Optimal Control in Partially Observable Complex Social Systems

Fan Yang
University at Buffalo
Buffalo, New York
fyang24@buffalo.edu

Bruno Lepri
FBK
Trento, Italy
lepri@fbk.eu

Wen Dong
University at Buffalo
Buffalo, New York
wendong@buffalo.edu

ABSTRACT

We live in a world full of complex social systems. Achieving optimal control in a complex social system is challenging due to the difficulty in modeling and optimization. To capture the complex social system dynamics accurately and succinctly, we model the decision-making problem as a partially observable discrete event decision process. To withstand the curse of dimensionality in high-dimensional belief state spaces and to optimize the problem in an amenable searching space, we investigate the connections between the value function of a partially observable decision process and that in the corresponding fully-observable scenario, and reduce the optimal control of a partially observable discrete event decision process to a policy optimization with a specially formed fully observable decision process and a belief state estimation. When tested in real-world transportation scenarios, in comparison with other state-of-the-art approaches, our proposed algorithm leads to the least average time on-road, the largest number of vehicles at work during work hours and the fewest training epochs to converge to the highest total rewards per episode.

ACM Reference Format:
Fan Yang, Bruno Lepri, and Wen Dong. 2020. Optimal Control in Partially Observable Complex Social Systems. 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

Organizations, cities, and more in general human societies are all examples of complex social systems, containing a large number of components that interact with each other and trigger the change of the system state [20]. For example, the transportation system is a complex one where different traffic conditions are formed through the movements and the interactions of each vehicle. An epidemic can be described as a complex system where a disease is spread through social face-to-face and proximity interactions among different people.

However, achieving optimal control in complex social systems remains a difficult problem, mainly due to two reasons: the difficulties in modeling and the difficulties in optimization. First, complex social systems usually contain a high-dimensional state space, complex state transitions, and only partial observability. It is still a challenge on how to accurately and succinctly modeling the dynamics of complex social systems. Second, complex social systems usually have sophisticated spatial-temporal dependencies, where different components could affect each other, and early decisions may have huge effects in the future. For example, in a transportation system, traffic congestion is formed when multiple vehicles simultaneously are choosing the same road, which will delay the driving time for others driving on the congested road, and which may take hours to dissipate. Due to the complicated spatial-temporal dependencies, it is not easy to optimize a complex social system optimal control problem.

In the literature, there are mainly two kinds of approaches to solve decision-making problems for complex social systems [12]. One kind is the simulation-based methods which reproduce the microscopic interactions through a simulation model, and use sampling-based algorithms to optimize the decision making process [17, 23, 25, 29]. These approaches generally can model the system dynamics with high fidelity but have difficulty in optimization due to the high variance. The other kind is the analytical-based methods, which depict the macroscopic dynamics of the system analytically, and use constrained optimization methods to solve the problem [7, 16]. However, when the system becomes complicated, which is often the case in real world, these methods often involve issues with modeling errors. Recently, Yang, Liu, and Dong have combined simulation and analytical approaches to propose a discrete event decision process (DEDP) which exhibits promising results in a real-world transportation scenario [32]. This is a great attempt, however, their approach assumes a decision making process able to acquire a full observation of the entire system.

To model the complex social system succinctly and accurately, we take inspiration from the work of Yang, Liu, and Dong [32], but we formulate the decision-making process in a social system as a partially observable discrete event decision process (PODEDP). Intuitively, their DEDP solution assumes perfect information, so it never performs information-gathering actions, and thus the policy heuristic formed with their solution is sub-optimal by acting overoptimistic in a PODEDP environment such the one of a real-world social system (e.g. a real-world transportation scenario). Moreover, compared with a partially observable Markov decision process (POMDP), a PODEDP describes the system transition dynamics more succinctly and accurately through a discrete event model [4, 14, 30] that captures the dynamics using a simulation process over microscopic interaction events of its components. We demonstrate this improvement through a comparison with an analytical approach based on a POMDP in the domain of transportation optimal control.

Directly solving a PODEDP in a complex social system is prohibitive due to the exploding state action spaces. To solve a PODEDP efficiently, we establish a connection between the value function in a PODEDP and that in a specially formed DEDP (SDEDP). With this connection, we split the process of optimizing the policy parameters, and the procedure of gathering information from partial observations and reduce the optimal control of a PODEDP to a SDEDP policy optimization and belief state estimation. To optimize

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.

the SDEDP policy parameters with formidable state space, we apply Taylor approximation to simplify the trajectory distribution, using a duality theorem [32] to recast optimal control as parameter learning, and introducing an approximate inference algorithm with marginal transition kernel. To track the belief state in complex social systems with exploding state space, we apply variational inference and maximize the variational lower bound [28]. We demonstrate that our method leads to a less-variance and a higher-performance solution in benchmarking with other simulation approaches in experimenting with traffic control problems.

This paper makes the following contributions: first of all, we model the complex social system dynamics succinctly and accurately with a PODEDP. Second, we develop a tractable solution to the PODEDP through a SDEDP policy optimization and a belief state estimation. Finally, we evaluate our proposed framework on a real-world transportation scenario and we demonstrate that PODEDP achieves higher system expected rewards, faster convergence, and lower variance of value function when compared with other state-of-the-art analytical and sampling approaches.

2 BACKGROUND

In this section, we review the discrete event model, the discrete event decision process (DEDP), and the duality theorem.

2.1 Discrete Event Model

A discrete event model specifies the dynamics of complex social systems through a sequence of stochastic events which individually described the microscopic state transitions, and which aggregated together described the macroscopic system state changes. The discrete event model is used to specify the dynamics of engineered systems (where it is known as a stochastic Petri-net [14]), of biochemical networks (where it is known as a stochastic kinetic model [30]), and of social networks (where it is known as discrete event simulation [4]). The discrete event model is also used in machine learning to infer the probability distribution of the latent state [6, 19, 31, 33] and learn complex system dynamics from noisy observations [3, 27]. However, the potential of a discrete event model has been only recently explored in optimizing the decision making processes and interventions in complex social systems [32].

In the following, we introduce the stochastic kinetic formulation of a discrete event model [30] as the foundation for formulating a (partially observable) discrete event decision process. A stochastic kinetic model is a biochemist's tool for capturing the control in a biological network through a set of chemical reactions [1]. Specifically, in a system with M species and V events, an event (chemical reaction) v is in the form of a production $\alpha_{v,1}\mathbf{X}^{(1)} + \cdots + \alpha_{v,M}\mathbf{X}^{(M)} \xrightarrow{c_v} \beta_{v,1}\mathbf{X}^{(1)} + \cdots + \beta_{v,M}\mathbf{X}^{(M)}$, where $\mathbf{X} = (\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(M)})$ denotes individuals belonging to the M species in the system. The production is interpreted as having rate coefficient c_v (probability per unit time, as time goes to 0) as well as $\alpha_{v,1}$ individuals of species 1, $\alpha_{v,2}$ individuals of species 2, ..., $\alpha_{v,M}$ individuals of species M interacting according to event v, that results in their removal from the system. At the same time, $\beta_{v,1}$ individuals of species 1, $\beta_{v,2}$ individuals of species 2, ..., $\beta_{v,M}$ individuals of species M are introduced into the system. As such, event v changes the populations by $\Delta_v = (\beta_{v,1} - \alpha_{v,1}, \cdots, \beta_{v,M} - \alpha_{v,M})$. In a typical system, each event usually involves only a few species, i.e., most of $\alpha_{v,1}$, $\beta_{v,1}$, ... are 0 for each v, resulting in a sparse network of interactions.

2.2 Discrete Event Decision Process

Recently, Yang, Liu, and Dong has applied the discrete event model in fully observed social system optimal control [32], where they formulated the optimal control problem using a discrete event decision process DEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$. In the above, S and A represents the state and action spaces with states $s_t = \left(s_t^{(1)}, ..., s_t^{(M)}\right) \in S$ and action $a_t = \left(a_t^{(1)}, ..., a_t^{(D)}\right) \in A$. $\mathcal{V} = \{\emptyset, 1, ..., V\}$ is the set of events and $v_t \in \mathcal{V}$ indicates the event taking at time t which changes system state by Δ_{v_t}, C is the function mapping event rate coefficients to actions $\boldsymbol{c} = (c_1, ..., c_V) = C(a_t)$. Using the stochastic kinetic formulation of a discrete event model, the transition kernel P specified by events $v_t \in \mathcal{V}$ associated with event rate coefficients c_v is defined as $P(s_{t+1}, v_t|s_t, a_t) = p(v_t \mid s_t, a_t)\delta_{s_{t+1}=s_t+\triangle v_t}$, where δ is an indicator function, and event v_t changes the system state by Δ_{v_t}. The probability of a event v_t happened conditioned on state s_t and action a_t could be represented as

$$p(v_t \mid s_t, a_t) = \begin{cases} h_v(s_t, c_v) & \text{if } v_t \neq \emptyset \\ 1 - \sum_{v=1}^{V} h_v(s_t, c_v) & \text{if } v_t = \emptyset \end{cases}$$
$$\text{where } h_v(s_t, c_v) = c_v \prod_{m=1}^{M} g_v^{(m)}(s_t^{(m)})$$

In the above, the event rate $h_v(s_t, c_v)$ is computed as the probability of one single event happens (event rate coefficient) c_v times a total of $\prod_{m=1}^{M} g_v^{(m)}(s_t^{(m)})$ ways for different individuals to meet. R represents the reward function $R(s_t) = \sum_{m=1}^{M} R_t^{(m)}(s_t^{(m)})$, and γ is the discount factor. A policy is defined deterministically as $a_t = \mu(s_t; \theta)$ or stochastically as $\pi = p(a_t|s_t; \theta)$. For a deterministic policy, the probability measure to a sample path is

$$p(\xi_T) \quad = p(s_0) \prod_{t=0}^{T-1} \left(p(v_t \mid s_t, \mu(s_t))\delta_{s_{t+1}=s_t+\Delta_{v_t}}\right)$$

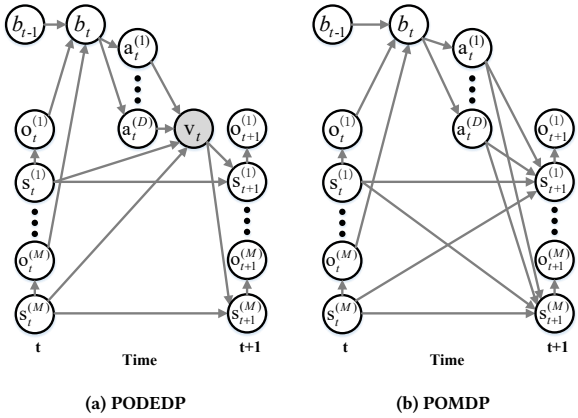
2.3 Duality Theorem

To cope with the high-dimensional state-action spaces, [32] developed a convex conjugate duality theorem between the log value (expected future reward) and the entropy of a proposal distribution. Specifically, in a DEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$, let T be a discrete time, m the component index, H the entropy function, and $r(T, m, \xi_T; \pi) = \gamma^T P(\xi_T; \pi) R_T^{(m)}(s_T^{(m)})$ the reward-weighted trajectory induced by policy π; define a proposal distribution $q(T, m, \xi_T)$ over the trajectories, then the following duality theorem is established.

***Theorem* 1.**

$$\log V^{\pi}(r) = \sup_{q} \left(\sum_{T,m,\xi_T} q(T, m, \xi_T) \log r(T, m, \xi_T; \pi) + \mathrm{H}(q) \right)$$

Theorem 1 reduces the policy evaluation and improvement in a policy optimization procedure into probabilistic inference and parameter learning [28]. With this Theorem, a DEDP could be solved using a policy-iteration paradigm around the duality form: solving the variational problem as policy evaluation and optimizing the target over parameter θ with a known mixture of finite-length trajectories as policy improvement.



(a) PODEDP (b) POMDP

Figure 1: The two time-slice structure of a PODEDP (left) and a POMDP (right). A PODEDP captures complex system dynamics by factoring complex interactions into a sequence of elementary events v_t (shaded node in the left panel)

3 METHODOLOGY

The two challenges for optimal control in complex social systems are the difficulty in modeling the complex dynamics and the difficulty in optimization. In this section, we develop a partially observable discrete event decision process (PODEDP) which captures the dynamics more succinctly and precisely than a POMDP. For optimization, a PODEDP is difficult to solve directly due to the formidable large searching spaces. We can neither solve a PODEDP through a DEDP because the value function of a DEDP is an upper bound of a PODEDP. Our method is to develop a specialy formed DEDP (SDEDP), which is more simple to solve than a PODEDP, to establish a connection between a PODEDP and a SDEDP, and to solve a PODEDP through a SDEDP policy optimization (Algorithm 1) and a PODEDP belief state estimation (Algorithm 2).

3.1 Partially Observable Discrete Event Decision Process

To capture non-linear and high-dimensional state transition dynamics as well as the partial observability of the system states, we formulate the real-world complex social system decision-making process as a PODEDP. Formally, a PODEDP (as shown in the left panel of Figure 1) is defined as a tuple PODEDP$\langle S, A, \Omega, \mathcal{V}, C, P, O, R, \gamma\rangle$, where $S, A, \mathcal{V}, C, P, R, \gamma$ has the same meaning as in a DEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$. The only additions to a DEDP are the observation space Ω and the associated observation function O.Ω is the observation space and $o_t \in \Omega$ the observation received at time t. O is the observation probability such as $P(o_t|s_t)$. We further define a belief state $b_t = p(s_t \mid o_{0:t})$ as the hidden state distribution conditioned on observation history, and a policy π parameterized by θ as a distribution of actions conditioned on belief state $\pi = p(a_t|b_t;\theta)$. The policy can be parameterized in any form, such as a neural network with the weights being θ. Solving a PODEDP involves optimizing the policy π through maximizing the expected future reward $\mathbb{E}_{\xi}(\sum_t \gamma^t R_t)$. In the above, $R_t = \sum_{s_t} b_t(s_t)R(s_t)$ is the reward associated with a belief state, and $\xi_T = (s_{0:T}, a_{0:T}, v_{0:T}, b_{0:T}, o_{0:T})$ is a length-T trajectory of a PODEDP with probabilistic measure $p(\xi_T)=p(s_0,o_0)\prod_t\left(\delta_{b_t=p(s_t|o_{0:t})}p(a_t|b_t)p(v_t|s_t,a_t)\delta_{s_{t+1}=s_t+\Delta_{v_t}}p(o_{t+1}|s_{t+1})\right)$, where δ is an indicator function.

A PODEDP makes a tractable expression of a complex social system through representing the transition kernel using microscopic events which could represent the macroscopic state changes when aggregated together, and the number of which grows linearly with the number of events. As a comparison, a POMDP represents the kernel with $p(s_{t+1} \mid s_t, a_t)$, which grows exponentially with the number of state-action variables, and which becomes intractable in complex social systems. Through introducing events, a PODEDP describes the dynamics of a complex social system more succinctly and accurately. In the experiments, we demonstrate this improvement through a comparison with a POMDP solver.

Comparing to a DEDP, a PODEDP does not assume fully observability of the system. Instead, it introduces observation variables and belief states to capture the partial observations and to infer the possible latent system states. As a result, a PODEDP is a more realistic description of a real world system and, meanwhile, introduces more complexity in solving it. For example, the Bellman Equation of a PODEDP is:

$$V^\star(b) \quad = \max_a \left[R(a,b) + \gamma \sum_{b'} p(b'(b,a,o)|b,a)V^\star(b'(b,a,o))\right] \ ,$$

where $p(b'(b,a,o)|b,a)$ is the transition of belief state:

$$p(b'(b,a,o)|b,a) = \frac{p(o|s')\sum_{s,v}p(v|s,a)\delta_{s'=s+\Delta_v}b(s)}{\sum_{s''}\sum_{s,v'}p(o|s'')p(v'|s,a)\delta_{s''=s+\Delta_{v'}}b(s)}.$$

The Bellman Equation of a DEDP is:

$$V^*(s) = \max_a \left[\sum_s R(a,s) + \gamma \sum_{v,s'} p(s',v|s,a)V(s')\right].$$

For a PODEDP$\langle S, A, \Omega, \mathcal{V}, C, P, O, R, \gamma\rangle$, we refer to a DEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$ as the corresponding DEDP of this PODEDP if these two decision processes share the same $S, A, \mathcal{V}, C, P, R, \gamma$ and start from the same initial state distribution $p(s_0)$. Note that in a PODEDP the policy takes only the stochastic form $\pi = p(a_t|b_t;\theta)$, but in a DEDP the policy could be stochastic $\pi = p(a_t|s_t;\theta)$ or deterministic $a_t = \mu(s_t;\theta)$. In the following sections, to simplify the discussion, we constrain our discussions on DEDP to a deterministic policy $a_t = \mu(s_t;\theta)$.

3.2 Optimal Control of a PODEDP

To overcome the burden of dimensionality and to search in high-dimensional belief state spaces in complex systems, we consider a PODEDP with policy parameterized by $p(a_t \mid b_t) = \sum_{s_t} b_t(s_t)\delta_{a_t=\mu(s_t)}$, where $\mu(s_t)$ is the deterministic policy for a DEDP. We solve a PODEDP by establishing the connections between the optimal value function of a PODEDP and that in a specially formed corresponding DEDP (SDEDP), and reducing the optimal control of a PODEDP to a SDEDP policy optimization and a belief state estimation, assuming a known initial belief state. All derivations and proofs are given in the Appendix.

***Theorem* 2.** *The optimal value function $V^\star(b)$ of a PODEDP$\langle S, A, \Omega, \mathcal{V}, C, P, O, R, \gamma\rangle$ is upper bounded by the expectation of the optimal value function $\tilde{V}^\star(s)$ of the corresponding DEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$ over the belief state $b(s)$: $V^\star(b) \le \sum_s b(s)\tilde{V}^\star(s)$*

Theorem 2 indicates that the optimum value function of a DEDP is an upper bound of that of a corresponding PODEDP. However, we can not solve a PODEDP through optimizing the value function

of a DEDP, since maximizing an upper bound of does not provide any performance guarantees to the original target function. To establish a bridge between a PODEDP and a DEDP, we introduce a specially formed DEDP.

A specially formed DEDP is defined as a tuple $SDEDP\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$, where $S, A, \mathcal{V}, C, P, R, \gamma$ has the same meaning as in a DEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$. The only difference is how the policy is defined. Let the policy in a DEDP be $a_t = \mu(s_t;\theta)$, the policy in a SDEDP is defined as the mean effect of actions over the distribution of states $\pi(a_t;\theta) = \sum_{s_t} p(s_t)\delta_{a_t=\mu(s_t;\theta)}$. For a PODEDP$\langle S, A, \Omega, \mathcal{V}, C, P, O, R, \gamma\rangle$, we refer to a SDEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$ as the corresponding SDEDP of this PODEDP if these two decision processes share the same $S, A, \mathcal{V}, C, P, R, \gamma$ and start from the same initial state distribution $p(s_0)$.

Comparing to a DEDP, the action of a SDEDP is a mean effect of actions over the distribution of states, which does not depend on specific individual states. Comparing to a PODEDP, a SDEDP has the same format of policy as in a PODEDP, but it directly observes the full system states instead of inferring it from partial observations. As a result, a SDEDP is a simplified version of a PODEDP which marginalized out the observations. The following theorem established the connection between a PODEDP and a SDEDP:

***Theorem* 3**. *Let $p(a_t \mid b_t)$ and $V(b)$ be the policy and value function of a PODEDP$\langle S, A, \Omega, \mathcal{V}, C, P, O, R, \gamma\rangle$ with policy parameterized by $p(a_t \mid b_t) = \sum_{s_t} b_t(s_t)\delta_{a_t=\mu(s_t)}$, where $\mu(s_t)$ be the deterministic policy in the corresponding DEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$. Let $\hat{\pi}(a_t;\theta) = \sum_{s_t} p(s_t)\delta_{a_t=\mu(s_t)}$ and $\hat{V}(s)$ and be the policy and value function of a corresponding SDEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$. Then $V(b)$ is equivalent to the expected value function $\sum_s b(s)\hat{V}(s)$.*

Theorem 3 shows that under a certain formulation of a PODEDP policy $p(a_t \mid b_t) = \sum_{s_t} b_t(s_t)\delta_{a_t=\mu(s_t)}$ the value function of a PODEDP is equivalent to that of a corresponding SDEDP. It provides the foundation of our PODEDP solver — we solve a PODEDP by training a SDEDP policy, estimating the belief state of the system from historical observations, and constructing a belief state policy satisfying the requirements of Theorem 2. The following corollary provides justifications for our method:

***Corollary* 1**. *In a PODEDP$\langle S, A, \Omega, \mathcal{V}, C, P, O, R, \gamma\rangle$ with policy parameterized by $\pi(a_t;\theta) = \sum_{s_t} p(s_t)\delta_{a_t=\mu(s_t;\theta)}$, let π^* be the optimal policy of this PODEDP and let $\hat{\pi}^*$ be the optimal policy of the corresponding SDEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$ where $\pi^*(a_t;\theta) = \sum_{s_t} p(s_t)\delta_{a_t=\mu^*(s_t;\theta)}$, then the policy $\pi^* = \hat{\pi}^*$ is an optimal policy of the PODEDP.*

With Corollary 1, we reduce solving a PODEDP to a SDEDP policy optimization and a belief state estimation, which is equivalent to optimizing the policy parameter θ under the mean effect of all possible observations in training stage (SDEDP policy optimization), and using specific observations $o_{0:t}$ to estimate the current belief state b_t and optimal policy π at testing stage (belief state estimation). Comparing to directly solving a PODEDP, our method split the process of optimizing the policy parameters and gathering the information from partial observations. Since we only use the observations at the testing stage, our method greatly reduces the variance introduced by partial observations during the training stage and eases the computation.

3.3 SDEDP Policy Optimization

In this section, we derive the algorithm for policy optimization in a SDEDP where the policy takes the form $p(a_t;\theta) = \sum_{s_t} p(s_t)\delta_{a_t=\mu(s_t;\theta)}$, and the state s_t and action a_t are continuous variables. To maintain a tractable solution, we apply a Taylor approximation to simplify the trajectory distribution, using a duality theorem to recasts optimal control as parameter learning, and applying variational inference and Bethe entropy approximation to reduce the exploding searching space.

***Theorem* 4**. *In a SDEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$ with policy $p(a_t;\theta) = \sum_{s_t} p(s_t)\delta_{a_t=\mu(s_t;\theta)}$ where the state s_t and action a_t are continuously variables and $\bar{s_t} = \mathbb{E}_{s_t}[s_t]$ is the expected state evaluated at time t, the first-order Taylor approximation of the trajectory distribution takes the form $\sum_{a_{0:T}} p(s_{0:T}, a_{0:T}, v_{0:T}) \approx p(s_0)\prod_{t=0}^{T-1} p(v_t \mid s_t, \mu(\bar{s_t};\theta))\delta_{s_{t+1}=s_t+\Delta_{v_t}}$.*

Theorem 4 simplifies the trajectory distribution a SDEDP. With Theorem 3, we denote a length-T trajectory ξ_T as

$$\begin{aligned} p(\xi_T) &= p(s_0)\prod_{t=0}^{T-1}\left(\delta_{a_t=\mu(s_t;\theta)}p(v_t \mid s_t, \mu(\bar{s_t};\theta))\delta_{s_{t+1}=s_t+\Delta_{v_t}}\right) \\ &= p(s_0)\prod_{t=0}^{T-1}\left(p(s_{t+1}, v_t \mid s_t, \mu(\bar{s_t};\theta))\right) \end{aligned}$$

In the following, we derive a variational inference algorithm to identify the optimal policy parameter θ of a DEDP with policy $a_t = \mu(\bar{s_t};\theta)$, based on the convex conjugate duality between the log expected future reward function and the entropy function of a distribution over finite-length DEDP trajectories. Applying Theorem 1, we have $\log V^{\pi}(r) = \sup_q \left(\sum_{T,m,\xi_T} q(T, m, \xi_T)\log r(T, m, \xi_T;\pi) + \mathrm{H}(q)\right)$, where $p(\xi_T) = p(s_0)\prod_{t=0}^{T-1}\left(p(s_{t+1}, v_t \mid s_t, \mu(\bar{s_t};\theta))\right)$. We will solve it using a EM paradigm: in policy evaluation we find a q that best approximate r; in policy improvement we find a parameter θ that maximize the log value function $\log V^{\pi}(r)$.

In policy evaluation, we apply variational inference and Bethe entropy approximation to relax the intractable searching with a exploding state space to a tractable one with mean field approximation of the state $q(s_t|T, m) = \prod_{\hat{m}=1}^{M} q(s_t^{(\hat{m})}|T, m)$. Applying the approximation, we average the effects of all other components into a projected marginal kernel $p(s_t^{(\hat{m})}, v_t|s_{t-1}^{(\hat{m})}, \mu(\bar{s}_{t-1};\theta))$. We further introduce a forward message $\alpha_t^{(\hat{m})}(s_t^{(\hat{m})})$ and a backward message $\beta_t^{(\hat{m})}(s_t^{(\hat{m})})$, and get a forward-backward algorithm.

$$\alpha_t^{(\hat{m})}(s_t^{(\hat{m})}) \propto \sum_{s_{t-1}^{(\hat{m})}, v_{t-1}} \alpha_{t-1}^{(\hat{m})}(s_{t-1}^{(\hat{m})}) \cdot p(s_t^{(\hat{m})}, v_{t-1}|s_{t-1}^{(\hat{m})}, \mu(\bar{s}_{t-1};\theta)) \quad (1)$$

$$\begin{aligned} \beta_t^{(\hat{m})}(s_t^{(\hat{m})}) &= \sum_m q(t,m)\beta_{t|t,m}^{(\hat{m})}(s_t^{(\hat{m})}) \\ &\quad + \sum_{s_{t+1}^{(\hat{m})}, v_t} p(s_{t+1}^{(\hat{m})}, v_t|s_t^{(\hat{m})}, \mu(\bar{s}_{t-1};\theta))\beta_{t+1}^{(\hat{m})}(s_{t+1}^{(\hat{m})}) \end{aligned} \quad (2)$$

In policy improvement, we maximize the log value function $\log V^{\pi}(r)$ with respect to parameter θ, using the forward backward messages inferred from the policy evaluation.

$$\begin{aligned} \frac{\partial \log V^{\pi}(r)}{\partial\theta} &= \sum_{t,s_t} \frac{\prod_{\hat{m}} \alpha_t^{(\hat{m})}(s_t^{(\hat{m})}, v_t=v)\beta_t^{(\hat{m})}(s_t^{(\hat{m})}, v_t=v)}{c_v} \frac{\partial c_v}{\partial\theta} \\ &- \sum_{t,s_t} \frac{\prod_{\hat{m}} \alpha_t^{(\hat{m})}(s_t^{(\hat{m})}, v_t=\emptyset)\beta_t^{(\hat{m})}(s_t^{(\hat{m})}, v_t=\emptyset)\cdot\prod_m g_v^m(s_t^{(m)})}{1-\sum_{v=1}^{V} c_v\cdot\prod_m g_v^m(s_t^{(m)})} \frac{\partial c_v}{\partial\theta} \end{aligned} \quad (3)$$

To summarize, the SDEDP policy optimization is as in Algorithm 1.

Algorithm 1 SDEDP Policy Optimization

Input: SDEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$, initial policy parameter θ
Output: Optimal policy parameter θ for $\mu(\tilde{s_t}; \theta)$
Procedure:

```
for i = 0, 1, 2, ... do
  for u = 0, 1, 2, ... do
    update the forward messages with Eq. (1).
    update backward messages with Eq. (2).
  end for
  Update parameter θ with the gradient computed in Eq. (3).
end for
```

3.4 Belief State Estimation

To estimate the belief state $b_t(s_t) = p(s_t \mid o_{0:t})$, we essentially infer the state distribution in a PODEDP from past observations up to now $o_{0:t}$. Exact inference in a complex social system is intractable due to the formidable state space. As such, we estimate the posterior distribution $p(s_{0:T}, a_{0:T}, v_{0:T} \mid o_{0:T})$ using an approximate distribution $q(s_{0:T}, a_{0:T}, v_{0:T})$, and apply the Bethe entropy approximation $q(s_t) = \prod_{\hat{m}=1}^{M} q(s_t^{(\hat{m})})$, where $q(s_t^{(\hat{m})})$ is the one-slice marginal involving only component $\hat{m}$. The variational lower bound admits:

$$
\begin{aligned}
&\text{ELBO} \\
&= \mathbb{E}_{q(s_{0:T}, a_{0:T}, v_{0:T})} \log\left(p(s_{0:T}, a_{0:T}, v_{0:T}, o_{0:T}) - q(s_{0:T}, a_{0:T}, v_{0:T})\right) \\
&= \sum_{s_{0:T}, a_{0:T}, v_{0:T}} \log\left(\frac{\prod_{t=1}^{T-1} p(s_t, a_{t-1}, v_{t-1}, o_t \mid s_{t-1}; \theta)}{\prod_{t=1}^{T} q(s_{t-1,t}, a_{t-1}, v_{t-1})}\right) \\
&\quad + \sum_{s_{0:T}, v_{0:T}} \log\left(\prod_{t=1}^{T-1} \prod_{\hat{m}} q(s_t^{(\hat{m})})\right) \\
&= \sum_{t=1}^{T-1} \sum_{\hat{m}} \sum_{s_t^{(\hat{m})}} q(s_t^{(\hat{m})}) \log q(s_t^{(\hat{m})}) \\
&\quad - \sum_{t=1}^{T-1} \sum_{s_{t-1,t}, a_{t-1}, v_{t-1}} q(s_{t-1,t}, a_{t-1}, v_{t-1}) \log\left(\frac{q(s_{t-1,t}, a_{t-1}, v_{t-1})}{p(s_t, a_{t-1}, v_{t-1}, o_t \mid s_{t-1}; \theta)}\right)
\end{aligned}
$$

Incorporating the consistency constraint of the approximate distribution q, the problem becomes

$$
\begin{aligned}
&\min \text{ over } q(s_t^{(\hat{m})}), q(s_{t-1,t}, a_{t-1}, v_{t-1}) \ \forall t \le T, m \\
&\sum_{t=1}^{T-1} \sum_{\hat{m}} \sum_{s_t^{(\hat{m})}} q(s_t^{(\hat{m})}) \log q(s_t^{(\hat{m})}) \\
&- \sum_{t=1}^{T-1} \sum_{s_{t-1,t}, a_{t-1}, v_{t-1}} q(s_{t-1,t}, a_{t-1}, v_{t-1}) \log\left(\frac{q(s_{t-1,t}, a_{t-1}, v_{t-1})}{p(s_t, a_{t-1}, v_{t-1}, o_t \mid s_{t-1}; \theta)}\right) \\
&\text{subject to:} \\
&\sum_{s_{t-1,t}, a_{t-1}, v_{t-1} \setminus s_t^{(\hat{m})}} q(s_{t-1,t}, a_{t-1}, v_{t-1}) \quad = q(s_t^{(\hat{m})})
\end{aligned}
$$

The inference algorithm finds the proposal distribution q that maximizes the variational lower bound. Applying Bethe entropy approximation and solving an optimization problem with the method of Lagrange multipliers, the belief state can be estimated as follows:

$$
b_t(s_t) = \prod_{\hat{m}=1}^{M} \alpha_t^{(\hat{m})}(s_t^{(\hat{m})}), \tag{4}
$$

where $\alpha_t^{(\hat{m})}(s_t^{(\hat{m})}) \propto \sum_{s_{t-1}^{(\hat{m})}, a_{t-1}, v_{t-1}} \alpha_{t-1}^{(\hat{m})}(s_{t-1}^{(\hat{m})}) p(s_t^{(\hat{m})}, o_t^{(\hat{m})} a_{t-1}, v_{t-1} \mid s_{t-1}^{(\hat{m})}; \theta)$ is the forward message.

Algorithm 2 Optimal Control of a PODEDP

Input: PODEDP$\langle S, A, \Omega, \mathcal{V}, C, P, O, R, \gamma\rangle$
Output: Optimal policy $\pi^*(a_t \mid b_t)$
Procedure:

- Optimize SDEDP policy $\mu^*(s_t)$ according to Algorithm 1.
- Estimate the belief state $b_t(s_t)$ according to Eq. (4).
- Construct the PODEDP optimal policy $\pi^*(a_t|b_t) = \sum_{s_t} b_t(s_t) \delta_{a_t = \mu^*(s_t)}$

To summarize, we solve a PODEDP by optimizing the corresponding SDEDP policies $\pi(a_t; \theta) = \sum_{s_t} p(s_t) \delta_{a_t = \mu(s_t; \theta)}$ using Algorithm 1, estimating the belief state $b_t(s_t)$ using Eq. (4), and constructing the belief state policy $\pi(a_t \mid b_t)$ according to Theorem 3. We give the optimal control of a PODEDP as Algorithm 2.

One of the limitations of our approach is that it requires knowing the events that are used to specify the system dynamics. As future work, we plan to investigate how to learn the events from scratch.

4 EXPERIMENTS

We benchmark our algorithm against other state-of-the-art algorithms in complex transportation system [32] control problems, which contains a SynthTown (inset figure in Fig. 2) and a Berlin scenario. The SynthTown consists of a synthesized network from MATSIM [10] with one home facility, one work facility, 23 road links, and 50 individuals. The Berlin contains a real-world network with 1530 locations and 9178 individuals. The goal is to find an optimal control of the movement of vehicles such that all vehicles spend less time on roads, going to facilities on time, and staying in facilities for enough amount of time.

Experimental Setup. The transportation system as a whole is modeled as a PODEDP, where the different locations are the components of the system, and the number of vehicles of each location are the states. To be specific, in a transportation system with M locations, the optimal control problem is modeled as a PODEDP$\langle S, A, \Omega, \mathcal{V}, C, P, O, R, \gamma\rangle$. The states $s_t = (s_t^{(1)}, \cdots, s_t^{(M)}, t)$ are the number of vehicles at M locations plus the current time t. We randomly select 10% vehicles as probe vehicles (in our experiment, different percentage of partial observations does not have an obvious influence on the obtained results). The observation variables $o_t = (o_t^{(1)}, \cdots, o_t^{(M)}, t)$ are the populations of these probe vehicles at the M locations and the current time t. The movement of vehicles from one location to the other is represented as event $p \cdot m_1 \overset{c_{m_1 m_2}}{\rightarrow} p \cdot m_2$, where m_1 is the current location, and m_2 is the next location, and $c_{m_1 m_2}$ is the event rate coefficient—the probability of one vehicle making the movement. The action variables a_t are the event rate coefficients of choosing the downstream locations at a crossing center, and those of leaving or entering facilities. To simulate different traffic conditions such as traffic congestion, we implement the state transition $p(s_{t+1}, v_t \mid s_t, a_t)$ following the traffic flow diagram in MATSIM. The reward function $R(s_t)$ is implemented following the Charypa-Nagel scoring function [10]. For our algorithm (PODEDP), we implement a policy with the format $p(a_t \mid b_t; \theta) = \sum_{s_t} b_t(s_t) \delta_{a_t = \mu(s_t; \theta)}$, where $\mu(s_t; \theta)$ is implemented as a neural network with weight θ.

Dataset	SynthTown					Berlin				
Metrics	**TRPE**	**EC**	**TOR**	**VOR**	**VAW**	**TRPE**	**EC**	**TOR**	**VOR**	**VAW**
PODEDP	**23.42**	**75**	**25.2**	**1.91**	**15.77**	**9.73**	**300**	**80.09**	**556.49**	**927.20**
AC	15.19	200	35.4	2.69	13.22	-14.31	-	679.21	4288.05	203.34
PG	10.94	200	55.8	4.24	12.39	-18.29	-	792.96	4907.31	178.65
GPS	14.11	150	27	1.98	11.18	-9.92	-	497.72	3317.92	122.80
DEDP	24.05	75	23.5	1.79	15.83	10.94	200	81.12	588.26	1185.94

Table 1: Performance comparison of algorithms in the following five metrics: total reward per episode (TRPE), epochs to converge (EC), average time (minutes) on road per vehicle (TOR), average number of vehicles on road per unit time (minutes) (VOR), and average number of vehicles at work during work hours (VAW).

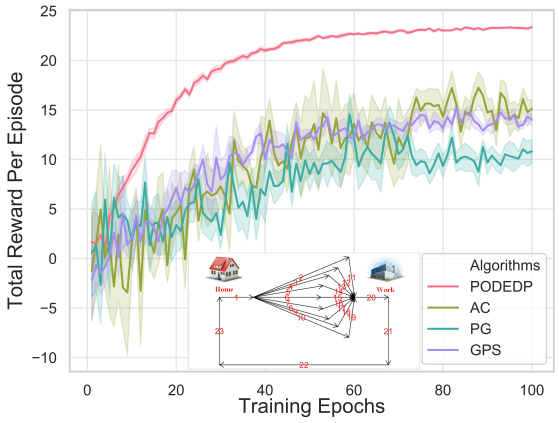


Figure 2: Training process on SynthTown

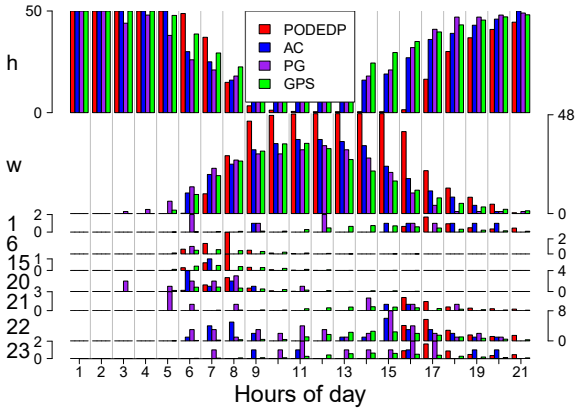


Figure 3: Average number of vehicles with trained policies

We benchmark our algorithm against other analytical and simulation methods. For the simulation method, we benchmark against a policy gradient (PG) [23] and an actor-critic (AC) [13] that sampling the actions and next states to reproduce the population flow. For the analytical method, we benchmark against a guided policy search (GPS) [17] that models the dynamics as a POMDP, approximating the transition dynamics with differential equations and solving the local policies analytically. All these algorithms implement the policy as a neural network with the historical inputs as observations. We further benchmark against an optimal control algorithm with a fully observed discrete event decision process (DEDP) [32].

Experimental Result. The comparison results in Table 1 indicates that our algorithm (PODEDP) performs best among all partial observation algorithms in almost all evaluation metrics for both SynthTown and Berlin scenario, such as the least average time on road, the largest number of vehicles at work during work hours, and the fewest training epochs to converge to the highest total rewards per episode (TRPE). For the Berlin scenario, the other algorithms did not converge in a reasonable number of epochs.

The optimal control under full observations (DEDP) obtains slightly higher TRPE than our algorithm (PODEDP). The DEDP assumes perfect information which does not perform information-gathering actions. On the other hand, the PODEDP assumes limited observations and solve a PODEDP through solving a SDEDP and tracking the belief state. The DEDP achieves slightly better TRPE because as indicated by Theorem 1, the optimal value function of a PODEDP is upper bounded by the expected optimal value function of the corresponding DEDP. Moreover, the TRPE of PODEDP is close to that of the DEDP, indicating our method learned a policy with a value function that is close to its upper bound.

To better compare the empirical performance and convergence ability of these algorithms, we look more deeply into the SynthTown scenario. As indicated in Figure 2, the TRPE-epoch curve of PODEDP is higher with less variance than that of other algorithms. Figure 3 presented the average number of vehicles of ten runs at each locations for each algorithms using the learned policy, which indicate how well each learned policies performed. As shown in this Figure, the PODEDP leads to the smallest amount of vehicles on roads, largest amount of vehicles at work during work hours (9 am - 5 pm), and largest amount of vehicles at home during rest hours (other hours), which indicates that the learned policy of our algorithm best satisfied the needs. By combining the accurate modeling of PODEDP with a tractable solution using variational inference, our method achieve the best performance. For other analytical method, GPS introduces modeling error when approximate the state transitions with differential equations. For other simulation methods, PG and AC introduce high variance in sampling.

In summary, for the complex traffic control problem, our algorithm outperformed other partially observable decision-making algorithms (GPS, PG, and AC), and achieved comparable performance against fully observed optimal control algorithm (DEDP), in both SynthTown and Berlin scenarios.

5 RELATED WORKS

A PODEDP is a special POMDP describing the dynamics of a complex network using microscopic events. POMDP solvers generally go into the following four categories: 1) iteratively applying Bellman optimality equation to get convex piece-wise linear value functions of the belief state, 2) identifying a deterministic/stochastic finite-state controller (FSC) through policy iteration, 3) invoking an online algorithm based on Monte Carlo tree search to reduce the exponential search space, or 4) designing a reinforcement learning algorithm with a memory of the latent state.

In category (1), exact algorithms can only be applied to small scenarios due to the exponential number of elements in the value function [5]. Approximate algorithms that approximate the value function for a subset of points in the belief state space suffer from computation complexity issues [24]. Heuristic algorithms solving a POMDP based on the corresponding MDP value functions do not perform information gatherings [8].

In category (2), different FSC algorithms have been proposed to solve a POMDP, including algorithms with no memory [11], algorithms based on deterministic [9], and stochastic FSC [26]. However, in a complex social system, the exploding number of the system states makes it infeasible for FSC algorithms to enumerate all combinations of the components and prevents these algorithms from performing fine-grained control.

in category (3), an online algorithm generally cooperates with an offline algorithm to guide the search in the most promising directions and to cut off after finite look-ahead steps [22]. A good offline algorithm such as ours could potentially lessen the number of look-ahead steps and better guide the search of an online algorithm.

In category (4), value-based and policy-based reinforcement learning algorithms have been proposed to solve a POMDP [2, 15, 18]. However, these algorithms suffer from sample efficiency issues [21].

In this paper, we develop a PODEDP solver through establishing the connections between the optimal value function of a PODEDP and a fully observable SDEDP, and reducing the optimal control of a PODEDP to SDEDP policy optimization and belief state estimation. To our best knowledge, this approach is novel.

6 CONCLUSIONS

In this paper, we developed a new framework to achieve optimal control in complex social systems. To capture the dynamics precisely and succinctly, we developed PODEDP for modeling social system decision-making processes. To optimize the problem with amenable searching space, we established the connection between the PODEDP value function and a SDEDP value function and reduced the optimal control of a PODEDP to SDEDP policy optimization and belief state estimation. We tested our framework in complex transportation scenarios against other state-of-the-art methods and demonstrated that our algorithm outperformed its competitors in multiple aspects.

7 APPENDIX

7.1 Proof of Theorems and Corollaries

Proof of ***Theorem*** **2.** . We use Bellman optimality equation.

$$
\begin{aligned}
V^\star(b) \quad &= \max_a \left[R(a,b) + \gamma \textstyle\sum_{b'} p(b'(b,a,o)|b,a) V^\star(b'(b,a,o)) \right] \\
&= \max_a \left[\textstyle\sum_s R(a,s) b(s) + \gamma \sum_o p(o|b,a) V^\star(b'(b,a,o)) \right] \\
&\le \max_a \left[\textstyle\sum_s R(a,s) b(s) + \gamma \sum_o p(o|b,a) \sum_{s'} (b'(b,a,o))(s') V(s') \right] \\
&= \max_a \left[\textstyle\sum_s R(a,s) b(s) + \gamma \sum_o p(o|b,a) \sum_{s'} \frac{p(s',o|b,a)}{p(o|b,a)} V(s') \right] \\
&= \max_a \left[\textstyle\sum_s R(a,s) b(s) + \gamma \sum_{s'} p(s'|b,a) V(s') \right] \\
&= \max_a \left[\textstyle\sum_s R(a,s) b(s) + \gamma \sum_{s',s} p(s'|s,a) b(s) V(s') \right] \\
&\le \textstyle\sum_s b(s) \max_a \left[\sum_s R(a,s) + \gamma \sum_{s'} p(s'|s,a) V(s') \right] \\
&\le \textstyle\sum_s b(s) \tilde{V}^\star(s).
\end{aligned}
$$

The third step in the previous proof is due to that the value function of a belief state PODEDP is a convex function. The 8th step is due to the definition of $V^\star$ for DEDP. In the second and fourth steps, b' is a deterministic function of b, a and y, and the probability of b' is the probability of observing y.

***Lemma* 1.** *Let $\pi(a_t \mid b_t)$ be the policy of a PODEDP$\langle S, A, \Omega, \mathcal{V}, C, P, O, R, \gamma \rangle$, and $\hat{\pi}$ be the policy of the corresponding SDEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma \rangle$ where $\hat{\pi}(a_t) = \pi(a_t \mid b_t)$. We further define $p(s_t \mid \pi)$ as the state distribution at time t following the policy π in a PODEDP, and $p(s_t \mid \hat{\pi})$ as the state distribution at time t following the policy $\hat{\pi}$ in the SDEDP. If $\pi(a_t \mid b_t) = \hat{\pi}(a_t)$, then the state distribution following the PODEDP policy is equivalent to the state distribution following the SDEDP policy $p(s_t \mid \pi) = p(s_t \mid \hat{\pi})$.*

Proof of ***Lemma*** **1.**. We prove this lemma using mathematical induction.

Since the DEDP is the corresponding DEDP of the PODEDP, these two decision processes start from the same initial distribution: $p(s_0 \mid \pi) = p(s_0 \mid \hat{\pi})$

Suppose $p(s_t \mid \pi) = p(s_t \mid \hat{\pi})$, then:

$$
\begin{aligned}
& p(s_{t+1}|\pi) \\
=& \textstyle\sum_{b_t,a_t,v_t,s_t} p(s_t|\pi)\pi(a_t|b_t)p(v_t|s_t,a_t)\delta_{s_{t+1}=s_t+\Delta v_t}\delta_{b_t=p(s_t|o_{0:t})} \\
=& \textstyle\sum_{b_t,a_t,v_t,s_t} p(s_t|\pi)\pi(a_t|b_t)p(v_t|s_t,a_t)\delta_{s_{t+1}=s_t+\Delta v_t}\delta_{b_{t-1}=p(s_{t-1}|\pi)} \\
=& \textstyle\sum_{b_t,a_t,v_t,s_t} p(s_t|\pi)\pi(a_t|b_t)p(s_{t+1},v_t|s_t,a_t)\delta_{b_{t-1}=p(s_{t-1}|\pi)} \\
=& \textstyle\sum_{b_t,a_t,v_t,s_t} p(s_t|\hat{\pi})\hat{\pi}(a_t)p(s_{t+1},v_t|s_t,a_t)\delta_{b_{t-1}=p(s_{t-1}|\pi)} \\
=& \textstyle\sum_{a_t,v_t,s_t} p(s_t|\hat{\pi})\hat{\pi}(a_t)p(s_{t+1},v_t|s_t,a_t) \\
=& p(s_{t+1}|\hat{\pi}).
\end{aligned}
$$

As such, this Lemma is proved.

Proof of ***Theorem*** **3.**. Using Lemma 1, we have:

$$
\begin{aligned}
V(b) \quad &= \textstyle\sum_t \gamma^t \mathbb{E}_{b_t,o_{0:t}} R(b_t) \\
&= \textstyle\sum_t \gamma^t \sum_{b_t,o_{0:t}} p(o_{0:t}|\pi)\delta_{b_t=p(s_t|o_{0:t})} R(b_t) \\
&= \textstyle\sum_t \gamma^t \sum_{b_t,o_{0:t}} p(o_{0:t}|\pi)\delta_{b_t=p(s_t|o_{0:t})} \sum_{s_t} R(s_t) b(s_t) \\
&= \textstyle\sum_t \gamma^t \sum_{o_{0:t}} p(o_{0:t}|\pi) \sum_{s_t} p(s_t|o_{0:t}) R(s_t) \\
&= \textstyle\sum_t \gamma^t \sum_{s_t} p(s_t|\pi) R(s_t) \\
&= \textstyle\sum_t \gamma^t \sum_{s_t} p(s_t|\hat{\pi}) R(s_t) \\
&= \mathbb{E}_{s\sim p(s|\pi)} \hat{V}(s) \\
&= \textstyle\sum_s b(s) \hat{V}(s),
\end{aligned}
$$

where $p(o_{0:t} \mid \pi)$ is the observation history distribution starting from initial until time t following policy π in a PODEDP, $p(s_t \mid \pi)$ is the state distribution at time t following policy π in the PODEDP, and $p(s_t \mid \hat{\pi})$ is the state distribution at time t following policy $\hat{\pi}$ in the corresponding DEDP.

Proof of ***Corollary*** **1.**. Let $V^*(b)$ be the value function for $\pi^*(a_t \mid b_t)$ in a PODEDP$\langle S, A, \Omega, \mathcal{V}, C, P, O, R, \gamma \rangle$, $V_{opt}(b)$ be the optimal

PODEDP value function, $\hat{V}^*(s)$ be the optimal value function induced by $\hat{\pi}^*(a_t \mid s_t)$ in the corresponding DEDP$\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$, where $\pi^*(a_t \mid b_t) = \sum_{s_t} \hat{\pi}^*(a_t \mid s_t) b_t(s_t)$. In the following, we will prove $V^*(b) = V_{opt}(b)$.

Using Theorem 2 and 3, we have $V_{opt}(b) \le \sum_s b(s)\hat{V}^\star(s) = V^*(b)$. On the other hand, since $V_{opt}(b)$ is the optimal value function in the PODEDP, we have $V_{opt}(b) \ge V^*(b)$. As such, $V^*(b) = V_{opt}(b)$.

*Proof of **Theorem** 4.*. The probability of a trajectory in a specially formed $DEDP\langle S, A, \mathcal{V}, C, P, R, \gamma\rangle$ with policy $p(a_t;\theta) = \sum_{s_t} p(s_t)\, \delta_{a_t=\mu(s_t;\theta)}$ can be written as:

$$\begin{aligned} p(\xi_T) &= p(s_0)\prod_{t=0}^{T-1}\Big(p(a_t;\theta)p(v_t|s_t,a_t)\delta_{s_{t+1}=s_t+\Delta_{v_t}}\Big) \\ &= p(s_0)\prod_{t=0}^{T-1}\Big(\textstyle\sum_{s'_t}\delta_{a_t=\mu(s'_t;\theta)}p(s'_t)p(v_t|s_t,a_t)\delta_{s_{t+1}=s_t+\Delta_{v_t}}\Big). \end{aligned}$$

Using the energy form

$$p(v_t|s_t,a_t;\theta) = \frac{\exp(w(v_t,s_t,a_t;\theta))}{\sum_{v'_t}\exp\big(w(v'_t,s_t,a_t;\theta)\big)} = f(v_t,s_t,a_t;\theta).$$

Let $\bar{a}_t = \sum_{a_t} p(a_t)a_t$, and use first-order Taylor expansion for f at point $(v_t, s_t, \bar{a}_t)$ then

$$\begin{aligned} & \mathbb{E}_{a_t} p(v_t|s_t,a_t) \\ = \;& \textstyle\sum_{a_t} p(a_t) f(v_t,s_t,a_t;\theta) \\ \approx \;& \textstyle\sum_{a_t} p(a_t) f(v_t,s_t,\bar{a}_t;\theta) + \sum_{a_t} p(a_t)\frac{\partial}{\partial v_t} f(v_t,s_t,\bar{a}_t;\theta)(v_t - v_t) \\ & + \textstyle\sum_{a_t} p(a_t)\frac{\partial}{\partial s_t} f(v_t,s_t,\bar{a}_t;\theta)(s_t - s_t) + \sum_{a_t} p(a_t)\frac{\partial}{\partial a_t} f(v_t,s_t,\bar{a}_t;\theta)(a_t - \bar{a}_t) \\ = \;& \textstyle\sum_{a_t} p(a_t) f(v_t,s_t,\bar{a}_t;\theta) + \frac{\partial}{\partial a_t} f(v_t,s_t,\bar{a}_t;\theta)\sum_{a_t} p(a_t)(a_t - \bar{a}_t) \\ = \;& f(v_t,s_t,\bar{a}_t;\theta) \\ = \;& p(v_t|s_t,\bar{a}_t). \end{aligned}$$

Let $\bar{s}_t = \sum_{s_t} p(s_t)s_t$, and use first-order Taylor expansion for μ at point $\bar{s}_t$ then

$$\begin{aligned} \bar{a}_t &= \textstyle\sum_{a_t} a_t p(a_t) \\ &= \textstyle\sum_{a_t} a_t \sum_{s_t} p(s_t)\delta(a_t = \mu(s_t)) \\ &= \textstyle\sum_{s_t} p(s_t)\mu(s_t) \\ &\approx \textstyle\sum_{s_t} p(s_t)\Big(\mu(\bar{s}_t) + \frac{\partial}{\partial s_t}\mu(s_t=\bar{s}_t)(s_t - \bar{s}_t)\Big) \\ &= \mu(\bar{s}_t). \end{aligned}$$

As such, $\mathbb{E}_{a_t} p(v_t|s_t,a_t) \approx p(v_t|s_t,\mu(\bar{s}_t))$.

Denoting the original MDP trajectory as $\xi'_T = (s_{0:T}, a_{0:T}, v_{0:T})$, and the trajectory marginalized over $a_{0:T}$ as $\xi_T = (s_{0:T}, v_{0:T})$, the probability of a MDP trajectory marginalized over $a_{0:T}$ becomes

$$\begin{aligned} p(\xi_T) &= \textstyle\sum_{a_{0:T}} p(\xi'_T) \\ &= \textstyle\sum_{a_{0:T}} p(s_0)\prod_{t=0}^{T-1}\Big(p(a_t;\theta)p(v_t|s_t,a_t)\delta_{s_{t+1}=s_t+\Delta_{v_t}}\Big) \\ &= p(s_0)\prod_{t=0}^{T-1}\Big(\textstyle\sum_{a_t} p(a_t;\theta)p(v_t|s_t,a_t)\delta_{s_{t+1}=s_t+\Delta_{v_t}}\Big) \\ &= p(s_0)\prod_{t=0}^{T-1}\Big(p(v_t|s_t,\mu(\bar{s}_t;\theta))\delta_{s_{t+1}=s_t+\Delta_{v_t}}\Big). \end{aligned}$$

7.2 Optimal Control of SDEDP

7.2.1 Derivation of Eq. (1) and Eq. (2). The derivation is similar to the one in [32], but differs in that we use a transition kernel $p(s_{t+1}, v_t \mid s_t, \mu(\bar{s}_t;\theta))$ instead of $p(s_{t+1}, v_t \mid s_t;\theta)$. In the context of a discret event model, this equals to having $c_v = \mu(\bar{s}_t;\theta)$ instead of $c_v = \theta$. Rearranging the terms and applying the approximations described in the main text, the target becomes the following:

$$\begin{aligned} & \textstyle\sum_T\sum_m\sum_{\xi_T} q(T,m,\xi_T)\log\Big(\gamma^T P(\xi_T;\theta)R_T^{(m)}\Big) + \mathrm{H}(q(T,m,\xi_T)) \\ = & \textstyle\sum_T\sum_m (q(T,m)\log\Big(\frac{\gamma^T}{q(T,m)}\Big) - \sum_{t=1}^{T-1}\sum_{s_{t-1,t},v_{t-1}} q(T,m,s_{t-1,t})\log\Big(\frac{q(s_{t-1,t},v_{t-1}|T,m)}{p(s_t,v_{t-1}|s_{t-1},\mu(\bar{s}_{t-1};\theta))}\Big) \\ & - \textstyle\sum_t\sum_m\sum_{s_{t-1,t},v_{t-1}} q(T,m,s_{t-1,t})\log\Big(\frac{q(s_{t-1,t},v_{t-1}|T,m)}{p(s_t,v_{t-1}|s_{t-1},\mu(\bar{s}_{t-1};\theta))R_t^{(m)}}\Big) \\ & + \textstyle\sum_{T,m}\sum_{t=1}^{T-1}\sum_{\hat{m}}\sum_{s_t^{(\hat{m})}} q(T,m,s_t^{(\hat{m})})\log q(s_t^{(\hat{m})}|T,m) \end{aligned}$$

We solve this maximization problem with the method of Lagrange multipliers. Taking derivative with respect to $q(s_{t-1,t}, a_t|T,m)$ and $q(s_t^{(\hat{m})}|T,m)$, and setting it to zero, denoting $\exp\left(\frac{\alpha^{(\hat{m})}_{t,s_t^{(\hat{m})},T,m}}{q(T,m)}\right)$ as $\alpha^{(\hat{m})}_{t|T,m}\big(s_t^{(\hat{m})}\big)$, $\exp\left(\frac{\beta^{(\hat{m})}_{t,s_t^{(\hat{m})},T,m}}{q(T,m)}\right)$ as $\beta^{(\hat{m})}_{t|T,m}\big(s_t^{(\hat{m})}\big)$. We can compute α, β through a forward-backward iterative approach:

$$\begin{aligned} &\text{forward:} && q(s_t^{(\hat{m})}|T,m) = \textstyle\sum_{s_{t-1}^{(\hat{m})},v_{t-1}} q(s_{t-1,t}^{(\hat{m})}, v_{t-1}|T,m) \\ &\Rightarrow && \alpha^{(\hat{m})}_{t|T,m}(s_t^{(\hat{m})}) = \frac{Z_t^{(\hat{m})}}{Z_t}\textstyle\sum_{s_{t-1}^{(\hat{m})},v_{t-1}} \alpha^{(\hat{m})}_{t-1|T,m}(s_{t-1}^{(\hat{m})})\cdot p(s_t^{(\hat{m})}, v_{t-1}|s_{t-1}^{(\hat{m})};\theta) \\ &\text{backward:} && q(s_{t-1}^{(\hat{m})}|T,m) = \textstyle\sum_{s_t^{(\hat{m})},v_{t-1}} q(s_{t-1,t}^{(\hat{m})}, v_{t-1}|T,m) \\ &\Rightarrow && \beta^{(\hat{m})}_{t-1|T,m}(s_{t-1}^{(\hat{m})}) = \frac{Z_t^{(\hat{m})}}{Z_t}\textstyle\sum_{s_t^{(\hat{m})},v_{t-1}} p(s_t^{(\hat{m})}, v_{t-1}|s_{t-1}^{(\hat{m})};\theta)\cdot\beta^{(\hat{m})}_{t|T,m}(s_t^{(\hat{m})}) \\ & && \text{for } t{=}T, \hat{m}{=}m \\ & && \beta^{(\hat{m})}_{t-1|T,m}(s_{t-1}^{(\hat{m})}) = \frac{Z_t^{(\hat{m})}}{Z_t}\textstyle\sum_{s_t^{(\hat{m})},v_{t-1}} p(s_t^{(\hat{m})}, v_{t-1}|s_{t-1}^{(\hat{m})};\theta)\cdot\beta^{(\hat{m})}_{t|T,m}(s_t^{(\hat{m})})R_T^{(m)} \end{aligned}$$

For policy improvement, we take the derivative of $\log V^\pi$ over the policy parameter θ, we get

$$\begin{aligned} \frac{\partial \log V^\pi}{\partial\theta} &= \sum_{T,m,t,v_t,s_t} q(T,m,v_t,s_t)\frac{\partial \log P(v_t|s_t,\mu(\bar{s}_t;\theta))}{\partial c_{v_t}}\frac{\partial c_{v_t}}{\partial\theta} \\ &= \sum_{t,s_t}\frac{\prod_{\hat{m}}\alpha_t^{(\hat{m})}(s_t^{(\hat{m})}, v_t=v)\beta_t^{(\hat{m})}(s_t^{(\hat{m})}, v_t=v)}{c_v}\frac{\partial c_v}{\partial\theta} \\ &\quad - \sum_{t,s_t}\frac{\prod_{\hat{m}}\alpha_t^{(\hat{m})}(s_t^{(\hat{m})}, v_t=0)\beta_t^{(\hat{m})}(s_t^{(\hat{m})}, v_t=0)\cdot\prod_m g_v^m(s_t^{(m)})}{1-\sum_{v=1}^V c_v\cdot\prod_m g_v^m(s_t^{(m)})}\frac{\partial c_v}{\partial\theta} \end{aligned}$$

7.3 Belief State Estimation

7.3.1 Derivation of Eq. (4). We solve this maximization problem with the method of Lagrange multipliers.

$$\begin{aligned} L &= \sum_{t=1}^{T-1}\textstyle\sum_{\hat{m}}\sum_{s_t^{(\hat{m})}} q(s_t^{(\hat{m})})\log q(s_t^{(\hat{m})}) \\ &\quad - \sum_{t=1}^{T-1}\sum_{s_{t-1,t},a_{t-1},v_{t-1}} q(s_{t-1,t},a_{t-1},v_{t-1})\log\Big(\frac{q(s_{t-1,t},a_{t-1},v_{t-1})}{p(s_t,a_{t-1},v_{t-1},o_t|s_{t-1};\theta)}\Big) \\ &\quad + \textstyle\sum_{t,\hat{m},s_{t-1}^{(\hat{m})}}\alpha^{\hat{m}}_{t-1,s_{t-1}^{(\hat{m})}}\Big(\sum_{s_{t-1,t},a_{t-1},v_{t-1}\setminus s_{t-1}^{(\hat{m})}} q(s_{t-1,t},a_{t-1},v_{t-1}) - q(s_{t-1}^{(\hat{m})})\Big) \end{aligned}$$

Taking the derivative and setting it to zero, we get

$$\begin{aligned} &\frac{\partial L}{\partial q(s_{t-1,t},a_{t-1},v_{t-1})} = 0 \quad\Rightarrow\quad \text{for } t{=}1,\ldots T{-}1 \\ &q(s_{t-1,t},a_{t-1},v_{t-1}) = \frac{1}{Z_t}\exp\Big(\textstyle\sum_{\hat{m}}\alpha^{(\hat{m})}_{t-1,s_{t-1}^{(\hat{m})}}\Big)\cdot p(s_t,a_{t-1},v_{t-1},o_t|s_{t-1};\theta) \\ &\frac{\partial L}{\partial q(s_t^{(\hat{m})})} = 0 \quad\Rightarrow\quad q(s_t^{(\hat{m})}) = \frac{1}{Z_T^{(m)}}\exp\Big(\textstyle\sum_{\hat{m}}\alpha^{(\hat{m})}_{t-1,s_{t-1}^{(\hat{m})}}\Big) \end{aligned}$$

Marginalizing over $q(s_{t-1,t}, a_{t-1}, v_{t-1})$, we have

$$\begin{aligned} & q(s_{t-1,t}^{(\hat{m})}, a_{t-1}, v_{t-1}) \\ &= \frac{1}{Z_t}\exp\Big(\textstyle\sum_{\hat{m}}\alpha^{(\hat{m})}_{t-1,s_{t-1}^{(\hat{m})}}\Big)\sum_{s_{t-1,t}^{(\hat{m}')};\hat{m}'\neq\hat{m}}\exp\Big(\sum_{\hat{m}'}\alpha^{(\hat{m}')}_{t-1,s_{t-1}^{(\hat{m}')}}\Big)\cdot p(s_t,a_{t-1},v_{t-1},o_t|s_{t-1};\theta) \\ &:= \frac{1}{Z_t}\exp\Big(\textstyle\sum_{\hat{m}}\alpha^{(\hat{m})}_{t-1,s_{t-1}^{(\hat{m})}}\Big)\cdot p(s_t^{(\hat{m})},a_{t-1},v_{t-1},o_t^{(\hat{m})}|s_{t-1}^{(\hat{m})};\theta) \end{aligned}$$

We denote $\exp\left(\alpha^{(\hat{m})}_{t-1,s_{t-1}^{(\hat{m})}}\right)$ as $\alpha_t^{(\hat{m})}\left(s_t^{(\hat{m})}\right)$, and can compute α through a forward approach:

$$\begin{aligned} & q(s_t^{(\hat{m})}) = \textstyle\sum_{s_{t-1}^{(\hat{m})},a_{t-1},v_{t-1}} q(s_{t-1,t}^{(\hat{m})}, v_{t-1}) \\ \Rightarrow\quad & \alpha_t^{(\hat{m})}(s_t^{(\hat{m})}) = \frac{Z_t^{(\hat{m})}}{Z_t}\textstyle\sum_{s_{t-1}^{(\hat{m})},a_{t-1},v_{t-1}}\alpha_{t-1}^{(\hat{m})}(s_{t-1}^{(\hat{m})})\cdot p(s_t^{(\hat{m})},a_{t-1},v_{t-1},o_t^{(\hat{m})}|s_{t-1}^{(\hat{m})};\theta) \end{aligned}$$

The belief state can be estimated as follows:

$$b_t(s_t) = p(s_t|o_{1:t}) \propto q(s_t) \propto \textstyle\prod_{\hat{m}=1}^M \alpha_t^{(\hat{m})}(s_t^{(\hat{m})})$$

REFERENCES

[1] Uri Alon. 2006. *An introduction to systems biology: design principles of biological circuits*. CRC press.

[2] Kai Arulkumaran, Antoine Cully, and Julian Togelius. 2019. Alphastar: An evolutionary computation perspective. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 314–315.

[3] Josh Bongard and Hod Lipson. 2007. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* 104, 24 (2007), 9943–9948.

[4] Andrei Borshchev. 2013. *The big book of simulation modeling: multimethod modeling with AnyLogic 6*. AnyLogic North America Chicago.

[5] Anthony R Cassandra, Leslie Pack Kaelbling, and Michael L Littman. 1994. Acting optimally in partially observable stochastic domains. In *Aaai*, Vol. 94. 1023–1028.

[6] Ido Cohn, Tal El-Hay, Nir Friedman, and Raz Kupferman. 2009. Mean field variational approximation for continuous-time Bayesian networks. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 91–100.

[7] Mehrdad Farajtabar, Yichen Wang, Manuel Gomez Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. 2015. Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems*. 1954–1962.

[8] Matthew L Ginsberg. 1999. GIB: Steps toward an expert-level bridge-playing program. In *IJCAI*. Citeseer, 584–593.

[9] Eric A Hansen. 1998. An improved policy iteration algorithm for partially observable MDPs. In *Advances in Neural Information Processing Systems*. 1015–1021.

[10] Andreas Horni, Kai Nagel, and Kay W Axhausen. 2016. The multi-agent transport simulation MATSim. *Ubiquity, London* 9 (2016).

[11] Tommi Jaakkola, Satinder P Singh, and Michael I Jordan. 1995. Reinforcement learning algorithm for partially observable Markov decision problems. In *Advances in neural information processing systems*. 345–352.

[12] Li Li, Ding Wen, and Danya Yao. 2014. A survey of traffic control with vehicular communications. *IEEE Transactions on Intelligent Transportation Systems* 15, 1 (2014), 425–432.

[13] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[14] Marco Ajmone Marsan, Gianfranco Balbo, Gianni Conte, Susanna Donatelli, and Giuliana Franceschinis. 1994. *Modelling with generalized stochastic Petri nets*. John Wiley & Sons, Inc.

[15] Andrew Kachites McCallum. 1996. *Reinforcement Learning with Selective Perception and Hidden State*. Ph.D. Dissertation. University of Rochester.

[16] Nicolas Meuleau, Leonid Peshkin, Kee-Eung Kim, and Leslie Pack Kaelbling. 1999. Learning finite-state controllers for partially observable environments. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 427–436.

[17] William H Montgomery and Sergey Levine. 2016. Guided policy search via approximate mirror descent. In *Advances in Neural Information Processing Systems*. 4008–4016.

[18] Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. 2017. Policy gradient with value function approximation for collective multiagent planning. In *Advances in Neural Information Processing Systems*. 4319–4329.

[19] Manfred Opper and Guido Sanguinetti. 2008. Variational inference for Markov jump processes. In *Advances in Neural Information Processing Systems*. 1105–1112.

[20] Scott E Page. 2015. What sociologists should know about complexity. *Annual Review of Sociology* 41 (2015), 21–41.

[21] Leonid Peshkin, Nicolas Meuleau, and Leslie Kaelbling. 2001. Learning policies with external memory. *arXiv preprint cs/0103003* (2001).

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

[23] John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. 2015. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*. 3528–3536.

[24] Sven Seuken and Shlomo Zilberstein. 2007. Memory-Bounded Dynamic Programming for DEC-POMDPs.. In *IJCAI*. 2009–2015.

[25] Sriram Srinivasan, Marc Lanctot, Vinicius Zambaldi, Julien Pérolat, Karl Tuyls, Rémi Munos, and Michael Bowling. 2018. Actor-critic policy optimization in partially observable multiagent environments. In *Advances in Neural Information Processing Systems*. 3426–3439.

[26] Marc Toussaint and Amos Storkey. 2006. Probabilistic inference for solving discrete and continuous state Markov Decision Processes. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 945–952.

[27] Dustin Tran, Rajesh Ranganath, and David Blei. 2017. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*. 5523–5533.

[28] Martin J Wainwright, Michael I Jordan, et al. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning* 1, 1–2 (2008), 1–305.

[29] Fei-Yue Wang. 2010. Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications. *IEEE Transactions on Intelligent Transportation Systems* 11, 3 (2010), 630–638.

[30] Darren J Wilkinson. 2011. *Stochastic modelling for systems biology*. CRC press.

[31] Zhen Xu, Wen Dong, and Sargur N Srihari. 2016. Using social dynamics to make individual predictions: variational inference with a stochastic kinetic model. In *Advances in Neural Information Processing Systems*. 2783–2791.

[32] Fan Yang, Bo Liu, and Wen Dong. 2019. Optimal control of complex systems through variational inference with a discrete event decision process. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 296–304.

[33] Boqian Zhang, Jiangwei Pan, and Vinayak A Rao. 2017. Collapsed variational Bayes for Markov jump processes. In *Advances in Neural Information Processing Systems*. 3749–3757.