Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)

Online Decision-Making for Scalable Autonomous Systems

Kyle Hollins Wray [1,2] and **Stefan J. Witwicki** [2] and **Shlomo Zilberstein** [1]
[1] College of Information and Computer Sciences, University of Massachusetts, Amherst, MA 01002
[2] Nissan Research Center - Silicon Valley, Sunnyvale, CA 94089
wray@cs.umass.edu, stefan.witwicki@nissan-usa.com, shlomo@cs.umass.edu

Abstract

We present a general formal model called MODIA that can tackle a central challenge for autonomous vehicles (AVs), namely the ability to interact with an unspecified, large number of world entities. In MODIA, a collection of possible decision-problems (DPs), known a priori, are instantiated online and executed as decision-components (DCs), unknown a priori. To combine the individual action recommendations of the DCs into a single action, we propose the lexicographic executor action function (LEAF) mechanism. We analyze the complexity of MODIA and establish LEAF's relation to regret minimization. Finally, we implement MODIA and LEAF using collections of partially observable Markov decision process (POMDP) DPs, and use them for complex AV intersection decision-making. We evaluate the approach in six scenarios within a realistic vehicle simulator and present its use on an AV prototype.

1 Introduction

There has been substantial progress with planning under uncertainty in partially observable, but fully modeled worlds. However, few effective formalisms have been proposed for planning in open worlds with an unspecified, large number of objects. This remains a key challenge for autonomous systems, particularly for *autonomous vehicles* (AVs). AV research has advanced rapidly since the DARPA Grand Challenge [Thrun *et al.*, 2006], which acted as a catalyst for subsequent work on low-level sensing [Sivaraman and Trivedi, 2013] and control [Dolgov *et al.*, 2010], as well as high-level route planning [Wray *et al.*, 2016a].

A critical missing component to enable autonomy in *long-term urban deployments* is the *mid-level intersection decision-making* (e.g., the second-to-second stop, yield, edge, or go decisions). As in many robotic domains, the primary challenges include the sheer complexity of real-world problems, wide variety of possible scenarios that can arise, and unbounded number of multi-step problems that will be actually encountered, perhaps simultaneously. These factors have limited the deployment of existing methods for mid-level decision-making [Ulbrich and Maurer, 2013; Brechtel *et al.*, 2014; Bai *et al.*, 2015; Jo *et al.*, 2015]. We present a scalable, realistic solution, with strong mathematical foundations, via decomposition into problem-specific decision-components.

Our primary motivation is to provide a *general* solution for AV decision-making at any intersection, including n-way stops, yields, left turns at green traffic lights, right turns at red traffic lights, etc. In this domain, the AV approaches the intersection knowing only the static features from the map, such as road, crosswalk, and traffic controller information. Any number of vehicles and pedestrians can arrive and interact around the intersection, all potentially relevant to decision-making and unknown a priori. The AV must make mid-level decisions, using *very limited hardware* resources, including when to stop, yield, edge forward, or go, based on all possible interactions among all vehicles including the AV itself. Vehicles can be occluded, requiring the use of information gathering actions based on belief over partial observability. Pedestrians can jaywalk, necessitating that motion forward is taken only under strong confidence they will not cross. Uncertainty regarding priority and right-of-way exists, and must be handled under stochastic changes. Vehicles and pedestrians can block one another's motion, and AV-related blocking conflicts must be discovered and resolved via motion-based negotiation.

We provide a general solution for domains concerning *multiple online decision-components with interacting actions* (MODIA). For the particularly difficult AV intersection decision domain, MODIA considers all vehicles and pedestrians as separate individual decision-components. Each component is a partially observable Markov decision process (POMDP) that maintains its own belief for that particular component problem and proposes an action to take at each time step. MODIA then employs an executor function to act as an action aggregator to determine the actual action taken by the AV. This decomposition enables a tractable POMDP solution, benefiting from powerful belief-based reasoning while only growing linearly in the number of encountered problems.

The primary contributions include: a formal definition of MODIA (Section 3), a rigorous analysis of the complexity and regret-minimization properties (Section 4), an AV intersection decision-making MODIA solution (Section 5), and an evaluation of the approach in simulation as well as integration with a real AV (Section 6). We begin with a review of POMDPs (Section 2), and conclude with a survey of related work (Section 7) and final reflections (Section 8).

2 Background Material

A *partially observable Markov decision process (POMDP)* is represented by the tuple $\langle S, A, \Omega, T, O, R\rangle$ [Kaelbling *et al.*, 1998]. S is a finite set of states. A is a finite set of actions. Ω is a finite set of observations. $T: S \times A \times S \rightarrow [0,1]$ is a state transition function such that $T(s,a,s') = Pr(s'|s,a)$. $O: A \times S \times \Omega \rightarrow [0,1]$ is an observation function such that $O(a,s',\omega) = Pr(\omega|a,s')$. $R: S \times A \rightarrow \mathbb{R}$ is a reward function. The agent does not observe the true state of the system, and instead makes observations while maintaining a *belief* over the true state denoted $b \in \triangle^{|S|}$. Given action $a \in A$ and subsequent observation $\omega \in \Omega$, belief b is updated to b' with: $b'(s') = \eta O(a,s',\omega) \sum_s T(s,a,s')b(s)$ for all $s' \in S$, with normalizing constant η. A *policy* maps beliefs to actions $\pi: \triangle^{|S|} \rightarrow A$. (Note: $\triangle^n$ is the standard n-simplex.) The value function $V: \triangle^{|S|} \rightarrow \mathbb{R}$ for a belief is the expected reward given a fixed policy π, a discount factor $\gamma \in [0,1]$, and a horizon h. Also, it is useful to define the *Q-value* of belief b given action a as $Q: \triangle^{|S|} \times A \rightarrow \mathbb{R}$ with $V(b) = Q(b, \pi(b))$. Since V^π is piecewise linear and convex, we describe it using sets of α*-vectors* $\Gamma = \{\alpha_1, \ldots, \alpha_r\}$ with each $\alpha_i = [\alpha_i(s_1), \ldots, \alpha_i(s_n)]^T$ and $\alpha_i(s)$ denoting value of state $s \in S$. The objective is to find optimal policy π^* that maximizes V denoted as V^*. Given an initial belief b^0, V^* can be iteratively computed for a time step t, expanding beliefs at each update resulting in belief b, by maximizing:

$$Q^t(b,a) = \sum_{s \in S} b(s)R(s,a) + \sum_{\omega \in \Omega} \max_{\alpha \in \Gamma^{t-1}} \sum_{s \in S} b(s) V^t_{sa\omega\alpha}$$

and $V^t_{sa\omega\alpha} = \gamma \sum_{s' \in S} O(a,s',\omega)T(s,a,s')\alpha(s')$; for $s \in S$, $\alpha^0(s) = \underline{R}/(1-\gamma)$ in $\Gamma^0 = \{\alpha^0\}$ with $\underline{R} = \min_{s,a} R(s,a)$.

3 Problem Formulation

We begin with a general problem description that considers a single autonomous agent that encounters any number of decision problems online during execution. This paper focuses on collections of POMDPs primarily for their general form, self-consistency, and space limitations. It can be generalized to other decision-making models in the natural way. Finally, Figure 1 depicts a complete MODIA example for AVs, and is referenced throughout this section for each concept.

3.1 Decision-Making with MODIA

The **multiple online decision-components with interacting actions (MODIA)** model describes a realistic single-agent *online* decision-making scenario defined by the tuple $\langle \mathcal{P}, \mathcal{A}\rangle$. $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ are **decision-problems (DPs)** that could be encountered during execution. For this paper, each $\mathcal{P}_i \in \mathcal{P}$ is a POMDP with $\mathcal{P}_i = \langle S_i, A_i, \Omega_i, T_i, O_i, R_i\rangle$ (Section 2) starting from an initial belief $b_i^0 \in \triangle^{|S_i|}$. We consider discrete *time steps* $t \in \mathbb{N}$ over the agent's entire lifetime. $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_z\}$ are z **primary actions** that are the true actions taken by the agent that affect the state of the external system environment. *Importantly, only $\mathcal{P}$ and $\mathcal{A}$ are known offline a priori.*

AV Example Figure 1 has two pre-solved intersection decision-components: single vehicle ($\mathcal{P}_1$) or pedestrian ($\mathcal{P}_2$). Each are POMDPs with actions (recommendations) 'stop' or 'go'. Primary actions $\mathcal{A}$ for the AV are also 'stop' or 'go'.

Online, the DPs are *instantiated* based on what the agent experiences in the external system environment. Due to the nature of actually executing multiple decision-making models (e.g., POMDPs) in real applications, there is no complete model for which, when, or how many DPs are instantiated, or even how long they are relevant.

Formally, the online **instantiations** in MODIA are defined by the tuple $\langle \mathcal{C}, \phi, \tau\rangle$. Over the agent's lifetime, there are n DP instantiations called **decision-components (DCs)** denoted as $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$, with both $\mathcal{C}$ and n *unknown* a priori. Let $\phi: \mathcal{C} \rightarrow \mathcal{P}$ denote the DP for each instantiation. Let $\tau: \mathcal{C} \rightarrow \mathbb{N} \times \mathbb{N}$ be the two time steps that each DC is instantiated and terminated. For notational convenience, for all $\mathcal{C}_i \in \mathcal{C}$, let $\tau_s(\mathcal{C}_i)$ and $\tau_e(\mathcal{C}_i)$ be the start and end times; we have $\tau_s(\mathcal{C}_i) < \tau_e(\mathcal{C}_i)$. Without loss of generality, we also assume for $i < j$, $\tau_s(\mathcal{C}_i) \leq \tau_s(\mathcal{C}_j)$. We call a DC $\mathcal{C}_i \in \mathcal{C}$ **instantiated** at time step $t \in \mathbb{N}$ if $t \in [\tau_s(\mathcal{C}_i), \tau_e(\mathcal{C}_i)]$. Any instantiated $\mathcal{C}_i \in \mathcal{C}$ includes POMDP $\phi(\mathcal{C}_i)$, its policy $\pi_i: \triangle^{|S_i|} \rightarrow A_i$, and its current belief state $b_i^{t_i} \in \triangle^{|S_i|}$ with *local* POMDP time step $t_i = t - \tau_s(\mathcal{C}_i)$.

AV Example (Continued) Online, the AV encounters an intersection and immediately (at time step 1) observes two vehicles and one pedestrian. Three DCs are instantiated; $\mathcal{C}_1$ and $\mathcal{C}_2$ are for each vehicle ($\phi(\mathcal{C}_1) = \phi(\mathcal{C}_2) = \mathcal{P}_1$), and $\mathcal{C}_3$ is for the pedestrian ($\phi(\mathcal{C}_3) = \mathcal{P}_2$). The start times for all $\mathcal{C}_i$ are $\tau_s(\mathcal{C}_i) = 1$; the end times $\tau_e(\mathcal{C}_i)$ are still unknown. Each POMDP $\mathcal{C}_i$, with $\phi(\mathcal{C}_i) = \mathcal{P}_j$: $b_i^0 = b_j^0$, $t_i = 1$, and $\pi_i = \pi_j$.

3.2 The MODIA Executor

With DPs and primary actions $\langle \mathcal{P}, \mathcal{A}\rangle$ (known a priori), and online execution of DCs $\langle \mathcal{C}, \phi, \tau\rangle$ (unknown a priori), the primary actions taken from $\mathcal{A}$ are determined by an action **executor** function $\epsilon: \bar{A} \rightarrow \mathcal{A}$ with $\bar{A} = (\bigcup_i A_i)^*$. (Note: X^* is a Kleene operator on a set X, and A_i is the set of actions for the POMDP from DP $\mathcal{P}_i$.) The executor takes DC action recommendations and converts them to a primary action taken by the agent in the external system environment. It also converts a primary action back to what that decision meant to individual DCs via their action sets. In this paper, we use the notation $\epsilon^{-1}: \mathcal{A} \rightarrow \bar{A}$ with $\epsilon_i^{-1}(\mathbf{a})$ referring to an individual $\mathcal{C}_i$'s action from POMDP $\phi(\mathcal{C}_i)$ for some $\mathbf{a} \in \mathcal{A}$.

It is important to note the requirement that the executor function ϵ must be able to map any tuple of actions taken from any combination of DPs, with any number of possible duplicates, to a primary action. MODIA is a class of problems that operates without any knowledge about which (or how many) DPs will be instantiated online.

AV Example (Continued) In Figure 1, all three DCs produce an action $\langle \bar{a}_1, \bar{a}_2, \bar{a}_3\rangle = \bar{a} \in \bar{A}$ at each time step. The example states $\bar{a}_1 = \bar{a}_3 = stop$ and $\bar{a}_2 = go$. The executor ϵ decides from $\bar{a}$ that $stop \in \mathcal{A}$ will be the primary action. It informs each DC $\mathcal{C}_i$ what the primary action means to $\mathcal{C}_i$ individually, simply $\epsilon_i^{-1}(stop) = stop$, for belief updates.

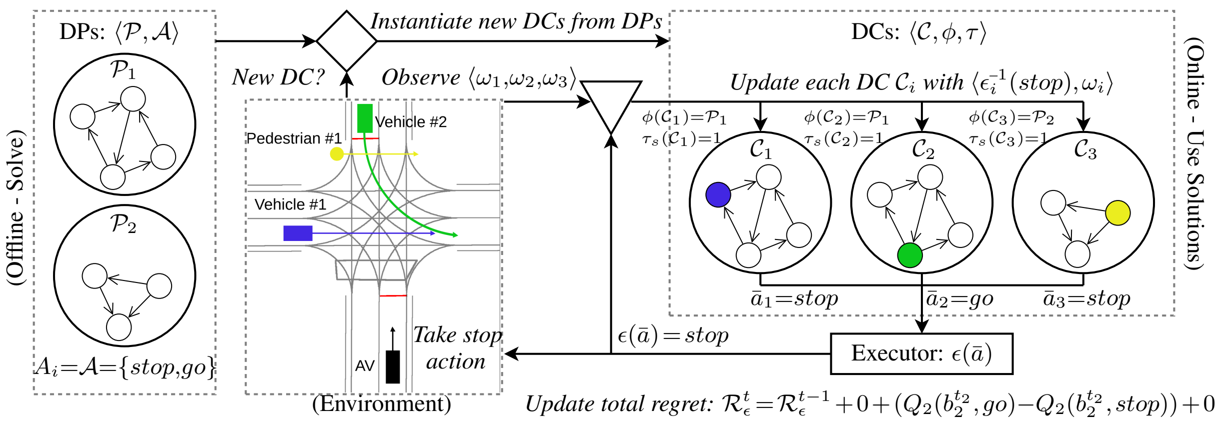

Figure 1: Example visualization of MODIA for AVs. Offline, the DPs (left) are solved: vehicles ($\mathcal{P}_1$) and pedestrians ($\mathcal{P}_2$). Online, the AV approaches an intersection in the environment (center). DCs (right) are instantiated from DPs based on 3 new observations: 2 vehicles ($\mathcal{C}_1$ and $\mathcal{C}_2$) and 1 pedestrian ($\mathcal{C}_3$). Each DC recommends an action ($\bar{a}$): 2 stops and 1 go. The executor decides: stop. The agent takes the action, resulting in regret for $\mathcal{C}_2$'s action in $\mathcal{R}^t_\epsilon$. New observations induce DC updates.

3.3 The MODIA Objective

The goal of the class of problems captured by MODIA is to design the DPs, primary action set, and executor so that it solves the online real-world problem (e.g., AVs). Prior work on *single*-POMDP online algorithms experimentally analyze their performance with simpler metrics such as average discounted reward (ADR) or run time [Somani *et al.*, 2013; Kurniawati and Yadav, 2016], and richer metrics such as error bound reduction (EBR) or lower bound improvement (LBI) [Ross *et al.*, 2008]. MODIA is an online *multi*-POMDP model that differs from these previous online single-POMDP solvers. We instead provide a concrete objective function to enable the analysis of this complex online problem within a theoretical context. Our problem domain does not contain a model for how DPs are instantiated as DCs, nor how long DCs remain active. Thus, the *objective* is to *minimize regret* experienced at each step for any given DC instantiations.

Formally, for $\langle\mathcal{P},\mathcal{A},\mathcal{C},\phi,\tau,\epsilon\rangle$, let $h\leq\tau_e(C_n)$ be a horizon, let $I^t=\{i\in\{1,\ldots,n\}|\tau_s(\mathcal{C}_i)\leq t\leq\tau_e(\mathcal{C}_i)\}$ denote the set of indexes for instantiated DCs, and let executor decision $\epsilon(\bar{a})=\mathbf{a}^t$ at time $t\in\{1,\ldots,h\}$ with primary action $\mathbf{a}^t\in\mathcal{A}$ and the tuple of all instantiated DC's actions $\bar{a}\in\bar{A}$, so for all $i\in I^t$, $\bar{a}_i=\pi_i(b_i^{t_i})$ with π_i, t_i, and $b_i^{t_i}$ from instantiated DC $\mathcal{C}_i\in\mathcal{C}$. The total **regret** $\mathcal{R}^h_\epsilon\in\mathbb{R}$ is:

$$\mathcal{R}^h_\epsilon=\sum_{t=1}^{h}\sum_{i\in I^t}Q_i(b_i^{t_i},\pi_i(b_i^{t_i}))-Q_i(b_i^{t_i},\epsilon_i^{-1}(\mathbf{a}^t)). \quad (1)$$

We refer to the regret at time t for all instantiated DCs in I^t as r^t_ϵ. Informally, a DC's regret in MODIA is the expected reward following the DC's desired policy's action, minus the realized expected reward following the executor's action.

AV Example (Continued) Executor ϵ selected $stop\in\mathcal{A}$, which has $\epsilon_i^{-1}(stop)=stop$ for all $\mathcal{C}_i\in\mathcal{C}$. Following each DC's desired action, only $\mathcal{C}_2$ chose go instead. This induces regret equal to $Q_2(b_2^{t_2},go)-Q_2(b_2^{t_2},stop)\geq 0$; $\mathcal{C}_1$ and $\mathcal{C}_3$ have 0 regret. $\mathcal{R}^t_\epsilon$ is updated accordingly.

3.4 LEAF for MODIA

So far we have described the general form of MODIA using a general executor. Now we examine a particular kind of executor with desirable regret-minimizing properties (shown in Section 4). Specifically, we can define a lexicographic preference over the individual actions suggested by each DC. Thus, each DC suggests an action, stored collectively as a tuple of action recommendations, and the executor only executes the best (in terms of preference) action from this set.

A **lexicographic executor action function (LEAF)** has two requirements regarding a MODIA's structure in $\langle\mathcal{P},\mathcal{A}\rangle$. First, let the primary actions $\mathcal{A}$ be factored with the *unique* action sets from among the DPs; formally, $\mathcal{A}=\bigtimes_i\Lambda_i$ with $\Lambda=\bigcup_j\{A_j\}$. Second, let $\succ_i$ be a lexicographic ordering over actions in these unique action sets $\Lambda_i\in\Lambda$. If a MODIA satisfies these two requirements, then for all $\bar{a}=\langle\bar{a}_1,\ldots,\bar{a}_x\rangle\in\bar{A}$ and $\mathbf{a}=\langle\mathbf{a}_1,\ldots,\mathbf{a}_y\rangle\in\mathcal{A}$, LEAF $\epsilon(\bar{a})=\mathbf{a}$ is defined by:

$$\mathbf{a}_i\succ_i\mathbf{a}, \quad \forall\mathbf{a}\in\{\mathbf{a}'\in\Lambda_i|\exists j \text{ s.t. } \bar{a}_j=\mathbf{a}'\} \quad (2)$$

for all $\Lambda_i\in\Lambda$, and $\epsilon(\emptyset)=\mathbf{a}$ for some fixed $\mathbf{a}\in\mathcal{A}$. Informally, $\bar{a}$ are the current desired actions from DCs, Λ_i is the unique action set, $\mathbf{a}$ are the resulting actions, and each $\mathbf{a}_i$ (from matching unique action set Λ_i) has the highest preference following $\succ_i$ from the available voted-upon actions. Similarly, the inverse executor extracts the relevant action factor taken by the system and distributes it to all DCs who have that action set; formally, for all $\mathcal{C}_i\in\mathcal{C}$, with $\phi(\mathcal{C}_i)=\mathcal{P}_\ell$, there exists an action $\bar{a}_j\in\Lambda_j=A_\ell$ such that for the primary action taken $\mathbf{a}\in\mathcal{A}$, $\epsilon_i^{-1}(\mathbf{a})=\bar{a}_j$. In summary, LEAF simply takes the most preferred action among those available.

AV Example (Continued) In the AV example, we have action sets $\{stop,go\}=A_1=A_2=\mathcal{A}=\Lambda_1$. Thus, it satisfies the first requirement: primary actions are composed of DP actions. For the second, we define a lexicographic preference $\succ_1$ (encouraging safety) over Λ_1 with $stop\succ go$. Now ϵ in Figure 1 is actually LEAF. Namely, the action $stop$ is the most preferred action desired among only the actions selected by the DCs. Thus, $stop$ is the result of the executor.

3.5 Risk-Sensitive MODIA

Now we also consider a specific kind of MODIA, with a form of monotonicity in an ordered relationship over actions and Q-values. Informally, we require DP's Q-values to be monotonic over actions with a penalty for selecting policy-violating high-risk actions. Formally, a MODIA is **risk-sensitive** with respect to a preference $\succ_i$, if for all j, b, a, and a': (1) if $a \succ_i a' \succeq_i \pi_j(b)$ then $Q_j(b,a) \leq Q_j(b,a')$, (2) if $\pi_j(b) \succ_i a$ then $Q_j(b,a) \leq \underline{Q}$ for sufficient penalty $\underline{Q}$.

AV Example (Continued) Action $stop$ makes no progress towards the goal while go does, so long as go is optimal, resulting in (1). Conversely, performing go when $stop$ is optimal produces a severe expected cost, resulting in (2).

4 Theoretical Analysis

Given DPs and primary actions $\langle \mathcal{P}, \mathcal{A} \rangle$, MODIA requires the selection of an executor to minimize regret accumulated over time, in addition to solving the DPs themselves. With n unknown a priori, as well as which and when DPs are instantiated as DCs, it is impossible to perform tractable planning techniques entirely offline; again, MODIA is an category of *online* decision-making scenarios. Assume, however, that a prescient oracle provided $\langle \mathcal{C}, \phi, \tau \rangle$ a priori. While this is an impossible scenario, it is useful to understand the worst-case complexity of exploiting this information in the underlying problem of selecting a regret-minimizing executor given this normally unobtainable information. Proposition 1 formally proves this complexity.

Proposition 1. *If $\langle \mathcal{C}, \phi, \tau \rangle$ is known a priori, then the complexity to compute the optimal executor ϵ^* is $O(n^2 zmh)$ with $z = |\mathcal{A}|$, $m = \max_i |A_i|$, and $h = \max_i \tau_e(\mathcal{C}_i)$.*

Proof. Must determine the worst-case complexity to fully define executor $\epsilon^* : \bar{A} \to \mathcal{A}$ to minimize regret Equation 1. In the worst-case, we must explore all relevant executors, and compute the regret for each, resulting in the optimal solution.

By executor definition in Section 3.2, $\bar{A} = (\bigcup_i A_i)^*$ and $z = |\mathcal{A}|$. Given $n = |\mathcal{C}|$, the maximum *realizable* set size of $\bar{A}$ is all unique potential actions, multiplied by the maximal number of unique DCs instantiated simultaneously. In the worst-case, $A_i \neq A_j$ for all $i \neq j$, so all possible actions must be considered for each; this order bound is $m = \max_i |A_i|$. Also, all combinations of instantiated DCs must be realized, so all $\tau(\mathcal{C}_i) \neq \tau(\mathcal{C}_j)$ for all $i \neq j$. In any order, n births, n deaths, and time no DCs instantiated; thus there are $2n+1$ in total. Hence, the number of potential executors is $O(znm)$.

In the worst-case scenario, $R_i(b_i^{t_i}, \pi_i(b_i^{t_i}))$ differs for every time step for all $\mathcal{C}_i \in \mathcal{C}$. Equation 1 requires $O(h \max_t I^t)$ operations. Given $\mathcal{C}$, $h = \max_i \tau_e(\mathcal{C}_i)$. By definition of I^t, $\max_t I^t \leq n$. Thus, the worst-case complexity to compute an optimal ϵ^* is $O(znm) \cdot O(hn) = O(n^2 zmh)$. □

With Proposition 1, we know this impossible oracular scenario's complexity is relatively high, but not exponential. This suggests a method for computing an optimal executor, under more realistic assumptions. Thus, let $\hat{\rho}$ be a given *model* for the hardest feature of MODIA: *online instantiation*. Let $\hat{\rho} : \hat{N}_n \times \hat{T}_n \times \hat{E}_n \times \hat{N}_n \times \hat{T}_n \to [0,1]$ define the probability that a particular set of instantiated DCs $\langle \hat{n}, \hat{\tau} \rangle \in \hat{N}_n \times \hat{T}_n$, and executor selection $\hat{\epsilon} \in \hat{E}_n$, results in a successor DC instantiation state $\langle \hat{n}', \hat{\tau}' \rangle \in \hat{N}_n \times \hat{T}_n$. Here, $\hat{N}_n = \{1, \ldots, k\}^n$ are instantiation indexes (defining ϕ), $\hat{T}_n = \{\hat{\tau} \in \{\alpha, \{1, \ldots, h\}^2, \omega\}^n | \forall i \in \mathbb{N}, \hat{\tau}_{is} < \hat{\tau}_{ie}\}$ are the instantiation start and end times (defining τ) including non-instantiated α and completed ω demarcations, and $\hat{E}_n = \{\epsilon : \bar{A} \to \mathcal{A} | |\bar{A}| \leq n\}$ are all valid executors (defining ϵ). Additionally, we must assume knowledge of a maximum number of DCs n and horizon h for decidability. Given this model, Proposition 2 proves the resulting MDP's optimal policy minimizes *expected* regret, and that the problem is unfortunately computationally intractable in practice.

Proposition 2. *If n, h, and model $\hat{\rho}$ are known a priori, then: (1) the resulting MDP's optimal policy π^* minimizes expected regret, and (2) its state space is exponential in n and k.*

Proof. We must show the construction of an MDP whose optimal policy minimizes expected regret and show its complexity in the *necessity* of an exponential state space.

Let $\langle \hat{S}, \hat{A}, \hat{T}, \hat{R} \rangle$ be a finite horizon MDP with horizon $\hat{h} = h+1$. States are $\hat{S} = \{\hat{s}^0\} \cup \hat{E}_n \times \hat{B}_n \times \hat{T}_n$ with $\hat{s}^0$ denoting the initial executor selection state and $\hat{B}_n = \{\hat{B} \in (\bigcup_i \hat{B}_i^h)^* | |\hat{B}| = n\}$ be all possible reachable beliefs for $\mathcal{P}_i$ in horizon h (denoted $\hat{B}_i^h$) for all possible instantiations. For notation, we use $\hat{s} = \langle \hat{\epsilon}, \hat{b}, \hat{\tau} \rangle$, each containing instantiated values $\hat{\epsilon}_i$, $\hat{b}_i$, $\hat{\tau}_{si}$, and $\hat{\tau}_{ei}$, as well as $\hat{\theta} : \hat{B}_n \to \hat{N}_n$ mapping beliefs to their original POMDPs' indices. Actions are executor selection $\hat{A} = \hat{E}_n$. State transitions $\hat{T} : \hat{S} \times \hat{A} \times \hat{S} \to [0,1]$ have two cases. First, $\hat{T}(\hat{s}^0, \hat{a}, \hat{s}') = [\hat{s}' = \langle \hat{a}, \emptyset, \emptyset \rangle]$ captures executor selection. Second, for $\hat{s} \neq \hat{s}^0$ we have:

$$\hat{T}(\hat{s}, \hat{a}, \hat{s}') = [(\hat{s} = \hat{s}^0 \wedge \hat{\epsilon}' = \hat{a}) \vee (\hat{s} \neq \hat{s}^0 \wedge \hat{\epsilon}' = \hat{\epsilon})]$$
$$\cdot \hat{\rho}(\hat{\theta}(\hat{b}), \hat{\tau}, \hat{\epsilon}, \hat{\theta}(\hat{b}'), \hat{\tau}') \prod_{i=1}^{n} [\hat{b}_i' = b_j^0 \wedge \hat{\tau}_i = \alpha \wedge \hat{\tau}_i' = 1]$$
$$\cdot \prod_{i=1}^{n} Pr(\hat{b}_i' | \hat{b}_i, \pi_j(\hat{b}_i)) [\hat{\tau}_i \in \mathbb{N} \wedge \hat{\tau}_i' = \hat{\tau}_i + 1] \prod_{i=1}^{n} [\hat{\tau}_i' = \omega \wedge \hat{b}_i' = \hat{b}_i]$$

with $j = \hat{\theta}_i(\hat{b})$. This captures executor state assignment, the instantiation model $\hat{\rho}$, the proper initialization of belief, the belief update for active DCs, and the termination of a DC. Rewards $\hat{R} : \hat{S} \times \hat{A} \to \mathbb{R}$ describe the negative regret, $\hat{R}(\hat{s}, \hat{a}) = \sum_i Q_j(\hat{b}_i, \hat{\epsilon}_i^{-1}(\mathbf{a}^t)) - Q_j(\hat{b}_i, \pi_j(\hat{b}_i))[\hat{\tau}_i \in \mathbb{N}]$ with $R(\hat{s}^0, \hat{a}) = 0$. By construction, this is MODIA, assuming $\hat{\rho}$, n, and h were provided. By assigning $\epsilon^* = \pi^*(\hat{s}^0)$, we minimize expected regret. In the worst-case, it necessitates modeling all n DC instantiation permutations (with replacement) of the k DPs, which is $O(k^n)$. □

This illustrates the importance of the original MODIA formulation. Even with the instantiation model of Proposition 2, the problem is still unscalable. And the knowledge needed to bound the number of active DCs (e.g., n and h) is generally unavailable a priori. This intrinsic lack of information motivated our formulation that minimizes the regret at each time step. Hence, the agent is guided by the optimal

DC policies from each instantiated DP, selecting the regret-minimizing action at each time step. Proposition 3 proves that LEAF minimizes the regret in risk-sensitive MODIA at each time step, enabling a tractable solution to MODIA.

Proposition 3. *If a MODIA is risk-sensitive, then LEAF minimizes regret r_ϵ^t for all t.*

Proof. By definition of regret r_ϵ^t for LEAF ϵ at time step t: $r_\epsilon^t = \sum_i Q_j(b_i^{t_i}, \pi_j(b_i^{t_i})) - Q_j(b_i^{t_i}, \epsilon_i^{-1}(\mathbf{a}^t))$ with $\phi(\mathcal{C}_i) = \mathcal{P}_j$. We must show for all $\tilde{\epsilon}$, $r_\epsilon^t \leq r_{\tilde{\epsilon}}^t$. For readability, hereafter, let $a_i = \epsilon_i^{-1}(\mathbf{a}^t)$, $\tilde{a}_i = \tilde{\epsilon}_i^{-1}(\tilde{\mathbf{a}}^t)$, $a_j^* = \pi_j(b_i^{t_i})$, and $b_i = b_i^{t_i}$. By definition of risk-sensitive, there always exists action a_j^* such that $Q_j(b_i, a_j^*) \geq \underline{Q}$. Thus, it is sufficient to show that for all $i \in I^t$, $Q_j(b_i, a_i) \geq Q_j(b_i, \tilde{a}_i)$, or there exists a $\mathcal{C}_i \in \mathcal{C}$ with $\phi(\mathcal{C}_i) = \mathcal{P}_j$ such that $Q_j(b_i, \tilde{a}_i) \leq \underline{Q}$. By risk-sensitivity and LEAF, consider 3 cases for ϵ and $\tilde{\epsilon}$.

Case 1: $a_i =_x \tilde{a}_i$ for $a_i, \tilde{a}_i \in \Lambda_x = A_j$. Trivially, we have $Q_j(b_i, a_i) = Q_j(b_i, \tilde{a}_i)$.

Case 2: $a_i \succ_x \tilde{a}_i$ has two cases. *Case 2.a*: If $a_i = a_j^*$, then by definition π_j's optimality, for any $\tilde{a}_i \in A_j$, $Q_j(b_i, a_i) = Q_j(b_i, a_j^*) \geq Q_j(b_i, \tilde{a}_i)$. *Case 2.b*: If $a_i \neq a_j^*$, then by LEAF Equation 2, $a_i \in \{a \in \Lambda_x | \exists u \text{ s.t. } \bar{a}_u = a\}$. Thus, by definition of $\bar{a} \in \bar{A}$, there exists this $u \neq i$ such that $a_i = a_u = a_v^*$ with $\phi(\mathcal{C}_u) = \mathcal{P}_v$. By risk-sensitivity, $a_v^* = a_u = a_i \succ_x \tilde{a}_i$ that implies $Q_v(b_u, \tilde{a}_i) \leq \underline{Q}$.

Case 3: $a_i \prec_x \tilde{a}_i$. By definition of risk-sensitivity, we have $\tilde{a}_i \succ_x a_i \succeq_x a_j^*$ and consequently $Q_j(b_i, \tilde{a}_i) \leq Q_j(b_i, a_i)$.

All cases proven. LEAF minimizes regret r_ϵ^t for any t. □

5 Application to Autonomous Vehicles

We apply MODIA and LEAF to this concrete problem of AV decision-making at intersections. The formulation expands on the numerous AV examples described in Section 3. Due to space considerations, we focus our attention strictly on defining vehicle-related DP (POMDP); however, pedestrian and other DPs follow in a similar manner. Overall, this AV robotic application serves to both ground our theoretical work and simultaneously present an actual solution to intersection decision-making in the real world.

The MODIA AV $\langle \mathcal{P}, \mathcal{A} \rangle$ defines $\mathcal{P}$ by converting *intersection types* (and *pedestrian types*) into POMDP DP. These *types* capture the static abstracted information. For example, intersection types contain features such as the number of road segments, lane information (incoming and outgoing), crosswalk locations, and traffic controller information. A DP is created for all lanes within all intersection types (and pedestrian types). Formally, for each such vehicle and intersection type, we define the DP POMDP $\langle S_i, A_i, \Omega_i, T_i, O_i, R_i \rangle = \mathcal{P}_i \in \mathcal{P}$. $S_i = S_{av}^\ell \times S_{av}^t \times S_{ov}^\ell \times S_{ov}^t \times S_{ov}^b \times S_{ov}^p$ describes the AV's location (approaching/at/edged/inside/goal) and time spent at location (short/long), as well as the other vehicle's location (approaching/at/edged/inside/empty), time spent at location (short/long), blocking (yes/no), and priority at intersection in relation to AV (ahead/behind), respectively. Actions are simply $A_i = \{stop, edge, go\}$, and encode movement by assigning desired velocity and goal points along the AV's trajectory within the intersection. Lower-level nuances in path planning [Wray *et al.*, 2016b] are optimized by other methods. $\Omega_i = \Omega_{av}^t \times \Omega_{av}^b \times \Omega_{ov}^t \times \Omega_{ov}^b$ primarily encode the noisy sensor updates in blocking detection (yes/no) but also if the time spent was updated (yes/no) for both the AV and other vehicle. $T_i : S_i \times A_i \times S_i \to [0,1]$ multiply the probabilities of a wide range of situations quantifiable and definable in the state-action space described. This includes multiplying probabilities for: (1) vehicle kindly lets AV have priority, (2) vehicle cuts AV off, (3) AV's success or failure of motion to an abstracted state based on its physical size, (4) a new vehicle arrives at an intersection lane, (6) time increments, (7) vehicle actually stops at stop sign or does a rolling stop, (8) vehicle is blocking the AV's path following the static intersection type's road structure, etc. Additionally, a dead end state (an absorbing non-goal self-loop) is reached when the AV and other vehicle both have state factor "inside" while also "blocking" each other. $O_i : A_i \times S_i \times \Omega_i \to [0,1]$ captures the sensor noise (e.g., determined via calibration and testing of the AV's sensors). This includes successful detections of: (1) other vehicle's crossing of physical locations mapped to abstracted states, (2) determining the blocking probability based on the location of the other vehicle, etc. $R_i : S_i \times A_i \to \mathbb{R}$ is defined as unit cost for all states, except the goal state.

The primary actions are $\mathcal{A} = \{stop, edge, go\}$ and simply describe the AV's movement along the desired trajectory. We define a lexicographic preference $\succ_1$ over this action set $stop \succ_1 edge \succ_1 go$. This preference formalizes the notion that if even one DC said to stop, then the AV should stop. Similarly, if at least one DC said to edge but none said stop, then the AV should cautiously edge forward. Otherwise, the AV should go. This enables us to apply LEAF because $A_i = \mathcal{A}$ for all A_i (even the pedestrian DPs) and we have lexicographic preference $\succ_1$. Lastly, the defined MODIA produces Q-values that satisfy risk-sensitivity.

6 Experimentation

We begin with experiments on six different intersections in an industry-standard vehicle simulation developed by Realtime Technologies, Inc. that accurately simulates vehicle dynamics with support for ambient traffic and pedestrians. We evaluated MODIA on real map data at six different intersections, each highlighting a commonly encountered real-world scenario. Table 1 describes each scenario by name and provides details regarding the road segments, vehicles, and pedestrians that exist. The number of potential incidents describes how many risks exist, which MODIA perfectly obviates. We compare a MODIA AV with *ignorant* and *naive* AV baseline algorithms. The ignorant AV follows the law but ignores the existence of all vehicles and pedestrians, acting as if the intersections are empty. The naive AV follows the law and cautiously waits until all others have cleared the intersection beyond 15 meters before attempting to go. These two baselines implement extremes of rule-based AVs [Jo *et al.*, 2015] and serve as a form of bound for AV behavior to understand MODIA AV's performance. We evaluate each by their time to complete an intersection, which includes the observations while approaching, decisions at the intersection, and travel within the intersection. In Table 1, we observe the MODIA AV successfully completes intersections faster than the cau-

Intersection Scenarios					MODIA		Baselines	
Name	RS	V	P	PI	$\|\mathcal{C}\|$	$\mathcal{M}$	$\mathcal{I}$	$\mathcal{N}$
Crosswalk Pedestrian	4	0	1	1	4	21.1	16.7	30.1
Vehicle & Pedestrian	3	1	1	1	3	16.8	13.6	37.1
Walk & Run Pedestrians	3	1	2	2	6	19.1	13.3	23.3
Multi-Vehicle Interaction	4	2	0	2	5	19.0	13.2	20.9
Bike Crossing	3	0	1	1	3	16.4	13.8	19.8
Jay Walker	4	0	1	1	4	17.7	14.4	24.3

Table 1: Results for six intersection problems described by the number of road segments (RS), vehicles (V), pedestrians (P), and potential incidents (PI). MODIA AV $\mathcal{M}$ (number of DCs $|\mathcal{C}|$) is compared with two baselines, ignorant $\mathcal{I}$ and naive $\mathcal{N}$, using their intersection completion times (seconds).

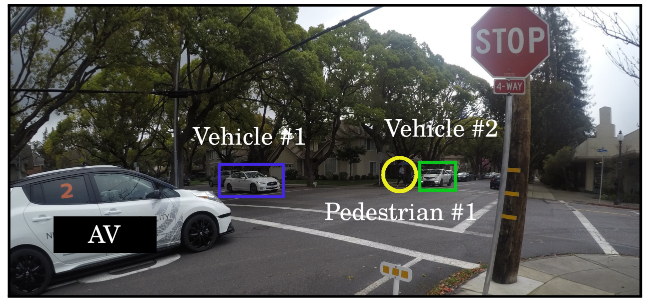

Figure 2: Our fully-operational AV prototype at a 4-way stop intersection that implements AV MODIA and LEAF.

tious naive AV. While the MODIA AV takes longer than the ignorant AV, the ignorant AV encounters each potential incident and the MODIA AV safely avoids them.

Figure 2 depicts a common 4-way intersection with our fully-operational AV prototype, which operates on real public roads and contains an implementation of MODIA and LEAF. This real-world scenario illustrates MODIA's success in addressing scalability concerns while simultaneously handling the nuanced aspects of online decision-making. Each described vehicle DP POMDP has 400 states (265 with additional pruning), with a rich well-structured belief space. In MODIA AVs, the POMDP's size is constant and applies to any intersection. In comparison, a single all-encompassing POMDP with these state factors quickly becomes utterly infeasible, and will vary greatly among intersections. For example, the 4-way stop from Figure 2 that only considers the AV and 3 other vehicles (no pedestrians) would the state space $S = S_{av}^{\ell} \times S_{av}^{t} \times_{i=1}^{3} (S_{ovi}^{\ell} \times S_{ovi}^{t} \times S_{ovi}^{b} \times S_{ovi}^{p})$. This has $|S| = 640{,}000$ states, exemplifying notions from Proposition 2. Conversely, MODIA AVs scale *linearly* with the number of vehicles, and would only be 795 states evenly distributed over three POMDPs. On modest hardware, a DP can take <1 minute to solve using *nova* [Wray and Zilberstein, 2015]. Monolithic POMDPs, like the one described, are unequivocally intractable; however, MODIA enables the now realized POMDP solution for AV decision-making.

7 Related Work

Previous work on an general models related to MODIA include architectures for mobile robots [Brooks, 1986; Rosenblatt, 1997] or other systems [Decker, 1996], and contain decision-components that produce actions, aggregated to a system action. They do not, however, naturally model uncertainty or have a general theoretical grounding. Forms of hierarchicies include action-based execution of child problems with multi-options [Barto and Mahadevan, 2003] and abstract machines [Parr and Russell, 1998]. Action-space partitioning that execute smaller MDPs [Hauskrecht *et al.*, 1998] and POMDPs [Pineau *et al.*, 2001] also exists. These do not model the online execution of an unknown number of decision-components for use in robotics. More application-focused work on action voting for simple POMDPs to solve intractable POMDPs have been used successfully [Yadav *et al.*, 2015]. Robotic applications of hierarchical POMDPs for an intelligent wheelchair decompose the problem into components [Tao *et al.*, 2009], or with two POMDP levels for vision-based robots [Sridharan *et al.*, 2010]. These practical methods work well but lack generalized mathematical foundations. Also, none of these present AV-specific solutions.

Previous work specific to AV decision-making includes simple rule-based or finite-state controller systems [Jo *et al.*, 2015], which are simple to implement but are brittle, difficult to maintain, and were unable to handle the abundant uncertainty in AV decision-making. Initial attempts using deep neural networks map raw images to control [Chen *et al.*, 2015] are slow to train and tend to fail rapidly when presented with novel situations. Mixed-observability MDPs for pedestrian avoidance also successfully use a decision-component approach (AV-pedestrian pairs) but provide limited theoretical work and do not extend to intersections [Bandyopadhyay *et al.*, 2013]. Using a single POMDP for all decision-making has been explored, including continuous POMDPs using raw spacial coordinates for mid-level decision-making [Brechtel *et al.*, 2014], online intention-aware POMDPs for pedestrian navigation [Bai *et al.*, 2015], and POMDPs for lane changes that use online approximate lookahead algorithms [Ulbrich and Maurer, 2013]. These approaches do not address the exponential complexity concerns (scalability), provide generalizable theoretical foundations, or enable simultaneous seamless integration of multiple different decision-making scenarios on a real AV, all of which are provided by MODIA.

8 Conclusion

MODIA is a principled theoretical model designed for direct practical use in online decision-making for autonomous robots. It has a number advantages over the direct use of a massive monolithic POMDP for planning and learning. Namely, it remains tractable by growing linearly in the number of decision-making problems encountered. Its component-based form simplifies the design and analysis, and enables provable theoretical results for this class of problems. MODIA is shown to successfully solve a challenging AV interaction problem. Future work will explore more executors and models beyond LEAF and risk-sensitive MODIA, develop additional AV-related DPs, and tackle other intractable robotic domains such as humanoid service robots using MODIA as a scalable online decision-making solution.

Acknowledgments

We thank the reviewers for their helpful comments, Liam Pedersen for valuable discussions, and Nissan Motor Co., Ltd. for supporting this work.

References

[Bai *et al.*, 2015] Haoyu Bai, Shaojun Cai, Nan Ye, David Hsu, and Wee Sun Lee. Intention-aware online POMDP planning for autonomous driving in a crowd. In *IEEE Int'l. Conf. on Robotics and Automation (ICRA)*, pages 454–460, 2015.

[Bandyopadhyay *et al.*, 2013] Tirthankar Bandyopadhyay, Chong Zhuang Jie, David Hsu, Marcelo H. Ang, Daniela Rus, and Emilio Frazzoli. Intention-aware pedestrian avoidance. In *Proc. of the 13th Int'l. Symposium on Experimental Robotics*, pages 963–977, 2013.

[Barto and Mahadevan, 2003] Andrew G. Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(4):341–379, 2003.

[Brechtel *et al.*, 2014] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs. In *Proc. of the 17th Int'l. IEEE Conf. on Intelligent Transportation Systems (ITSC)*, pages 392–399, 2014.

[Brooks, 1986] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23, 1986.

[Chen *et al.*, 2015] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. DeepDriving: Learning affordance for direct perception in autonomous driving. In *Proc. of the 15th IEEE Int'l. Conf. on Computer Vision (ICCV)*, pages 2722–2730, 2015.

[Decker, 1996] Keith Decker. TAEMS: A framework for environment centered analysis & design of coordination mechanisms. *Foundations of Distributed Artificial Intelligence*, pages 429–448, 1996.

[Dolgov *et al.*, 2010] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. *The Int'l. Journal of Robotics Research*, 29(5):485–501, 2010.

[Hauskrecht *et al.*, 1998] Milos Hauskrecht, Nicolas Meuleau, Leslie Pack Kaelbling, Thomas Dean, and Craig Boutilier. Hierarchical solution of Markov decision processes using macro-actions. In *Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 220–229, 1998.

[Jo *et al.*, 2015] Kichun Jo, Junsoo Kim, Dongchul Kim, Chulhoon Jang, and Myoungho Sunwoo. Development of autonomous car–Part II: A case study on the implementation of an autonomous driving system based on distributed architecture. *IEEE Transactions on Industrial Electronics*, 62(8):5119–5132, 2015.

[Kaelbling *et al.*, 1998] Leslie P. Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998.

[Kurniawati and Yadav, 2016] Hanna Kurniawati and Vinay Yadav. An online POMDP solver for uncertainty planning in dynamic environment. In *Proc. of the 16th Int'l. Symposium Robotics Research (ISRR)*, pages 611–629, 2016.

[Parr and Russell, 1998] Ronald Parr and Stuart Russell. Reinforcement learning with hierarchies of machines. In *Proc. of the 10th Conf. on Advances in Neural Information Processing Systems (NIPS)*, pages 1043–1049, 1998.

[Pineau *et al.*, 2001] Joelle Pineau, Nicholas Roy, and Sebastian Thrun. A hierarchical approach to POMDP planning and execution. In *Proc. of the ICML Workshop on Hierarchy and Memory in Reinforcement Learning*, 2001.

[Rosenblatt, 1997] Julio K. Rosenblatt. DAMN: A distributed architecture for mobile navigation. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3):339–360, 1997.

[Ross *et al.*, 2008] Stéphane Ross, Joelle Pineau, Sébastien Paquet, and Brahim Chaib-Draa. Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, 32:663–704, 2008.

[Sivaraman and Trivedi, 2013] Sayanan Sivaraman and Mohan M. Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1773–1795, 2013.

[Somani *et al.*, 2013] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. DESPOT: Online POMDP planning with regularization. In *Proc. of the 26th Conf. on Advances in Neural Information Processing Systems (NIPS)*, pages 1772–1780, 2013.

[Sridharan *et al.*, 2010] Mohan Sridharan, Jeremy Wyatt, and Richard Dearden. Planning to see: A hierarchical approach to planning visual actions on a robot using POMDPs. *Artificial Intelligence*, 174(11):704–725, 2010.

[Tao *et al.*, 2009] Yong Tao, Tianmiao Wang, Hongxing Wei, and Diansheng Chen. A behavior control method based on hierarchical POMDP for intelligent wheelchair. In *Proc. of IEEE/ASME Int'l. Conf. on Advanced Intelligent Mechatronics (AIM)*, pages 893–898, 2009.

[Thrun *et al.*, 2006] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.

[Ulbrich and Maurer, 2013] Simon Ulbrich and Markus Maurer. Probabilistic online POMDP decision making for lane changes in fully automated driving. In *Proc. of the 16th Int'l. IEEE Conf. on Intelligent Transportation Systems (ITSC)*, pages 2063–2067, 2013.

[Wray and Zilberstein, 2015] Kyle H. Wray and Shlomo Zilberstein. A parallel point-based POMDP algorithm leveraging GPUs. In *Proc. of the 2015 AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents*, pages 95–96, 2015.

[Wray *et al.*, 2016a] Kyle H. Wray, Luis Pineda, and Shlomo Zilberstein. Hierarchical approach to transfer of control in semi-autonomous systems. In *Proc. of the 25th Int'l. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 517–523, 2016.

[Wray *et al.*, 2016b] Kyle H. Wray, Dirk Ruiken, Roderic A. Grupen, and Shlomo Zilberstein. Log-space harmonic function path planning. In *Proc. of the 29th IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems (IROS)*, pages 1511–1516, 2016.

[Yadav *et al.*, 2015] Amulya Yadav, Leandro S. Marcolino, Eric Rice, Robin Petering, Hailey Winetrobe, Harmony Rhoades, Milind Tambe, and Heather Carmichael. Preventing HIV spread in homeless populations using PSINET. In *Proc. of the 27th Conf. on Innovative Applications of Artificial Intelligence (IAAI)*, pages 4006–4011, 2015.