Monte Carlo Methods for Exact & Efficient Solution of the Generalized Optimality Equations

Pedro A. Ortega[1], Daniel A. Braun[2] and Naftali Tishby[3]

***Abstract*— Previous work has shown that classical sequential decision making rules, including expectimax and minimax, are limit cases of a more general class of bounded rational planning problems that trade off the value and the complexity of the solution, as measured by its information divergence from a given reference. This allows modeling a range of novel planning problems having varying degrees of control due to resource constraints, risk-sensitivity, trust and model uncertainty. However, so far it has been unclear in what sense information constraints relate to the complexity of planning. In this paper, we introduce Monte Carlo methods to solve the generalized optimality equations in an efficient & exact way when the inverse temperatures in a generalized decision tree are of the same sign. These methods highlight a fundamental relation between inverse temperatures and the number of Monte Carlo proposals. In particular, it is seen that the number of proposals is essentially independent of the size of the decision tree.**

I. INTRODUCTION

Decision trees, also known as game trees, are an essential tool in decision theory, operations research, artificial intelligence and robotics for representing probabilistic planning problems [1], [2]. In particular, decision trees are at the heart of adaptive control, reinforcement learning, path planning, experimental design, active learning and games. In robotics, decision trees have been applied, for example, to solve problems of navigation, sensory classification, knowledge sharing and linguistic planning [3], [4], [5], [6]. Interestingly, the decision rule depends on the kind of system the agent is interacting with. So, for instance, if the agent is controlling a *stochastic, neutral* system, then it has to apply the *Expectimax* rule [7]; if it is competing against an *adversarial* system, then it has to apply the *Minimax* rule; and if it is controlling a hybrid system containing both adversarial and stochastic responses, it has to use the *Expectiminimax* rule (Figure I). Once the correct decision tree is formulated, the optimal control command is calculated using dynamic programming [8]: starting from the leaves, values are recursively aggregated using either the maximum, expectation or minimum operators.

In [9], the aforementioned decision trees have been shown to be limit cases of a more general class based on the free energy framework for bounded rational planning [10]. This generalization is based on the observation that the *free energy* between two information states can instantiate a family of aggretation operators that includes the maximum, the expectation and the minimum operators as special cases, alongside bounded-rational operators that encapsulate *information limitations* in the control process *due to* resource constraints, risk-sensitivity, trust and model uncertainty. These generalized decision trees extend the work pioneered by Kappen [11], [12], Todorov [13], [14], Ortega & Braun [15] and Tishby & Polani [16] by allowing decision trees to mix different operators.

The contribution of this paper is to show how to *exactly* solve generalized decision trees using Monte Carlo methods *without* visiting all the leaves of the tree. This result is based on the fact that one can obtain optimal actions without having to explicitly calculate the optimal distribution by identifying the sampling processes implicitly defined in the optimality equations. This is of fundamental importance because it opens up the possibility of obtaining *exact and efficient* solutions to a whole new range of control problems that have never been tackled before.

This paper is structured as follows. In Section II we provide the preliminaries to understand general decision trees. Section III is the core contribution of this paper, namely a rejection sampling and a Metropolis-Hastings method for solving generalized decision trees. Simulations and experimental results are presented in Section IV. The final section discusses the methods and ends with concluding remarks.

II. PRELIMINARIES TO BOUNDED RATIONAL PLANNING

A. One-Step Decisions

In [15], [17], [10] it was shown that a bounded rational planning problem can be formalized based on the *free energy* between two information states. Formally, the *planning problem* is modeled as a tuple $(\mathcal{X}, \alpha, Q, U)$, where: $\mathcal{X}$ is the set of possible *outcomes* or realizations; $\alpha \in \mathbb{R}$ is a parameter called the *inverse temperature*; Q is a prior probability distribution over $\mathcal{X}$ representing a *prior policy* (also known as uncontrolled dynamics); and $U : \mathcal{X} \to \mathbb{R}$ is a real-valued mapping of outcomes called the *utility function*. The solution of the problem is given by a posterior probability P over the

*This study was funded by the Emmy Noether Grant BR 4164/1-1, the Israeli Science Foundation center of excellence, the DARPA MSEE project and the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI) and by a grant from the U.S. Department of Transportation Research Innovative Technology Administration.

[1]Pedro A. Ortega is a Postdoctoral Research Fellow at the GRASP Robotics Lab, University of Pennsylvania, Philadelphia, USA ope@seas.upenn.edu

[2]Daniel A. Braun is group leader at the Max Planck Institute for Intelligent System and Biological Cybernetics, Tübingen, Germany. daniel.braun@tuebingen.mpg.de

[3]Naftali Tishby is the director of the Interdisciplinary Center for Neural Computation (ICNC) and a professor at the School of Engineering and Computer Science at the Hebrew University of Jerusalem, Israel. tishby@cd.huji.ac.il

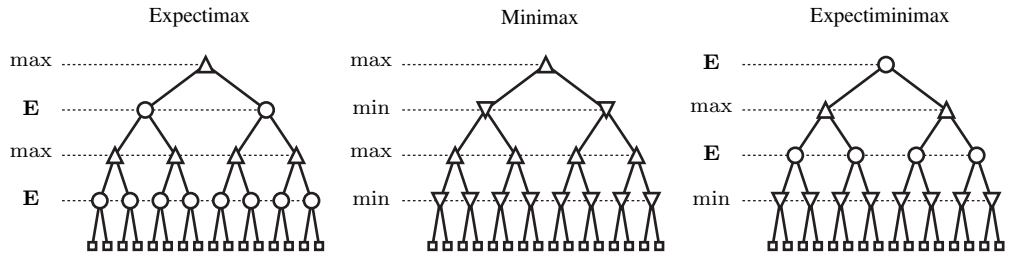

Fig. 1. Illustration of Expectimax, Minimax and Expectiminimax in decision trees representing three different interaction scenarios. The internal nodes can be of three possible types: maximum ($\triangle$), minimum ($\triangledown$) and expectation ($\circ$). The optimal decision is calculated recursively using dynamic programming.

outcomes $\mathcal{X}$ that optimizes the free energy functional

$$F_\alpha[\tilde{P}] := \underbrace{\sum_x \tilde{P}(x)U(x)}_{\text{Expected Utility}} - \underbrace{\frac{1}{\alpha}\sum_x \tilde{P}(x)\log\frac{\tilde{P}(x)}{Q(x)}}_{\text{Information Costs}}. \quad (1)$$

The inverse temperature $\alpha \in \mathbb{R}$ parameterizes the agent's amount of control or degree of influence over the outcome $x \in \mathcal{X}$: $\alpha > 0$ means that this influence is favorable; $\alpha = 0$ means no influence at all; and $\alpha < 0$ means that the influence is adverse. The optimal solution $\tilde{P} = P$, known as the *equilibrium distribution*, is given by

$$P(x) = \frac{1}{Z}Q(x)e^{\alpha U(x)}, \quad \text{where} \quad Z = \sum_x Q(x)e^{\alpha U(x)}. \quad (2)$$

The normalizing constant Z is known as the *partition function*. The inspection of (1) reveals that the free energy encapsulates a fundamental decision-theoretic trade-off: it corresponds to the expected utility, regularized by the additional information cost of representing the final distribution P using the base distribution Q. The inverse temperature plays the role of the conversion factor between units of information and units of utility. This planning scheme is of particular appeal from a Bayesian point of view, as the posterior policy can be thought of as arising from a belief update that treats utilities as evidence towards the alternatives with a precision given by the inverse temperature.

Inserting (2) into (1) yields the *certainty-equivalent* of the planning problem

$$F_\alpha[P] = \frac{1}{\alpha}\log Z_\alpha = \frac{1}{\alpha}\log\left(\sum_x Q(x)e^{\alpha U(x)}\right), \quad (3)$$

which represents how much the stochastic outcome is worth to the agent. Obviously, the more the agent is in control, the more valuable the outcome. This is seen as follows: for different choices of α, the value and the equilibrium distribution take the following limits,

$$\begin{aligned}
&\alpha \to +\infty \quad \tfrac{1}{\alpha}\log Z_\alpha = \max_x U(x) \qquad P(x) = \mathcal{U}_{\max}(x)\\
&\alpha \to 0 \qquad \tfrac{1}{\alpha}\log Z_\alpha = \sum_x Q(x)U(x) \quad P(x) = Q(x)\\
&\alpha \to -\infty \quad \tfrac{1}{\alpha}\log Z_\alpha = \min_x U(x) \qquad P(x) = \mathcal{U}_{\min}(x),
\end{aligned}$$

where $\mathcal{U}_{\max}$ and $\mathcal{U}_{\min}$ are the uniform distribution over the maximizing and minimizing subsets

$$\begin{aligned}
\mathcal{X}_{\max} &:= \{x \in \mathcal{X} : U(x) = \max_{x'} U(x')\}\\
\mathcal{X}_{\min} &:= \{x \in \mathcal{X} : U(x) = \min_{x'} U(x')\}
\end{aligned}$$

respectively. Here, we clearly see that the inverse temperature α plays the role of a boundedness parameter and that the single expression $\frac{1}{\alpha}\log Z$ is a generalization of the classical concept of value in control.

There are many ways of representing the same control pattern. Two planning problems are said to be equivalent iff they have the same prior and posterior policy distributions, and the same certainty-equivalent. The following theorem characterizes equivalent planning problems.

Theorem 1. *Two planning problems $(\mathcal{X}, \alpha, Q, U)$ and $(\mathcal{X}, \beta, Q, V)$ with partition functions Z_α and Z_β respectively are equivalent iff*

$$\alpha U(x) - \log Z_\alpha = \beta V(x) - \log Z_\beta. \quad (4)$$

In particular, the following corollary is crucial for the construction of generalized decision trees.

Corollary 1. *For any planning problem, there exists always a unique equivalent planning problem with a prespecified inverse temperature.*

B. Sequential Decisions

The previously outlined bounded rational framework can be extended to multiple steps by interpreting outcomes as *trajectories*, i.e. $x = x_1, \ldots, x_T$. These are essentially the planning problems considered by Kappen and Todorov in the KL-control framework. We generalize this to planning problems where the agent can have varying degrees of

control in each state, and represent these as generalized decision trees.

A *generalized decision tree* [9] is a tuple $(T, \mathcal{X}, \beta, Q, R, V)$ where:

- $T \in \mathbb{N}$ is the *horizon*, i.e. the depth of the tree;
- $\mathcal{X}$ is the alphabet of interactions, defining the set of *states* $\mathcal{X}^* := \bigcup_{t=0}^{T} \mathcal{X}^t$ (i.e. the nodes of the tree), where the subset $\mathcal{X}^T \subset \mathcal{X}^*$ is the set of terminal states;
- $\beta(x_{\leq t})$ is the *inverse temperature* in the state $x_{\leq t} \in \mathcal{X}^*$;
- $Q(x_t|x_{<t})$ is *prior probability* of moving from state $x_{<t}$ to state $x_{\leq t} = x_{<t}x_t$;
- $R(x_t|x_{<t})$ is the *reward* obtained when moving from state $x_{<t}$ to state $x_{\leq t}$;
- $V(x_{\leq T})$ is the *value* of the terminal state $x_{\leq T}$,

where we have used the shorthands $x_{<t} := x_1, \ldots, x_{t-1}$ and $x_{\leq t} := x_{<t}x_t$.

Generalized decision trees only differ from classical decision trees in that the former have node-specific inverse temperatures instead of having decision and chance nodes. In order to solve them, we need to extremize the following functional.

Theorem 2. *The free energy of a generalized decision tree is given by:*

$$F_\alpha[P] = \sum_{x_{\leq T}} P(x_{\leq T}) \Big\{ \sum_{t=1}^{T} G(x_t|x_{<t}) + V(x_{\leq T}) \Big\} + C \quad (5)$$

where C is a constant independent of P and where

$$G(x_t|x_{<t}) := R(x_t|x_{<t}) - \frac{1}{\beta(x_{<t})} \log \frac{P(x_t|x_{<t})}{Q(x_t|x_{<t})} \quad (6)$$

is the information-constrained instantaneous reward.

The proof relies on applying Theorem 1 to the individual nodes of a decision tree with homogeneous temperatures to obtain a decision tree with heterogeneous temperatures. Note that the constant C in (5) can be dropped, because it does not affect the resulting equilibrium distribution.

Theorem 3. *The solution to the free energy is given by*

$$P(x_t|x_{<t}) = \frac{1}{Z(x_{<t})} Q(x_t|x_{<t}) \exp\Big\{ \beta(x_{<t}) W(x_{\leq t}) \Big\},$$

where the partition functions of the terminal and internal states, respectively, are recursively defined as

$$Z(x_{\leq T}) = \exp\Big\{ \beta(x_{\leq t}) V(x_{\leq T}) \Big\}$$

$$Z(x_{<t}) = \sum_{x_t} Q(x_t|x_{<t}) \exp\Big\{ \beta(x_{<t}) W(x_{\leq t}) \Big\},$$

where $W(x_{\leq t})$ is shorthand for

$$W(x_{\leq t}) := R(x_t|x_{<t}) + \frac{1}{\beta(x_{\leq t})} \log Z(x_{\leq t}),$$

i.e. the instantaneous reward plus the value of the future.

III. Solving the Generalized Optimality Equations

Classical decision trees are typically solved using dynamic programming. With a decision tree of depth T and alphabet $\mathcal{X}$, this would require $\mathcal{O}(|\mathcal{X}|^T)$ operations, which can quickly become intractable. A brute-force approach for solving generalized decision trees that computes the values recursively has the same time complexity. However, we can do better. In the bounded rational case, solving the generalized optimality equations amounts to *sampling* from the equilibrium distribution P, given a sampler Q. Directly sampling from P is intractable because it requires computing the partition function. Therefore, we propose two basic sampling schemes:

1) *Rejection sampling*, for the case when we want to obtain a sample that meets a prespecified target value. The number of proposals will depend on this target.
2) *Metropolis-Hastings*, for the case when we want to specify the number of proposals. The target value will depend on the amount of proposals.

We first discuss the methods for solving one-step decisions and then generalize them to sequential decisions that have *either only positive or only negative inverse temperatures*.

A. Basic Rejection & Metropolis Sampling

If we set a desired target value, then we can use rejection sampling to obtain the sample x. This works as follows: draw first a sample x from Q, then accept with probability

$$A(x|V^*) = \min\big\{1, e^{\alpha(U(x)-V^*)}\big\}, \quad (7)$$

where $V^* \in \mathbb{R}$ is the target value.

Theorem 4. *Rejection sampling with acceptance probability* (7) *produces the correct distribution as long as $V^* \geq \max_x\{U(x)\}$ when $\alpha \geq 0$ and $V^* \leq \min_x\{U(x)\}$ when $\alpha \leq 0$.*

If we do not want to fix a target value but instead we prefer fixing the number of proposals, then we can run a Markov chain and use a Metropolis scheme to obtain a sample from P. This is done as follows. Given a current state x, we propose the next state x' by sampling it from Q and then accept the transition $x \to x'$ with probability

$$A(x'|x) = \min\Big\{1, e^{\alpha(U(x')-U(x))}\Big\}. \quad (8)$$

Otherwise the stay at x. We repeat this for a fixed number of iterations and then return the last state as a sample. Notice that the Metropolis sampler can be seen as a rejection sampler where the target is given by the utility of the previous step.

Theorem 5. *The stationary distribution of the Markov chain with acceptance probability* (8) *is the equilibrium distribution* (2)*.*

Equations (7) and (8) can intuitively thought of as sampling challenges where the difficulty is mainly controlled by the inverse temperature α—the closer α is to zero, the easier it is to accept a proposal.

B. Sampling in Generalized Decision Trees

To obtain a sample from the posterior of a generalized decision tree, we can use the same Monte Carlo schemes as in the one-step decision case. However, there is an important caveat. While in the previous case there is a single inverse temperature governing the difficulty of obtaining a sample, in generalized decision trees we have one for each node—the root node being the one that characterizes the overall planning ability of the agent. Therefore, any sampling algorithm must take these heterogeneous control restrictions into account. In what follows, we derive a recursive sampling algorithm that renders the sampling process practical by equalizing the inverse temperatures but simultaneously corrects this distortion by altering the number of accepted proposals it requires in order to accept a sample. This algorithm only works when the inverse temperatures in the decision tree have the same sign—although the magnitudes are allowed to differ.

For this, we start our analysis by considering the marginal distribution of the first step. Given a target value V^*, to obtain a sample from

$$P(x_1) = \frac{1}{Z(\varepsilon)} Q(x_1) \exp\big\{\beta(\varepsilon) R(x_1) + \tfrac{\beta(\varepsilon)}{\beta(x_1)} \log Z(x_1)\big\}$$

we can first sample $x_1' \sim Q(x_1)$, and then accept it with probability a, where a is the acceptance probability of the tail:

$$\begin{aligned} a &= \frac{\exp\big\{\beta(\varepsilon) R(x_1') + \tfrac{\beta(\varepsilon)}{\beta(x_1')} \log Z(x_1')\big\}}{\exp\big\{\beta(\varepsilon) V^*\big\}} \\ &= \left(\frac{Z(x_1')}{\exp\big\{\beta(x_1')[V^* - R(x_1')]\big\}} \right)^{\frac{\beta(\varepsilon)}{\beta(x_1')}} =: z^{\frac{\beta(\varepsilon)}{\beta(x_1')}}. \end{aligned}$$

This result has a convenient operational interpretation. Define the *temperature ratio* as $\xi := \beta(\varepsilon)/\beta(x_1')$. Since the inverse temperatures have the same sign, $\xi > 0$, and if we assume that $z \leq 1$, then accepting the sample x_1' is equivalent to generating ξ *consecutive Bernoulli successes* with bias z (we will see further down how to generate these). In turn, since z is equal to

$$\begin{aligned} &\frac{Z(x_1')}{\exp\big\{\beta(x_1')[V^* - R(x_1')]\big\}} \\ &= \frac{\sum_{x_2'} Q(x_2'|x_1') \exp\big\{\beta(x_1') R(x_2'|x_1') + \tfrac{\beta(x_1')}{\beta(x_{\leq 2}')} \log Z(x_{\leq 2}')\big\}}{\exp\big\{\beta(x_1')[V^* - R(x_1')]\big\}}, \end{aligned}$$

generating a Bernoulli success is equivalent to first generating $x_2' \sim Q(x_2|x_1')$ and then accepting with probability a', where

$$a' = \left(\frac{Z(x_{\leq 2}')}{\exp\big\{\beta(x_{\leq 2}')[V^* - R(x_1') - R(x_2'|x_1')]\big\}} \right)^{\frac{\beta(x_1')}{\beta(x_{\leq 2}')}}$$

is the probability of the tail rooted at $x_1' x_2'$. It is easily seen how to recursively extend this process for generating $x_3', x_4', \ldots$ until reaching a leave x_T'. Essentially, when a parent node has a different temperature from its child node, then the previous procedure "equalizes" them by demanding either more ($\xi > 1$) or less ($\xi < 1$) accepted samples from the child node in order to accept the sample from the parent node.

a) Generating a non-integer amount of consecutive Bernoulli successes: To make this algorithm practical, we need to determine an efficient way to generate an arbitrary, possibly non-integer amount ξ of consecutive Bernoulli successes. This can be done by first generating $\lfloor \xi \rfloor$ Bernoulli trials in the obvious way, and then generating the remaining $(\xi - \lfloor \xi \rfloor)$ using the following theorem.

Theorem 6. *Let x be a Bernoulli random variate with bias $(1 - f_N)$ where*

$$f_N = \sum_{n=1}^{N} b_n, \qquad \text{and}$$
$$b_n = (-1)^{n+1} \frac{\xi(\xi-1)(\xi-2)\cdots(\xi-n+1)}{n!}$$

for $0 < \xi < 1$ and where N is a Geometric random variate with probability of success p. Then, x is a Bernoulli random variate with bias p^ξ.

b) Summary of the algorithm: We now state the recursive rejection sampling algorithm. To obtain a sample from $Z(x_{<t})$ with target value $V(x_{<t})^*$:

1) Obtain a sample $x' \sim Q(x_t|x_{<t})$.
2) *Base case:* If $x_{<t}x'$ is a terminal node, then accept with probability

$$\exp\Big\{\beta(x_{<T})\big(R(x_T|x_{<T}) + V(x_{\leq T}) - V^*(x_{\leq t})\big)\Big\},$$

   otherwise reject.
3) *Recursion:* if $x_{<t}x'$ is not a terminal node, then attempt to generate $\xi = \beta(x_{<t})/\beta(x_{<t}x')$ accepted samples from $Z(x_{<t}x')$ with target value $V^*(x_{<t}x') := V^*(x_{<t}) - R(x'|x_{<t})$. If *all* of them are accepted, then return *any* of the generated paths; otherwise reject.

This is initialized by setting $V^*(\varepsilon)$ equal to our initial target value V^*, and then generating a sample from $Z(\varepsilon)$. If the sample gets accepted, then we choose any of the generated trajectories $x'_{\leq t}$ as our accepted sample. Analogously to the one-step decision case, the Metropolis sampler uses the recursive rejection sampler as the acceptance step.

IV. Experimental Results

We have conducted three experiments. The first one verifies that the proposed Monte Carlo methods generate the correct distribution. The second experiment investigates the relation between the difficulty of generating a sample, the number of outcomes, and the inverse temperature. Finally, we apply the Metropolis sampler to a navigation planning example. It must be stressed that, in the literature, there exists no planning algorithm that can calculate the optimal policy of a generalized decision tree.

A. Experimental Validation of Monte Carlo Methods

We compared the equilibrium distribution obtained by Monte Carlo simulation with the true equilibrium distribution—see Figure 2, panels a, b & c. For this, we first created a decision tree of depth 3 with branching factor 10, totalling an amount of 1000 leaves. The tree's transition probabilities, rewards and inverse temperatures were chosen randomly. Panels a and b compare the true equilibrium distribution (solid blue) against the simulated equilibrium distribution using both rejection sampling (dash-dotted red) and Metropolis (dashed green) in a regular plot and a semi-log plot respectively. Panel c shows the corresponding relative deviation curves ($d(x) := \log \frac{p(x)}{\hat{p}(x)}$) for the two simulations. The outcomes have been sorted in ascending order to ease the interpretation.

We found that these simulations were very accurate, confirming the validity of our algorithms. In the case rejection sampling, we have found that choosing a target value that is too high increases the number of rejected proposals. In the Metropolis-Hastings sampler, we have found that the Markov chain has to be run roughly three times longer than rejection sampling in order to obtain a sample from the equilibrium distribution with high probability.

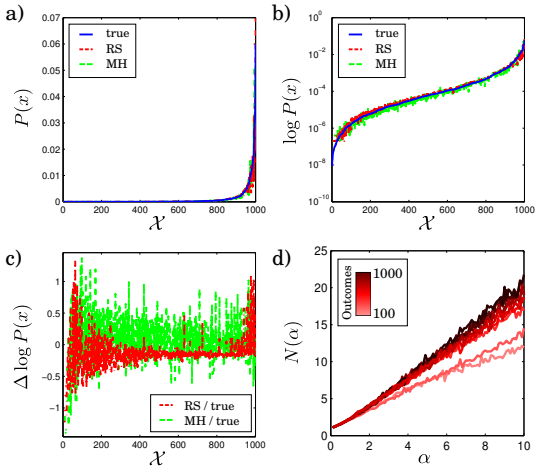

Fig. 2. Panels a,b & c: Comparison of the true versus simulated equilibrium distribution. Panel d: Average number of rejected proposals before acceptance as a function of the inverse temperature.

B. Number of Proposals

We investigated the relationship between the average number of rejected proposals, the number of outcomes, and the inverse temperature. In order to do so, we have created a total of ten one-step decision trees of increasing number of outcomes. The transition probabilities and rewards were drawn uniformly. Then, for each decision tree, we then simulated the equilibrium distribution as a function of the inverse temperature α, and then calculated the average number of rejections before acceptance. The resulting curves are shown in Figure 2, Panel d. Ten curves are shown, corresponding to one-step decision trees with 100, 200, ..., 1000 outcomes. These curves show a remarkable fact: as the number of outcomes increases, the proposal curves converge to a limit curve. Hence, the number of proposals essentially depend on the inverse temperature α, and not on the number of outcomes. This suggests that the inverse temperature controls the effective number of alternatives in the decision problem. Instead, dynamic programming must visit all the leaves of the tree in the worst case.

C. Navigation Planning with Limited Control

We have applied the Metropolis-Hastings sampler in a toy planning problem. A vehicle has to be remotely controlled using an antenna with limited range through a landscape with quadratic cost. The strength of the signal of the antenna limits the ability to control the vehicle, which would follow a dynamics $x(t)$ governed by a velocity vector $v(t)$ evolving as a random walk when uncontrolled:

$$x(t) = x(t-1) + v(t) \cdot dt, \qquad v(t) = v(t-1) + \nu \cdot dt,$$

(i.e. integration using the Euler method with time discretization dt) where ν is normally distributed with mean zero and a diagonal covariance matrix. Notice that the corresponding decision tree has an uncountably infinite branching factor. The signal strength was modeled with a location-dependent inverse temperature. We sampled 30 trajectories from the equilibrium distribution using Metropolis-Hastings (1000 iterations) for 3 starting locations having the same distance from the goal but different initial signal strength. The trajectories are shown in Figure 3, panels a–c. In the map, the black contours model the inverse temperature/signal strength, and the red contours the local reward (the minimum is at $[0, 1]$). Panels d–f show the mean evolution and error bars (one standard deviation) of the trajectories' reward curves during the Monte Carlo simulation. It is seen that a strong signal (first column) leads to better controlled future projections, whereas a low signal (right column) significantly hampers the ability to control the vehicle.

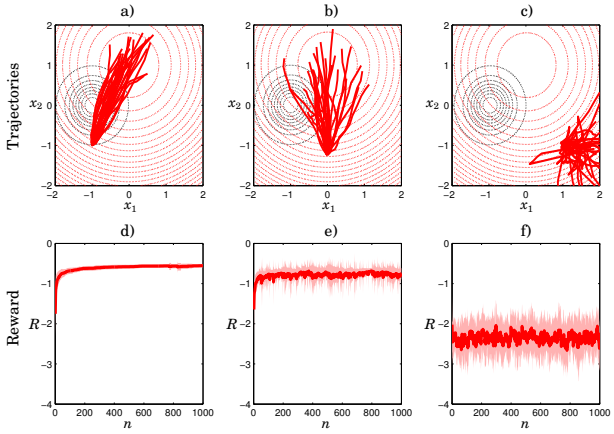

Fig. 3. Navigation planning with limited control under three initial conditions. Panels a–c show the projected trajectories, where the red and black contours encode the reward and inverse temperature landscapes respectively. Panels d–f contain the mean evolution of the Monte Carlo simulation generating the trajectories.

V. Discussion and Conclusions

A. Very large and negative temperature ratios

The proposed sampling methods work well when the temperature ratios between two subsequent states are strictly positive at all times, which is the case when all the inverse temperatures in the tree have the same sign. However, when the temperature change tends to infinity $\xi \to \infty$, then the number of required samples from the child node grows unboundedly. This can only happen when the inverse temperature of a child node tends to zero. However in this case, any of the child node's samples get accepted, and so one can interpret this process as essentially estimating the typical realization of the uncontrolled process.

In the case when the temperature ratio is negative ($\xi < 0$), then our interpretation in terms of Bernoulli trials breaks down—since it would correspond to generating a negative amount of consecutive Bernoulli successes. This restriction implies that we cannot solve generalized decision trees mixing cooperative and adversarial transitions.

B. Number of Proposals

Our second experiment has suggested that the inverse temperature controls the effective number of alternatives considered by the agent. The following theorem tells us how many proposal samples from Q are needed in order to generate a sample from the equilibrium distribution in a one-step decision.

Theorem 7. *Let $\delta > 0$ be a constant. The number of proposals n_α needed to achieve a probability $1 - \delta$ of acceptance is given by*

$$n_\alpha = \frac{\log \delta}{\log(1 - p_\alpha)}$$

where

$$p_\alpha = \frac{Z_\alpha}{\exp\{\alpha V^*\}} = \frac{\sum_x Q(x) \exp\{\alpha U(x)\}}{\exp\{\alpha V^*\}},$$

as long as $V^* \geq \max_x\{U(x)\}$ whenever $\alpha \geq 0$ or $V^* \leq \min_x\{U(x)\}$ whenever $\alpha \leq 0$.

Importantly, if we interpret $\mathcal{X}$ as a discretization of a continuous domain Ω endowed with a prior probability density $q(\omega)$ and bounded utility density $u(\omega)$, then the partition function Z_α corresponds to a discrete approximation to the partition function over Ω. It is easily seen that in this case, the number n_α of samples does not depend on the number of outcomes $|\mathcal{X}|$.

C. Conclusions

The presented sampling schemes for generalized decision trees operationalize the free energy for bounded rational control. This has two implications. First, we can solve a novel class of control problems under information constraints due to resource constraints, risk-sensitivity, trust and model uncertainty. Second, we have shown how the trade off between value and information encapsulated in the free energy functional can be exploited algorithmically. In particular, to find the optimal solution to a generalized decision tree, we do not need to visit all its branches. Rather, the amount of branches to be explored is directly controlled by the inverse temperatures of the internal nodes. This is in stark contrast to dynamic programming, which needs to visit all the branches to obtain an exact solution. More generally though, we believe that our work casts some light onto the problem of bounded-rational control [18]. In particular, our results suggest an intricate relationship between the degree of control of an agent, its value thresholds, and the effective number of alternatives it is contrasting during planning.

Acknowledgments: The authors thank Cardinal for his contribution of Theorem 6.

References

[1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice-Hall, Englewood Cliffs, NJ, 2010.

[2] M. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1999.

[3] G. S. Hamzei and D. Mulvaney, "Self-organising fuzzy decision trees for robot navigation: An online learning approach," in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, vol. 3, 1998, pp. 2332–2337 vol.3.

[4] S. Koo, J.-G. Lim, and D.-S. Kwon, "Online touch behavior recognition of hard-cover robot using temporal decision tree classifier," in *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*, 2008, pp. 425–429.

[5] D. Wilking and T. Röfer, "Realtime Object Recognition Using Decision Tree Learning," in *RoboCup 2004: Robot Soccer World Cup VIII*, ser. Lecture Notes in Computer Science, D. Nardi, M. Riedmiller, C. Sammut, and J. Santos-Victor, Eds. Springer Berlin Heidelberg, 2005, vol. 3276, pp. 556–563.

[6] H. He, T. McGinnity, S. Coleman, and B. Gardiner, "Linguistic Decision Making for Robot Route Planning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 203–215, 2013.

[7] D. Michie, "Game-playing and game-learning automata," *Advances in Programming & Non-Numerical Computation*, pp. 183–200, 1966.

[8] R. Bellman, "Dynamic Programming," Princeton, NJ, 1957.

[9] P. Ortega and D. Braun, "Free Energy and the Generalized Optimality Equations for Sequential Decision Making," in *European Workshop on Reinforcement Learning (EWRL10)*, 2012.

[10] P. A. Ortega and D. A. Braun, "Thermodynamics as a Theory of Decision-Making with Information Processing Costs," *Proceedings of the Royal Society A 20120683*, 2013.

[11] H. Kappen, "A linear theory for control of non-linear stochastic systems," *Physical Review Letters*, vol. 95, p. 200201, 2005.

[12] H. Kappen, V. Gómez, and M. Opper, "Optimal control as a graphical model inference problem," *Machine Learning*, vol. 1, pp. 1–11, 2012.

[13] E. Todorov, "Linearly solvable Markov decision problems," in *Advances in Neural Information Processing Systems*, vol. 19, 2006, pp. 1369–1376.

[14] ——, "Efficient computation of optimal actions," *Proceedings of the National Academy of Sciences U.S.A.*, vol. 106, pp. 11 478–11 483, 2009.

[15] P. Ortega and D. Braun, "Information, utility and bounded rationality," in *Lecture notes on artificial intelligence*, vol. 6830, 2011, pp. 269–274.

[16] N. Tishby and D. Polani, *Perception-Action Cycle*. Springer New York, 2011, ch. Information Theory of Decisions and Actions, pp. 601–636.

[17] P. Ortega, "A unified framework for resource-bounded autonomous agents interacting with unknown environments," Ph.D. dissertation, Department of Engineering, University of Cambridge, UK, 2011.

[18] H. Simon, *Models of Bounded Rationality*. Cambridge, MA: MIT Press, 1984.