Coordinated Exploration in Concurrent Reinforcement Learning

Maria Dimakopoulou [1] **Benjamin Van Roy** [1]

Abstract

We consider a team of reinforcement learning agents that concurrently learn to operate in a common environment. We identify three properties – *adaptivity*, *commitment*, and *diversity* – which are necessary for efficient coordinated exploration and demonstrate that straightforward extensions to single-agent optimistic and posterior sampling approaches fail to satisfy them. As an alternative, we propose *seed sampling*, which extends posterior sampling in a manner that meets these requirements. Simulation results investigate how per-agent regret decreases as the number of agents grows, establishing substantial advantages of seed sampling over alternative exploration schemes.

1. Introduction

The field of reinforcement learning treats the design of agents that operate in uncertain environments and learn over time to make increasingly effective decisions. In such settings, an agent must balance between accumulating near-term rewards and probing to gather data from which it can learn to improve longer-term performance. A substantial literature, starting with (Kearns & Singh, 2002), has developed reinforcement learning algorithms that address this trade-off in a provably efficient manner. Until recently, most provably efficient exploration algorithms (e.g., (Jaksch et al., 2010)) have been based on UCB (upper confidence bound) approaches. Over the past few years, new approaches that build on and extend PSRL (posterior sampling for reinforcement learning) (Strens, 2000) have proved advantageous in terms of statistical efficiency (Osband et al., 2013; Osband & Van Roy, 2017a;b; 2014a;b). PSRL operates in a simple and intuitive manner. An agent learns over episodes of interaction with an uncertain environment, modeled as a Markov decision process (MDP). The agent is uncertain about the transition probabilities and rewards of the MDP and refines estimates as data is gathered. At the start of each episode, the agent samples an MDP from its current posterior distribution. This sample can be thought of as a random statistically plausible model of the environment given the agent's initial beliefs and data gathered up to that time. The agent then makes decisions over the episode as though the environment is accurately modeled by the sampled MDP.

In concurrent reinforcement learning (Silver et al., 2013; Pazis & Parr, 2013; Guo & Brunskill, 2015; Pazis & Parr, 2016), multiple agents interact with different instances of the same unknown environment, have perfect information sharing with one another, and learn in parallel how to achieve a common goal. The collection of agents can be viewed alternatively as a single agent with multiple threads. Hence, this literature is distinct from the multi-agent systems literature, which is often approached from a game-theoretic perspective with agents having different roles or objectives (e.g., (Vlassis, 2007; Chalkiadakis & Boutilier, 2003; Mertikopoulos & Zhou, 2018)) and without exploration considerations.

One might consider two straightforward extensions of PSRL to the concurrent setting. In one, at the start of each episode, each agent samples an independent MDP from the current posterior, which is conditioned on all data accumulated by all agents. Then, over the course of the episode, each agent follows the decision policy that optimizes its sampled MDP. The problem with this approach is that each agent does not benefit over the duration of the episode from the potentially vast quantities of data gathered by his peers. An alternative – which we will refer to as *Thompson resampling* – would be to have each agent independently sample a new MDP at the start of each time period within the episode, as done in (Kim, 2017). The new sample would be from a posterior distribution additionally conditioned on data gathered by all agents so far. However, as discussed in (Russo et al., 2017), this naive extension is disruptive to the agents' ability to explore the environment thoroughly. In particular, an agent may have to apply a coordinated sequence of actions over multiple time periods in order to adequately probe the environment. When MDPs are resampled independently over time periods, agents are taken off course. On the other hand, Guo & Brunskill (2015), Pazis & Parr (2013), and Pazis & Parr (2016) have proposed and studied UCB exploration schemes for concurrent reinforcement learning. Advantages of PSRL over UCB in single-agent contexts by

[1]Stanford University, California, USA. Correspondence to: Maria Dimakopoulou <madima@stanford.edu>, Benjamin Van Roy <bvr@stanford.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

themselves motivate extension to concurrent reinforcement learning. A more important motivating factor is that UCB approaches sometimes do not coordinate exploration in an effective manner, as our results show. The issue is that UCB approaches are deterministic, and as such, they do not diversify agent behaviors to effectively divide and conquer when there are multiple facets of the environment to explore.

The approaches we have discussed highlight potentially conflicting needs to adapt to new information, maintain the intent with which an agent started exploring the environment and diversify the exploratory effort among agents. Efficient coordinated exploration calls for striking the right balance. In this paper, we present a variation of PSRL – *seed sampling* – that accomplishes this. To focus on the issue of coordinated exploration, we consider a single-episode reinforcement learning problem in which multiple agents operate, making decisions and progressing asynchronously. In this context, we study the rate at which per-agent regret vanishes as the number of agents increases. Through this lens, we demonstrate that seed sampling coordinates exploration in an efficient manner and can dramatically outperform other approaches to concurrent reinforcement learning. In particular, the rate at which regret decays appears to be robust across problems, while for each alternative, there are problems where regret decays at a far slower rate.

We envision multiple application areas where *seed sampling* is poised to play an important role. One is in web services, where the idea is that each user is served by an agent. As agents interact with users, they share data and learn from each others' experiences. This data can be used to optimize the quality of each user's lifetime experience. The paper of (Silver et al., 2013) studied this application, introducing concurrent TD-learning. But to do this in an efficient manner, it is important to take exploratory actions and to diversify such actions across users. Seed sampling is able to structure such exploration in a systematic and robust manner. The control of autonomous vehicles presents another important application area. Here, each agent manages a single vehicle, and again, the agents learn from each other as data is gathered. The goal could be to optimize a combination of metrics, such as fuel consumption, safety, and satisfaction of transportation objectives. Efficient coordinated exploration can be key to the performance of the team.

2. Problem Formulation

Consider a time-homogeneous, single-episode MDP, which is identified by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho, H)$, where $\mathcal{S}$ is the finite state space, $\mathcal{A}$ is the finite action space, $\mathcal{R}$ is the reward model, $\mathcal{P}$ is the transition model, ρ is the initial state distribution and H is the horizon. Consider K agents, who explore and learn to operate in parallel in this common environment. Each kth agent begins at state $s_{k,0}$ and takes an action at arrival times $t_{k,1}, t_{k,2}, \ldots, t_{k,H}$ of an independent Poisson process with rate $\lambda = 1$. At time $t_{k,m}$, the agent takes action $a_{k,m}$, transitions from state $s_{k,m-1}$ to state $s_{k,m}$ and observes reward $r_{k,m}$. The agents are uncertain about the transition structure $\mathcal{P}$ and/or the reward structure $\mathcal{R}$, over which they share common priors. There is clear value in sharing data across agents, since there is commonality across what agents aim to learn. Agents share information in real time and update their posterior, so that when selecting an action at time $t_{k,m}$, the kth agent can base his decision on observations made by all agents prior to that time.

Denote as $\mathcal{T} = \left\{0, \ldots, \max_{k\in\{1,\ldots,K\}} t_{k,H}\right\}$. We will define all random variables with respect to a filtered probability space $(\Omega, \mathbb{F}, (\mathbb{F}_t)_{t\in\mathcal{T}}, \mathbb{P})$. As a convention, variables indexed by t are $\mathbb{F}_t$-measurable and therefore, variables indexed by k, m are $\mathbb{F}_{t_{k,m}}$-measurable.

The total reward accrued by the agents is $\sum_{k=1}^{K}\sum_{m=1}^{H} r_{k,m}$ and the expected mean regret per agent is defined by

$$\text{BayesRegret}(K) = \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\sum_{m=1}^{H}(R^* - r_{k,m})\right]$$

where R^* is the optimal reward.

We now consider some examples that illustrate this problem formulation.

Example 1 (**Maximum Reward Path**). Consider an undirected graph with vertices $\mathcal{V} = \{1, \ldots, N\}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The probability of any two vertices being connected is p. Let $\theta \in \Re_+^{|\mathcal{E}|}$ be the vector of edge weights. We treat θ as a vector with an understanding that edges are sorted according to a fixed but arbitrary order. The state space $\mathcal{S}$ is the set of vertices $\mathcal{V}$ and the action space from each vertex $v \in \mathcal{V}$ is the set of edges incident to v. When action $(v, u) = e \in \mathcal{E}$ is taken from state v, the agent transitions deterministically to state u and observes reward r_e, which is a noisy observation of the weight of edge e, such that $\mathbb{E}[r_e|\theta] = \theta_e$. The K agents are uncertain about the edge weights and share a common $\mathcal{N}(\mu_0, \Sigma_0)$ prior over $\ln\theta$. Denote as $e_{k,m} = (v_{k,m-1}, v_{k,m})$ the mth edge of the kth agent's path traversed at time $t_{k,m}$. For each $m = 1, \ldots, H$ the agent observes a reward $r_{k,m}$, distributed according to $\ln r_{k,m}|\theta \sim \mathcal{N}(\ln\theta_{e_{k,m}} - \sigma^2/2, \sigma^2)$. The K agents start from the same vertex $v \in \mathcal{V}$. The objective is, starting from vertex v, to traverse the path $(v_0 = v, v_1, \ldots, v_H)$ that maximizes $\sum_{m=1}^{H}\theta_{(v_{m-1},v_m)}$, i.e., to find the maximum reward path from vertex v with exactly H edges.

Example 2 (**Bipolar Chain**). Consider the directed graph of Figure 1. The chain has an even number of vertices, N, $\mathcal{V} = \{0, 1, \ldots, N-1\}$. The endpoints of the chain are absorbing. The set of edges is $\mathcal{E} = \{(v, v+1), \forall v =$

$1, \ldots, N-3\} \cup \{(v+1, v), \forall v = 1, \ldots, N-3\} \cup (1,0) \cup (N-2, N-1)$. The leftmost edge $e_L = (1,0)$ has weight θ_L and the rightmost edge $e_R = (N-2, N-1)$ has weight θ_R, such that $|\theta_L| = |\theta_R| = N$ and $\theta_R = -\theta_L$. All other edges $e \in \mathcal{E} \backslash \{e_L, e_R\}$ have weight $\theta_e = -1$. The agents do not know whether $\theta_L = N, \theta_R = -N$ or $\theta_L = -N, \theta_R = N$ and they share a common prior that assigns probability $p = 0.5$ to either scenario. Each one of the K agents starts from vertex $v_S = N/2$. Denote as $e_{k,m}$ the edge traversed at the mth step of the kth agent's path and $\theta_{k,m}$ the respective weight. Further, denote as $t_{k,h}$, $1 \leq h \leq H$ the time at which the kth agent reaches either endpoint with the hth traversal being the last one in the agent's path. The kth agent's objective is to maximize $\sum_{m=1}^{h} \theta_{k,m}$. The optimal reward is $R^* = N/2$ if $\theta_L = N, \theta_R = -N$ and $R^* = N/2 + 1$ if $\theta_L = -N, \theta_R = N$ because the leftmost endpoint $v_L = 0$ is one vertex further from the start $v_S = N/2$ than the rightmost endpoint $v_R = N - 1$ and requires the traversal of one more penalizing edge with weight -1.

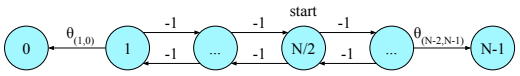


Figure 1: Graph of "Bipolar Chain" example

Example 3 (Parallel Chains). Consider the directed graph of Figure 2. Starting from vertex 0, each of the K agents chooses one of the C chains. Once a chain is chosen, the agent cannot switch to another chain. All the edges of each chain c have zero weights, apart from the edge incoming to the last vertex of the chain, which has weight θ_c. Let $\theta \in \Re^C$ be the vector of these edge weights for the C chains. The K agents are uncertain about θ, over which they share a common $\mathcal{N}(\mu_0, \Sigma_0)$ prior. Denote as c_k the chain chosen by the kth agent. When traversing the last edge at time $t_{k,H}$, the agent observes reward $r_{k,H}$ distributed according to $r_{k,H}|\theta \sim \mathcal{N}(\theta_{c_k}, \sigma^2)$. For all other transitions at times $t_{k,m}$, $m = 1, \ldots, H-1$, the kth agent observes reward $r_{k,m} = 0$. The objective is to choose the chain with the maximum reward.

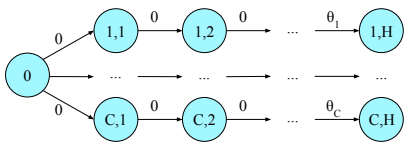


Figure 2: Graph of "Parallel Chains" example

3. Algorithms

Three properties are necessary for efficient coordinated exploration in concurrent reinforcement learning:

Property 1 (Adaptivity). *Adapt as data becomes available to make effective use of new information.*

Property 2 (Commitment). *Maintain the intent to carry out probing action sequences that span multiple periods.*

Property 3 (Diversity). *Divide-and-conquer learning opportunities among agents.*

As we discuss in Section 3.1, straightforward extensions of provably efficient single-agent reinforcement learning algorithms fail to meet these requirements. In Section 3.2, we introduce the concept of *seed sampling*, which leads to algorithms that simultaneously satisfy these three properties.

All algorithms we consider share some common structure, which we will now describe. The K concurrent agents share a prior distribution $\mathcal{F}_0$ of the MDP $\mathcal{M}$. Denote by $\mathcal{F}_t$ the posterior distribution, given the history of observations $\mathcal{H}_{t-1}$ available up to time t. At each time $t_{k,m}$, the agent generates an MDP $\mathcal{M}_{k,m}$, computes the optimal policy $\pi_{k,m}$ for $\mathcal{M}_{k,m}$, takes a single action $a_{k,m} = \pi_{k,m}(s_{k,m-1})$, transitions to state $s_{k,m}$ and observes reward $r_{k,m}$. The observation $(s_{k,m-1}, a_{k,m}, s_{k,m}, r_{k,m})$ is used to update the shared posterior distribution of $\mathcal{M}$. Therefore, at time $t_{k',m'} > t_{k,m}$, the k'th agent can use the knowledge gained from this observation in order to take his m'th action. The key difference between the studied algorithms is how each kth agent forms his MDP $\mathcal{M}_{k,m}$ at time $t_{k,m}$.

3.1. Baseline Algorithms

First, we discuss the straight-forward adaptation of provably efficient single-agent reinforcement learning algorithms to the concurrent reinforcement learning setting. However, neither of these baselines achieve coordinated exploration in concurrent reinforcement learning, either because the agents, when adapting to new information, do not maintain the level of intent required to ensure thorough exploration or because the agents do not diversify their exploratory effort in a manner that mutually benefits their common learning.

3.1.1. Thompson Resampling

At time $t_{k,m}$, the kth agent samples MDP $\mathcal{M}_{k,m}$ from the posterior $\mathcal{F}_{t_{k,m}}$. If at time $t_{k,m}$ of the mth action of the kth agent and at time $t_{k',m'}$ of the m'th action of the k'th agent the posterior is the same, $\mathcal{F}_{t_{k,m}} \equiv \mathcal{F}_{t_{k',m'}}$, the kth agent and the k'th agent will form a different MDP. Therefore, the agents will diversify their exploration efforts. However, resampling an MDP independently at each time period may break the agent's commitment to a sequence of actions that extend over multiple time periods. This commitment is necessary for learning in an environment with delayed consequences, and hence the learning performance may suffer.

To demonstrate the importance of Property 2 (Commitment), consider the Bipolar Chain example of Section 2. Assume that the kth agent samples an MDP at time $t_{k,1}$, in which the left-most edge is positive and the right-most edge is negative.

Therefore, the kth agent decides to move left at $t_{k,1}$. When the kth agent re-samples an MDP at time $t_{k,2}$, the left-most edge may now be negative and the right-most edge may now be positive due to randomness, even if no other agent has gathered information to warrant this change in the sampled MDP. As a consequence, the kth agent moves right at $t_{k,2}$, undoing his previous move, incurring unnecessary cost and most importantly delaying the traversal of either the left-most or the right-most edge, which would produce information valuable for all agents.

3.1.2. CONCURRENT UCRL

At time $t_{k,m}$, the kth agent forms confidence bounds for the reward structure $\mathcal{R}$ and the transition structure $\mathcal{P}$ that define the set of statistically plausible MDPs given the posterior $\mathcal{F}_{t_{k,m}}$. The kth agent chooses the MDP $\mathcal{M}_{k,m}$ that maximizes the achievable average reward subject to these confidence bounds. This algorithm is deterministic and does not suffer from the flaw of Thompson resampling. Note, however, that if at time $t_{k,m}$ of the mth action of the kth agent and at time $t_{k',m'}$ of the m'th action of the k'th agent the posterior is the same, $\mathcal{F}_{t_{k,m}} \equiv \mathcal{F}_{t_{k',m'}}$, the kth agent and the k'th agent will form the same MDP. Therefore, the agents may not always diversify their exploration efforts.

To demonstrate the importance of Property 3 (Diversity), consider the Parallel Chains example of Section 2 and assume that the parallel chains' last edge weights, θ_c, are independent. Further, assume that for any pair of chains c, c', the prior means of θ_c, $\theta_{c'}$ are the same, $\mu_{0,c} = \mu_{0,c'}$, but the prior variances of θ_c, $\theta_{c'}$ differ, $\sigma_{0,c} \neq \sigma_{0,c'}$. Then, UCRL will direct all K agents to the chain with the maximum prior variance and will not diversify exploratory effort to the other $C-1$ chains. As the horizon H gets larger, the learning performance benefits less and less from an increased number of parallel agents and the expected mean regret per agent does not improve due to lack of diversity.

3.2. Seed Sampling Algorithms

We now present the concept of *seed sampling*, which offers an approach to designing efficient coordinated exploration algorithms that satisfy the three aforementioned properties. The idea is that each concurrent agent independently samples a random seed, such that the mapping from seed to MDP is determined by the prevailing posterior distribution. Independence among seeds diversifies exploratory effort among agents (Property 3). If the mapping is defined in an appropriate manner, the fact that the agent maintains a consistent seed leads to a sufficient degree of commitment (Property 2), while the fact that the posterior adapts to new data allows the agent to react intelligently to new information (Property 1).

In the subsections that follow, we discuss ways to define the mapping from seed and posterior distribution to sample. Note that the mappings we present represent special cases that apply to specific problem classes. The idea of seed sampling is broader and can be adapted to other problem classes, as we will explore further in Section 4.

Let $\mathcal{G}_0$ be a deterministic function mapping a seed $z \sim \mathcal{Z}$ to MDP $\mathcal{M}_0$, such that $\mathcal{M}_0 \sim \mathcal{M}$. At time t, the deterministic function mapping $\mathcal{G}_t$ is generated based on $\mathcal{G}_0$ and the history of observations $\mathcal{H}_{t-1}$ available up to this time. At the beginning of the episode, each kth agent samples seed $z_k \sim \mathcal{Z}$. At time $t_{k,m}$, the kth agent samples an MDP according to $\mathcal{M}_{k,m} = \mathcal{G}_{t_{k,m}}(z_k)$. The intuition behind the seed sampling algorithms is that each agent forms its own sample of the MDP at each time period, which is distributed according to the posterior over $\mathcal{M}$ based on all agents' observations, while the randomness injected to the agent's samples remains fixed throughout the horizon, allowing the agent to maintain the necessary commitment to action sequences that span multiple time periods.

3.2.1. EXPONENTIAL-DIRICHLET SEED SAMPLING

The agents are uncertain about the transition structure $\mathcal{P}$ over which they hold a common Dirichlet prior $\mathcal{F}_0^{\mathcal{P}}$. The prior over the transition probabilities associated with each state-action pair (s,a) is Dirichlet-distributed with parameters $\alpha_0(s,a) = (\alpha_0(s,a,s'),\ \forall s' \in \mathcal{S})$. The Dirichlet parameters are incremented upon each state transition and at time t, the posterior given the history of observations $\mathcal{H}_{t-1}$ is $\mathcal{F}_t^{\mathcal{P}}$. At time t, the transition probabilities from the state-action pair (s,a) is Dirichlet-distributed with parameters $\alpha_t(s,a) = (\alpha_t(s,a,s'),\ \forall s' \in \mathcal{S})$. At the beginning of the episode, each kth agent samples $|\mathcal{S}|^2|\mathcal{A}|$ sequences of independent and identically distributed seeds $z_{k,s,a,s'} = (z_{k,s,a,s',i},\ i = 1,2,\ldots,)$ such that $z_{k,s,a,s',i} \sim \text{Exp}(1)$. The mapping from seed to transition structure is defined as $\mathcal{G}_t(z) := \Big\{ p_{s,a}(s') = \sum_{i=1}^{\alpha_t(s,a,s')} z_{s,a,s',i} \Big/ \sum_{\tilde{s}\in S} \sum_{i=1}^{\alpha_t(s,a,\tilde{s})} z_{s,a,\tilde{s},i},\ \forall (s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S} \Big\}$. Then, each $p_{s,a}$ in $\mathcal{G}_t(z)$ is Dirichlet distributed with parameters $\alpha_t(s,a)$ due to the fact that (a) if $Y_1,\ldots,Y_d$ are independently distributed Gamma random variables with shape parameters $a_1,\ldots,a_d$, then $X = (X_1,\ldots,X_d)$ with $X_i = Y_i / \sum_{j=1}^d Y_j$ is d-dimensional Dirichlet distributed with parameters $a_1,\ldots,a_d$, and (b) any Gamma with shape parameter a can be represented as the sum of a Exp(1) random variables (Gentle, 2013). The transition structure of the sampled MDP of the kth agent at time $t_{k,m}$ is given by $\mathcal{G}_{t_{k,m}}(z_k)$.

3.2.2. STANDARD-GAUSSIAN SEED SAMPLING

The agents are uncertain about the parameters $\theta \in \Re^{|\mathcal{S}||\mathcal{A}|}$ of the reward structure, over which they share a common

normal or lognormal prior $\mathcal{F}_0^{\mathcal{R}}$ with parameters μ_0 and Σ_0. The posterior over θ at time t, given the history of observations $\mathcal{H}_{t-1}$ available up to this time, is $\mathcal{F}_t^{\mathcal{R}}$ and is normal or lognormal with parameters μ_t and Σ_t. In either case, conjugacy properties result in simple rules for updating the posterior distribution's parameters upon each observation in $\mathcal{H}_{t-1}$. Consider Example 1 and Example 3 of Section 2. At the beginning of the episode, each kth agent samples seed $z_k = \mathcal{N}(0, I)$. In the case of normal prior, as in Example 3 (Parallel Chains), the mapping from seed to the reward structure's parameters is defined as $\mathcal{G}_t(z) := \mu_t + D_t z$, where D_t is the positive definite matrix such that $D_t^T D_t = \Sigma_t$. Then, $\mathcal{G}_t(z)$ is a multivariate normal with mean vector μ_t and covariance matrix Σ_t (Gentle, 2009). In the case of lognormal prior, as in Example 1 (Maximum Reward Path), the mapping from seed to the reward structure's parameters is $\mathcal{G}_t(z) := \exp(\mu_t + D_t z)$, where D_t is defined as before. Similarly, $\mathcal{G}_t(z)$ is a multivariate lognormal with parameters μ_t and Σ_t. The reward structure of the sampled MDP of the kth agent at time $t_{k,m}$ has parameters $\hat{\theta}_{k,m} = \mathcal{G}_{t_{k,m}}(z_k)$.

3.2.3. Martingalean-Gaussian Seed Sampling

The agents are uncertain about the parameters $\theta \in \Re^{|\mathcal{S}||\mathcal{A}|}$ of the reward structure, over which they share a common normal or lognormal prior $\mathcal{F}_0^{\mathcal{R}}$ with parameters μ_0 and Σ_0. Define seed $z = (\hat{\theta}_0, w)$ with distribution $\mathcal{Z}$ such that $\hat{\theta}_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$ and $\{w_j : j = 0, 1, \dots\}$ is an IID sequence of $\mathcal{N}(\mu_w, \sigma_w^2)$. At time t, the history up to this time $\mathcal{H}_{t-1}$ consists of observations $\{(s_j, a_j, s'_j, r_j), j = 1, \dots, |\mathcal{H}_{t-1}|\}$. The deterministic mapping $\mathcal{G}_t$ from seed $z = (\hat{\theta}_0, w)$ to reward structure parameters is a model fit to the sample $\hat{\theta}_0$ from the prior and the observations in $\mathcal{H}_{t-1}$ randomly perturbed by w. In the case of normal prior, $\{w_j : j = 0, 1, \dots\}$ is an IID sequence of $\mathcal{N}(0, \sigma^2)$ and $r_j|\theta \sim \mathcal{N}\left(\theta_{(s_j,a_j)}, \sigma^2\right)$. The mapping at time t from seed $z = (\hat{\theta}_0, w)$ to the reward structure's parameters is defined as $\mathcal{G}_t(z) := \arg\min_\rho \Big((\rho - \hat{\theta}_0)^T \Sigma_0^{-1} (\rho - \hat{\theta}_0) + \frac{1}{\sigma^2} \sum_{j=1}^{|\mathcal{H}_{t-1}|} (o_j^T \rho - r_j - w_j)^2\Big)$, where o_j is the one-hot vector $|\mathcal{S}||\mathcal{A}| \times 1$, whose positive element corresponds to the state-action pair of the jth observation in $\mathcal{H}_{t-1}$, r_j is the reward of the jth observation in $\mathcal{H}_{t-1}$ and w_j is a component of the seed which corresponds to the perturbation of the reward of the jth observation in $\mathcal{H}_{t-1}$. In the case of lognormal prior, $\{w_j : j = 0, 1, \dots\}$ is an IID sequence of $\mathcal{N}(-\sigma^2/2, \sigma^2)$ and $\ln r_j|\theta \sim \mathcal{N}\left(\ln \theta_{(s_j,a_j)} - \sigma^2/2, \sigma^2\right)$. The mapping at time t from seed $z = (\hat{\theta}_0, w)$ to the reward structure's parameters is similar as in the normal case, but instead of fitting to the rewards r_j, we fit to $\ln r_j$.

Consider again Example 3 (Parallel Chains) and Example 1 (Maximum Reward Path). At the beginning of the episode, each kth agent samples seed $z_k = (\hat{\theta}_{k,0}, w_k)$ distributed according to $\mathcal{Z}$. At time $t_{k,m}$, the kth agent generates $\hat{\theta}_{k,m} = \mathcal{G}_t(z_k)$, which is a model fit to his sample $\hat{\theta}_{k,0}$ from the prior and to the observations in the history $\mathcal{H}_{t_{k,m}-1}$ perturbed by w_k, $\hat{\theta}_{k,m} = \left(O^T O + \sigma^2 \Sigma_0^{-1}\right)^{-1} \left(O^T (R + W^k) + \sigma^2 \Sigma_0^{-1} \hat{\theta}_{k,0}\right)$, where O is the $\left|\mathcal{H}_{t_{k,m}-1}\right| \times |\mathcal{S}||\mathcal{A}|$ matrix whose jth row is o_j^T, R is the $\left|\mathcal{H}_{t_{k,m}-1}\right| \times 1$ vector whose jth element is r_j in the normal prior case and $\ln r_j$ in the lognormal prior case and W^k is the $\left|\mathcal{H}_{t_{k,m}-1}\right| \times 1$ vector whose jth element is $w_{k,j} \sim \mathcal{N}(0, \sigma^2)$ in the normal prior case and is $w_{k,j} \sim \mathcal{N}(-\sigma^2/2, \sigma^2)$ in the lognormal prior case.

Proposition 1. *Conditioned on $\mathbb{F}_{t_{k,m}-1}$, $r_{k,m}$ and $s_{k,m}$, $\hat{\theta}_{k,m}$ is distributed according to the posterior of θ.*

Proposition 2. *For each agent k, denote as $\mathcal{T}_k = \{0, \dots, t_{k,H}\}$ and consider a probability measure $\tilde{\mathbb{P}}_k$ defined on $(\Omega, \mathbb{F}, (\mathbb{F}_t)_{t \in \mathcal{T}_k})$, for which $\hat{\theta}_{k,0}$ is deterministic, θ is distributed $\mathcal{N}(\hat{\theta}_{k,0}, 2\Sigma_0)$. Then, $\hat{\theta}_{k,m}$ is a martingale with respect to $\tilde{\mathbb{P}}_k$.*

Proposition 1 follows from Lemma 1 in the supplement of (Lu & Van Roy, 2017) and is core to sampling an MDP that follows the posterior distribution based on the data gathered by all agents (Property 1). Proposition 2 follows from the definitions and motivates the name of this algorithm.

4. Computational Results

In this section, we present computational results that demonstrate the robustness of seed sampling algorithms of Section 3.2 versus the baseline algorithms of Section 3.1.[1] In sections 4.1 and 4.2, we present two simple problems that highlight the weaknesses of concurrent UCRL and Thompson resampling and demonstrate how severely performance may suffer due to violation of any among Properties 1, 2, 3. In Section 4.3, we demonstrate the relative efficiency of seed sampling in a more complex problem.

4.1. Bipolar Chain

Consider the directed graph of Figure 1 and the description of Example 2 in Section 2. The agents' objective is to maximize the accrued reward. However, the agents do not know whether the leftmost edge e_L has weight $\theta_L = N$ or whether the rightmost edge has weight $\theta_R = N$. The agents share a common prior that assigns equal probability $p = 0.5$ to either scenario. When any of the K agents traverses e_L or e_R for the first time, all K agents learn the true values of θ_L, θ_R. Denote as T the time when the true MDP is revealed. The horizon is $H = 3N/2$. The horizon is selected in such a way, so that if an agent picks the wrong direction and moves in every time period towards

[1] For a demo, see: https://youtu.be/xjGK-wm0PkI

the wrong endpoint, if the true values of θ_L, θ_R are revealed before the wrong endpoint is reached, this agent has enough time to correct the trajectory and reach the correct endpoint. The optimal reward is $R^* = N/2$ if $\theta_L = N, \theta_R = -N$ and $R^* = N/2 + 1$ if $\theta_L = -N, \theta_R = N$ because the leftmost endpoint $v_L = 0$ is one vertex further from the start $v_S = N/2$ than the rightmost endpoint $v_R = N - 1$ and requires the traversal of one more penalizing edge with weight -1. We now examine how seed sampling, concurrent UCRL and Thompson resampling behave in this setting.

In seed sampling, each kth agent samples a seed $z_k \sim$ Bernoulli(p), which remains fixed for the entire duration of the episode. The mapping from seed $z_{k,m}$ to sampled MDP $\mathcal{M}_{k,m}$ at time $t_{k,m} < T$ is determined by $\hat{\theta}_{L,k,m}$ and $\hat{\theta}_{R,k,m}$ which are defined as $\hat{\theta}_{L,k,m} = N \cdot \text{sign}(z_k - 0.5)$, $\hat{\theta}_{R,k,m} = -\hat{\theta}_{L,k,m}$. After one of the K agents traverses e_L or e_R, the sampled MDP of each kth agent who has not terminated is the true MDP, $\hat{\theta}_{L,k,m} = \theta_L, \quad \hat{\theta}_{R,k,m} = \theta_R$, satisfying Property 1. Note that in seed sampling, among the agents who start before the true MDP is revealed, i.e., $\{k : t_{k,0} < T\}$, half go left and half go right in expectation, satisfying Property 3. Thanks to the seed $z_{k,m}$ the sampled MDP $\mathcal{M}_{k,m}$ remains fixed in every step of the kth agent's trajectory until the true MDP is learned. Therefore, all agents commit to reaching either the left or the right endpoint of the chain depending on the seed they sampled, until the correct direction is discovered by one of the agents, satisfying Property 2. When the correct direction is revealed, the horizon $H = 3N/2$ allows all agents who have picked the wrong direction but have not yet terminated to change their trajectory and eventually to reach the correct endpoint.

In concurrent UCRL, the agents are initially optimistic that they can achieve the maximum attainable reward, which is $N/2 + 1$ in the scenario that the rightmost edge (i.e., the closest one to the start) has the positive weight, $\theta_R = N$. Each kth agent at time $t_{k,m} < T$ chooses an MDP that is defined as $\hat{\theta}_{L,k,m} = -N$, $\hat{\theta}_{R,k,m} = N$. Therefore, all agents who start before the true MDP is revealed, i.e., $\{k : t_{k,0} < T\}$ go right, violating Property 3. Note that, in this particular example, diversification is not essential to exploration, since going towards a single direction will still reveal the true MDP. When the correct direction is revealed and it is not the rightmost endpoint, all agents who have not terminated change their trajectory and eventually reach the correct endpoint. If the correct direction is the rightmost endpoint, no agent has to change trajectory.

In Thompson resampling, each kth agent at time $t_{k,m} < T$ samples an MDP which is defined as $\hat{\theta}_{L,k,m} = N \cdot \text{sign}\left(\text{Bernoulli}(p) - 0.5\right)$, $\hat{\theta}_{R,k,m} = -\hat{\theta}_{L,k,m}$. Note that for two subsequent time periods $t_{k,m} < t_{k,m+1} < T$, the sampled MDP of the kth agent may differ due to randomness in drawing a Bernoulli(p) sample each time. Therefore, the kth agent who decided to go towards one direction at time $t_{k,m}$ may change his decision and go towards the opposite direction at time $t_{k,m+1}$, violating Property 2. In this setting, violation of Property 2 is detrimental to exploration. The horizon H of each one of the K agents is consumed to meaningless oscillations and it is very difficult for any agent to reach either endpoint of the chain. The larger the number of vertices in the chain is, the more unlikely becomes for any agent to ever discover the true MDP.

Figure 3 shows the mean regret per agent for $N = 100$ number of vertices in the chain as the number of concurrent agents increases. The figure presents averages over hundreds of simulations. As the number of agents increases, in seed sampling and concurrent UCRL more and more agents have not yet started their trajectory or moved further away from the start towards the wrong direction the moment the true MDP is revealed. Therefore the mean regret per agent decreases. On the other hand, in Thompson resampling, the lack of commitment to exploring either direction prevents the true MDP to be discovered before the horizon expires, even for a very large number of agents. As a result the mean regret per agent does not improve.

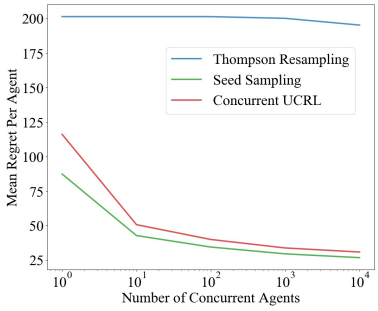


Figure 3: Performance of the algorithms of Section 3 in the "Bipolar Chain" example with $N = 100$ vertices and $H = 150$ horizon in terms of mean regret per agent as the number of agents increases.

4.2. Parallel Chains

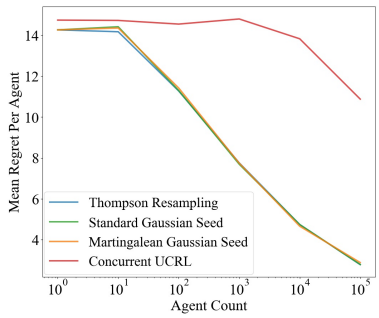
Consider the directed graph of Figure 2 and the description of Example 3 in Section 2. The agents' objective is to maximize the accrued reward by choosing the chain whose last edge has the largest weight $c^* = \text{argmax}_c \theta_c$. Recall that the weight of the last edge of chain c is denoted as θ_c and $\theta = (\theta_1, \ldots, \theta_C)$ When the last edge of chain c is traversed, the reward is a noisy observation of θ_c such that $r_c|\theta \sim \mathcal{N}(\theta_c, \sigma^2)$. However, the agents do not know the true value θ and they share a common, well-specified prior on it. Assume that all the θ_c, $c = 1, \ldots, C$ are independent and that the prior on the cth chain's last edge weight is $\mathcal{N}(\mu_c, \sigma_c^2)$. Further assume that $\forall c \in \{1, \ldots, C\}$, the prior mean is the same, $\mu_c = \mu_0$, and the prior variance increases as we move from higher to lower chains, $\sigma_c^2 = \sigma_0^2 + c$.

In this setting, martingalean-Gaussian seed sampling, standard-Gaussian seed sampling and Thompson resampling are expected to have identical performance. Thanks to sampling the seeds independently, the martingalean-Gaussian seed sampling and standard-Gaussian seed sampling agents construct MDPs in which different chains appear to be optimal. This is also the case for Thompson resampling agents, who sample their MDPs independently from the prior. As a result, the martingalean-Gaussian seed sampling, standard-Gaussian seed sampling and Thompson resampling agents are directed to different chains and satisfy Property 3. Note that, unlike martingalean-Gaussian seed sampling and standard-Gaussian seed sampling, Thompson resampling does not satisfy Property 2 but in this setting this does not impact the learning performance. A Thompson resampling agent k may draw an MDP at time $t_{k,0}$ in which chain c is optimal, but due to resampling at time $t_{k,1}$ his belief may change and another chain $c' \neq c$ may appear to be optimal. However, since transitions from one chain to another are not possible in the directed graph of Figure 2, the agent has no choice but exploring his initial choice, which is chain c. Even if inherently Thompson resampling lacks the commitment of martingalean-Gaussian seed sampling and standard-Gaussian seed sampling, the structure of the problem forces the Thompson resampling agents to maintain intent and perform equally well.

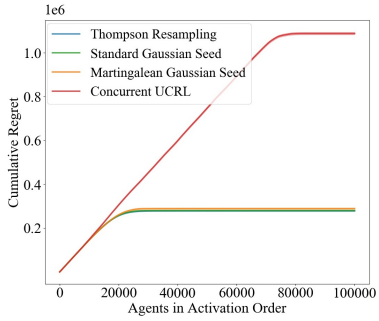
On the other hand, the concurrent UCRL agents are optimistic that they can achieve the maximum attainable reward and they are all directed to chain C for which the upper confidence bound of the weight of the last edge is the largest. Once enough agents have traversed the last edge of chain C and the posterior variance on θ_C becomes lower than the prior variance on θ_{C-1}, the optimistic driven behavior directs the agents who have not left the source to chain $C-1$. As long as there are agents who have not left the source, this way of exploration repeats until some agents are directed to chain 1. The violation of Property 3 leads to a wasteful allocation of the agents' exploratory effort, as all agents are directed to gather similar information. This is detrimental to learning performance, as an agent k with a later activation time $t_{k,0}$ will not have all the information to make the optimal choice of chain that he could have made if the agents who started before him had explored all the chains.

Consider the specification of the problem with $C = 10$ chains, horizon (or equivalently number of vertices in each chain) $H = 5$, $\theta_c \sim \mathcal{N}(0, 100 + c)$, $\forall c \in \{1, \ldots, C\}$ and likelihood of observed reward when the last edge of chain c is traversed $r_c|\theta_c \sim \mathcal{N}(\theta_c, 1)$. Figure 4a shows the mean regret per agent achieved by the algorithms as the number of agents increases and Figure 4b shows the cumulative regret of $100,000$ concurrent agents, with the agents ordered in ascending activation time $t_{k,0}$. Both figures present averages over hundreds of simulations.

The figures demonstrate that UCB approaches to concurrent reinforcement learning do not efficiently coordinate, and as a result, performance may suffer severely. In order for concurrent UCRL to achieve the same mean regret per agent that martingalean-Gaussian seed sampling, standard-Gaussian seed sampling and Thompson resampling achieve with 100 agents, 100,000 agents are required.



(a) Mean regret per agent as the number of agents increases.



(b) Cumulative regret of 100,000 concurrent agents ordered in ascending activation time with 95% confidence interval.

Figure 4: Performance of the algorithms of Section 3 in the "Parallel Chains" example with $C = 10$ chains, $H = 5$ number of vertices per chain, $\theta_c \sim \mathcal{N}(0, 100 + c)$, $\forall c \in \{1, \ldots, 10\}$.

4.3. Maximum Reward Path

We now present the performance of the algorithms in a more complex problem. Consider the description of Example 1. The agents start from the same vertex and their goal is to make H edge traversals that will return the highest reward. Initially, the agents do not know the edge weights. The edge weights are assumed to be independent and the agents share a common prior $\mathcal{N}(\mu_e, \sigma_e^2)$ over $\ln \theta_e$, $\forall e \in \mathcal{E}$. Every time edge e is traversed, the observed reward r_e is distributed according to $\ln r_e|\theta_e \sim \mathcal{N}(\ln \theta_e - \sigma^2/2, \sigma^2)$ and the common posterior of all agents is updated according to $\mu_e \leftarrow \frac{\sigma^2 \mu_e + \sigma_e^2 \left(\ln r_e + \sigma^2/2\right)}{\sigma_e^2 + \sigma^2}$ and $\sigma_e^2 \leftarrow \frac{\sigma_e^2 \sigma^2}{\sigma_e^2 + \sigma^2}$. The kth agent, at time $t_{k,m}$, $m = 1, \ldots, H$, constructs MDP $\mathcal{M}_{k,m} = \hat{\theta}_{k,m} = \{\hat{\theta}_{k,m,e}, e \in \mathcal{E}\}$ and from vertex $v_{k,m}$ computes the maximum reward path of $H - m + 1$ steps.

In Thompson resampling, the kth agent's sampled MDPs

at time $t_{k,m}$ and time $t_{k,m+1}$ may differ significantly. Part of this difference is due to the fact that for some edges observations were made between $t_{k,m}$ and $t_{k,m+1}$. However, $\mathcal{M}_{k,m}$ and $\mathcal{M}_{k,m+1}$ may also have different weights for edges that were not traversed between $t_{k,m}$ and $t_{k,m+1}$ due to randomness. Therefore, the kth agent may be enticed to redirect towards an edge which has a large weight in $\mathcal{M}_{k,m+1}$ but did not have a large weight in $\mathcal{M}_{k,m}$, even if this change in beliefs is not substantiated by true observations. As a result, Thompson resampling agents violate Property 2 and are susceptible to myopic behavior, which harms the agents' ability to explore deep in the graph in order to identify the maximum reward path of fixed length.

In concurrent UCRL, the agents are immune to the distractions suffered by Thompson resampling agents, as they construct deterministic upper confidence bounds from the common posteriors on the edge weights. However, the kth agent at time $t_{k,m}$ and the k'th agent at time $t_{k',m'} > t_{k,m}$ have identical beliefs on all the edges that have not been observed between $t_{k,m}$ and $t_{k',m'}$. Therefore, the path chosen by k'th agent may be very similar or even identical to the path chosen by kth agent. This lack of diversity (Property 3) delays the exploration of the graph's edges and the identification of the maximum reward path of fixed length.

In standard-Gaussian seed sampling or martingalean-Gaussian seed sampling, each agent samples a seed independently and constructs an MDP by using this seed and the mapping detailed in Section 3.2.2 or Section 3.2.3 respectively. The fact that each agent samples a seed independently leads, thanks to randomness, to MDPs with very different edge weights for the edges that have not been traversed yet. As a result, agents pursue diverse paths. At the same time, maintaining a constant seed ensures that each agent adjusts his beliefs in subsequent time periods in a manner that is consistent with the observations made by all agents and not driven by further randomness that would be distracting to the agent's exploration.

Consider the specification of the problem in which we sample Erdős-Rényi graphs with number of vertices $N = 100$ and edge probability $p = 2 \ln N/N$. The edge weights θ are independent and the common prior of the agents on the edge weights is $\ln \theta_e \sim \mathcal{N}(0, 4)$, $\forall e \in \mathcal{E}$. When edge e is traversed, the observed reward r_e is distributed according to $\ln r_e | \theta_e \sim \mathcal{N}(\ln \theta_e - 0.005, 0.01)$. The horizon (i.e. length of the maximum reward path to be found) is $H = 10$.

In Figure 5, we show the performance of all algorithms. The results are averaged over hundreds of simulations. Standard-Gaussian seed sampling and martingalean-Gaussian seed sampling achieve the lowest cumulative regret, as they adhere to all the properties of coordinated exploration. Concurrent UCRL follows with 49.9% higher cumulative regret than the seed sampling algorithms. Concurrent UCRL does not satisfy the diversity property, and incurs much larger cumulative regret than the seed sampling algorithms, but does better than Thompson resampling because, unlike the latter, it satisfies the commitment property which is essential in deep exploration. Thompson resampling has 191% higher cumulative regret than the seed sampling algorithms.

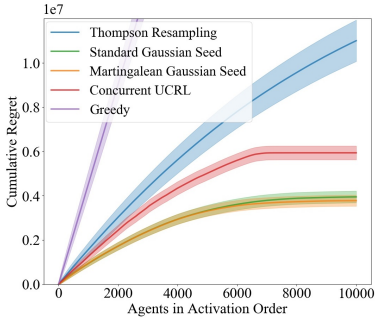


Figure 5: Performance of the algorithms of Section 3 and of the Greedy algorithm in the "Maximum Reward Path" example with $N = 100$ nodes, $p = 2 \ln N/N$ edge probability, $\ln \theta_e \sim \mathcal{N}(0, 4)$, $\forall e \in \mathcal{E}$, $H = 10$ horizon in terms of cumulative regret of 10,000 concurrent agents ordered in ascending activation time with 95% confidence interval.

5. Closing Remarks

Concurrent reinforcement learning is poised to play an important role across many applications, ranging from web services to autonomous vehicles to healthcare, where each agent is responsible for a user, vehicle or patient. To learn efficiently in such settings, agents should coordinate the exploratory effort. We presented three properties that are essential to efficient coordinated exploration: real-time *adaptivity* to shared observations, *commitment* to carry through with action sequences that reveal new information, and *diversity* across learning opportunities pursued by different agents. We demonstrated that optimism-based approaches fall short with respect to diversity, while naive extensions of Thompson sampling lack the requisite level of commitment. We proposed *seed sampling*, a novel extension of PSRL that does satisfy these properties. We presented several seed sampling schemes, customized for particular priors and likelihoods, but the seed sampling concept transcends these specific cases, offering a general approach to more broadly designing effective coordination algorithms for concurrent reinforcement learning. Much work remains to be done on this topic. For starters, it would be useful to develop a mathematical theory that sharpens understanding of the efficiency and robustness of seeding schemes. Beyond that, work is required to develop seeding schemes that operate in conjunction with practical scalable reinforcement learning algorithms that approximate optimal value functions and policies for problems with intractable state and action spaces.

Acknowledgments

The Ph.D. research of Maria Dimakopoulou at Stanford University is supported by the "Arvanitidis in Memory of William K. Linvill" Stanford Graduate Fellowship and by the Onassis Foundation.

References

Chalkiadakis, G. and Boutilier, C. Coordination in multiagent reinforcement learning: A Bayesian approach. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 709–716. ACM, 2003.

Gentle, J. E. *Computational statistics*. Springer Science & Business Media, 2009.

Gentle, J. E. *Random number generation and Monte Carlo methods*. Springer Science & Business Media, 2013.

Guo, Z. and Brunskill, E. Concurrent PAC RL. In *AAAI*, pp. 2624–2630, 2015.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Kearns, M. J. and Singh, S. P. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3): 209–232, 2002.

Kim, M. J. Thompson sampling for stochastic control: The finite parameter case. *IEEE Transactions on Automatic Control*, 62(12):6415–6422, 2017.

Lu, X. and Van Roy, B. Ensemble sampling. In *Advances in Neural Information Processing Systems*, 2017.

Mertikopoulos, P. and Zhou, Z. Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming*, 2018.

Osband, I. and Van Roy, B. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, pp. 1466–1474, 2014a.

Osband, I. and Van Roy, B. Near-optimal reinforcement learning in factored MDPs. In *Advances in Neural Information Processing Systems*, pp. 604–612, 2014b.

Osband, I. and Van Roy, B. On optimistic versus randomized exploration in reinforcement learning. In *The Multidisciplinary Conference on Reinforcement Learning and Decision Making*, 2017a.

Osband, I. and Van Roy, B. Why is posterior sampling better than optimism for reinforcement learning? In *International Conference on Machine Learning*, pp. 2701–2710, 2017b.

Osband, I., Russo, D., and Van Roy, B. (More) efficient reinforcement learning via posterior sampling. In *Advances In Neural Information Processing Systems*, pp. 3003–3011. 2013.

Pazis, J. and Parr, R. PAC optimal exploration in continuous space Markov decision processes. In *AAAI*. Citeseer, 2013.

Pazis, J. and Parr, R. Efficient PAC-optimal exploration in concurrent, continuous state mdps with delayed updates. In *AAAI*. Citeseer, 2016.

Russo, D., Van Roy, B., Kazerouni, A., Osband, I., and Wen, Z. A tutorial on Thompson sampling. *arXiv preprint arXiv:1707.02038*, 2017.

Silver, D., Newnham, L, B., Weller, S., and McFall, J. Concurrent reinforcement learning from customer interactions. In *International Conference on Machine Learning*, pp. 924–932, 2013.

Strens, M. J. A. A Bayesian framework for reinforcement learning. In *International Conference on Machine Learning*, pp. 943–950, 2000.

Vlassis, N. A concise introduction to multiagent systems and distributed artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, pp. 1–71, 2007.