

Hierarchical State Abstractions for Decision-Making Problems with Computational Constraints

Daniel T. Larsson, Daniel Braun, Panagiotis Tsiotras

Abstract—In this semi-tutorial paper, we first review the information-theoretic approach to account for the computational costs incurred during the search for optimal actions in a sequential decision-making problem. The traditional (MDP) framework ignores computational limitations while searching for optimal policies, essentially assuming that the acting agent is perfectly rational and aims for exact optimality. Using the free-energy, a variational principle is introduced that accounts not only for the value of a policy alone, but also considers the cost of finding this optimal policy. The solution of the variational equations arising from this formulation can be obtained using familiar Bellman-like value iterations from dynamic programming (DP) and the Blahut-Arimoto (BA) algorithm from rate distortion theory. Finally, we demonstrate the utility of the approach for generating hierarchies of state abstractions that can be used to best exploit the available computational resources. A numerical example showcases these concepts for a path-planning problem in a grid world environment.

I. INTRODUCTION

Markov Decision Processes (MDP) is the standard framework for optimal sequential decision making under uncertainty [1]. However, this framework assumes that the decision maker, customary referred to as the “agent,” is perfectly rational [2]. That is, the agent is assumed to have unlimited computational and/or time resources to process all information and compute the optimal solution unhindered. In many practical applications this is a strong and often unrealistic assumption. For example, biological systems are adaptive in nature and excel at finding solutions to complex problems that may not be globally optimal but are “good enough” given the constraints and/or available sensory information [2], [3].

Over the past several years considerable effort has been devoted to developing a framework that accounts for the available computational constraints. The idea is to account for the cost of computation not after the fact, as is traditionally the case, but from the beginning, as part of the problem formulation. This stream of research has led to the development of *information-constrained MDPs* [3]–[5] that utilize concepts from information theory in order to model

the computational limitation(s) of the agent [6], [7]. The approach also has strong connections with the thermodynamic point of view of the theory of computation [8]–[10].

The introduction of the added constraints (computational and otherwise) to the decision process is meant to model an agent that does not have access to unlimited resources, and hence will select actions that are not optimal in the traditional sense of maximizing value or utility, but rather in the sense of maximizing (expected) value or utility *for a given processing cost* [2], [4], [7]. Such a decision maker, who alters its behavior based on its resources, is said to be a *bounded rational* agent [2], [11]–[13]. In this work, we apply this framework to develop decision-making algorithms for path-planning problems while also accounting for the information processing costs of computing the optimal actions.

The information-limited MDP problem is formally posed as a constrained optimization problem, where the objective is to maximize the value function subject to a constraint that penalizes the difference between the resulting policy and a known prior (computationally cheap) policy. This constraint is often represented using the Kullback-Leibler (KL) divergence [3], [4], [7]. The unconstrained optimization problem obtained by introducing Lagrange multipliers likens that of the free-energy function from thermodynamics [10] and is hence called the “free energy” of the information limited decision process [4], [7], [10], [12]. In this formulation the goal is to find a policy that maximizes the free energy at each step of the process. The solution to this problem can be found numerically through value-iteration type of algorithms or backward recursion [4], [7], [14] and follows closely the process for finding the optimal value function using dynamic programming (DP) [1], [14], [15].

The framework can be augmented so that, across *all states on average*, the transformation (information) cost of “molding” this prior into the optimal policy is reduced [3], [4]. The resulting optimization problem is analogous to that of the Blahut-Arimoto algorithm used extensively in rate distortion theory [4], [16].

The concept of mutual information between states and actions is important and is a topic that does not seem to have been exploited extensively in the controls community. Most importantly, it leads to a direct analogy of the amount of information an action carries about the state. For example, a perfectly rational decision maker will solve the optimization problem producing a policy that is *state specific*. This means knowing what action was selected is very informative regarding what state the agent was in when

The work of DL and PT has been supported by ONR award N00014-13-1-0563 and of DB by project ERC STG BRISC 678082.

D. Larsson is a PhD student with the D. Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA. Email: daniel.larsson@gatech.edu

D. Braun is a Professor with the Institute of Neural Information Processing, Ulm University, D-89081 Ulm, Germany. Email: daniel.braun@uni-ulm.de

P. Tsiotras is a Professor with the D. Guggenheim School of Aerospace Engineering and the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA. Email: tsiotras@gatech.edu

action was taken [3]. However, a bounded rational agent must consider the resources required to find the optimal action and hence finds policies that are relatively generic and not state-specific. This classification of more informative versus less informative states leads in a natural way to a series of state space *abstractions*, which are discussed in greater detail in this paper. Such abstractions can be utilized in order to focus the available computational resources in a top-down, task-specific manner.

The use of hierarchical abstractions has been explored to find efficient solutions to path-planning problems [17]–[23] subject to computational constraints. In those references, hierarchies of multi-resolution abstractions were created to speed-up replanning. The difference with the current approach is that while in those references these abstractions were imposed externally (and perhaps rather axiomatically) by the designer, here they arise intrinsically, as the result of the limited available computational resources of the agent.

The scope of this paper is twofold. First, we provide an overview of the existing framework of the variational free-energy principle that allows the construction of state abstractions, which can then be utilized to solve sequential decision making problems subject to computational constraints. Second, we propose a novel modification to the Blahut-Arimoto algorithm, allowing us to handle a closed-loop optimization problem with the existing framework. Finally, the proposed approach is demonstrated on a typical path-planning problem in a grid world. This is the first work, as far as the authors know, where the information-theoretic bounded rationality framework is used for multi-resolution hierarchical abstraction in sequential decision problems.

II. PROBLEM FORMULATION

A. Preliminaries of Standard Markov Decision Process

We consider sequential decision-making problems in the presence of uncertainty. Markov Decision Processes (MDPs) is the standard mathematical framework to solve such problems in discrete time. We will use x_t to denote the system state and u_t the control input at time t , where $t = 0, 1, 2, \dots$. The system state is an element of the state set X ($x_t \in X$) and the control input is selected from the set of admissible inputs, $u_t \in U(x_t)$. The system dynamics are modeled via a collection of transition probabilities $P(x_{t+1}|x_t, u_t)$, which represent the probability that the agent, being at state x_t and selecting input u_t at time step t , will find itself at state x_{t+1} at the next instance of time. Each action results in a reward $R(x_{t+1}, x_t, u_t)$ which the agent receives for selecting input u_t while at state x_t and at time t and transitioning to state x_{t+1} at time $t + 1$.

The objective of the infinite-time MDP is to find a policy which maximizes the future discounted expected reward. By policy we mean a map that for each state x_t provides a probabilistic distribution over actions, $\pi(u_t|x_t)$. Since this paper is limited to infinite horizon problems, we assume that our policy is stationary, that is, π is independent of t .

More formally, we seek to maximize the objective function

$$V^\pi(x_0) \triangleq \lim_{T \rightarrow \infty} \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \gamma^t R(x_{t+1}, x_t, u_t) \right], \quad (1)$$

where $\gamma \in [0, 1)$ is the *discount factor*, x_0 is the initial state and u_t is selected according to the provided policy π at each time step. Equation (1) can be interpreted as the value-to-go provided the system starts at state x_0 and executes the stationary policy π . The expectation in equation (1) is over all future trajectories $\tau = (x_0, u_0, x_1, u_1, \dots, x_T)$ of length T starting at x_0 , which under the Markov property assumption is given as

$$\Pr(u_0, x_1, u_1, x_2, \dots, x_T | x_0) = \prod_{t=0}^{T-1} \pi(u_t | x_t) P(x_{t+1} | x_t, u_t). \quad (2)$$

The objective is then to find the optimal distribution over admissible inputs at each state, in order to maximize the objective function in (1). Introducing the shorthand notation $x = x_t$, $u = u_t$, and $x' = x_{t+1}$, expanding (1) and making use of (2), we obtain

$$V^\pi(x) = \sum_{u \in U(x)} \pi(u|x) \sum_{x' \in X} P(x'|x, u) \times [R(x', x, u) + \gamma V^\pi(x')]. \quad (3)$$

We then write the optimal value function as [7]

$$V^*(x) = \max_{\pi} V^\pi(x), \quad (4)$$

which can be equivalently written as

$$V^*(x) = \max_{u \in U(x)} \sum_{x' \in X} P(x'|x, u) [R(x', x, u) + \gamma V^*(x')], \quad (5)$$

with the optimal policy given by $\pi^* = \operatorname{argmax}_{\pi} V^\pi(x)$.

We also define the state-action value function for any policy π as [1]

$$Q^\pi(x, u) \triangleq \sum_{x' \in X} P(x'|x, u) [R(x', x, u) + \gamma V^\pi(x')]. \quad (6)$$

Similarly, the optimal state-action value function is defined as

$$Q^*(x, u) \triangleq \sum_{x' \in X} P(x'|x, u) [R(x', x, u) + \gamma V^*(x')]. \quad (7)$$

In the case of a perfectly rational decision maker, the agent will deterministically select the optimal action that maximizes equation (7). That is, the agent will act greedily with respect to $Q^*(x, u)$. In this paper, we distinguish between a general stochastic policy and one that is deterministic by denoting optimal deterministic policies as $\Gamma^*(u|x)$ and all other optimal policies as $\pi^*(u|x)$. Then, a perfectly rational agent will choose the policy $\Gamma^*(u|x) = \delta(u - u^*(x))$, where $u^*(x) = \operatorname{argmax}_u Q^*(x, u)$.

B. The Information-Limited Markov Decision Process

The problem formulation presented above does not consider the “effort” required to find the optimal policy $\Gamma^*(u|x)$, which maximizes (1) [4], [7]. In order to model the effort required to obtain the (global) maximum, we provide the agent with a prior choice distribution ($\rho(u|x)$) and then limit the amount by which the posterior policy ($\pi(u|x)$) is permitted to differ from $\rho(u|x)$ [3], [4], [7]. The difference between the two distributions is measured using the Kullback-Leibler (KL) divergence [24]

$$D_{\text{KL}}(\pi(u|x)||\rho(u|x)) \triangleq \sum_{u \in U(x)} \pi(u|x) \log \frac{\pi(u|x)}{\rho(u|x)}. \quad (8)$$

The total discounted information cost for a decision maker starting from any initial state x and following the policy $\pi(u|x)$ is then defined as [7]

$$D^\pi(x) \triangleq \lim_{T \rightarrow \infty} \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \gamma^t \log \frac{\pi(u_t|x_t)}{\rho(u_t|x_t)} \right], \quad (9)$$

where the expectation in equation (9) is with respect to the resulting trajectory, given by equation (2). The goal is now to not only maximize the value alone, but maximize the trade-off between the value and information cost [3], [4], [7], [12]. This is a constrained optimization problem, which can be solved using Lagrange multipliers. The free energy for a given policy $\pi(u|x)$ is defined as

$$F^\pi(x; \beta) \triangleq V^\pi(x) - \frac{1}{\beta} D^\pi(x), \quad (10)$$

where $\beta > 0$. The objective is now to find the policy $\pi^*(u|x)$ which maximizes equation (10). Substituting known quantities and expanding terms, we obtain the mapping $\mathcal{B} : \mathbb{R}^{|X|} \rightarrow \mathbb{R}^{|X|}$ defined as

$$\begin{aligned} \mathcal{B}[F](x; \beta) \triangleq & \max_{\pi(u|x)} \sum_{u \in U(x)} \pi(u|x) \times \\ & \left[\sum_{x' \in X} P(x'|x, u) R(x', x, u) - \frac{1}{\beta} \log \frac{\pi(u|x)}{\rho(u|x)} \right. \\ & \left. + \gamma \sum_{x' \in X} P(x'|x, u) F(x'; \beta) \right]. \end{aligned} \quad (11)$$

The proof of how this mapping results from equation (10) can be found in [7]. The optimal value of the free energy $F^*(x; \beta)$ is the fixed point of the equation $F^* = \mathcal{B}[F^*]$. Similarly to (6), we define

$$Q_F(x, u; \beta) \triangleq \sum_{x' \in X} P(x'|x, u) \left[R(x', x, u) + \gamma F(x'; \beta) \right], \quad (12)$$

and $Z(x; \beta) \triangleq \sum_{u \in U(x)} \rho(u|x) e^{\beta Q_F(x, u; \beta)}$. As in [7], [10], equation (11) can be shown to be equivalent to

$$\mathcal{B}[F](x; \beta) = \frac{1}{\beta} \log Z(x; \beta). \quad (13)$$

This is iteratively applied until convergence to obtain the optimal free energy $F^*(x; \beta)$. For a given value of β the

optimal policy for the infinite horizon problem is then

$$\pi^*(u|x) = \frac{\rho(u|x) e^{\beta Q_F^*(x, u; \beta)}}{Z^*(x; \beta)}, \quad (14)$$

where $Q_F^*(x, u; \beta)$ and $Z^*(x; \beta)$ are given by

$$\begin{aligned} Q_F^*(x, u; \beta) &= \sum_{x' \in X} P(x'|x, u) \left[R(x', x, u) + \gamma F^*(x'; \beta) \right], \\ Z^*(x; \beta) &= \sum_{u \in U(x)} \rho(u|x) e^{\beta Q_F^*(x, u; \beta)}. \end{aligned} \quad (15)$$

Using this framework we are able to model a wide variety of agents with limited resources by varying the resource parameter, β . Hence, β can be viewed as a parameter which reflects the agents computational abilities [5]. We now discuss the limiting cases of the resource parameter, β .

Case I ($\beta \rightarrow 0$): In this case the information term in equations (10) and (11) becomes dominant and the process becomes mainly concerned with reducing the information cost $D^\pi(x)$. Note that the KL divergence is always nonnegative-definite [24] and hence obtains its minimum at zero. This leads to $\pi^* = \rho$. That is, the prior distribution is directly “copied” to the posterior policy and represents an agent that is not able to deviate from its initial notion of an optimal action, indicative of a decision maker with no computational resources [5].

Case II ($\beta \rightarrow \infty$): In the limit as $\beta \rightarrow \infty$ the information cost in equations (10) and (11) is effectively removed. The process will focus on finding a policy that maximizes $V^\pi(x)$. This means that the agent will behave more like its non-information limited counterpart. We therefore expect to recover the non-information limited policy in the limit (i.e., $\pi^* \rightarrow \Gamma^*$), which is indicative of an agent that has unlimited computational resources [5].

III. THE INFORMATION LIMITED MDP AND THE MODIFIED BLAHUT-ARIMOTO ALGORITHM

In this section we address the question of whether we can find an optimal a priori action distribution $\rho(u|x)$ so as to minimize the information cost across all states *on average*. Note that the information limited MDP does not modify $\rho(u|x)$ when searching for the optimal policy π^* . Instead, it will find the policy that best trades the value and information for a given $\rho(u|x)$. The problem now becomes one of finding both the posterior policy and prior action distribution so as to maximize equation (10) on average across all system states.

Averaging equation (10) over system states and assuming a state-independent prior distribution over actions ($\rho(u|x) = \rho(u)$), we obtain a relation analogous with that of the rate distortion function from information theory [3], [4], [7], [16]. The relation between equation (16) and that of the rate distortion function in [16] is quite intimate, as the averaging over system states of (10) gives rise to the direct appearance of the mutual information between actions and states [3], [24]. Optimizing the averaged free energy over both π and ρ we arrive at the Blahut-Arimoto (BA) algorithm. Formally,

the problem can be stated as

$$\max_{\rho} \sum_{x \in X} p(x) \times \left(\max_{\pi} \sum_{u \in U(x)} \pi(u|x) \left[Q_F(x, u; \beta) - \frac{1}{\beta} \log \frac{\pi(u|x)}{\rho(u)} \right] \right). \quad (16)$$

Optimizing equation (16) with respect to $\rho(u)$ is a convex optimization problem [3], [24], and is numerically solved by recursively iterating between the following relations until convergence [3], [4], [16]

$$\pi^*(u|x) = \frac{\rho^*(u) e^{\beta Q_F^*(x, u; \beta)}}{\sum_u \rho^*(u) e^{\beta Q_F^*(x, u; \beta)}}, \quad (17)$$

$$\rho^*(u) = \sum_x p(x) \pi^*(u|x). \quad (18)$$

Here $p(x)$ is the (fixed) probability distribution over states and π^* is found in the same manner as described in the previous section. The proof can be found in [3], [4].

Similarly to the previous section, an agent utilizing the BA algorithm for path planning will find a policy that deviates in various amounts to the provided prior action distribution depending on the value of β . Now, however, the prior action distribution is *not* static and is updated each iteration to be consistent with the posterior policy, π (see equations (17)-(18)). Note that at low β the agent becomes mainly concerned with minimizing the mutual information between states and actions, as given by $\sum_{x \in X} p(x) D^\pi(x)$ in (16). This results in a policy that is generic with respect to the state, leading to state abstractions as shown in [3]. That is, the agent will focus its efforts to find a single action that yields a high reward on average across the state space.

This is no issue for agents that have access to all actions in every state or that have enough resources (high β) to sufficiently alter ρ into π . However, for agents that are resource limited and have admissible actions spaces that are state dependent, this behavior may lead to a posterior policy that assigns non-zero probability to actions that are not available in a certain states. To remedy this issue, we propose an extension to the BA algorithm by altering the update equations in (17)-(18) as follows

$$\begin{aligned} \pi^*(u|x) &= \frac{\rho^*(u|x) e^{\beta Q_F^*(x, u; \beta)}}{\sum_u \rho^*(u|x) e^{\beta Q_F^*(x, u; \beta)}}, \\ \hat{\pi}^*(u) &= \sum_x p(x) \pi^*(u|x), \\ \rho^*(u|x) &= \frac{\hat{\pi}^*(u) e^{Q_p(x, u)}}{\sum_u \hat{\pi}^*(u) e^{Q_p(x, u)}}. \end{aligned} \quad (19)$$

By introducing the penalty function $Q_p(x, u)$, we ensure that the agent may only deliberate among available actions in each state while forming its posterior policy π^* . The penalty function can then be constructed so that $Q_p(x, u) \rightarrow -\infty$ for actions that are inadmissible in a given state. In the numerical example that follows, we use $Q_p(x, u) = \pm 100$ for actions that are available/unavailable, respectively.

IV. NUMERICAL EXAMPLE

We demonstrate the previous algorithms by applying them to a path-planning problem in a grid world. Consider an agent navigating over a 4×4 square (the world environment) that is full of obstacles, as shown in Fig. 1. The goal is to safely navigate from the start state (cell 1) to the goal state (cell 16) while avoiding the obstacles. At each time step $t = 0, 1, \dots$ the agent can choose its actions from the set $U(x_t) \subseteq \{\text{UP, DN, L, R, A-UP, A-DN}\}$. The actions A-UP and A-DN do not correspond to physical movement, but rather encode changes in the ‘‘perceptual representation’’ of the environment that are motivated by the sensing and/or computational resources of the agent. Note that not all actions are available at every state, as shown in Fig. 3.

The space the agent operates in (and reasons about) is an augmented grid world with two abstraction levels, as shown in Fig. 1. Specifically, the world states 1,2,5,6 are all abstracted by the additional new (perceptual) state 17, states 3,4,7,8 are all abstracted by the new state 18 and so on, as shown in Figs. 1 and 2. Similarly, perceptual states 17,18,19,20 can all be abstracted, in turn, by the new (root) state 21 (X). This situation is similar to a standard quad-tree decomposition for this grid world scenario, although one can envision different methods to generate the corresponding abstractions. Here we do not allow the agent to abstract to the root state, and hence the final search space includes 20 states.

For each state $x \in \{1, 2, \dots, 20\}$ the set of available actions $U(x) \subseteq \{\text{UP, DN, L, R, A-UP, A-DN}\}$ includes only actions that will not *intentionally* move the agent into an obstacle (see Fig. 3). Since the transitions are generally probabilistic, there is a possibility that the agent may end up in *any* neighboring state. Thus, actions have a success probability of 80%, with the remaining 20% distributed among the neighboring states. For example, selecting the action UP in state 1 will with 80% probability move the agent to state 5, while the remaining 20% is distributed over states 2 and 6 using this action. The exception to this rule are the special actions A-UP and A-DN. These special actions are deterministic and can be viewed as the agent’s ability to toggle a sensor suite consisting of two variable resolution sensors on and off. The agent must pay an increased price for navigating the grid world at the finest resolution level, due to the demand to process an increased amount of information. The actions A-UP and A-DN can thus also be viewed as actions that are taken when the agent has limited computational resources, in which case it will favor higher level (coarser) abstractions since these correspond to search spaces of lower cardinality and hence are presumably easier to navigate in.

In terms of transitions for special actions A-UP and A-DN, we assume that if an agent abstracts up, it will find itself in the corresponding aggregate state at the next epoch. Conversely, if an agent decides to abstract down, it will find itself in one of the underlying states with equal probability at the next time step. For example, executing A-UP from state

1 will deterministically move the agent to state 17, while executing A-DN from state 17 will move the agent to either state 1, 2, 5, or 6 with equal probability.

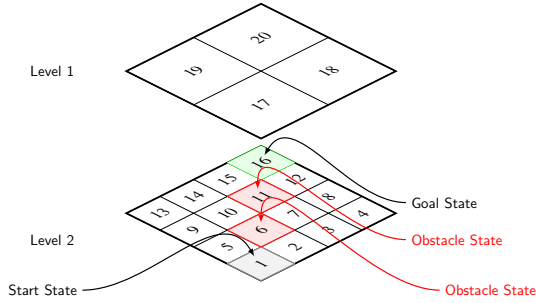


Fig. 1. Grid layering with state numbering. Start state indicated in gray, obstacle states in red with goal state in green.

The rewards function is constructed as follows. The agent pays a cost of -0.55 for every action corresponding to UP, DN, L, R within level 2, and -0.15 for these actions in level 1. To abstract up, the agent must pay a cost of approximately -2 , and -1 to abstract down. It receives $+2$ for reaching the goal and -10 for navigating into an obstacle. The discount factor is 0.95 . It should be noted that goal and obstacles are *absorbing* states, hence navigation into obstacles is catastrophic. Also, since goal state 16 is accessible from state 20, we assume that the reward the agent receives for this transition is $R_{\text{abstractDN}} + R_{\text{goal}}$ to correspond to both the price of switching to the higher resolution sensor and finding itself at the goal state.

In the discussion that follows all quantities that refer to x_0 are the values of the variable at state 1. For example, $V^\pi(x_0)$ is the expected value of starting from state 1 and following the given policy $\pi(u|x)$. As a measure to compare the performance of the other algorithms and to better understand

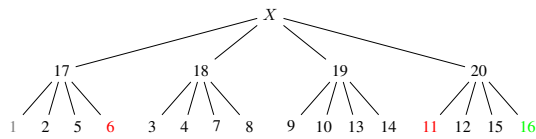


Fig. 2. Hierarchical description of states with coloring consistent with description in Fig. 1.

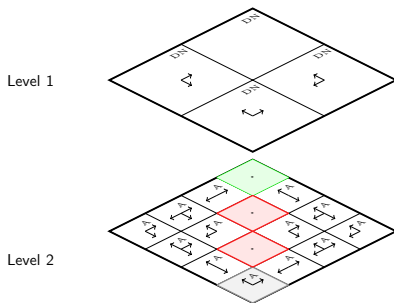


Fig. 3. Visualization of the available actions at each state. Actions “A” and “DN” represent abstract up and abstract down, respectively. Arrows indicate movements within a given level.

how the resource parameter alters the agent’s behavior, we first present results for the deterministic case ($\beta \rightarrow \infty$). The deterministic policy obtained from solving the standard MDP problem is shown in Fig. 4a and has an associated value of $V^\pi(x_0) = -4.22$. It is important to note that the policy depicted in Fig. 4a by Γ^* represents a rational agent. That is, the policy Γ^* would be followed by an agent who has sufficient processing capabilities to find the optimal actions and does so on the finest grid level, electing to not abstract to a coarser resolution for path planning. It will be seen that, as the agent’s computational abilities are reduced, it begins to favor planning at the coarser grid level while its resulting policy becomes increasingly suboptimal with respect to Γ^* .

Shown in Fig. 4b is the prior action distribution, $\rho(u|x)$. The prior distribution, ρ , can be viewed as the policy the agent would follow in case it had no resources to find another, possibly better, policy [7]. Thus, if the agent has insufficient resources to deliberate (limit as $\beta \rightarrow 0$), ρ becomes the posterior policy π , as seen when comparing Figs. 4b and 6b.

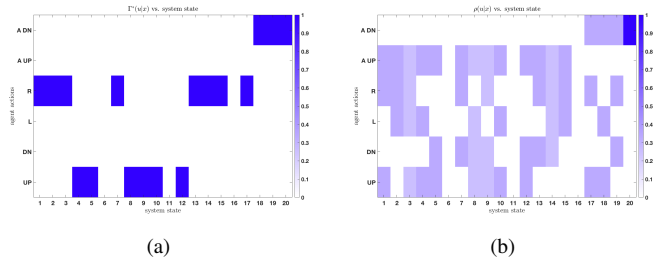


Fig. 4. Deterministic policy, $\Gamma^*(u|x)$ (left) and a priori action distribution, $\rho(u|x)$ (right).

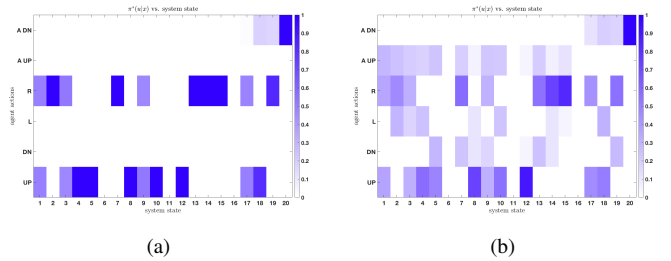


Fig. 5. $\pi^*(u|x)$ from the BA-IL algorithm (left) and $\pi^*(u|x)$ from the IL algorithm (right) for $\beta = 0.5$.

Figures 4-6 display the policies obtained for various values of β for both the information-limited (IL) and the Blahut-

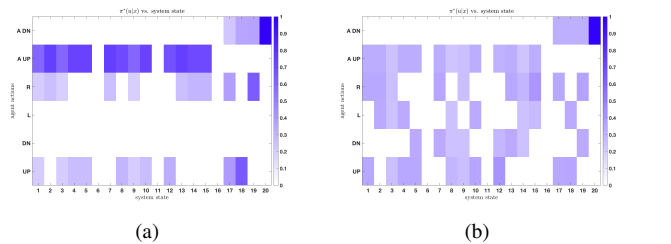


Fig. 6. $\pi^*(u|x)$ from the BA-IL algorithm (left) and $\pi^*(u|x)$ from the IL algorithm (right) for $\beta = 0.05$.

TABLE I
VALUE OF POLICY π^* FOR VARIOUS VALUES OF β .

$V^\pi(x_0)$ IL-BA	$V^\pi(x_0)$ IL	β
-10.5	-11	0.05
-4.33	-7.39	0.5
-4.27	-4.35	5
-4.22	-4.22	100

Arimoto information limited (BA-IL) algorithms. More precisely, the figures show the distribution over actions for a given state and value of β (and the distribution over states, $p(x)$, in the case of the BA algorithm). In each of these figures, the y-axis corresponds to the action space whereas the x-axis displays the possible system states, with state numbering consistent with Fig. 1. Furthermore, the shading of a given state-action pair represents the probability that the action will be selected when the system finds itself in that state. All plots are configured so that the darker shading represents a higher probability of an action being selected when following the policy, with no shading representing zero probability. Column n of these plots can therefore be considered a probability distribution over actions for the n^{th} state, where $n \in \{1, 2, \dots, 20\}$.

Table I shows the corresponding values of following policies obtained from the algorithms for a variety of β . Note that as $\beta \rightarrow \infty$ the value of the policy π^* approaches that of Γ^* , although does so non-linearly [7]. This is also observed in Figs. 5-6 as the resulting policies (π^*) approach Γ^* as β is increased. We also note that the policy becomes less deterministic as $\beta \rightarrow 0$. Furthermore, as β is reduced, abstractions emerge from the solution of the BA algorithm. This can be seen in Fig. 6a since, when compared to the non-BA policy in Fig. 6b, we see that the agent elects to select the same action in many states. That is, the resulting optimal policy π^* from the BA algorithm is relatively *generic* with respect to the state and thus the state and action spaces have low mutual information. Because of the additional step in optimizing the prior distribution in the BA algorithm, a resource restricted agent following a BA optimal policy achieves higher value for a given β when compared to the same agent following an optimal policy provided by the information-limited algorithm, as seen in Table I. Finally, it should be noted from Fig. 6 that as the information parameter is reduced, the agent begins to path plan at the coarser grid level, indicative that less computationally complex solutions are sought at the cost of complete optimality.

V. CONCLUSION

We have presented an overview of an information-theoretic approach to account for computational resources in sequential decision making problems modeled by MDPs. It is shown how the limited information processing costs give rise to state hierarchical abstractions, which can then be used to solve the decision problem at the best state representation “granularity” level. We have demonstrated how the approach can be applied to a typical path-planning problem. It is shown that the resulting optimal policy changes significantly as a

function of the processing constraints. In the process, we proposed a simple extension to the classical Blahut-Arimoto algorithm from rate distortion theory that is useful in many engineering applications.

REFERENCES

- [1] R. S. Sutton and A. G. Barto, *Reinforcement Learning*. MIT Press, 1998.
- [2] B. L. Lipman, *Information Processing and Bounded Rationality: a Survey*. Wiley on behalf of the Canadian Economics Association, Feb 1995, vol. 28, no. 1, pp. 42–67.
- [3] T. Genewein, F. Leibfried, J. Grau-Moya, and D. A. Braun, “Bounded rationality, abstraction, and hierarchical decision-making: An information-theoretic optimality principle,” *Frontiers in Robotics and AI*, vol. 2, 2015.
- [4] N. Tishby and D. Polani, “Information theory of decisions and actions,” in *Perception-Action Cycle: Model, Architectures and Hardware*, ser. Springer Series in Cognitive and Neural Systems 1. Springer Science & Business Media, 2011, ch. 19, pp. 601 – 636.
- [5] J. Grau-Moya, F. Leibfried, T. Genewein, and D. A. Braun, “Planning with information-processing constraints and model uncertainty in Markov decision processes,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 475–491.
- [6] P. A. Ortega and D. A. Braun, *Information, Utility and Bounded Rationality*. Springer-Verlag Berlin Heidelberg, 2011, pp. 269–274.
- [7] J. Rubin, O. Shamir, and N. Tishby, “Trading value and information in MDPs,” in *Decision Making with Imperfect Decision Makers*. Springer-Verlag Heidelberg, 2012, ch. 3, pp. 57 – 74.
- [8] C. H. Bennett, “The thermodynamics of computation - a review,” *International Journal of Theoretical Physics*, vol. 21, no. 12, pp. 905–940, 1982.
- [9] R. P. Feynman, *Feynman Lectures on Computation*. Addison-Wesley Longman Publishing Co., Inc., 1998.
- [10] P. A. Ortega and D. A. Braun, “Thermodynamics as a theory of decision-making with information-processing costs,” in *Proceedings of the Royal Society*. Royal Society, 2013.
- [11] H. A. Simon, *Models of Bounded Rationality*. MIT Press, 1982, vol. 3.
- [12] D. A. Braun, P. A. Ortega, E. Theodorou, and S. Schaal, “Path integral control and bounded rationality,” *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2011.
- [13] E. Horvitz and S. Zilberstein, “Computational tradeoffs under bounded resources,” *Artificial Intelligence*, vol. 126, no. 1, pp. 1–4, 2001.
- [14] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Athena Scientific, 2007, vol. 2.
- [15] R. Bellman, “A Markovian decision process,” *Journal of Mathematics and Mechanics*, vol. 6, no. 5, pp. 679–684, 1957.
- [16] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” in *37th Annual Allerton Conference on Communication, Control and Computing*, 1999.
- [17] R. Cowlagi and P. Tsiotras, “Multiresolution path planning with wavelets: A local replanning approach,” in *American Control Conference*, Seattle, WA, June 11-13, 2008, pp. 1220–1225.
- [18] —, “Multi-resolution path planning: Theoretical analysis, efficient implementation, and extensions to dynamic environments,” in *49th IEEE Conference on Decision and Control*, Atlanta, GA, Dec. 15–17, 2010, pp. 1384–1390.
- [19] —, “Multiresolution path- and motion planning for autonomous agents via wavelet-based cell decompositions,” *IEEE Transactions on Systems Man and Cybernetics, Part B: Cybernetics*, vol. 42, no. 5, pp. 1455–1469, October 2012.
- [20] P. Tsiotras, D. Jung, and E. Bakolas, “Multiresolution hierarchical path-planning for small UAVs using wavelet decompositions,” *Journal of Intelligent and Robotic Systems*, vol. 66, no. 4, pp. 505–522, June 2012.
- [21] R. Cowlagi and P. Tsiotras, “Hierarchical motion planning with kinodynamic feasibility guarantees,” *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 379–395, April 2012.
- [22] S. Behnke, *Local Multiresolution Path Planning*, ser. Lecture Notes in Computer Science. Berlin: Springer, 2004, vol. 3020, pp. 332–343.
- [23] S. Kambhampati and L. S. Davis, “Multiresolution path planning for mobile robots planning for mobile robots,” *IEEE Journal of Robotics and Automation*, pp. 135–145, 1986.
- [24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, 2006.