Approximate Multiagent Reinforcement Learning for On-Demand Urban Mobility Problem on a Large Map

Daniel Garces[1], Sushmita Bhattacharya[1], Dimitri Bertsekas[2], Stephanie Gil[1]

***Abstract*— In this paper, we focus on the autonomous multiagent taxi routing problem for a large urban environment where the location and number of future ride requests are unknown a-priori, but can be estimated by an empirical distribution. Recent theory has shown that a rollout algorithm with a stable base policy produces a near-optimal stable policy. In the routing setting, a policy is stable if its execution keeps the number of outstanding requests uniformly bounded over time. Although, rollout-based approaches are well-suited for learning cooperative multiagent policies with considerations for future demand, applying such methods to a large urban environment can be computationally expensive due to the large number of taxis required for stability. In this paper, we aim to address the computational bottleneck of multiagent rollout by proposing an approximate multiagent rollout-based two phase algorithm that reduces computational costs, while still achieving a stable near-optimal policy. Our approach partitions the graph into sectors based on the predicted demand and the maximum number of taxis that can run sequentially given the user's computational resources. The algorithm then applies instantaneous assignment (IA) for re-balancing taxis across sectors and a sector-wide multiagent rollout algorithm that is executed in parallel for each sector. We provide two main theoretical results: 1) characterize the number of taxis m that is sufficient for IA to be stable; 2) derive a necessary condition on m to maintain stability for IA as time goes to infinity. Our numerical results show that our approach achieves stability for an m that satisfies the theoretical conditions. We also empirically demonstrate that our proposed two phase algorithm has equivalent performance to the one-at-a-time rollout over the entire map, but with significantly lower runtimes.**

I. Introduction

Self-driving taxis are currently operating in multiple cities, including Austin, Phoenix, and San Francisco [1], with possibilities of being deployed to more cities in the near future [2]. This widespread deployment of autonomous taxis creates new opportunities for improved on-demand mobility through coordinated routing and planning, and poses interesting new practical and theoretical problems for the field of robotics. For instance, the ability of autonomous taxis to communicate with each other and with a centralized server allows for the orchestration of fleet-wide coordinated plans that result in more requests being serviced [3].

Coordination plans have been studied in the literature in the form of the Dynamic Vehicle Routing (DVR) problem [4] with stochastic demand, where the location and number of future requests is unknown a-priori. However, due to the size of the problem and the complexity associated with the stochasticity of the demand, there are still many research opportunities related to the design of better and faster algorithms to learn cooperative plans that take into account future requests and maximally use taxi fleets. Approaches in the literature have mainly focused on immediate demand [5], [6] and sector level routing [7], [8], [9], [10], [11], abstracting away either the stochasticity of the demand, or the complexity associated with "fine-grained" street/intersection level decisions. Other works, including our previous work [12], have considered using reinforcement learning methods, particularly rollout-based approaches [13], [14], [15], to tackle fine-grained routing decisions. These rollout-based methods as defined in [14] are comprised of three major components: 1) a one-step lookahead cost minimization where the immediate future is simulated using Monte-Carlo (MC) approximation for all potential controls, 2) a future cost approximation for each potential control based on a truncated application of a simple to compute policy known as the base policy for a finite time horizon, and 3) a terminal cost approximation that compensates for the truncated application of the base policy. Recent theory [14], [15] shows that rollout's one-step lookahead cost minimization acts as a Newton step, and hence provides super linear convergence to the optimal policy. In particular, as long as the base policy is close to the optimal policy with a reasonable competitive factor [16] and it is stable, then rollout-based approaches learn a stable near-optimal policy. This theoretical result makes rollout-based algorithms very well-suited for tackling the fine-grained routing problem. In the routing setting, a policy is said to be stable if its execution results in the number of outstanding requests being uniformly bounded over time. Applying these rollout methods to a large urban environment, however, poses a unique set of challenges that we aim to address in this paper.

A major challenge of dealing with a city-scale environment is the large volume of requests that enters the system, which then requires a large number of taxis to guarantee stability. This large number of taxis makes the application of a multiagent (one-at-a-time) rollout scheme, as proposed in our previous work [12], computationally prohibitive. In this paper, we address this computational bottleneck by proposing an approximation to the one-at-a-time rollout algorithm that keeps computational costs below user-defined constraints, while still maintaining stability and the Newton-step property of rollout. Our proposed method reduces the computational

[1]Daniel Garces, Sushmita Bhattacharya, and Stephanie Gil are with the REACT lab, Harvard University, Boston, MA, USA (e-mail: {dgarces, sushmita_bhattacharya,sgil}@g.harvard.edu)

[2]Dimitri Bertsekas is with the Department of Electrical Engineering and Computer Science, Arizona State University, AZ, USA (e-mail: dimitrib@mit.edu).

This work was supported by ONR YIP grant #N00014-21-1-2714, NSF grant #2114733, AFOSR grant #FA9550-22-1-0223, Amazon Research Award, and in part by Apple Scholars in AI/ML Ph.D. Fellowship Program.

cost of executing one-at-a-time rollout with a large number of taxis by partitioning the map into disjoint sectors based on expected demand and the maximum number of taxis that can be run sequentially given the user's indicated computational resources. Our method then executes a two-phase algorithm composed of a high level planner and multiple low level planners that are run in parallel. The high level planner routes taxis across sectors based on the current and estimated future demand, while the low level planners route taxis within each sector by employing one-at-a-time rollout with instantaneous assignment with reassignment (IA-RA) as the base policy. We choose IA-RA as the base policy since it is 2-competitive[1] [16], which facilitates the super-linear convergence of one-at-a-time rollout to the optimal policy [15]. We also theoretical results for a sufficient condition on the total number of taxis m that will guarantee IA-RA to be stable. Compared to previous work [17], [18], [19], [20], our analysis uses the full stochasticity of the system and assumes that the pickups and dropoffs are jointly distributed. In addition, for the case where pickups and dropoffs can be assumed independent, we also provide a necessary condition on m for asymptotic stability of IA-RA as time goes to infinity, building on the results proposed in [19]. We empirically demonstrate that our approach results in a significantly lower computational cost and comparable performance as one-at-a-time rollout over the entire map, and we verify that stability is achieved for fleet sizes lying within the range given by our theoretical results.

II. Problem formulation

In this section, we present the formulation of a large scale multiagent taxicab routing and pickup problem as a discrete time, finite horizon, stochastic Dynamic Programming problem that plans over a city-scaled street network. In the following subsections, we provide definitions for our environment, requests, state and control spaces, the concept of stability, and the challenges associated with the large scale.

A. Environment

We assume that autonomous taxis are deployed in an urban environment with a known fixed street topology. The environment is hence represented as a directed graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ corresponds to the set of street intersections in the map numbered 1 through n, while $E \subseteq \{(i,j)|i,j \in V\}$ corresponds to the set of directed streets that connect intersections i and j. The set of neighboring intersections to intersection i is denoted as $\mathcal{N}_i = \{j|j \in V, (i,j) \in E\}$. We also assume that the environment can be divided into K sectors, where each sector $s_k \subseteq V$ and $s_k \neq \emptyset$, $\forall k \in \{1, \ldots, K\}$, such that $V = \bigcup_{k=1}^{K} s_k$ and $s_k \cap s_h = \emptyset, \forall h \neq k$.

B. Requests

We define a ride request r as a tuple $r = \langle \rho_r, \delta_r, t_r, \phi_r \rangle$, where $\rho_r \in V$ and $\delta_r \in V$ correspond to the nearest intersection to the request's desired pickup and drop-off locations, respectively; t_r corresponds to the time at which the request was placed into the system; and $\phi_r \in \{0, 1\}$ is an indicator, such that $\phi_r = 1$ if the request has been picked up by a vehicle, $\phi_r = 0$ otherwise. We model the number of requests that enter the system at time t as a random variable η_t, which has the same distribution as random variable η with an unknown underlying distribution p_η. We assume p_η is fixed for the entire length of the time horizon T, and its estimated probability distribution, denoted $\tilde{p}_\eta$, can be estimated from historical trip data. We denote the set of ride requests that enter the system at time t as $\mathbf{r_t}$. Here the cardinality of the set of new requests at time t is $|\mathbf{r_t}| = \eta_t$. We model the pickup intersection for an arbitrary request r as the random variable ρ_r. Similarly, we model the drop-off intersection for request r as the random variable δ_r. We assume that requests are independent and identically distributed (i.i.d), and hence we drop the subscripts when talking about their distributions. Random variables ρ and δ are jointly distributed and have unknown underlying probability distributions p_ρ and $p_{\delta|\rho}$, respectively. We assume these distributions do not change over the entire length of the time horizon T. We denote the marginal distribution of δ as p_δ. We also denote the estimated categorical distributions for pickup locations, conditional dropoff locations, and the marginal dropoff locations as $\tilde{p}_\rho$, $\tilde{p}_{\delta|\rho}$, and $\tilde{p}_\delta$, respectively. These categorical distributions are estimated using historical trip data. We define $\bar{\mathbf{r}}_\mathbf{t} = \{r | r \in \mathbf{r_{t'}}, \phi_r = 0, 1 \leq t' \leq t\}$ as the set of outstanding ride requests that have not yet been picked up by any taxi at time t.

C. State and control space

We assume there is a total of m taxis and all taxis can perfectly observe all requests, and other taxis' locations and occupancy status. We assume that all of the taxis remain inside the predefined street network G, and they are able to traverse any edge in G in a single time step. We represent the state of the system at time t as a tuple $x_t = \langle \vec{\nu_t}, \vec{\tau_t}, \bar{\mathbf{r}}_\mathbf{t} \rangle$. We define $\vec{\nu_t} = [\nu_t^1, \ldots, \nu_t^m]$ as the list of locations for all m taxis at time t, where $\nu_t^\ell \in V$ corresponds to the index of the closest intersection to the geographical position of taxi ℓ. We define $\vec{\tau_t} = [\tau_t^1, \ldots, \tau_t^m]$ as the list of time remaining in the current trip for all m taxis. If taxi ℓ is available, then it has not picked up a request and hence $\tau_t^\ell = 0$, otherwise $\tau_t^\ell \in \mathbb{N}^+$. The initial location of an arbitrary taxi ℓ at time $t = 0$ is given by random variable ξ_ℓ. All ξ_ℓ for $\ell \in \{1, \ldots, m\}$ are assumed to be independent and identically distributed with known underlying distribution p_ξ.

We denote the control space for taxi ℓ at time t as $\mathbf{U}_t^\ell(x_t)$. If the taxi is available (i.e. $\tau_t^\ell = 0$), then $\mathbf{U}_t^\ell(x_t) = \{\mathcal{N}_{\nu_t^\ell}, \nu_t^\ell, \psi_r\}$, where ψ_r corresponds to a special pickup control that becomes available if there is a request $r \in \bar{\mathbf{r}}_\mathbf{t}$ with pickup at the location of taxi ℓ (i.e. $\rho_r = \nu_t^\ell$). If the taxi is currently servicing a request r (i.e. $\tau_t^\ell > 0$), then $\mathbf{U}_t^\ell(x_t) = \{\zeta\}$, where ζ corresponds to the next hop in shortest path between taxi ℓ's current location ν_t^ℓ and the destination of the request δ_r. The controls available to all m taxis at time t, $\mathbf{U}_t(x_t)$, is expressed as the Cartesian product of local control sets for each taxi, such that $\mathbf{U}_t(x_t) =$

[1] An $\alpha-$competitive policy never produces a cost greater than α times the optimal cost for any input [16]

$\mathbf{U}_t^1(x_t) \times \cdots \times \mathbf{U}_t^m(x_t)$.

D. Stability of a policy

We define a policy $\pi = \{\mu_1, \dots \mu_T\}$ as a set of functions that maps state x_t into control $u_t = \mu_t(x_t) \in \mathbf{U}_t(x_t)$. Using a similar formulation as in [20], we define the total distance to be traveled in service of a request r_q with index q given a policy π as $W_{r_q,\pi} = d(l_{r_q,\pi}, \rho_{r_q}) + d(\rho_{r_q}, \delta_{r_q})$, where $l_{r_q,\pi}$ is the location of a taxi assigned to request r_q based on policy π, and $d : V \times V \to \mathbb{N}^+$ is a function that gives the length of the shortest path between two locations. We define the total distance to be traveled in service of all the requests that enter the system for the entire time horizon T as $Z_{\pi,T} = \sum_{t=1}^{T} \sum_{q=R_{t-1}+1}^{R_t} W_{r_q,\pi}$, where the random variable $R_t = \sum_{t'=1}^{t} \eta_{t'}$ represents the total number of requests that have entered the system until time t. It is important to note that $R_0 = 0$. We define the total distance that can be covered by a fleet of m taxis as $m \cdot T$ since each taxi can travel unit distance at each time step. Assuming that we have at least as many available taxis at each time step as incoming requests, a given policy π is said to be stable if, for a fixed fleet size of m taxis, the expected number of outstanding requests is uniformly bounded. Hence, a policy π is stable as long as the distance to be traveled in service of all the requests that enter the system according to policy π is less than or equal to the total distance that can be covered by a fleet of m taxis. In other words, for a policy π to be stable (following a similar argument as in [20]), the expected total distance for servicing all requests should be upper bounded by the distance covered by taxis, i.e., $E[Z_{\pi,T}] \leq m \cdot T$.

E. Challenges of a large scale multi-agent problem

We are interested in learning a cooperative pickup and routing policy on a city-scale map that minimizes the total wait time for all requests over a finite horizon of length T. We denote the state transition function as f, such that $x_{t+1} = f(x_t, u_t, \eta, \rho, \delta)$, where x_{t+1} is the resulting state after control $u_t \in \mathbf{U}_t(x_t)$ has been applied from state x_t. We define the stage cost $g_t(x_t, u_t, \eta, \rho, \delta) = |\overline{\mathbf{r}}_\mathbf{t}|$ as the number of outstanding requests at time t. We denote the cost of executing policy π from initial state x_1 as $J_\pi(x_1) = E\left[g_T(x_T) + \sum_{t=1}^{T-1} g_t(x_t, \mu_t(x_t), \eta, \rho, \delta)\right]$, where $g_T(x_T) = |\overline{\mathbf{r}_\mathbf{T}}|$ is the terminal cost. Since the control space for the problem grows exponentially with the number of taxis, obtaining an optimal policy through the Bellman equations is intractable. For this reason, we consider policy improvement schemes, such as one-at-a-time rollout [13], [14], which solve several smaller lookahead optimizations to obtain a lower cost policy that improves upon a base policy and has a control space that scales linearly with the number of taxis instead of exponentially. We define base policy $\pi = \{\mu_1, \dots \mu_T\}$ as an easy to compute heuristic that is given. One-at-a-time rollout finds an approximate policy $\tilde{\pi} = \{\tilde{\mu}_1, \dots \tilde{\mu}_T\}$, where $\tilde{\mu}_t(x_t) = (\tilde{\mu}_t^1(x_t), \dots, \tilde{\mu}_t^m(x_t))$, $t = [1, \dots, T]$. For state x_t, $\tilde{\mu}_t$ is found by solving m minimizations for $\ell \in [1, \dots, m]$ as follows:

$$\tilde{\mu}_t^\ell(x_t) \in \underset{u_t^\ell \in \mathbf{U}_t^\ell(x_t)}{\operatorname{argmin}} E[g_t(x_t, u_t, \eta, \rho, \delta) + \tilde{J}_{\pi,t+1}(x_{t+1})], \tag{1}$$

where $u_t = (\tilde{\mu}_t^1(x_t) : \tilde{\mu}_t^{\ell-1}(x_t), u_t^\ell, \mu_t^{\ell+1}(x_t) : \mu_t^m(x_t))$, and $\tilde{J}_{\pi,t+1}(x_{t+1}) = |\overline{\mathbf{r}}_{\mathbf{t+1+t_h}}| + \sum_{t'=t+1}^{t+t_h} g_{t'}(x_{t'}, \mu_{t'}(x_{t'}), \eta, \rho, \delta)$ is a cost approximation derived from t_h applications of the base policy π from state x_{t+1}, with a terminal cost approximation $|\overline{\mathbf{r}}_{\mathbf{t+1+t_h}}|$.

To apply one-at-a-time rollout to a large city-scale problem, we design an algorithm that approximates this rollout scheme, but incurs a lower computational cost that satisfies user defined computational constraints. Our algorithm is given in Sec. III. We find a sufficiently large fleet size m for which a reasonable base policy π is stable, such that $E[Z_{\pi,T}] \leq m \cdot T$, as defined in Sec. II-D. In particular, we are interested in the stability of the policy π_{base} associated with IA-RA, as this policy is 2-competitive [16] and hence our approximate rollout approach obtains a near-optimal policy.

III. Approximation algorithm for multiagent rollout

In this section we propose an approximate algorithm for multiagent rollout (see Eq. 1). Our proposed method is composed of a two-phase planning scheme that reduces the computational cost of one-at-a-time rollout through partitioning of the map using the demand distribution. We take into account user defined computational constraints in the form of the maximum number of taxis that can be run by one-at-a-time rollout in each sector m_{lim}, and the length of the planning horizon t_h (longer planning horizon result in longer runtimes). The algorithm is detailed in Algorithm 1. The proposed two-phase algorithm also takes as input m the total number of taxis in the fleet. We provide theoretical bounds on m in Sec. IV and calculated values in practice in Sec. V-C.

The first routine in Algorithm 1 is denoted as $get\_partitions$ and it places the center of each partition on the map. $get\_partitions$ solves a capacitated facility location problem [21], where the capacity for each partition center is set to be m_{lim}, and then the expected number of requests for the ride service during the entire time horizon is used as the demand. The $get\_partitions$ routine then assigns each node to the closest partition center using weighted k-means, where the weights of the nodes are given by the probability distribution of pickups. This routine guarantees that the size of each partition is inversely proportional to the density of requests.

At each time-step, the $High\_level\_planner$ re-balances the taxis between partitions using an instantaneous assignment of taxis to current and expected future requests for the next t_h time-steps as given by a certainty equivalence approximation. It returns the controls for taxis that are expected to go across regions u_h^g, as well as the list of high level taxis $\hat{m}$, and $\hat{d}$ the set of locations for the high level taxis to move towards. The $Low\_level\_planner$, on the other hand, plans for routing and pickup controls for

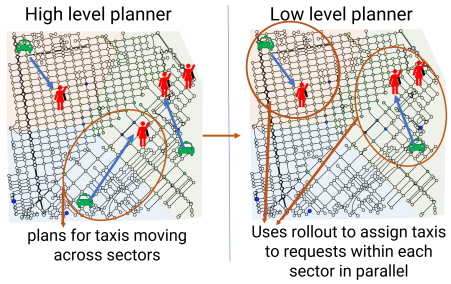

Fig. 1. Our two phased approach executed on a map with 3 sectors.

taxis that remain in their original sectors according to the high level planner. The *Low_level_planner* executes one-at-a-time rollout with base policy IA-RA as defined in Eq. 1 to obtain $\tilde{u}_t^k$ the control of taxis in sector k at time t.

After partitioning the graph, the state x_t consists of K sub-states $\{x_t^k\}_{k=1}^K$, one corresponding to a partition $k \in \{1,\dots,K\}$ of the graph. The state transition of partition k is given by, $x_{t+1}^k = f^k(x_t^k, u_t^k, u_h^g(t,k), \eta, \rho, \delta)$. The control u_t can be separated as $\{u_t^k, u_h^g(t,k)\}_{k=1}^K$, where the control component u_t^k corresponds to the taxis that are local to partition k. The control component $u_h^g(t,k)$ corresponds to the controls of taxis coming into partition k at time t as given by the higher level planner. Since we consider the length of outstanding requests as the stage cost, we have $|\bar{\mathbf{r}}_\mathbf{t}| = g_t(x_t, u_t, \eta, \rho, \delta) = \sum_{k=1}^K g_t^k(x_t^k, u_t^k, u_h^g(t,k), \eta, \rho, \delta) = \sum_{k=1}^K |\bar{\mathbf{r}}_\mathbf{t}^\mathbf{k}|$, where $\bar{\mathbf{r}}_\mathbf{t}^\mathbf{k} \subseteq \bar{\mathbf{r}}_\mathbf{t}$, and $\forall r \in \bar{\mathbf{r}}_\mathbf{t}^\mathbf{k}, \rho_r \in s_k$. The cost of our two-phase policy π_{2P} is given by

$$J_{\pi_{2P}}(x_1) = E[\sum_{t=1}^{T}\sum_{k=1}^{K} g_t^k(x_t^k, \tilde{u}_t^k, u_h^g(t,k), \eta, \rho, \delta)]$$

Algorithm 1: Two-phase Planner

Input: Initial state x_1, maximum number of taxis per sector m_{lim}, fleet size m, planning horizon t_h
Output: policy π_{2P} that gives routing/pickup strategy for all taxis in the system

$K \leftarrow \frac{m}{m_{\text{lim}}}$
$\{s_k\}_{k=1}^K \leftarrow get\_partitions(m_{\text{lim}}, K, G, \eta, \rho, \delta)$
$\hat{d} \leftarrow \{\}, \hat{m} \leftarrow []$
for *each time planning step* $t \in [1,\dots,T]$ **do**
  $u_h^g, \hat{m}, \hat{d} \leftarrow High\_level\_planner(x_t, \eta, \rho, \delta, \pi_{\text{base}}, t_h, \hat{m}, \hat{d}, \{s_k\}_{k=1}^K)$
  for *each sector* $s_k, k \in \{1,\dots,K\}$ *in parallel* **do**
    $\tilde{u}_t^k \leftarrow Low\_level\_planner(x_t^k, u_h^g(t,k), \eta, \rho, \delta, \pi_{\text{base}}, t_h, \hat{m}, \hat{d})$
  set $\mu_{2P,t}(x_t) = \{\tilde{u}_t^k, u_h^g(t,k)\}_{k=1}^K$
  $x_{t+1} \sim f(x_t, \mu_{2P,t}(x_t), \eta, \rho, \delta)$
set $\pi_{2P} = \{\mu_{2P,t}\}_{t=1}^T$
return π_{2P}

Figure 1 shows the two phased approach with an example with 3 taxis and 4 outstanding requests.

IV. Theoretical Results

In this section, we provide a sufficient condition for choosing a fleet size m that will make the policy π_{base}, instantaneous assignment with reassignment (IA-RA) at each time step, a stable policy. We also provide an asymptotic necessary condition on m for the stability of π_{base} as $T \to \infty$. **Due to space constraints we provide proof ideas for all the results. Complete proofs can be found in our extended technical report [22].**

A. Sufficient condition for stability of π_{base}

We are interested in finding the sufficient conditions on the fleet size m that guarantee the stability of policy π_{base} such that the relation $E[Z_{\pi_{\text{base}},T}] \leq m \cdot T$ always holds. To do so, we first analyze the policy $\hat{\pi}$ referred to as random instantaneous assignment, where taxis are randomly assigned to requests. Under this policy, a taxi does not move until it has been assigned to a request. Once a taxi is assigned to a request, the taxi cannot be assigned to other requests until it has serviced the originally assigned request. By having a random assignment of requests to taxis, $l_{r_q,\hat{\pi}}$ for an arbitrary request r_q becomes a random variable instead of a deterministic function of the requests in the system and the locations of all the taxis. The randomness in $\hat{\pi}$ also makes the request's pickup location ρ_{r_q} and the location of the taxi assigned to the request $l_{r_q,\hat{\pi}}$ independent, making the analysis easier. Using this policy, we can find an upper bound on $E[Z_{\hat{\pi},T}]$, and choose m such that $m \cdot T$ is greater than or equal to the upper bound, making $\hat{\pi}$ a stable policy by definition. We then show that the IA-RA policy π_{base} where the assignment is given by a matching algorithm, like the auction algorithm [23] or the modified JVC algorithm [24], results in a smaller service distance than $\hat{\pi}$, i.e., $Z_{\pi_{\text{base}},T} \leq Z_{\hat{\pi},T}$. This implies $E[Z_{\pi_{\text{base}},T}] \leq E[Z_{\hat{\pi},T}] \leq m \cdot T$, and hence π_{base} constitutes a stable policy for the sufficiently large fleet size m found in the analysis of the stability of $\hat{\pi}$. We present the formal claim for the sufficient conditions on m for the stability of $\hat{\pi}$ below in the following lemma.

Lemma 1: Let the random variable l_{rand} with support V represent the location of the random taxi that gets assigned to a request after that taxi has previously served a different request. Define $D_{\max} \triangleq \max\{E[d(\xi,\rho)], E[d(l_{\text{rand}},\rho)]\} + E[d(\rho,\delta)]$. If the fleet size m satisfies $m \geq E[\eta] \cdot D_{\max}$, then the policy associated with a random instantaneous assignment of taxis to requests, $\hat{\pi}$, constitutes a stable policy such that $E[Z_{\hat{\pi},T}] \leq m \cdot T$.

The proof idea goes as follows. First, we split the total distance required to service requests in two cases: 1) the distance associated with taxis that are servicing their first request, and 2) the distance associated with taxis that have already serviced at least one request. Since $\hat{\pi}$ is a random assignment of taxis to requests, the location of the assigned taxis and the pickup of the corresponding request are independent. Moreover, within these two cases, they are identically distributed. Using this, we upper bound the expected distance for these two cases by $D_{\max}$. We then combine the two cases to obtain the claim of the lemma.

Notice that all the terms given in $D_{\max}$ can be calculated in practice using historical data. We use the result from lemma 1 to show that the same m chosen to guarantee stability of $\hat{\pi}$ serves as a sufficiently large m to guarantee stability of π_{base} which is formalized in Theorem 1.

Theorem 1: Assume that the fleet size m satisfies the condition given in lemma 1. Then the policy π_{base}, which corresponds to instantaneous assignment with reassignment (IA-RA) at each time step, is a stable policy such that $E[Z_{\pi_{\text{base}},T}] \leq m \cdot T$, for a finite horizon $T > 0$.

The proof sketch is as follows: first we define $\bar{\pi}$ as the policy associated with instantaneous assignment (IA) with commitment to the initial assignment and then we show that with $\hat{\pi}$, the distance traveled per assigned request is at least as long as that with $\bar{\pi}$. This follows directly from the definition of $\bar{\pi}$, since IA produces a match of taxis to requests that minimizes the distance between the assigned taxis and their respective requests. We then show that $\bar{\pi}$ results in longer or equal distance traveled per assigned request than π_{base}. This follows directly from the structure of the reassignment, which only happens if the distance associated with the new assignment is the minimum distance of all possible assignments at that time step. Finally, since we know that $\hat{\pi}$ is stable for a fleet size m as given in lemma 1, then we can conclude that π_{base} is also stable as π_{base} results in a smaller or equal distance traveled.

B. Necessary condition for stability of π_{base}

We are interested in finding the necessary condition for stability of policy π_{base} asymptotically as $T \to \infty$. For this reason, we want to find a lower bound on $E[Z_{\pi_{\text{base}},T}/T]$. Choosing a fleet size m smaller than this bound would make the policy π_{base} asymptotically unstable, i.e., $E[Z_{\pi_{\text{base}},T}] > m \cdot T$ as $T \to \infty$. To do so, we first find a lower bound for $E[Z_{\pi_{\text{base}},T}/T]$, the expected travel distance associated with servicing the requests that enter the system per time step, and then we apply a limit as $T \to \infty$ to obtain an expression for the asymptotic lower bound. The following theorem states this result formally.

Theorem 2: Let $WD(p_\delta, p_\rho)$ denote the first Wasserstein distance [25] between probability distributions p_δ and p_ρ with support Ω, such that:

$$WD(p_\delta, p_\rho) = \inf_{\gamma \in \Gamma(p_\delta, p_\rho)} \int_{x,y \in \Omega} ||y - x|| d\gamma(x, y)$$

Where $||\cdot||$ is the euclidean metric, and $\Gamma(p_\delta, p_\rho)$ is the set of measures over the product space $\Omega \times \Omega$ having marginal densities p_δ and p_ρ, respectively. Define $D_{\min} \triangleq WD(p_\delta, p_\rho) + E[d(\rho, \delta)]$. Assume that the random variables for pickups ρ and drop-offs δ are independent and we have a fleet of size $m < E[\eta] \cdot D_{\min}$. Then, the policy π_{base}, which corresponds to instantaneous assignment with reassignment (IA-RA) at each time step, is asymptotically unstable, i.e., $E[Z_{\pi_{\text{base}},T}] > m \cdot T$ as $T \to \infty$.

The proof sketch is as follows: First, we lower bound the expected distance required to service a request using π_{base} by the sum of the expected distance between the pickup and the dropoff of the request and the average distance associated with the solution for the bipartite matching problem. We then lower bound the average distance associated with the solution for the bipartite matching problem with the average distance for the solution of the Euclidean bipartite matching problem. Finally, after applying the limit as $T \to \infty$ and using the results presented in [19], we lower bound the average distance of the solution to the euclidean bipartite matching problem by $WD(p_\delta, p_\rho)$ and obtain the claim in the theorem.

V. Numerical studies

In this section we evaluate the performance of our algorithm using a real taxi data set for the city of San Francisco [26]. We compare the performance of our algorithm against three benchmarks: a greedy policy, instantaneous assignment with reassignment (IA-RA), and a rollout-based algorithm over the entire map as proposed in [12]. We provide a comparison of run-time of our two-phase approach and the rollout-based approach [12] to empirically verify the reduction in run-time associated with our two-phase approach. We verify our theoretical results in the number of taxis in the fleet required for stability by executing our algorithm for larger time horizons and plotting the number of outstanding requests at each time step. We empirically verify that for m chosen in the range given by Theorem 1, and Theorem 2, our proposed approach is stable in the sense that the number of outstanding requests is uniformly bounded over time.

A. Experimental Setup

Our numerical results consider a section of $1500m \times 1500m$ in San Francisco with 859 nodes and 1959 edges. For the comparison studies we consider a horizon length of $T = 60$, while for the stability results we consider $T = 180$. All experiments were executed in an AMD Threadripper PRO WRX80. All individual results correspond to an average over 20 different trials with different instantiations of the random variables.

B. Estimating probability distributions

For our experiments, we estimate $\tilde{p}_\eta$, $\tilde{p}_{\rho|\eta}$, and $\tilde{p}_\delta$ using historical trip data from several taxis in San Francisco [26]. We divide the historical data in 1-hour intervals, where each time step t spans 1 minute. We empirically estimate $\tilde{p}_\eta$ by using the number of requests that arrive at each time step within each 1-hour time span. The distributions $\tilde{p}_\rho$ and $\tilde{p}_{\delta|\rho}$ are derived from the relative frequency of historical requests that originated and ended inside the map.

C. Calculated values for theoretical results

For our experiments, we consider $\tilde{p}_\eta$ for an hour in which $E[\eta] = 1$ (we get around 60 requests per hour). For simplicity, we assume that ξ is distributed according to the marginal probability distribution p_δ, and hence we find that $E[d(\xi, \rho)] \approx 15$. We use $p_{l_{\text{rand}}}$ and p_ρ to calculate $E[d(l_{\text{rand}}, \rho)] \approx 13$. We use p_ρ and $p_{\delta|\rho}$ to calculate $E[d(\rho, \delta)] \approx 15$. From this we get that the sufficient number of taxis for stability of our two-phase approach is $m \geq \max(15, 13) + 15 = 30$ from Theorem 1. We approximate the Wasserstein distance $WD(\delta, \rho)$ using the procedure suggested in [20]. We obtain $WD(\delta, \rho) \approx 1.87$.

From this, we get that asymptotically, the minimum number of taxis needed for stability as $T \to \infty$ is $m > 1.87 + 15$, rounding to next integer $m \geq 17$ (Theorem 2).

D. Implementation details for two-phase approach

We execute 2000 MC simulations with certainty equivalence to approximate the expected cost associated with each potential control in the one-step lookahead step of the rollout for the local planner. We also consider a planning horizon $t_h = 10$ for the rollout, and a capacity of $m_{\text{lim}} = 10$ taxis per sector, based on the computational resources available.

E. Benchmarks

In this section, we discuss the details of the benchmarks to be used as comparisons for our performance results.

Greedy policy: Each taxi moves towards its closest request without coordinating with other taxis. This method does not consider future demand.

Instantaneous assignment (IA-RA): It solves a matching problem between available taxis and outstanding requests at every time step using an auction algorithm [27], [28]. This method does not consider future demand.

One-at-a-time rollout-based global routing: performs rollout over the entire map using the procedure described in the scalability section of [12]. We set the planning horizon to $t_h = 10$ as suggested in the paper. We run the same number of MC simulations as with our approach. This method considers expected future demand.

F. Performance results

This section includes the results for the performance and the execution time of our two-phase approach.

As shown in Fig. 2, our method results in a comparable performance to the rollout-based global routing [12]. For lower number of taxis, when $m < 17$, our method is unstable. After we surpass 17 taxis, standard IA-RA starts being stable and performs similarly to the rollout-based global routing [12]. As shown in the graphs, for $m \geq 30$, our proposed method results in a lower cost than IA-RA, resulting in a 5% to 18% improvement, sometimes even outperforming the rollout-based global routing thanks to the smaller sampling space associated with each sector. Since both rollout-based methods are running the same number of MC simulations, a smaller sample space leads to better approximation of the expectation in Eq.(1).

To better understand the advantages of our proposed method, we compare the execution time of our proposed two-phase approach to the rollout-based global routing.

Fig. 3 shows our method results in significantly lower runtimes than the rollout-based global routing. Execution time for our method still grows linearly with the number of taxis, but with a smaller slope. This shows that partitioning the map and solving sub-problems in parallel results in a faster execution with similar wait time compared to one-at-a-time rollout over the entire map.

G. Stability of two-phase approach

Fig. 4 shows the stability results of our two-phase approach with various number of taxis over a horizon of 3 hours. Without enough taxis, $m < 17$, for which IA-RA is shown to be unstable, our approach shows an increasing number of outstanding requests over time. However, with sufficient numbers of taxis ($m = 25, 35$), we see that both the IA-RA policy and our two-phase approach has a bounded number of outstanding requests over a large horizon of 180 minutes.

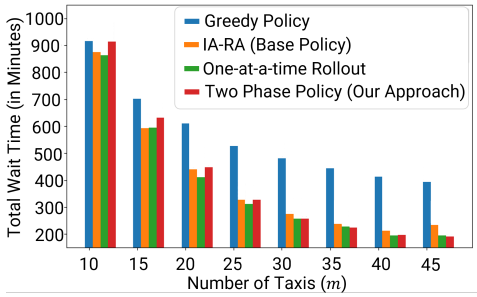

Fig. 2. Total wait time over all requests of our two-phase approach and the benchmarks.

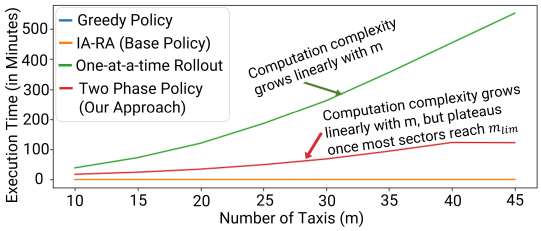

Fig. 3. Execution time comparison between our two-phase approach and the benchmarks.

VI. Conclusion

In this paper, we propose an approximation algorithm that allows us to apply one-at-a-time rollout to a large scale urban environment. We provide a necessary and a sufficient conditions for the total fleet size m to make the instantaneous assignment base policy stable, which is key to guarantee rollout's convergence to a near-optimal policy. We also verify this results in simulation with a real dataset [26]. As future work, we plan on relaxing the assumption of unit travel time for the taxis in the fleet. Even though the algorithm would still work for this more realistic setting, we need to derive new theoretical results to take into account this change.

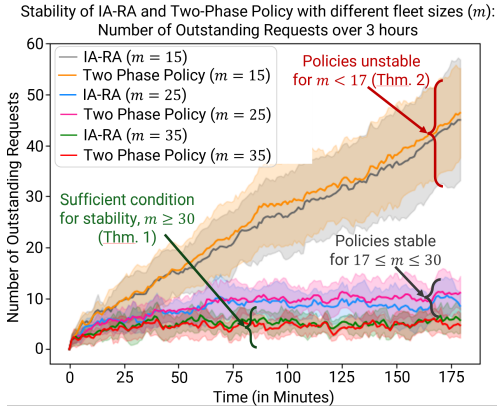

Fig. 4. Stability of IA-RA and two-phase policy in terms of the means (lines) and standard deviations (shaded regions) of the number of outstanding requests.

REFERENCES

[1] N. Bidarian, "Regulators give green light to driverless taxis in san francisco," *CNN*, 2023. [Online]. Available: https://www.cnn.com/2023/08/11/tech/robotaxi-vote-san-francisco/index.html

[2] J. Muller, "Robotaxis hit the accelerator in growing list of cities nationwide," *Axios*, 2023. [Online]. Available: https://www.axios.com/2023/08/29/cities-testing-self-driving-driverless-taxis-robotaxi-waymo

[3] D. Kondor, I. Bojic, G. Resta, F. Duarte, P. Santi, and C. Ratti, "The cost of non-coordination in urban on-demand mobility," *Scientific Reports*, vol. 12, 03 2022.

[4] G. Berbeglia, J.-F. Cordeau, and G. Laporte, "Dynamic pickup and delivery problems," *European Journal of Operational Research*, vol. 202, no. 1, pp. 8–15, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221709002999

[5] R. Duan and S. Pettie, "Linear-time approximation for maximum weight matching," *J. ACM*, vol. 61, no. 1, 2014. [Online]. Available: https://doi.org/10.1145/2529989

[6] D. Bertsimas, P. Jaillet, and S. Martin, "Online vehicle routing: The edge of optimization in large-scale applications," *Oper. Res.*, vol. 67, pp. 143–162, 2019.

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

[8] M. Lowalekar, P. Varakantham, and P. Jaillet, "Online spatio-temporal matching in stochastic and dynamic domains," *Artificial Intelligence*, vol. 261, pp. 71–112, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370218302030

[9] R. Iglesias, F. Rossi, K. Wang, D. Hallac, J. Leskovec, and M. Pavone, "Data-driven model predictive control of autonomous mobility-on-demand systems," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6019–6025.

[10] D. Gammelli, K. Yang, J. Harrison, F. Rodrigues, F. Pereira, and M. Pavone, "Graph meta-reinforcement learning for transferable autonomous mobility-on-demand," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2913–2923.

[11] T. Enders, J. Harrison, M. Pavone, and M. Schiffer, "Hybrid multi-agent deep reinforcement learning for autonomous mobility on demand systems," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 1284–1296.

[12] D. Garces, S. Bhattacharya, S. Gil, and D. Bertsekas, "Multiagent reinforcement learning for autonomous routing and pickup problem with adaptation to variable demand," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2023.

[13] D. Bertsekas, "Multiagent reinforcement learning: Rollout and policy iteration," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 249–272, 2021.

[14] ——, *Rollout, Policy Iteration, and Distributed Reinforcement Learning*, ser. Athena scientific optimization and computation series. Athena Scientific., 2020. [Online]. Available: https://books.google.com/books?id=Hbo-EAAAQBAJ

[15] ——, *Lessons from AlphaZero for Optimal, Model Predictive, and Adaptive Control*. Nashua, NH, USA: Athena Scientific, 2022.

[16] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International journal of robotics research*, vol. 23, no. 9, pp. 939–954, 2004.

[17] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: A queueing-theoretical perspective," *The International Journal of Robotics Research*, vol. 35, no. 1–3, p. 186–203, jan 2016. [Online]. Available: https://doi.org/10.1177/0278364915581863

[18] M. Vazifeh, P. Santi, G. Resta, S. Strogatz, and C. Ratti, "Addressing the minimum fleet problem in on-demand urban mobility," *Nature*, vol. 557, 05 2018.

[19] K. Treleaven, M. Pavone, and E. Frazzoli, "Asymptotically optimal algorithms for one-to-one pickup and delivery problems with applications to transportation systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2261–2276, 2013.

[20] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone, "Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in singapore," *Road Vehicle Automation. Lecture Notes on Mobility*, pp. 229–245, 04 2014.

[21] L.-Y. Wu, X.-S. Zhang, and J.-L. Zhang, "Capacitated facility location problem with general setup cost," *Computers & Operations Research*, vol. 33, no. 5, pp. 1226–1241, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054804002357

[22] D. Garces, S. Bhattacharya, D. Bertsekas, and S. Gil, "Approximate multiagent reinforcement learning for on-demand urban mobility problem on a large map (extended version)," 2023. [Online]. Available: https://arxiv.org/abs/2311.01534

[23] D. Bertsekas, "Constrained multiagent rollout and multidimensional assignment with the auction algorithm," 2020. [Online]. Available: https://arxiv.org/abs/2002.07407

[24] D. F. Crouse, "On implementing 2d rectangular assignment algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1679–1696, 2016.

[25] L. Rüschendorf, "The wasserstein distance and approximation theorems," *Probability Theory and Related Fields*, vol. 70, no. 1, pp. 117–129, 1985.

[26] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD dataset epfl/mobility (v. 2009-02-24)," Downloaded from https://crawdad.org/epfl/mobility/20090224, Feb. 2009.

[27] D. Bertsekas, "A distributed algorithm for the assignment problem," *Lab. for Information and Decision Systems Report*, 05 1979.

[28] ——, *Network Optimization: Continuous and Discrete Models*, ser. Athena scientific optimization and computation series. Athena Scientific, 1998. [Online]. Available: https://books.google.com/books?id=qUUxEAAAQBAJ