Hierarchical Multiagent Reinforcement Learning for Maritime Traffic Management

Arambam James Singh
Singapore Management University
arambamjs.2016@smu.edu.sg

Akshat Kumar
Singapore Management University
akshatkumar@smu.edu.sg

Hoong Chuin Lau
Singapore Management University
hclau@smu.edu.sg

ABSTRACT

Increasing global maritime traffic coupled with rapid digitization and automation in shipping mandate developing next generation maritime traffic management systems to mitigate congestion, increase safety of navigation, and avoid collisions in busy and geographically constrained ports (such as Singapore's). To achieve these objectives, we model the maritime traffic as a large multiagent system with individual vessels as agents, and VTS (Vessel Traffic Service) authority as a regulatory agent. We develop a hierarchical reinforcement learning approach where vessels first select a high level action based on the underlying traffic flow, and then select the low level action that determines their future speed. We exploit the nature of collective interactions among agents to develop a policy gradient approach that can scale up to large real world problems. We also develop an effective multiagent credit assignment scheme that significantly improves the convergence of policy gradient. Extensive empirical results on synthetic and real world data from one of the busiest port in the world show that our approach consistently performs significantly better than the previous best approach.

ACM Reference Format:
Arambam James Singh, Akshat Kumar, and Hoong Chuin Lau. 2020. Hierarchical Multiagent Reinforcement Learning for Maritime Traffic Management. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

Recent study by United Nations on maritime transport shows that global port activity and cargo handling capacity have expanded rapidly over the years [22]. Furthermore, rapid digitization and automation are transforming both shipping and port operations for enhanced performance, sustainable operations and safety. Technologies such as e-Navigation [14] aim to enhance the safety of maritime navigation by digitizing both on-board and shore-based operations, and automating communication among vessels and vessel traffic services (VTS). Furthermore, autonomous ships are on the horizon that promise to further enhance safety and reduce cost by removing the human element from certain operations, and allow for more efficient use of space in ship design and fuel efficiency [22, 23, 28]. Despite such advances, a key bottleneck remains—that of limited navigable space in some of the busiest port waters such as Singapore strait. Unlike air and road traffic which can expand the network capacity, navigable sea space in busy ports remains inherently limited by geographical features, and constantly under pressure due to land reclamation [29]. Given such hard resource constraints, increasing maritime traffic requires developing new traffic management systems that can effectively coordinate vessel movements to maintain safety of navigation, avoid near miss situations and reduce collision risks in busy port waters by exploiting improved digitization and automation in the maritime ecosystem. Our work precisely addresses such challenges.

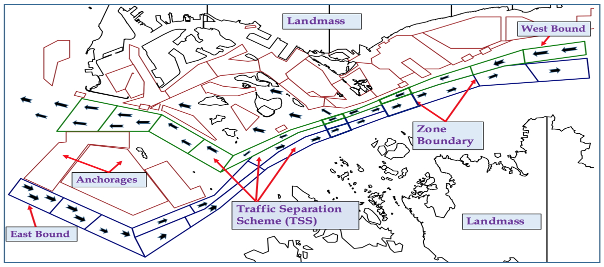


Figure 1: Navigation map of Singapore Strait overlayed on Electronic navigation chart (ENC) with color-coded features (best viewed electronically)

As a case study, we focus on Singapore strait, which is one of the busiest shipping areas in the world, providing the shortest route between the Indian Ocean and the South China Sea. This makes it a popular route for oil tankers and cargo ships [10, 19]. Figure 1 shows the navigation map of the strait. Our focus is on the *traffic separation scheme* (TSS), which is the key maritime highway including a set of mandatory one-direction routes designed to reduce collision risk among vessels either transitioning through the strait or entering *fairway* which leads vessels to their destinations such as *berths* or *anchorage* area. The TSS can be further divided into smaller *zones* as shown in figure 1. Even though we focus on the Singapore strait, our work is applicable to other busy ports as well which have their own TSS as mandated by the International Maritime Organization [15].

Traffic control system: The current traffic is regulated by the VTS authority in TSS. VTS officers continuously monitor the traffic, and provide advices and warnings to vessels which are likely to be involved in a near-miss (a high risk proximity situation) or create hotspots in the near future (10-15 minutes). The current system does not proactively regulate the traffic to avoid forming hotspots in the first place. Our goal is to develop a multiagent traffic control system that provides vessels a recommended duration to cross each *zone* in the TSS based on the prevailing traffic conditions such that the traffic intensity is within a specified limit in each zone, and maximizing traffic throughput while maintaining safety of navigation. A multiagent control is highly desirable as each vessel (or an agent) is controlled by a different ship master, with VTS authority being

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand. © 2020 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

the regulatory agent(RA). A policy that can be executed by vessels in a decentralized fashion is key to the practical adoption of our techniques. Our model assumes a collaborative setting with navigation safety as the common interest shared by all vessels, along with the RA. It is noted that in practice, a regulatory authority does not prefer to provide direct navigation guidance to vessels, as this can lead to liability issues (e.g., vessels collide even after following RA's guidance). Due to this, we follow a *centralized-learning* and *decentralized-execution* paradigm. Vessels can execute the learned decentralized policies based only on their local observations without requiring direct feedback from the RA. The RA uses a central simulator to train decentralized policies, which ensures that RA can reliably evaluate the joint-policy as it has global view during the learning phase. As a result, learning converges faster, and is stable as opposed to each vessel learning in a decentralized fashion. Thus, the role of RA is critical during the learning phase.

We also highlight challenges that make maritime traffic control different than road traffic. Unlike the road traffic, movement of vessels is highly dynamic in nature due to vessel condition and weather factors. Therefore, we need to model uncertainty in the movement of vessels. There are no traffic lights in port waters; vessels cannot stop completely while in-transit, they must maintain a minimum cruising speed and have a maximum speed limit. Furthermore, any traffic control strategy must scale to a large number of agents as hundreds of vessels navigate through TSS each day.

Our contributions: Motivated by the insight that road traffic can be categorized into 5-6 qualitative classes, called *level of service* (ranging from free flow to traffic breakdown) [27] to convey the congestion level of the road traffic, we envision that maritime traffic can similarly be described using multiple level of services (LoS). Corresponding to the prevailing traffic, a suitable LoS can be determined, and agents can take the best action for the chosen LoS. However, accurately mapping traffic to a particular LoS is challenging. We therefore develop a hierarchical reinforcement learning approach that first learns a policy over high-level actions (each such *meta action* can be thought of as corresponding to a LoS) directly from data generated using a simulator. Each meta action maps to the low level policy that tells a vessel the recommended duration to cross a zone. We optimize both high-level and low-level policies using the policy gradient method.

Our approach is scalable to realistic instances with hundreds of agents, and is executable in a decentralized setting where agents only observe traffic in their local neighborhood. Standard policy gradient is very slow to converge due to the presence of large number of agents resulting in high variance of gradient estimates. We therefore also develop a *multiagent credit assignment* method that accurately determines the contribution of each meta action to the overall traffic management objective. We empirically test on several synthetic instances, and real-world data which consists of all vessel movements in Singapore strait over a 6 months period (consisting of more than 14 million unique position records). We show that our hierarchical policy gradient approach significantly outperforms the previous best method [30] consistently, providing about 30%-40% improvement in solution quality in several settings.

Related Work: Several existing works address maritime traffic management. Expert system and rule-based traffic modeling techniques have been presented in [12, 13, 16]. Mathematical programming based methods exist to optimize efficiency of port operations [18]. Scheduling methods are developed to increase traffic efficiency while maintaining safety of navigation [1, 6, 36]. Multiagent path finding is used to re-route vessels that are in close quarter situation [33]. However, these approaches do not model the uncertainty present in environment, and often require a centralized control. In contrast, our approach can model the uncertainty and partial observability in the maritime domain, and provides decentralized policies.

Closely related to our approach is the policy gradient approach for maritime traffic management in [30]. The key difference in our solution strategy is the use of meta actions and optimizing a policy over them. Using such meta actions enables better exploration of the state-space while learning policies. As a result, our approach gives significantly better solution quality than [30] over a range of synthetic and real-world problems.

Our work is motivated by hierarchical reinforcement learning (HRL) [2, 5, 26, 32], a framework for control with temporally extended actions. High level actions can take variable amount of time to complete, unlike primitive actions which are executed at every time step. One key benefit of HRL is structured exploration, i.e exploration using higher level actions rather than just primitive actions. However, existing approaches that extend HRL to multiagent systems are limited in scalability to a few agents [35]. In contrast, our approach exploits the fact that vessels in maritime traffic can be considered homogenous (or belonging to a few types) affecting each other only via their *collective presence* (such as congestion). Exploiting such collective nature of interactions enables scalability to large number of agents.

2 MODEL DEFINITION

We use a similar traffic control model used in [30] with the addition of meta actions. In practice, vessels in port waters may belong to different types such as ferries, barges, pilot boats among others. However, our primary focus is on tankers and cargo vessels which are the largest vessel in size. Alleviating congestion for such vessels is critical as due to their size, they are much less maneuverable than smaller vessels, and can quickly result in unsafe situations without proper traffic management.

We assume a total of M vessels, and a planning horizon H. The planning horizon can coincide with the peak traffic window, or our approach can be used in a rolling window basis. Our main area of interest is TSS where majority of the traffic activity occurs as vessels enter and leave the port through TSS. The TSS is divided into unidirectional zones $z \in Z$ to ensure traffic safety as shown in figure 1. A directed acyclic graph is an input for the model where nodes are zones and edges represent the traffic flow.

Zone classification: The zone set Z is further categorized as follows. We introduce a *dummy zone* $z_d \in Z$ which contains all the future vessels that will arrive in the port waters within the planning window. *Source zones* $z \in Z_{\text{src}} \subset Z$ are zones from where vessels enter the TSS. There can be multiple source zones. E.g., zones adjacent to the extreme ends of TSS where new vessels enter; vessels in berths re-enter TSS through intersection zones (where traffic from TSS can flow towards berths and vice-versa) after cargo loading and unloading; similarly for vessels in anchorages. The dummy zone z_d sends vessels at different time periods to source zones (vessel arrival distribution can be learned from the data).

Terminal zones $z \in Z_{\text{ter}} \subset Z$ represent zones such as berth, anchorages or port water boundary. After entering the TSS, some vessels head directly to berths, some enter anchorage area waiting for berth spots, and some vessels transit through the port at the other end without entering to berth or anchorage.

Planning zones $z \in Z$ are the main zones for which we optimize the travel time to control congestion and delay.

Vessel model and state-space: Let the state of a vessel m at time t be denoted using s_t^m. The vessel can be *newly arrived* at zone z, or *in-transit* through zone z given that navigating a zone may take multiple time steps.

- For in-transit state, we define $s_t^m = \langle z, z', \tau\rangle$, where $z \in Z$ is current zone vessel is transiting through, $z' \in Z$ is the next zone vessel is heading to, and τ is time remaining to reach z'.
- For newly arrived state, we define $s_t^m = \langle z, \emptyset, \emptyset\rangle$, where $z \in Z$ is the new zone vessel just entered. For vessels in such state, next zone z', and τ, the time-to-next zone, are not yet decided.

Vessel observation: Based on its local state s_t^m and the global state $\mathbf{s}_t^m$, vessel m receives the observation o_t^m. If vessel m is in zone z, and $\mathrm{n}_t^{\mathrm{tot}} = \langle \mathrm{n}_t^{\mathrm{tot}}(z) \forall z\rangle$ be the count table representing total number of vessels present in different zones (we show how to compute it later), then agent m's observation is $o(z, \mathrm{n}_t^{\mathrm{tot}})$. Typically, in a partially observable setting, this observation corresponds to the counts of all vessels in zone z and local neighborhood of z. Such observation is easily obtained using the on-board radars in vessels.

Vessel decision making: When a vessel m is newly-arrived at a zone z, it needs to take two actions—*direction action* a_t^m to decide which zone z' to go to next; and *navigation action* ω_t^m to decide how much time to take to navigate to z'. We describe them next.

When vessel m is newly arrived at zone z (or $s_t^m = \langle z, \emptyset, \emptyset\rangle$), it samples its next zone $a_t^m = z'$ from the distribution $\alpha(z'|z)$. In several ports, destinations vessels are headed to are limited (e.g., berths, anchorages, transit-through). Furthermore, for large vessels such as tankers and cargos (which are our focus), spatial movement is restricted to only deep water routes. Therefore, to optimize the average traffic flow, we assume $\alpha(z'|z)$ as an input parameter which can be learned from the data, and do not optimize such spatial direction decision of vessels, similar to [30]. Our solution approach does not require direct access to this distribution; instead it uses samples from the traffic simulator, which may utilize α.

When a vessel is in-transit (or $s_t^m = \langle z, z', \tau\rangle$), it can only take a *dummy* direction action.

Next, we discuss how temporal movement of vessels is modeled (and optimized) in our approach. Intuitively, vessels in TSS can safely move between certain minimum and maximum speeds (based on our discussion with domain experts, in Singapore strait, it is [5knot, 15knot]). When a vessel starts navigating (say from zone z to z'), the traffic control can specify the time vessel should take to perform this navigation action. Navigation time can be converted to raw speed by considering the distance from z to z'. However, for effective practical implementation, we leave it to the ship captain to adjust vessel's speed such that the navigation action takes place as per the recommended duration. Moreover, the actual time required to navigate may not be exactly same as the recommended time. We also model this navigation uncertainty as discussed next.

Meta actions and navigation duration: After a newly-arrived vessel m has taken the direction action $a^m = z'$, it takes a meta action ω^m based on its local observation of the traffic or (s^m, o^m). A meta action ω maps to the low level action $\beta_\omega^{zz'}$, which is a continuous control parameter that determines how long vessel m should take to navigate to z'.

Given the control input $\beta_\omega^{zz'}$ to vessel m, [30] show that the time required to navigate to z' (say τ) can be sampled from the distribution $p^{\text{nav}}(\cdot|z, z'; \beta_\omega^{zz'})$. Intuitively, the parameter $\beta_\omega^{zz'}$ can be interpreted as providing the average travel time to go to z' from z. However, given the movement uncertainty, a vessel may take sometimes more or less time than $\beta_\omega^{zz'}$ to navigate. In the absence of any other information about vessel's characteristics, one can use the concept of *maximum entropy* distribution which has mean $\beta_\omega^{zz'}$ [17]. Given that a vessel requires a minimum and maximum travel time ($t_{\min}, t_{\max}$ resp. based on hard speed limits), the maximum entropy distribution with a specified mean and bounded support is shown to be the binomial distribution [11]. Therefore, similar to [30], we assume p^{nav} is a binomial distribution with its outcome $\Delta \in \{0, \ldots, (t_{\max} - t_{\min})\}$. The realized time required by the vessel m to navigate to z' is $\tau = t_{\min} + \Delta$. We provide empirical validation of this assumption using an expanded dataset than [30] by simulating traffic using a learned binomial distribution from the data, and showing that the simulated traffic produces traffic intensity similar to the actual observed traffic.

State transition function: For a newly-arrived vessel m at zone z with state $s_t^m = \langle z, \emptyset, \emptyset\rangle$, let the direction action be z' and meta action be ω. The set of next possible states are $\{\langle z, z', \tau\rangle \forall \tau \in \{t_{\min}, .., t_{\max}\}\}$. The transition probability is:

$$\phi\Big(\langle z, z', \tau\rangle | \langle z, \emptyset, \emptyset\rangle, \langle z', \omega\rangle\Big) = p^{\text{nav}}(\tau | z, z'; \beta_\omega^{zz'})\mathbb{I}(\tau \in \{t_{\min}, .., t_{\max}\})$$

where $\mathbb{I}$ is the indicator function; $\langle z', \omega\rangle$ is the joint action.

If the vessel m is in-transit from zone z to z' at time t ($s_t^m = \langle z, z', \tau\rangle$), then it can only take a dummy action until it finishes the navigation and reaches the zone z'. The transition function of such a vessel is deterministic and depends on the value of τ. There are two cases. If $\tau = 1$, then the remaining time to reach z' is 1 time step, therefore the next state is $\langle z', \emptyset, \emptyset\rangle$. If $\tau > 1$, then the vessel would still remain in-transit in zone z, but τ decreases by 1:

$$\phi\Big(\langle z', \emptyset, \emptyset\rangle | \langle z, z', \tau\rangle\Big) = 1 \text{ iff } \tau = 1 \quad (1)$$

$$\phi(\langle z, z', \tau - 1\rangle | \langle z, z', \tau\rangle) = 1 \text{ iff } \tau > 1 \quad (2)$$

We assume that terminal zones are absorbing states and have no outgoing transitions. We also assume that all vessels have the same transition function (as they are of the same type).

Reward function: The joint-reward is based on two components—congestion and delay. Each zone z is a limited capacity resource; its capacity C_z is the number of vessels that are allowed to transit at any time with sufficiently safe margins. Such capacities can be set by the VTS authority. There is a penalty imposed when the capacity of a zone is violated. To increase traffic throughput, a delay penalty is also imposed unless the vessel is at its final destination (or a terminal zone). Let $\mathrm{n}_t^{\mathrm{tot}}(z)$ denote the total number of vessels (either in-transit or newly-arrived) in zone z. The reward given to an agent m which is in zone z is:

$$r_t^m = -C(z, \mathrm{n}_t^{\mathrm{tot}}) = -\Big[w_r \cdot \max\Big((\mathrm{n}_t^{\mathrm{tot}}(z) - C_z), 0\Big) + w_d\Big] \quad (3)$$

where $w_r, w_d > 0$ are the resource and delay penalties. Since all agents have the same reward function, we can compute the overall reward at time t as $r_t = -\sum_z \mathrm{n}_t^{\mathrm{tot}}(z) C(z, \mathrm{n}_t^{\mathrm{tot}})$.

2.1 Policy representation

As noted earlier, we do not optimize the direction decision. We assume all vessels share a common policy—$\langle \boldsymbol{\mu}_\theta, \boldsymbol{\pi}_\nu \rangle$. The policy $\boldsymbol{\mu}_\theta$ is the meta action selection policy, and $\boldsymbol{\pi}_\nu$ is the low level navigation decision policy.

Meta action policy: We have $\boldsymbol{\mu}_\theta = \{\mu_{\theta zz'} \forall z, z'\}$ where meta action policy $\mu_{\theta zz'}$ is associated with the zone pair (z, z'). Intuitively, this represents a traffic control system where each zone z controls its outgoing traffic via the policy $\mu_{\theta zz'} \forall z' \in \mathrm{Nb}(z)$ where $\mathrm{Nb}(z)$ are the immediate neighbors of z. Let Ω be the set of all (discrete) meta actions (for simplicity, assume each zone pair has the same number of meta actions). If a vessel is newly-arrived at zone z and its direction action is z', it samples its meta action as $\omega \sim \mu_{\theta zz'}(o^m)$, where o^m is the observation received by the vessel. That is, the policy μ returns a probability distribution over Ω.

Low level policy: We have $\boldsymbol{\pi}_\nu = \{\pi_{\nu zz'} \forall_{z,z'}\}$. Consider a vessel m is newly-arrived at zone z and its direction decision is z' and meta action is ω, then $\beta_\omega^{zz'} = \pi_{\nu zz'}(o^m, \omega)$, where $\beta_\omega^{zz'}$ is the parameter that controls the realized navigation duration $\tau \sim p^{\mathrm{nav}}(\cdot|z, z'; \beta_\omega^{zz'})$. We note that each policy $\pi_{\nu zz'}$ is a *deterministic* policy that maps the agent's observation and meta action to $\beta_\omega^{zz'}$. As mentioned earlier, we interpret $\beta_\omega^{zz'}$ as the average recommended time to move from zone z to z'. However, given that p^{nav} is binomial distribution, we interpret $\beta_\omega^{zz'}$ as the success probability in binomial distribution (essentially, the average travel time is $t_{\min} + (t_{\max} - t_{\min})\beta_\omega^{zz'}$).

Previous work [30] involved only optimizing low level deterministic policy π. In contrast, our policy representations involve optimizing over both discrete meta actions and continuous low level actions simultaneously , which encodes a challenging decision making setting. However, a key benefit of our policy representation is that there is better exploration possible while learning as μ is a stochastic policy over meta actions. In addition, we can also use entropy based penalties over μ that further encourage exploration. Such exploration is difficult in a purely deterministic policy setting of [30]. As a result, our approach is able to provide much better solution quality than [30].

2.2 Count based value function

Let $(\boldsymbol{s}_{1:H}, \boldsymbol{a}_{1:H}, \boldsymbol{\omega}_{1:H})$ be a complete trajectory of joint states, direction actions and meta actions of all the agents. The learning objective can be defined as follows:

$$J(\boldsymbol{\mu}_\theta, \boldsymbol{\pi}_\nu) = \mathbb{E}_{\boldsymbol{s}_{1:H}, \boldsymbol{a}_{1:H}, \boldsymbol{\omega}_{1:H}}\left[\sum_{t=1}^{H} r_t \;\Big|\; \boldsymbol{\mu}_\theta, \boldsymbol{\pi}_\nu\right] \quad (4)$$

Computing the above expression is challenging even for a fixed policy. We need to sample the complete trajectory of each vessel multiple times, which can quickly become intractable given that we have hundreds of vessels. Fortunately, sampling individual agent trajectories is not required. In maritime traffic, transition, reward and observation functions do not depend on the identities of vessels; rather on their aggregate influence on each other, which can be summarized by different types of agent counts. The framework presented above is an instance of collective decentralized POMDPs [24], which is a formal model for collective multiagent planning. Working with count-based information is also a sufficient statistic for planning in the maritime case [30]. The main benefit of learning with count abstractions is that it is highly scalable w.r.t. the number of agents as individual agent trajectories need not be sampled. We next show different count statistics our learning approach uses.

Let $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{\omega}_t) = \langle s_t^m, a_t^m, \omega_t^m \rangle_{m \in 1:M}$ denote the joint state, direction action, and meta action for all vessels at time t.

- For vessels that are in-transit, we define counts: $\mathrm{n}_t^{\mathrm{txn}}(z, z', \tau) = \sum_{m=1}^{M} \mathbb{I}(s_t^m = \langle z, z', \tau \rangle), \forall z, z', \tau$. The table is $\mathrm{n}_t^{\mathrm{txn}} = \Big(\mathrm{n}_t^{\mathrm{txn}}(z, z', \tau) \forall z, z', \tau\Big)$.
- To count newly arrived vessels in a zone z, we define: $\mathrm{n}_t^{\mathrm{arr}}(z) = \sum_{m=1}^{M} \mathbb{I}(s_t^m = \langle z, \emptyset, \emptyset \rangle)$. The table is $\mathrm{n}_t^{\mathrm{arr}} = \Big(\mathrm{n}_t^{\mathrm{arr}}(z) \forall z\Big)$
- To count newly arrived vessels in a zone z which decide to go to z', we define: $\mathrm{n}_t^{\mathrm{nxt}}(z, z') = \sum_{m=1}^{M} \mathbb{I}(s_t^m = \langle z, \emptyset, \emptyset \rangle, a_t^m = z')$. Table is $\mathrm{n}^{\mathrm{nxt}} = \Big(\mathrm{n}^{\mathrm{nxt}}(z, z') \forall z, z'\Big)$
- To count newly arrived vessels at a zone z which plan to move to z' and chooses meta action ω: $\mathrm{n}_t^{\mathrm{mta}}(z, z', \omega) = \sum_{m=1}^{M} \mathbb{I}(s_t^m = \langle z, \emptyset, \emptyset \rangle), a_t^m = z', \omega_t^m = \omega)$. Count table $\mathrm{n}_t^{\mathrm{mta}} = \Big(\mathrm{n}_t^{\mathrm{mta}}(z, z', \omega) \forall z, z', \omega\Big)$
- To count newly arrived vessels at z, who decide to go to z', choose meta action ω and would take τ time steps to reach z', we have: $\tilde{\mathrm{n}}_t(z, z', \omega, \tau) = \sum_{m=1}^{M} \mathbb{I}(s_t^m = \langle z, \emptyset, \emptyset \rangle, a_t^m = z', \omega_t^m = \omega, s_{t+1}^m = \langle z, z', \tau \rangle)$. The table is $\tilde{\mathrm{n}}_t = \Big(\tilde{\mathrm{n}}_t(z, z', \omega, \tau) \forall z, z', \omega, \tau\Big)$

Based on above counts, we can also compute total number of agents in a zone z (either newly-arrived or in-transit) as $\mathrm{n}_t^{\mathrm{tot}}(z) = \mathrm{n}_t^{\mathrm{arr}}(z) + \sum_{z', \tau} \mathrm{n}_t^{\mathrm{txn}}(z, z', \tau)$.

The relation between all the above count tables is shown in the figure 2. We show in the supplemental material[1] how to sample such counts directly without sampling individual agent trajectories. The value function can now be computed by expectation over counts:

$$J(\boldsymbol{\mu}_\theta, \boldsymbol{\pi}_\nu) = \mathbb{E}_{\mathbf{n}_{1:H}}\left[\sum_{t=1}^{H} r(\mathbf{n}_t) \;\Big|\; \boldsymbol{\mu}_\theta, \boldsymbol{\pi}_\nu\right] \quad (5)$$

Simulator design: We note that our solution approach (presented next) requires only count samples from the maritime traffic simulator that essentially samples from the graphical model in figure 2. Our approach can be categorized as centralized learning and decentralized execution approach. The learning takes place in a centralized setting, but the policies can be executed in a decentralized and partially observable setting [20].

An example use case of the system is as follows. The RA trains decentralized policies in a centralized manner using the traffic simulator. The learned policy parameters are then distributed to vessels as they enter the strait waters, or enter a new zone. Vessels

[1] http://jamesarambam.github.io/files/aamas20_sup.pdf

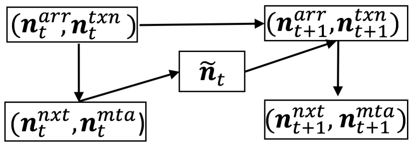

(n_t^{arr}, n_t^{txn})
$(n_{t+1}^{arr}, n_{t+1}^{txn})$
$\tilde{n}_t$
(n_t^{nxt}, n_t^{mta})
$(n_{t+1}^{nxt}, n_{t+1}^{mta})$

Figure 2: Bayes net showing relationship among counts at time step t and $t+1$. Each variable is a count table

can then use the learned policy and their local observations to get the next action. Due to uncertainty in the environment, some deviation in the realized travel time is captured by the model. In practice, due to human factors, some vessels may not heed the policy's recommendations. In our ongoing work, we plan to develop robust policies which take into account such human factors using methods such as quantal response theories [21].

3 META ACTION POLICY GRADIENT

We follow the policy gradient scheme [31] where we take the gradient of objective (5) w.r.t. policy parameters $\boldsymbol{\theta}$ and $\boldsymbol{\nu}$, and adjust parameters towards the direction of the gradient. We focus on computing gradients w.r.t. meta-policy parameters $\boldsymbol{\theta}$; gradients w.r.t. the low level policy parameters $\boldsymbol{\nu}$ is shown in previous work [30]. The key difference is that meta-policy μ is a stochastic policy whereas previous work only optimized the deterministic low-level policy π. Furthermore, we show that following standard meta-policy gradients results in very slow convergence and provides poor solution quality. We therefore develop a multiagent credit assignment technique based on ideas presented in traffic light control in road networks [4, 34]. Without effective credit assignment, the contribution of a particular meta action ω is difficult to ascertain towards the overall objective, and as a result gradients are not very informative. The zone-based value function we develop for meta actions effectively pinpoints the contribution of different meta actions, and results in faster convergence than vanilla policy gradient method.

Theorem 3.1. *The meta action policy gradient $\nabla_{\theta_{zz'}} J(\boldsymbol{\mu_\theta}, \boldsymbol{\pi_\nu})$ for each zone pair (z, z') is given as:*

$$\mathbb{E}_{\mathbf{n}_{1:H}}\Big[\sum_{t=1}^{H-1}\Big(\sum_{\omega} \mathrm{n}_t^{\mathrm{mta}}(z,\, z',\, \omega)\nabla_{\theta_{zz'}} \log \mu_{\theta_{zz'}}(\omega | o(z, \mathrm{n}_t^{\mathrm{tot}}))G_t\Big)\Big]$$

where $G_t = \sum_{t'=t}^{H} r(\mathbf{n}_{t'})$ is the total empirical return.

Proof sketch. From (5), we have expected objective:

$$J(\boldsymbol{\mu_\theta}, \boldsymbol{\pi_\nu}) = \sum_{\mathbf{n}_{1:H}} P(\mathbf{n}_{1:H}; \boldsymbol{\mu}_\theta, \boldsymbol{\pi}_\nu)\Big(\sum_{t'=1}^{H} r(\mathbf{n}_{t'})\Big)$$

Using the above expression, gradient $\nabla_{\theta_{zz'}} J(\boldsymbol{\mu_\theta}, \boldsymbol{\pi_\nu})$ is:

$$= \sum_{\mathbf{n}_{1:H}} P(\mathbf{n}_{1:H})\nabla_{\theta_{zz'}} \log P(\mathbf{n}_{1:H}; \boldsymbol{\mu}_\theta, \boldsymbol{\pi}_\nu)\Big(\sum_{t'=1}^{H} r(\mathbf{n}_{t'})\Big)$$

$$= \mathbb{E}_{\mathbf{n}_{1:H}}\Big[\nabla_{\theta_{zz'}} \log P(\mathbf{n}_{1:H}; \boldsymbol{\mu}_\theta, \boldsymbol{\pi}_\nu)\Big(\sum_{t'=1}^{H} r(\mathbf{n}_{t'})\Big)\Big] \tag{6}$$

The joint count distribution $P(\mathbf{n}_{1:H}; \boldsymbol{\mu}_\theta, \boldsymbol{\pi}_\nu)$ from figure 2 can be represented as:

$$= P(\mathrm{n}_1)\prod_{t=1}^{H-1} P(\mathrm{n}_t^{\mathrm{nxt}} \mid \mathrm{n}_t^{\mathrm{arr}})P(\mathrm{n}_t^{\mathrm{mta}} \mid \mathrm{n}_t^{\mathrm{nxt}}; \boldsymbol{\mu}_\theta)P(\tilde{\mathrm{n}}_t | \, \mathrm{n}_t^{\mathrm{mta}}; \boldsymbol{\pi}_\nu)P(\mathrm{n}_{t+1} \,|\tilde{\mathrm{n}}_t)$$

Intuitively, meta policy parameters $\boldsymbol{\theta}$ only affect those vessels that have already sampled next zone z' and are sampling for meta action ω using policy μ. Therefore only the term $P(\mathrm{n}_t^{\mathrm{mta}} \mid \mathrm{n}_t^{\mathrm{nxt}}; \boldsymbol{\mu}_\theta)$ is affected by θ; rest are constants w.r.t. θ. Using this information:

$$\nabla_{\theta_{zz'}} \log P(\mathbf{n}_{1:H}) = \sum_{t=1}^{H-1} \nabla_{\theta_{zz'}} \log P(\mathrm{n}_t^{\mathrm{mta}} \mid \mathrm{n}_t^{\mathrm{nxt}}; \boldsymbol{\theta}) \tag{7}$$

The distribution $P(\mathrm{n}_t^{\mathrm{mta}} \mid \mathrm{n}_t^{\mathrm{nxt}}; \boldsymbol{\theta})$ is a product of multinomial distributions, one for each zone pair. We can show that if $\mathrm{n}_t^{\mathrm{nxt}}(z, z')$ vessels have decided to go from z to z', counts $\mathrm{n}_t^{\mathrm{mta}}(z, z', \omega)$ are generated from $\mathrm{n}_t^{\mathrm{nxt}}(z, z')$ using a multinomial distribution as:

$$\mathrm{n}_t^{\mathrm{mta}}(z, z', \cdot) \sim \mathrm{Mul}\left(\mathrm{n}_t^{\mathrm{nxt}}(z, z'), \mu_{\theta_{zz'}}(\omega | o(z, \mathrm{n}_t^{\mathrm{tot}}))\forall\omega\right) \tag{8}$$

The gradient $\nabla_{\theta_{zz'}} \log P(\mathrm{n}_t^{\mathrm{mta}} \mid \mathrm{n}_t^{\mathrm{nxt}})$ consequently involves derivating the multinomial distribution corresponding the zone pair (z, z'):

$$\nabla_{\theta_{zz'}} \log P(\mathrm{n}_t^{\mathrm{mta}} \mid \mathrm{n}_t^{\mathrm{nxt}}; \boldsymbol{\theta}) = \sum_{\omega} \mathrm{n}_t^{\mathrm{mta}}(z, z', \omega)\times \nabla_{\theta_{zz'}} \log \mu_{\theta_{zz'}}(\omega | o(z, \mathrm{n}_t^{\mathrm{tot}}))$$

Using the above expression in (7), $\nabla_{\theta_{zz'}} \log P(\mathbf{n}_{1:H}; \boldsymbol{\theta}, \boldsymbol{\nu})$ is:

$$\sum_{t=1}^{H-1}\sum_{\omega} \mathrm{n}_t^{\mathrm{mta}}(z, z', \omega)\nabla_{\theta_{zz'}} \log \mu_{\theta_{zz'}}(\omega | o(z, \mathrm{n}_t^{\mathrm{tot}})) \tag{9}$$

Substituting (9) in (6), we get:

$$\nabla_{\theta_{zz'}} J(\boldsymbol{\mu_\theta}, \boldsymbol{\pi_\nu}) = \mathbb{E}_{\mathbf{n}_{1:H}}\Big[\sum_{t=1}^{H-1}\sum_{\omega} \mathrm{n}_t^{\mathrm{mta}}(z, z', \omega)\cdot \nabla_{\theta_{zz'}} \log \mu_{\theta_{zz'}}(\omega | o(z, \mathrm{n}_t^{\mathrm{tot}})) \cdot \Big(\sum_{t'=1}^{H} r(\mathbf{n}_{t'})\Big)\Big]$$

In order to reduce the variance of the gradient estimator we can use the below equivalent expression:

$$\nabla_{\theta_{zz'}} J(\boldsymbol{\mu_\theta}, \boldsymbol{\pi_\nu}) = \mathbb{E}_{\mathbf{n}_{1:H}}\Big[\sum_{t=1}^{H-1}\sum_{\omega} \mathrm{n}_t^{\mathrm{mta}}(z, z', \omega)\cdot \nabla_{\theta_{zz'}} \log \mu_{\theta_{zz'}}(\omega | o(z, \mathrm{n}_t^{\mathrm{tot}})) \cdot \Big(\sum_{t'=t}^{H} r(\mathbf{n}_{t'})\Big)\Big] \tag{10}$$

Notice that $\sum_{t'=t}^{H} r(\mathbf{n}_{t'})$ is nothing but the empirical return G_t, which completes our proof. □

Entropy based exploration: Various studies have shown that introducing policy entropy in the learning process improves exploration and makes the policy more robust [8, 9, 25, 37]. Therefore, we introduce policy entropy as a regulariser in the policy loss. The return G_t in (10) is replaced with $\big(G_t + \eta\mathcal{H}(\mu_{\theta_{zz'}}(\cdot | o(z, \mathrm{n}_t^{\mathrm{tot}})))\big)$, where $\mathcal{H}$ is the entropy of meta policy, and η is a parameter that determines the relative importance between the entropy and the return. Note here that introduction of expected entropy in the policy network loss is not possible in the work of [30], as their policy is deterministic.

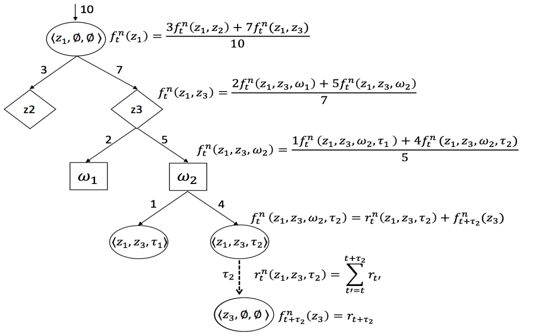

10
$\langle z_1, \emptyset, \emptyset \rangle$ $f_t^n(z_1) = \frac{3f_t^n(z_1,z_2) + 7f_t^n(z_1,z_3)}{10}$
3
7
z2
z3
$f_t^n(z_1,z_3) = \frac{2f_t^n(z_1,z_3,\omega_1) + 5f_t^n(z_1,z_3,\omega_2)}{7}$
2
5
$f_t^n(z_1,z_3,\omega_2) = \frac{1f_t^n(z_1,z_3,\omega_2,\tau_1) + 4f_t^n(z_1,z_3,\omega_2,\tau_2)}{5}$
ω_1
ω_2
1
4
$f_t^n(z_1,z_3,\omega_2,\tau_2) = r_t^n(z_1,z_3,\tau_2) + f_{t+\tau_2}^n(z_3)$
$\langle z_1, z_3, \tau_1 \rangle$
$\langle z_1, z_3, \tau_2 \rangle$
τ_2
$r_t^n(z_1,z_3,\tau_2) = \sum_{t'=t}^{t+\tau_2} r_{t'}$
$\langle z_3, \emptyset, \emptyset \rangle$ $f_{t+\tau_2}^n(z_3) = r_{t+\tau_2}$

Figure 3: Individual value function computation for meta actions

We omit the proof for the gradient w.r.t. low level policy parameters ν as it is similar to [30]. Final gradient expression is:

$$\nabla_{\nu^{zz'}} J(\boldsymbol{\mu_\theta}, \boldsymbol{\pi_\nu}) = \mathbb{E}_{\mathbf{n}_{1:H}}\Bigg[\sum_{t=1}^{H-1}\sum_{\omega,\tau} \tilde{\mathrm{n}}_t(z,z',\omega,\tau)\Big((\tau - t_{\min}^{zz'})\cdot$$
$$\nabla_{\nu^{zz'}} \log \beta_\omega^{zz'} + (t_{\max}^{zz'} - \tau)\nabla_{\nu^{zz'}} \log(1-\beta_\omega^{zz'})\Big)\cdot G_t\Bigg]$$

The policy trained with global empirical return G_t is known to be sample inefficient and results in poor solution quality particularly in multiagent settings because the global reward signal does not address the multiagent credit assignment problem [3, 7]. Therefore, we next address this issue for meta actions.

Credit assignment with meta actions: To address the multiagent credit assignment problem, we propose a solution approach motivated from car-based value function used for traffic light control [34]. The car-based value function essentially estimates the total expected reward of each car until they reach their destination given their current traffic light setting. The value function of a traffic light is the sum of car-based value functions of all cars that are waiting in queue at the particular light.

In the maritime traffic case, consider all vessels that enter a zone z, decide to move to z' and choose meta action ω at time t, then the value function of the zone intersection $\langle z, z'\rangle$ at time t is the expected reward obtained by all such agents. It is given as:

$$V_t^{zz'}(\mu_{\theta^{zz'}}, \pi_{\nu^{zz'}}) = \mathbb{E}_{\boldsymbol{s}_{1:H}, \boldsymbol{a}_{1:H}, \boldsymbol{\omega}_{1:H}}\Big[\sum_{m=1}^{M}\mathbb{I}\big[s_t^m = \langle z, \emptyset, \emptyset\rangle,$$
$$a_t^m = z', \omega_t^m = \omega\big]\cdot\Big(\sum_{t'=t}^{H} r_{t'}^m\Big)\Big] \quad (11)$$

Computing (11) requires sampling joint agent trajectories, which is not a scalable approach when there are a large number of agents. However, we can show (proof omitted) that the count based framework allows to compute the same by directly sampling counts:

$$V_t^{zz'}(\mu_{\theta^{zz'}}, \pi_{\nu^{zz'}}) = \mathbb{E}_{\mathbf{n}_{1:H}}\Big[\sum_{\omega} \mathrm{n}_t^{\mathrm{mta}}(z,z',\omega)\cdot f_t^{\mathrm{n}}(z,z'\omega)\Big] \quad (12)$$

where $f_t^n(z, z', \omega)$ is a total average return a vessel receives until it reaches its destination given its state at time t is $\langle z, \emptyset, \emptyset\rangle$, and action taken is (z', ω). We refer $f_t^n(z, z', \omega)$ as *individual meta value function* (IMVF). Next, we show a dynamic programming approach to compute IMVF given count samples $\mathrm{n}_{1:H}$.

We first illustrate intuitively given a count sample $\mathbf{n}_{1:H}$, how we can compute the IMVF. In figure 3, 10 vessels enter zone z_1, and 3 vessels decide to move to zone z_2, and 7 to zone z_3. The value function f_t^{n} is the weighted average of the value received when taking action z_2 and z_3. Of the 7 vessels who decided to move to z_3, 2 take meta action ω_1 and 5 take meta action ω_2. The value function $f_t^{\mathrm{n}}(z_1, z_3)$ is again the weighted average of the values received upon taking action ω_1 and ω_2. Out of 5 vessels who took meta action ω_2, 4 will take time τ_2 to cross z_1 and will reach z_3 at time $t + \tau_2$ and 1 vessel would reach z_3 at time $t + \tau_1$. The 4 vessels whose current state is $\langle z_1, z_3, \tau_2\rangle$ (bottom right, second last node) will stay in zone z_1 until τ_2 time units and accumulate reward $r_t^{\mathrm{n}}(z_1, z_3, \tau_2)$ as shown in figure 3. This whole process can summarized below with the following dynamic programming equations (proof omitted):

$$r_t^{\mathrm{n}}(z, z', \tau) = \sum_{t'=t}^{t+\tau} -C(z, \mathrm{n}_{t'}^{\mathrm{tot}}),\ \forall\tau \in [t_{\min}^{zz'}, t_{\max}^{zz'}]$$
$$f_t^{\mathrm{n}}(z, z', \tau) = r_t^{\mathrm{n}}(z, z', \tau) + f_{t+\tau}^{\mathrm{n}}(z')$$
$$f_t^{\mathrm{n}}(z, z', \omega, \tau) = \frac{f_t^{\mathrm{n}}(z, z', \tau)\cdot\tilde{\mathrm{n}}_t(z, z', \omega, \tau)}{\sum_\omega \tilde{\mathrm{n}}_t(z, z', \omega, \tau)}$$
$$f_t^{\mathrm{n}}(z, z', \omega) = \frac{\sum_{\tau=t_{\min}^{zz'}}^{t_{\max}^{zz'}} f_t^{\mathrm{n}}(z, z', \omega, \tau)\cdot\tilde{\mathrm{n}}_t(z, z', \omega, \tau)}{\sum_{\tau=t_{\min}^{zz'}}^{t_{\max}^{zz'}} \tilde{\mathrm{n}}_t(z, z', \omega, \tau)}$$
$$f_t^{\mathrm{n}}(z, z') = \frac{\sum_\omega f_t^{\mathrm{n}}(z, z', \omega)\cdot\mathrm{n}_t^{\mathrm{mta}}(z, z', \omega)}{\sum_\omega \mathrm{n}_t^{\mathrm{mta}}(z, z', \omega)}$$
$$f_t^{\mathrm{n}}(z) = \frac{\sum_{z'} f_t^{\mathrm{n}}(z, z')\cdot\mathrm{n}_t^{\mathrm{nxt}}(z, z')}{\sum_{z'} \mathrm{n}_t^{\mathrm{nxt}}(z, z')}$$

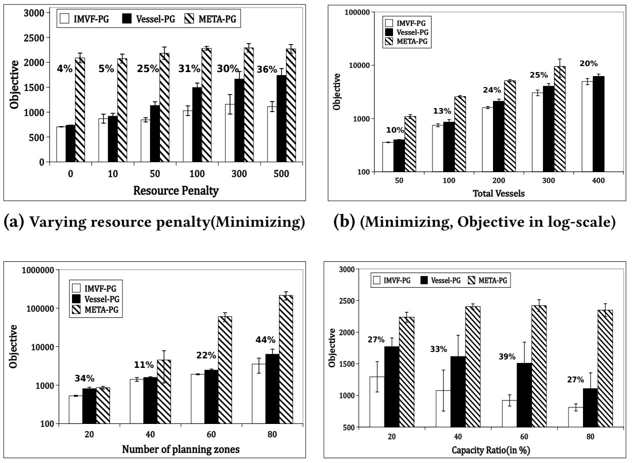
With IMVF we can perform efficient credit assignment as it gives a clearer training signal. Thus, the new policy gradient expression with credit assignment uses $f_t^n(z, z', \omega)$ instead of the global return G_t in theorem 3.1. For low level policy gradient, we similarly replace G_t with IMVF. We can estimate both low level and meta policy gradients using count samples $\mathbf{n}_{1:H}$ generated from the simulator, and parameters can be moved towards the direction of gradients.

4 EXPERIMENTS

We evaluate our approach on both synthetic and real world instances. Synthetic experiments compare against different baselines by varying different features of instances (e.g., number of agents, zone capacities, number of zones). Experiments based on real world scenarios are to measure performance of our approach on mitigating hotspots and improving throughput. We use the following approaches:

- **Vessel-PG** : Previous best approach by [30] for maritime traffic
- **Meta-PG** : Vanilla policy gradient version of meta action policy (Meta-PG) trained using the empirical return G_t
- **IMVF-PG** : Our approach with meta action policy trained using individual meta value functions

Synthetic experiments: We generate semi-random connected directed graphs where each edge denotes a zone which has a minimum and maximum travel time along with a maximum capacity of vessels it can accommodate at any time. Vessels enter the graph through different source zones, and they follow an arrival rate. To provide a fair comparison, we use same neural network based policy hyper-parameter settings for all three approaches. More details



(a) Varying resource penalty(Minimizing)

(b) (Minimizing, Objective in log-scale)

(c) (Minimizing, Objective in log-scale)

(d) Varying capacity(Minimizing)

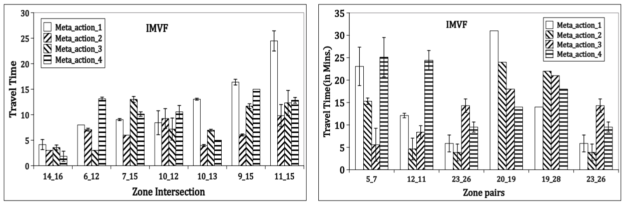
Figure 4: (a-d) show results for synthetic instances (lower value is better). Percentage values in graphs show the improvement by IMVF-PG over Vessel-PG for the respective setting.

on the experimental settings are provided in the supplemental material. In experiments, we measure total objective which combines both delay and congestion cost (a lower value is better). For all synthetic (and real world experiments) we use 4 meta actions and delay penalty $w_d = 1$. We experimented with varying number of meta actions, and observed that increasing beyond 4 meta actions provided little marginal gain.

We compute improvement in solution quality by our approach IMVF-PG over Vessel-PG using $\frac{\text{Vessel-PG} - \text{IMVF-PG}}{\text{Vessel-PG}}$; Meta-PG was often much worse than other approaches.

Figure 4a shows results for experiment with *varying resource penalty* (w_r), 100 vessels, max capacity of each zone is uniformly sampled between [5, 10], vessels arrival time at source zones is sampled uniformly between [1, 20]. Each data point is an average of 10 runs (we also show standard deviations). We observe that performance gap between IMVF-PG and Vessel-PG grows as we increase resource penalty—IMVF-PG achieves improvement of 31% and 36% on settings $w_r = 100, 500$ respectively over Vessel-PG . This is because with increasing resource penalty, tighter coordination is required among vessels, which is better achieved by our approach IMVF-PG using better exploration using meta actions. The standard policy gradient Meta-PG without the credit assignment is much worse, which confirms the benefits of our credit assignment scheme.

Figure 4b shows result for experiments with *varying vessels population* (M), capacity for each zone is uniformly sampled from [5, 50] for all population sizes. We increased the capacity from the previous setting as the number of vessels is much larger than in figure 4a. We omit the result for Meta-PG for $M = 400$ as it finds poor solution which distorts the graph for other approaches. In this setting, IMVF-PG is able to achieve improvement of 24%, 25% on $M = 200, 300$ respectively over Vessel-PG . Our approach IMVF-PG was consistently better than previous best Vessel-PG for all other agent populations also.



(a) Synthetic instances

(b) Real world instances

Figure 5: (a-b) shows results for multi-modal behavior of travel time in the maritime traffic.

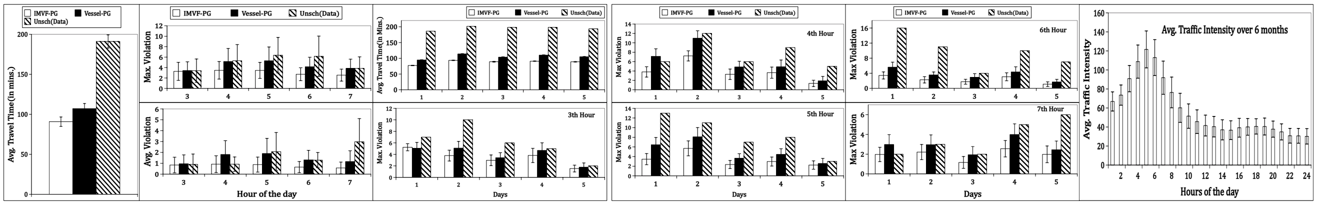
For figure 4c, we vary the *number of planning zones*, fix the total number of vessels $M = 100$, and capacity is uniformly sampled from [5, 10]. In this setting, we observe improvements (by IMVF-PG over Vessel-PG) of 22% and 44% on total planning zones 60, 80. For other zone sizes also, our approach IMVF-PG is the best among baselines.

Figure 4d shows results on *varying max capacity* of zones. We fix the total number of vessels to 100 and maximum capacity is set to 50. Then, for each zone we uniformly sample from [5, capacity ratio × 50]. E.g., for 20% capacity ratio, the capacity range becomes [5, 10]. In this setting, we see consistent significant improvement of IMVF-PG over Vessel-PG for all capacity settings—the improvement for 60% capacity is 39%. Furthermore, with increasing capacity, our approach provides consistently decreasing objective, which is highly desirable confirming that our approach can use increased capacity to better coordinate vessels in reducing the congestion.

In figure 5a, we show results on multi-modal behavior of travel time in the maritime traffic for our IMVF-PG approach. This figure shows that different meta actions indeed encode different navigation behaviors for different zone intersections. On the x-axis, we show the different zone intersections—14_16 denotes intersection between zone $z = 14$ and $z' = 16$. For each zone intersection, we show the mean of the binomial travel time distribution for each meta action ω on the y-axis. We observe that meta actions in all zone intersections show multi modal behavior. Each meta action has a different average travel time. For different traffic intensities, the meta action policy selects the most appropriate meta action for vessels to use. Using our learning process, we do not have to predefine what should be behavior encoded by different meta actions; it is learned automatically during the course of training.

Real data experiments: We also test on real data data gathered in Singapore strait. We use a total of 6 months datasets from 1 Jan 2017-30th June 2017, total of 180 days. We use 150 days for training and 30 days for testing. Our data contains AIS record of vessels voyaging through Singapore strait, which is one of the busiest ports of the world. The AIS record contains a timestamp, vessel unique id, lat-long position, speed over ground, direction and navigation status (anchored/sailing etc). We have data for every few seconds for majority of vessel in the strait totaling about 14 million records. Our dataset is 50% larger than the one used by [30].

Training: From the training datasets we learn the following input parameters—arrival distribution, initial vessel counts, and direction distribution α, zone capacities C_z, minimum $t^{zz'}_{\min}$ and maximum $t^{zz'}_{\max}$ travel time for each zone intersection. From the



(a) Avg. Travel Time (b) Resource Violation (c) Avg. Travel Time (top), and 3rd Hour violations (bottom) (d) 4th and 5th Hour violations (e) 6th and 7th Hour violations (f) Traffic

Figure 6: Results on real world historical data

arrival distribution we get information about new vessels entering the planning area at each time step. Initial count distribution gives us number of vessels that are present in the various zones when we start our system at $t = 1$. The maximum number of vessels present in a zone z over all days and at any point in time is treated as the max capacity of the zone. With set zone capacities for our traffic optimization to 50%MaxCapacity (MaxCapacity can be different for different zones). This ensures that our approach tries to increase safety of navigation by decreasing the traffic congestion. All the learned parameters are embedded into the simulator which provides count samples to the learning approaches. We set resource penalty $w_r = 500$ and delay penalty $w_d = 1$ (this combination worked well empirically). We train our policy mainly for the peak traffic intensity period as shown in figure 6f; i.e from 3rd hour−7th hour.

Testing: We test our learned policy on 30 different days comprising the test dataset. Each day is unique with its arrival rate and initial vessel counts. Figure(6a) and (6b) results are average over 30 days. Figure (6a) shows results on total travel time for east-bound route (as shown in figure 1) measuring average time (in minutes) a vessel takes to navigate through the east bound route from one end to the other during peak hours. The result shows that both IMVF-PG and Vessel-PG are able to perform well against the unscheduled (Unsch.) traffic (which is the essentially the replay of the historical data). Our approach IMVF-PG is able to further reduce the travel time over Vessel-PG . Results were similar for the west-bound route. Our observations suggest that there is significant scope for better traffic coordination in real world data. Our discussions with domain experts also confirm that an effective way to decrease the congestion is to sail through TSS as fast as possible (but within given min and max travel times). Our approach IMVF-PG validates this observation in simulation also.

Figure(6b) results show the maximum violations (top) and average violations (below) per minute for different (peak) hours over 30 days. If there is one more vessel in a zone than its capacity, then it is counted as one violation. We observe that IMVF-PG is able to effectively reduce the violations significantly on the peak hour (5th and 6th). Previous approach Vessel-PG also performs well against Unsch., but sub-par against our approach IMVF-PG . This result is significant as it shows that our approach can significantly increase the safety of navigation while keeping traffic throughout high, which is our study's main goal.

In figures (6c-bottom) to (6e), we show capacity violation results for top 5 busiest days for the 5 peak hours period (3rd−7th hour). We observe that in all 5 days and 5 peak hours, IMVF-PG is able to reduce congestion, and significantly reduce travel time (shown in figure 6c-top) effectively over both Vessel-PG and Unsch. In figure 5b, we also observe the multi-modal behavior of vessels in real world problem scenarios as well where different meta actions encode different average travel times for different zone intersections.

Simulator Validity: We also evaluated the count based simulator model against observed real data count. We use the same 150 days of training dataset to estimate the travel time parameter $\beta^{zz'}$. Then for testing, we use the remaining 30 test days. We evaluate accuracy for the peak hour period (3rd - 7th hour). For each day, we start with the initial count of the test day and use the learned parameter $\beta^{zz'}$ to simulate traffic movement using the simulator. For each hour, we then compute the RMSE value of the generated counts with the actual observed counts from data over all the zones. The average RMSE value overall 30 days was fairly low for each hour—4.8 for hour 3, 5.5 for hour 4, 6.6 for hour 5, 6.7 for hour 6, and 7.8 for hour 7. Our RMSE values are fairly low for each peak hour period given that more than 70 vessels are present in different zones during peak hours (as shown in figure 6f).It shows that the simulator and travel time assumptions are fairly accurate to describe aggregate traffic.

Conclusion: We have presented a new approach for maritime traffic control in geographically constrained ports such as Singapore's. Our key objective was to make congested waters safer by reducing the traffic intensity, while keeping traffic throughput high. We developed a hierarchical learning approach that used the notion of high level meta actions, which intuitively correspond to different traffic situations. Each meta action provided a mapping to a low level navigation action that provided vessels a recommended travel time to cross a zone. Using such high level and low level policies, we showed both theoretical advantages (such as better exploration while learning using the meta policy), and empirically validated our approach on both synthetic and a large real world datasets.

ACKNOWLEDGMENTS

This research is supported by the Agency for Science, Technology and Research (A*STAR), Fujitsu Limited and the National Research Foundation Singapore as part of the A*STAR-Fujitsu- SMU Urban Computing and Engineering Centre of Excellence. The support given by the Maritime and Port Authority of Singapore to this study is also acknowledged. First author's travel is supported by SMU Presidential Doctoral Fellowship.

REFERENCES

[1] Lucas Agussurja, Akshat Kumar, and Hoong Chuin Lau. 2018. Resource-Constrained Scheduling for Maritime Traffic Management. In *AAAI Conference on Artificial Intelligence*. AAAI Press, 6086–6093.

[2] Christopher Amato, George D Konidaris, and Leslie P Kaelbling. 2014. Planning with macro-actions in decentralized POMDPs. In *International conference on Autonomous agents and multi-agent systems*. Association for Computing Machinery (ACM), 1273–1280.

[3] Drew Bagnell and Andrew Y Ng. 2006. On local rewards and scaling distributed reinforcement learning. In *Advances in Neural Information Processing Systems*. 91–98.

[4] Bram Bakker, Shimon Whiteson, Leon Kester, and Frans C. A. Groen. 2010. Traffic Light Control by Multiagent Reinforcement Learning Systems. In *Interactive Collaborative Information Systems*. Springer, 475–510.

[5] Andrew G Barto and Sridhar Mahadevan. 2003. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems* 13, 1-2 (2003), 41–77.

[6] Saumya Bhatnagar, Akshat Kumar, and Hoong Chuin Lau. 2019. Decision Making for Improving Maritime Traffic Safety Using Constraint Programming. In *International Joint Conference on Artificial Intelligence*. 5794–5800.

[7] Yu Han Chang, Tracey Ho, and Leslie Pack Kaelbling. 2004. All learning is local: Multi-agent learning in global reward games. In *Advances in Neural Information Processing Systems*. 807–814.

[8] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*. 1352–1361.

[9] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*. PMLR, 1861–1870.

[10] Marcus Hand. 2017. Malacca and S'pore Straits traffic hits new high in 2016, VLCCs fastest growing segment. http://www.seatrade-maritime.com/news/asia/malacca-and-s-pore-strait-traffic-hits-new-high-in-2016-vlccs-fastest-growing-segment.html. (2017).

[11] P. Harremoes. 2001. Binomial and Poisson distributions as maximum entropy distributions. *IEEE Transactions on Information Theory* 47, 5 (2001), 2039–2041.

[12] Kazuhiko Hasegawa. 1993. Knowledge-based automatic navigation system for harbour manoeuvring. In *Ship Control Systems Symposium*. 67–90.

[13] Kazuhiko Hasegawa, Go Tashiro, Seiji Kiritani, and Koji Tachikawa. 2001. Intelligent marine traffic simulator for congested waterways. In *IEEE International Conference on Methods and Models in Automation and Robotics*. 632–636.

[14] IMO. 2019. E-navigation. http://www.imo.org/en/OurWork/Safety/Navigation/Pages/eNavigation.aspx. (2019).

[15] IMO. 2019. Ships' routeing. http://www.imo.org/en/OurWork/Safety/Navigation/Pages/ShipsRouteing.aspx. (2019).

[16] A. N. Ince and E. Topuz. 2004. Modelling and Simulation for Safe and Efficient Navigation in Narrow Waterways. *Journal of Navigation* 57, 1 (2004), 53–71.

[17] E. T. Jaynes. 1957. Information Theory and Statistical Mechanics. *Phys. Rev.* 106 (1957), 620–630.

[18] Elena Kelareva, Sebastian Brand, Philip Kilby, Sylvie Thiébaux, Mark Wallace, et al. 2012. CP and MIP Methods for Ship Scheduling with Time-Varying Draft.. In *International Conference on Automated Planning and Scheduling*. 110–118.

[19] Annabelle Liang and Wong Maye-E. 2017. Busy waters around Singapore carry a host of hazards. https://www.navytimes.com/news/your-navy/2017/08/22/busy-waters-around-singapore-carry-a-host-of-hazards/. (2017).

[20] Ryan Lowe, YI WU, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems*. 6379–6390.

[21] Richard D. McKelvey and Thomas R. Palfrey. 1995. Quantal Response Equilibria for Normal Form Games. *Games and Economic Behavior* 10, 1 (1995), 6–38.

[22] United Nations. 2019. Review of Maritime Transport. In *United Nations Conference on Trade and Development*. https://unctad.org/en/PublicationsLibrary/rmt2019_en.pdf

[23] Future Nautics. 2016. Autonomous Ships | White Paper. https://www.sipotra.it/old/wp-content/uploads/2017/05/Autonomous-Ships.pdf. (2016).

[24] Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. 2017. Collective Multiagent Sequential Decision Making Under Uncertainty. In *AAAI Conference on Artificial Intelligence*. 3036–3043.

[25] Brendan O'Donoghue, Rémi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. 2016. PGQ: Combining policy gradient and Q-learning. *arXiv preprint arXiv:1611.01626* (2016). http://arxiv.org/abs/1611.01626

[26] Doina Precup. 2000. Temporal abstraction in reinforcement learning. *Doctoral Dissertation, University of Massachusetts Amherst* (2000). https://scholarworks.umass.edu/dissertations/AAI9978540

[27] Roger P Roess and Elena S Prassas. 2014. *The Highway Capacity Manual: A Conceptual and Research History* (1 ed.). Springer International Publishing.

[28] Rolls-Royce. 2016. Remote and Autonomous Ship-The next steps. https://www.rolls-royce.com/~/media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/aawa-whitepaper-210616.pdf. (2016).

[29] Lim Tin Seng. 2017. Land From Sand: Singapore's Reclamation Story. http://www.nlb.gov.sg/biblioasia/2017/04/04/land-from-sand-singapores-reclamation-story/. (2017).

[30] Arambam James Singh, Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. 2019. Multiagent Decision Making For Maritime Traffic Management. In *AAAI Conference on Artificial Intelligence*. 6171–6178.

[31] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *International Conference on Neural Information Processing Systems*. 1057–1063.

[32] Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112, 1-2 (1999), 181–211.

[33] Teck-Hou Teng, Hoong Chuin Lau, and Akshat Kumar. 2017. Coordinating Vessel Traffic to Improve Safety and Efficiency. In *International Conference on Autonomous Agents and MultiAgent Systems*. 141–149.

[34] Marco Wiering. 2000. Multi-Agent Reinforcement Learning for Traffic Light Control. In *International Conference on Machine Learning*. 1151–1158.

[35] Yuchen Xiao, Joshua Hoffman, Tian Xia, and Christopher Amato. 2019. Multi-Robot Deep Reinforcement Learning with Macro-Actions. *arXiv preprint arXiv:1909.08776* (2019).

[36] Jinfen Zhang, Tiago A Santos, C Guedes Soares, and Xinping Yan. 2017. Sequential ship traffic scheduling model for restricted two-way waterway transportation. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment* 231, 1 (2017), 86–97.

[37] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum entropy inverse reinforcement learning.. In *AAAI Conference on Artificial Intelligence*. 1433–1438.