off-peak hours when the demand is low, and little rebalancing is needed. To address this problem, we have incorporated a truncation into the optimization problem, which ensures that the Gaussian distribution is bounded to be greater than or equal to zero (Algorithm 1). Similar truncation methods have be investigated in Jensen et al. (2013) study on bounded likelihood functions in GPR, as well as Swiler et al. (2020) survey paper on constrained GPR.

The computational complexity of GPR is cubic in the number of data points ($\mathcal{O}(n^3)$) (Rasmussen and Williams, 2005), making it unsuitable for large datasets. However, GPyTorch reduces the computational complexity to be squared in the number of data points ($\mathcal{O}(n^2)$). This, combined with GPU acceleration, results in an acceptable computation burden for our case studies. In this study, the average computational training time per prediction for GPR was 4.82 s (Table 3). Since we

Table 2
Pick-up time, total-,rebalance-, and pick-up-distances for fixed fleet size of 300 vehicles and each control algorithm.

Metric	MPC-Oracle	CCO MPC	GBM	MPC-fixed
Mean pickup time (s)	205	211	276	230
Median pickup time (s)	171	173	236	191
Total distance (km)	52,645	52,842	51,361	54,858
Rebalance distance (km)	7,307	7,206	0	9,970
Pickup distance (km)	7,464	7,757	13,482	6,983

Table 3
Computational running time for single execution of Chance Constrained MPC (CCMPC) for routing and Gaussian Process Regression (GPR) for travel demand prediction.

Method	Sample	Mean (s)	Median (s)	STD (s)	Max (s)
CCMPC	143	0.093	0.078	0.047	0.281
GPR	886	4.82	4.82	0.112	6.87

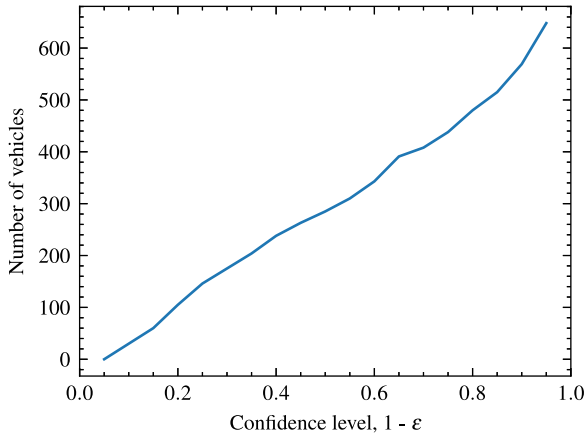

Fig. 5. Minimal fleet size, optimal solution from Eq. (20), for different confidence $1-\varepsilon$.

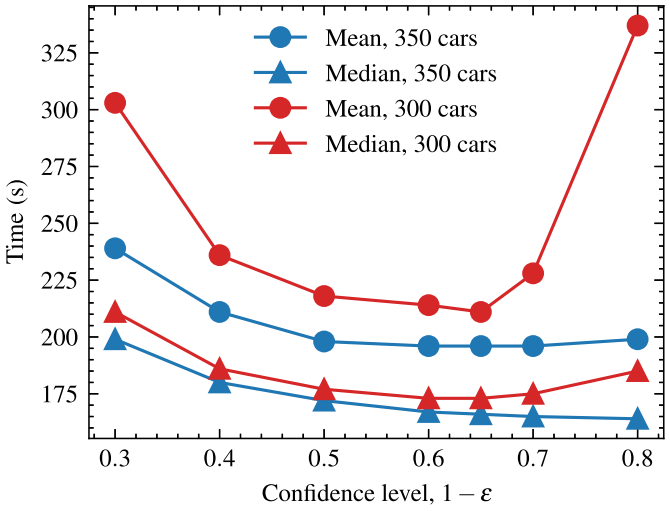

Fig. 6. Mean and median wait time for different confidence bounds for fixed fleet size of 300 and 350 vehicles using CCMPC Algorithm 1.

predict one hundred different flows when the city is divided into ten stations, the total computational time is around 482 s.

4.3. Minimal fleet size for different confidence levels

By solving the optimization problem, Eq. (20) we get the minimal required fleet size to keep the imbalance in each station below some threshold, k, with probability $1-\varepsilon$.

Intuitively, the minimal required fleet size should increase with higher probability, $1-\varepsilon$, which is also the case in Fig. 5. The results in Fig. 5 are from solving optimization Problem (20) with threshold $k=0$. When $1-\varepsilon$ is increasing the bound on the travel demand, $\lambda_{ij}(t)$, increases and hence a larger fleet size is required.

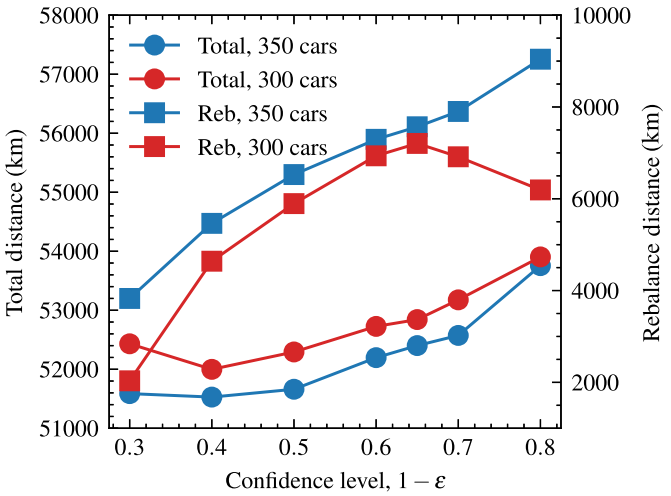

Fig. 7. Total and rebalancing distance for different confidence bounds for fixed fleet size of 300 and 350 vehicles.

4.4. Confidence bound in CCMPC

In the CCMPC (Algorithm 1), there are four variables that can be tuned according to the specific scenario; the objective cost weights ($c^{\mathrm{r}}_{ij}(t)$, c_λ), the horizon length T, the confidence bound $1-\varepsilon$, and the fleet size. We have chosen to have a horizon of 3 h and the cost weights are set according to operating costs, where the rebalancing cost is relative to the distance and the imbalance cost is relative to customer wait time. With a fixed horizon and cost weights, we study how the confidence bound in the chance constraint optimization Problem (7) affects the performance for fleet sizes of 300 and 350 vehicles. Two performance metrics are evaluated; the pick-up time (Fig. 6), which indicates the service level provided; and the distances driven (Fig. 7), which represent operating cost.

By studying the mean and median pick-up time as a function of confidence level for 300 vehicles, it is evident that there is an optimal confidence level at around 0.65 where both of these measures are minimized (Fig. 6). The maximum mean wait time is reached for a confidence level of 0.8 and the second-highest for a confidence level of 0.3. When the confidence level is increased, the number of predicted requests between the different stations' increase. Hence, a higher confidence level requires a larger fleet size, which we concluded in Section 4.3. For a fleet size of 300 vehicles there is not enough control input for confidence level of 0.8 and the rebalancing hence decreases (Fig. 7). When the vehicle fleet is increased to 350 vehicles, there is possibility for more rebalancing and the rebalancing increases for confidence level of 0.8. However, the performance in terms of median and mean pick-up time is similar or worse then for confidence level of 0.65. The extra rebalancing is not improving the service level since it rebalance more then it have too. For low confidence levels the number of requests are underestimated, hence the rebalancing decreases for lower confidence levels (Fig. 7). Even though the rebalancing distance is the lowest for confidence level of 0.3, the total distance is the lowest for confidence level of 0.4 which has more than the doubled rebalancing distance compared to confidence level of 0.3 (Fig. 7). Beyond 0.4, the distance grows due to stricter and stricter chance constraints forcing vehicles to drive towards customers. A confidence level of 0.65 has the maximal rebalancing distance of 7,206 km and the total distance is only 844 km more than the minimum total distance for fleet size of 300 vehicles. This indicates that the rebalancing at confidence level of 0.65 decreases the mean and median wait time at a low cost. Hence, this confidence level is considered to be optimal for this scenario. For this confidence level the median wait time is reduced by 4% compared to using only the mean prediction of the GPR.

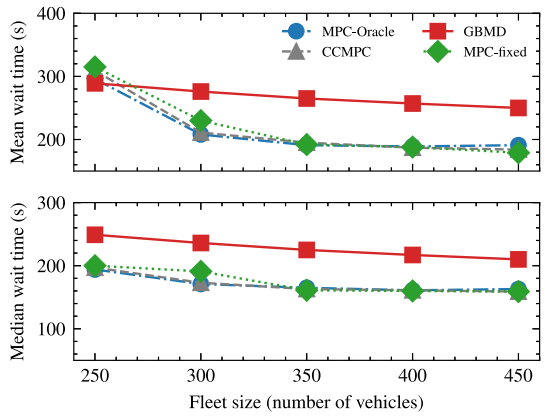

Fig. 8. Mean and median wait times as function of fleet size for different control algorithms.

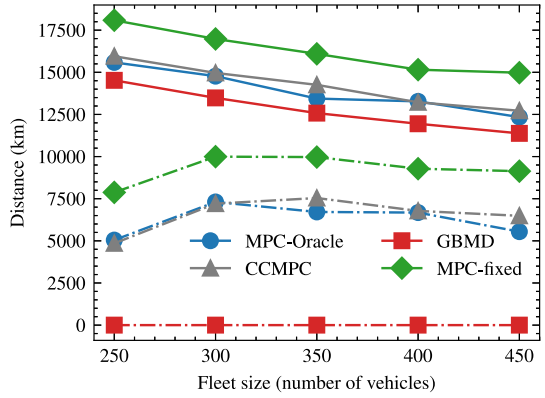

Fig. 9. Driving distance as function of fleet size for different control algorithms. Solid lines correspond to empty distance, i.e., distance driven without customers, and dash-dotted line corresponds to rebalance distance.

4.5. Comparative evaluation of AMoDs

The performance of the CCMPC (Algorithm 1) is benchmarked against three different control algorithms:

- *MPC-Oracle* – A non-causal controller where the future travel demand, $\lambda_{ij}(t)$, is known for all t in Eq. (6), i.e. the performance of this controller is an upper limit for the performance of the proposed algorithm.
- *MPC-FixedDemand* – This is a causal controller, see Eq. (6), with a fixed future travel demand, all future travel demands are set equal to the last known travel demand, i.e., $\lambda_{ij}(t) = \lambda_{ij}(t_0 - 1)\ \forall\ t \in [t_0, T + t_0]$.
- *Global Bipartite Matching Dispatcher (GBM)* – This controller solves the bipartite problem to match available vehicles with customer request using the Hungarian algorithm (Kuhn, 1955). The cost of matching a request with a vehicle is the distance between them. The controller does simply react to the current demand and does not perform any rebalancing.

A comparison of the performance, pick-up time, and distance driven, for different control algorithms as a function of fleet size can be seen in Figs. 8 and 9. It is apparent that the best performing algorithm in terms of wait time is the MPC-Oracle, which is expected. However, for a fleet size of 300 vehicles the best performing causal algorithm is the CCMPC with only a few seconds more than mean and median pick-up time (Fig. 8). For a fleet size of 300 vehicles, the mean pickup time for CCMPC is 24% lower than the GBM (Table 2), which is the worst performing algorithm in terms of pickup time. This is also expected since GBM is a reactive algorithm. On the other hand, MPC-fixed has a mean pickup time that is only 19 s longer compared to CCMPC. (Table 2). However, the MPC-Fixed has a rebalancing distance that is 2,764 km more than CCMPC. Since the fixed demand prediction is not accurate, a lot of vehicles will unnecessarily be rebalanced. When the fleet size is above 300 vehicles, the performance of MPC-Oracle, CCMPC, and MPC-fixed in terms of pick-up time is similar because a large fleet size compensates for a bad controller as the MPC-Fixed. A lot of vehicles can be rebalanced without affecting the pick-up-time since there is an oversupply of vehicles in the system. Therefore, the mean and median pickup-time is similar but the total distance driven is still more for the MPC-Fixed. From an operator's perspective, it is desired to keep the fleet size as low as possible because of cost savings. Therefore, a fleet size of 300 vehicles seems to be optimal in terms of cost and performance (Figs. 8 and 9).

4.6. Computational complexity

The number of variables in the optimization problem is proportional to the number of stations and the time horizon, resulting in a computational complexity of $\mathcal{O}(N^2T)$ for the number of variables and $\mathcal{O}(4N^2T + NT)$ for the number of constraints. The low computational running time of 0.093 s (on average, Table 3) is due to the optimization problem being TU (totally unimodular), as well as the fact that the number of variables and constraints is solely dependent on the number of stations and the time horizon.

5. Conclusion and future work

In this study, we have proposed a predictive chance constraint rebalancing approach for autonomous mobility-on-demand (AMoD) services, which is applied to the use case of ride-hailing. We first introduce a commonly used model for this service where the service area is discretized into smaller areas, called stations. The model consists of constraints for the imbalance and vehicle conservation. Based on the model, a model predictive controller (MPC) is formulated with the multi-objective to minimize the rebalance distance for vehicles and the imbalance in each station. The travel demand is predicted using Gaussian Process regression (GPR). GPR, in contrast to other proposed prediction methods, is superior for small datasets and provides a confidence bound on the prediction. We account for uncertainties in the travel demand prediction by formulating a chance constraint MPC (CCMPC). The CCMPC is relaxed using the GPR prediction and the use of the separable model. The proposed algorithm was benchmarked using the high fidelity transport simulator AMoDeus and real taxi data from San Francisco (Ruch et al., 2018). Our results show the importance of incorporating the confidence bound of the demand prediction. By tuning the confidence bound, the median wait time is reduced by 4% compared to using only the mean prediction of the GPR. We showed that the CCMPC is performing close to optimal performance and that is significantly better than a reactive controller. The performance and computational efficiency of the proposed method implies that it would be useful for real-time control. There are many important directions to consider for future work including embedding traffic and limited range into the model as well as more case studies for different cities. Furthermore, future research could explore the potential benefits of utilizing multivariate Gaussian process regression to model the continuous spatial properties of travel demand. Additionally, the incorporation of endogenous traffic and induced travel demand, which could affect the travel demand in a given area and therefore the service quality.

Replication and data sharing

The data used in this paper can be downloaded from https://ieee-dataport.org/open-access/crawdad-epflmobility. The simulation software used in this paper can be accessed at https://github.com/a

modeus-science/amod and downloaded for use.

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.

Acknowledgement

This work was co-funded by Vinnova, Sweden through the project: Simulation, analysis and modeling of future efficient traffic systems; the FFI program Efficient and Connected Transport Systems. Balázs Kulcsár acknowledged the partial support of the Area of Advanced Transport.

Appendix. Total Unimodular

There are certain cases where the optimal solution of the LP relaxation of an MILP is guaranteed to be integral. Consider the following MILP:

$$\begin{aligned} &\text{minimize}_x && c^\top \mathbf{y} \\ &\text{subject to} && \\ & && A\mathbf{y} \le b \\ & && \mathbf{y} \in \mathbb{N} \end{aligned}$$

If the A matrix is totally unimodular (TU), then the linear programming (LP) relaxation will always have one integral solution (Hoffman and Kruskal, 2010). We aim to show that the chance constraint optimization problem in Eq. (19) is TU. We can write the active constraints in the optimization problem (Eq. (19)) with the following matrix constraint:

$$\overbrace{\begin{bmatrix} B & C & 0 \\ 0 & D & E \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} \mathbf{x}^{\mathrm{r}} \\ \mathbf{x}^{\mathrm{c}} \\ \mathbf{s} \end{bmatrix}}^{y} = \overbrace{\begin{bmatrix} \varphi \\ \lambda \\ 0 \\ 0 \\ 0 \end{bmatrix}}^{b} \tag{A1}$$

where B and C represent the vehicle conservation constraint (6d), and D and E are matrices chosen to represent the imbalance constraint in Eq. (19b). The vectors $\boldsymbol{x}^{\mathrm{r}}$, $\boldsymbol{x}^{\mathrm{c}}$, and $\boldsymbol{s}$ are vectors of all decision variables $x^{\mathrm{r}}_{ij}(t)$, $x^{\mathrm{c}}_{ij}(t)$, and $s_{ij}(t)$. The matrix D is an identity matrix. Matrices B, C, D, and E are TU because each column of these matrices consists of at most two non-zero entries (+1 and −1) (Schrijver, 1998). Since we are minimizing $\boldsymbol{x}^{\mathrm{r}}$ and $\boldsymbol{s}$, we can neglect the decision variable $\boldsymbol{x}^{\mathrm{c}}$ by rewriting the constraint:

$$D\mathbf{x}^{\mathrm{c}} + E\mathbf{s} = \lambda \Leftrightarrow \mathbf{x}^{\mathrm{c}} = D^{-1}\lambda - D^{-1}E\mathbf{s} \tag{A2}$$

Therefore, we can simplify Eq. (21) to

$$\overbrace{\begin{bmatrix} B & 0 & -CD^{-1}E \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} \mathbf{x}^{\mathrm{r}} \\ \mathbf{x}^{\mathrm{c}} \\ \mathbf{s} \end{bmatrix}}^{y} = \overbrace{\begin{bmatrix} \varphi - D^{-1}\lambda \\ 0 \\ 0 \\ 0 \end{bmatrix}}^{b}, \tag{A3}$$

which $\boldsymbol{A}$ is a TU matrix (Schrijver, 1998). Therefore, both $\boldsymbol{x}^{\mathrm{r}}$ and $\boldsymbol{s}$ are integer, and thus, so is $\boldsymbol{x}^{\mathrm{c}}$ because all values in Eq. (22) are integer.

References

Braverman, A., Dai, J.G., Liu, X., Ying, L., 2019. Empty-car routing in ridesharing systems. Oper. Res. 67 (5), 1437–1452.
Charnes, A., Cooper, W.W., 1959. Chance-constrained programming. Manag. Sci. 6 (1), 73–79.
Cplex, I.I., 1987. 'User's Manual for Cplex'. IBM. https://www.ibm.com/docs/en/icos/20.1.0?topic=cplex-users-manual.
Gammelli, D., Peled, I., Rodrigues, F., Pacino, D., Kurtaran, H.A., Pereira, F.C., 2020. Estimating latent demand of shared mobility through censored Gaussian processes. Transport. Res. C Emerg. Technol. 120, 102775.
Gardner, J., Pleiss, G., Weinberger, K.Q., Bindel, D., Wilson, A.G., 2018. Gpytorch: blackbox matrix-matrix Gaussian process inference with gpu acceleration. Adv. Neural Inf. Process. Syst. 31, 7587–7597.
George, D.K., 2012. Stochastic Modeling and Decentralized Control Policies for Large-Scale Vehicle Sharing Systems via Closed Queueing Networks. Ph.D. Dissertation. Columbus, OH: The Ohio State University.
Grosso, J., Ocampo-Martínez, C., Puig, V., Joseph, B., 2014. Chance-constrained model predictive control for drinking water networks. J. Process Control 24 (5), 504–516.
Guo, X., Caros, N.S., Zhao, J., 2021. Robust matching-integrated vehicle rebalancing in ride-hailing system with uncertain demand. Transp. Res. Part B Methodol. 150, 161–189.
He, L., Hu, Z., Zhang, M., 2020. Robust repositioning for vehicle sharing. Manuf. Serv. Oper. Manag. 22 (2), 241–256.
Hewing, L., Kabzan, J., Zeilinger, M.N., 2019. Cautious model predictive control using Gaussian process regression. IEEE Trans. Control Syst. Technol. 28 (6), 2736–2743.
Hoffman, A.J., Kruskal, J.B., 2010. Integral Boundary Points of Convex Polyhedra. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 49–76.
Horni, A., Nagel, K., Axhausen, K.W., 2016. The Multi-Agent Transport Simulation MATSim. Ubiquity Press.
Iglesias, R., Rossi, F., Wang, K., Hallac, D., Leskovec, J., Pavone, M., 2018. Data-driven model predictive control of autonomous mobility-on-demand systems. In: '2018 IEEE International Conference on Robotics and Automation (ICRA)'. IEEE, pp. 6019–6025.
Jensen, B.S., Nielsen, J.B., Larsen, J., 2013. Bounded Gaussian process regression. In: '2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)'. IEEE, pp. 1–6.
Kocijan, J., 2016. Modelling and Control of Dynamic Systems Using Gaussian Process Models. Cham: Springer International Publishing.
Kuhn, H.W., 1955. The Hungarian method for the assignment problem. Nav. Res. Logist. Q. 2 (1–2), 83–97.
Lacombe, R., Gros, S., Murgovski, N., Kulcsár, B., 2021. Bilevel optimization for bunching mitigation and eco-driving of electric bus lines. IEEE Trans. Intell. Transport. Syst. 23 (8), 10662–10679.
Litman, T., 2020. Autonomous vehicle implementation predictions: Implications for transport planning, Technical report. Victoria Transport Policy Institute, 2020-1-9.

Mao, C., Liu, Y., Shen, Z.-J.M., 2020. Dispatch of autonomous vehicles for taxi services: a deep reinforcement learning approach. Transport. Res. C Emerg. Technol. 115, 102626.
Miao, F., Han, S., Lin, S., Wang, Q., Stankovic, J.A., Hendawi, A., Zhang, D., He, T., Pappas, G.J., 2017. Data-driven robust taxi dispatch under demand uncertainties. IEEE Trans. Control Syst. Technol. 27 (1), 175–191.
Ono, M., Williams, B.C., 2008. Iterative risk allocation: a new approach to robust model predictive control with a joint chance constraint. In: '2008 47th IEEE Conference on Decision and Control'. IEEE, pp. 3427–3432.
OpenStreetMap contributors, 2017. Planet OSM. https://planet.osm.org.
Pavone, M., Smith, S.L., Frazzoli, E., Rus, D., 2012. Robotic load balancing for mobility-on-demand systems. Int. J. Robot Res. 31 (7), 839–854.
Pereira, M., Boyraz Baykas, P., Kulcsár, B., Lang, A., 2022. Parameter and density estimation from real-world traffic data: a kinetic compartmental approach. Transp. Res. Part B Methodol. 155, 210–239.
Piorkowski, M., Sarafijanovic-Djukic, N., Grossglauser, M.. Crawdad Epfl/mobility. https://ieee-dataport.org/open-access/crawdad-epflmobility.
Prékopa, A., 2013. Stochastic Programming, 324. Springer Science & Business Media, pp. 233–269.
Raposo, A., et al.. The Future of Road Transport. https://publications.jrc.ec.europa.eu/repository/handle/JRC116644.
Rasmussen, C.E., Williams, C., 2005. Gaussian Processes for Machine Learning. The MIT Press.
Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., Aigrain, S., 2013. Gaussian processes for time-series modelling. Phil. Trans. Math. Phys. Eng. Sci. 371 (1984), 20110550.
Ruch, C., Gächter, J., Hakenberg, J., Frazzoli, E., 2020. The +1 method: model-free adaptive repositioning policies for robotic multi-agent systems. IEEE Trans. Netw. Sci. Eng. 7 (4), 3171–3184.
Ruch, C., Hörl, S., Frazzoli, E., 2018. Amodeus, a simulation-based testbed for autonomous mobility-on-demand systems. In: '2018 21st International Conference on Intelligent Transportation Systems (ITSC)'. IEEE, HI, USA, pp. 3639–3644.
Salazar, M., Rossi, F., Schiffer, M., Onder, C.H., Pavone, M., 2018. On the interaction between autonomous mobility-on-demand and public transportation systems. In: '2018 21st International Conference on Intelligent Transportation Systems (ITSC)'. IEEE, HI, USA, pp. 2262–2269.
Schrijver, A., 1998. Theory of Linear and Integer Programming. John Wiley & Sons.
Shapiro, A., Dentcheva, D., Ruszczynski, A., 2021. In: Lectures on Stochastic Programming: Modeling and Theory, 3rd edn. Society for Industrial and Applied Mathematics, Philadelphia, PA.
Swiler, L.P., Gulian, M., Frankel, A.L., Safta, C., Jakeman, J.D., 2020. A survey of constrained Gaussian process regression: approaches and implementation challenges. J. Mach. Learn. Model. Comput. 1 (2), 119–156.
Tsao, M., Iglesias, R., Pavone, M., 2018. Stochastic model predictive control for autonomous mobility on demand. In: '2018 21st International Conference on Intelligent Transportation Systems (ITSC)'. IEEE, pp. 3941–3948.
United Nations Publications, 2019. World Urbanization Prospects: the 2018 Revision. UN.
Van Ackooij, W., Zorgati, R., Henrion, R., Möller, A., 2011. Chance constrained programming and its applications to energy management. In: Stochastic Optimization-Seeing the Optimal for the Uncertain. IntechOpen, pp. 291–320.
Van Parys, B.P.G., Goulart, P.J., Kuhn, D., 2016. Generalized gauss inequalities via semidefinite programming. Math. Program. 156 (1), 271–302.
Varga, B., Tettamanti, T., Kulcsár, B., Qu, X., 2020. Public transport trajectory planning with probabilistic guarantees. Transp. Res. Part B Methodol. 139, 81–101.
Walukiewicz, S., 2013. Integer Programming, vol. 46. Springer Science & Business Media.
Xie, Y., Zhao, K., Sun, Y., Chen, D., 2010. Gaussian processes for short-term traffic volume forecasting. Transport. Res. Rec. 2165 (1), 69–78.
Zardini, G., Lanzetti, N., Pavone, M., Frazzoli, E., 2022. Analysis and control of autonomous mobility-on-demand systems. Ann. Rev. Control, Robot. Auton. Systems 5, 633–658.
Zhang, R., Rossi, F., Pavone, M., 2016. Model predictive control of autonomous mobility-on-demand systems. In: '2016 IEEE International Conference on Robotics and Automation (ICRA)'. IEEE, pp. 1382–1389.
Zhao, K., Tarkoma, S., Liu, S., Vo, H., 2016. Urban human mobility data mining: an overview. In: '2016 IEEE International Conference on Big Data (Big Data)', pp. 1911–1920.

Sten Elling Tingstad Jacobsen received his M.S. degree in applied physics from Chalmers University of Technology, Sweden, in 2020. He is currently working on his Ph.D. jointly at Volvo Cars AB and the Department of Electrical Engineering, Chalmers University of Technology, Sweden. His research interests include optimization, machine learning, control, and their application to transportation systems and the automotive area.

Anders Lindman received his Ph.D. degree in computational materials physics from Chalmers University of Technology, Sweden, 2017. He joined Volvo Car Corporation in 2018 where he is active in the areas of Vehicle Energy Efficiency and Future Mobility.

Balázs Kulcsár received his M.Sc. degree in traffic engineering and the Ph.D. degree from Budapest University of Technology and Economics (BUTE), Budapest, Hungary, in 1999 and 2006, respectively. He has been a Researcher/Post-Doctor with the Department of Control for Transportation and Vehicle Systems, BUTE, the Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN, USA, and with the Delft Center for Systems and Control, Delft University of Technology, Delft, the Netherlands. He is currently a Professor with the Department of Electrical Engineering, Chalmers University of Technology, Göteborg, Sweden. His main research interest focuses on traffic network modeling and control.