Exploration Bonuses and Dual Control

Peter Dayan[1]
CBCL
Department of Brain and Cognitive Sciences
E25-210, MIT
Cambridge, MA 02139

Terrence J Sejnowski
Howard Hughes Medical Institute
The Salk Institute
PO Box 85800
San Diego, CA 92186-5800
Department of Biology
University of California at San Diego
La Jolla, CA 92093

Running head: Exploration bonuses

Key words: reinforcement learning, dynamic programming, exploration bonus certainty equivalence

Abstract

Finding the Bayesian balance between exploration and exploitation in adaptive optimal control is in general intractable. This paper shows how to compute suboptimal estimates based on a certainty equivalence approximation (Cozzolino *et al*, 1965) arising from a form of dual control. This systematizes and extends existing uses of exploration bonuses in reinforcement learning (Sutton, 1990). The approach has two components: a statistical model of uncertainty in the world and a way of turning this into exploratory behavior. This general approach is applied to two-dimensional mazes with moveable barriers and its performance is compared with Sutton's DYNA system.

1 Introduction

Reinforcement learning techniques are often applied in problems whose challenge stems from the ignorance of the solvers about the nature of their environments. For instance, an agent running around a maze might not know where the goal is, where the boundaries are, where the barriers are, or whether any of these have changed over time. The optimal control for such problems has a 'dual' nature (Fe'ldbaum, 1965). The agent has to devote some effort, but not too much, to *exploring* the world in order to *exploit* it proficiently.

[1] Telephone number: (617) 252 1693 email: dayan@psyche.mit.edu

Arranging for just the right amount and type of exploration in reinforcement learning systems is, in general, intractable – dual control problems can be seen as partially observable Markov decision processes (POMDPs), whose adverse computational characteristics are all too well understood (Monahan, 1982; Lovejoy 1991). The balance between exploration and exploitation is one of the major issues in learning control, and systematic methods for addressing it are sorely required.

In reinforcement learning, various approaches to addressing the tradeoff between exploration and exploitation have been studied empirically (see Thrun, 1992, for a review). One popular technique is to be optimistic in parts of the state-space that have never been explored, or, if the environment can change over time, have not been explored recently (Sutton, 1990; Moore & Atkeson, 1993; Christiansen, Mason & Mitchell, 1991). Sutton's DYNA system does this explicitly by adding to the immediate value of each state-action pair a number that is a function of this how long it has been since the agent has tried that action in that state. Sutton (1990) called this number an exploration bonus. The agent is therefore encouraged to plan and try long-ignored actions. DYNA calculates the long term values of state-action pairs using a temporal difference method for approximating solutions to dynamic programming (Barto, Sutton & Watkins, 1989; Watkins, 1989). Moore & Atkeson (1993) achieve something similar to exploration bonuses with an agent that uses a different model of the environment from the one actually experienced. In this altered model, unfamiliar states have a possible transition to a fictitious and attractive absorbing state.[2]

Although these methods perform well in certain circumstances, none of them is explicitly based on a model of how the agent is uncertain about its world. It is the form of this uncertainty that should drive exploration. Optimal exploration is just optimal experimental design (Fedorov, 1972) applied to sequential decision problems (Cohn, 1994). The agent's initial uncertainty about the transition or reward structure of the world should drive the initial experimentation, and, if the world can change stochastically over time, then this further source of uncertainty should drive continuing exploration. Unfortunately, it is computationally intractable for the agent to determine the correct experiments even if it knows what it does not know (*eg* even if it knows which transitions it is unsure about). The various exploration bonuses mentioned above can be seen as heuristic methods for changing behavior to reflect ignorance. We sought a more systematic approach that decouples two facets of the problem: the uncertainty about the environment, and a (sub-optimal) method of turning this into exploratory actions.

As with POMDPs, we start from a probabilistic characterization of the uncertainty the agent has about the world. The agent uses Bayes' rule to update this probability distribution as it gains experience. Consider, for example, a maze in which the agent has to learn which actions are effective in each state (*ie* which actions do not attempt to take it

[2]Moore and Atkeson (1993) do this to arrange initial exploration, but one could equally well invent such transitions as a function of the time since the agent last visited a state.

across barriers from that state), and in which these action efficacies are changing over time in a stochastic manner. Consider an action that would allow a significant shortcut if it were effective, but has recently been observed to be barred. To the degree that the agent is uncertain about whether this action is now effective again, states leading to it acquire an apparent exploration bonus. This encourages the agent to explore the transition. A method is presented that uses a form of certainty equivalence approximation, since it is intractable to determine the optimal solution. Certainty equivalence approximations involve using the mean values of random variables in place of the random variables themselves in expressions that determine the appropriate actions. For some linear quadratic control problems with uncertainty about the state, it is exactly correct to perform planning using the mean value of the state, ignoring its variance in choosing the actions. Certainty equivalence is inexact in almost all non-linear problems.

The next section describes the new method in the context of finite state environments; Section 3 illustrates how it works in three tasks.

2 Exploration in Finite State Worlds

The General Case

Consider the case of optimal control in a nonstationary absorbing finite state Markovian environment. The state set is $\mathcal{X}$, the set of actions available to the agent is $\mathcal{A}$,[3] and the probability that the agent makes a transition to state $y \in \mathcal{X}$ on taking action $a \in \mathcal{A}$ at state $x \in \mathcal{X}$ is $p^t_{xy}(a)$, where t is the trial number, *ie* the number of times the agent has moved from a starting state to one of the absorbing states. The environment is nonstationary in the sense that between trials, the transition probabilities can change in a Markovian manner, according to a probability distribution $\mathcal{U}[p^{t+1}_{xy}(a)|p^t_{xy}(a)]$. The starting state may either be the same for each trial or determined randomly from one trial to the next. The cost of taking action a at state x is $c_x(a)$, and the task for the agent is to minimize the expected summed (and possibly discounted) cost.

There are various options for the form of the ignorance the agent has about its environment. A convenient one that we adopt below is to assume that the agent knows a) its own state $x^t(n) \in \mathcal{X}$ at all times n within trial t, b) the costs $c_x(a)$ of taking the actions, c) the states that are absorbing, and d) the distribution governing the way the transition probabilities change. The agent does not know $p^t_{xy}(a)$ – it has to infer these on the basis of the transitions that it experiences and its knowledge as to how they change. This ignorance turns a Markov decision problem into a POMDP.

[3]We assume that all actions are admissible at every nonabsorbing state, *ie* there is no state dependence as to which actions the agent can take.

In this POMDP, the unknown transition probabilities $p^t_{xy}(a)$ are themselves treated as random quantities, and the agent maintains a distribution over them which it updates in the light of information collected from the environment. At time n during trial t, this distribution is called $\mathcal{T}^{t,n}[p_{xy}(a)]$, and assigns joint probabilities to the complete set of transition probabilities. The agent uses Bayes' rule to update this distribution both during a single trial as it observes transitions, and at the end of a trial. The updates at the end of each trial are based on the distribution $\mathcal{U}$, which specifies random changes in all the transition probabilities. A complete description of the state of knowledge of the agent at time n in trial t consists of the distribution $\mathcal{T}^{t,n}[p_{xy}(a)]$ together with the current state $x^t(n)$. This is also known as the *information state* for the agent, and is what is used in POMDPs to generate optimal controls through dynamic programming (Meier, 1965; Rishel, 1970; Striebel, 1965; Bertsekas & Shreve, 1978; see Kumar, 1985 for a review).

An analogy may be helpful. Consider tossing a coin with an unknown probability θ of coming up tails. If an agent started with some prior distribution over θ, then, on observing a sequence of heads and tails, it can use Bayes' rule to produce a posterior probability distribution over θ that describes its full state of knowledge about the propensity of the coin to come up tails. This case is like the finite state case, where applying action a_1 can take the agent from x_1 to either x_2 or x_3. θ is like the probability that the action leads the agent to x_2 – and the agent's initial ignorance about the transition structure of the world is like its ignorance about θ. In the coin tossing case, $\mathcal{U}$ is just a form of delta function – the probability of getting tails does not change between coin flips.

In assessing the value of trying an action at one state as part of dynamic programming, the agent has to calculate the collective probabilities of the transitions to succeeding states, and also from those states on all potential paths. This involves probabilities over the power set of the set of all possible transitions, and becomes intractable for large problems. Note also that the distributions $\mathcal{T}^{t,n}[p_{xy}(a)]$, *ie* the beliefs of the agent, involve real numbers. This means that even though the actual state of the agent in the maze comes from a finite set, dynamic programming has to be performed in a *continuous* state space, in which the transition from one set of beliefs to another after performing an action or completing a run through the maze, is governed by Bayes rule.

Performing dynamic programming to solve a POMDP is highly computationally expensive (Monahan, 1982). We therefore make a form of certainty equivalence approximation, in which the stochasticity in the world is simplified through the use of expectations. The agent does not know the actual transition probabilities $p_{xy}(a)$, but only the distribution $\mathcal{T}^{t,n}$ over the transition probabilities. Under the approximation, it calculates the mean values $q^{t,n}_{xy}(a) = E_{\mathcal{T}^{t,n}}[p_{xy}(a)]$ for all the transition probabilities and uses those instead. In the coin flipping analogy, this is like working out the mean $\hat{\theta}$ of the posterior distribution for θ, the probability that the coin comes up tails, and using $\hat{\theta}$ as if it were the true probability that the coin comes up tails. Note that although $q^{t,n}_{xy}(a)$ and $\hat{\theta}$ are both generated by taking expectations, they themselves specify *probabilities.* This approximation was first suggested by Cozzolino, Gonzalez-Zubieta & Miller (1965).

Using the mean transition probabilities $q_{xy}^{t,n}(a)$ is a slightly unconventional certainty equivalence approximation — the agent plans as if the environment behaves like its statistical mean, although this mean is still a stochastic process. Note that in using this mean process, the approximation fails to directly account for the fact that if a transition on the way to the goal is blocked, then the agent will have to choose some other route to that goal. Experiments with a less crude approximation that takes this into account are hampered by their computational expense. Note also that in calculating the value of one state, the uncertainties as to how the transition probabilities at successor states came to influence the values of those successor states are ignored. Worse, for some models, it can be that the mean value $q_{xy}^{t,n}(a)$ is not even a possible value for the transition probability. In a version of the coin example, one might know that with probability 0.5 the coin always comes up tails and with probability 0.5, the coin always comes up heads. Under the approximation, the coin would be treated as if it was fair – 50% of the throws should produce tails and 50% of the throws heads – even though this cannot be. Also, in practice, it is only possible to calculate the means efficiently in environments for which there are simple sufficient statistics for the $\mathcal{T}^{t,n}[p_{xy}(a)]$ such that the agent does not need to record the complete history of all the transitions it has made (Striebel, 1965). However, this restriction does not render the exploration problem computationally trivial.

Under the approximation, the agent performs synchronous value iteration in the assumed 'mean' process:

$$\hat{V}_{\alpha+1}^{t,n}(x) = \min_{a\in\mathcal{A}} \left\{ c_x(a) + \gamma \sum_{y\in\mathcal{X}} q_{xy}^{t,n}(a)\hat{V}_{\alpha}^{t,n}(y) \right\} \tag{1}$$

where $0 \leq \gamma \leq 1$ is the discount factor and α is the number of the dynamic programming iteration. On convergence, the agent picks one of the actions that minimizes the right hand side of this equation, performs it, uses Bayes' rule to compute the posterior distribution over the transition probabilities based on the transition actually observed, and calculates $q_{xy}^{t,n+1}(a)$. The process then repeats. This amounts to doing dynamic programming exactly in a different controlled Markov process — one in which the probabilities of the transitions are equal to their subjective means. Thus, the value iteration update is guaranteed to converge, provided that all states x that appear to it to be absorbing ($q_{xx}^{t,n}(a) = 1\forall a$) have at least one action available whose cost is zero. There are environments for which arranging this could be problematical: for example, the agent may explore states from which it cannot escape. We assume benign worlds in which this does not occur. Note that the algorithm itself has no free parameters, even though the model of change in the world may have several. It is standard to allow the agent to perform synchronous dynamic programming using its model of the world for indirect approaches to solving Markov decision problems under conditions of incomplete knowledge of the transition probabilities (Sato *et al*, 1982; Kumar, 1985).

Depending on the model of how the transition probabilities can change over time, this algorithm automatically incorporates a form of exploration bonus or penalty (Sutton, 1990). Consider the case in which some particular transition from state x to state y would be highly advantageous in the sense of providing a cheap path. If the agent's expected probability $q_{xy}^{t,n}(a)$ for some action a is high, then state x itself comes to have a low expected value (cost), and so through the application of the synchronous dynamic programming (equation 1), the agent will plan to visit state x and attempt action a. The reduction in the cost of state x is the equivalent of the exploration bonus, a connection that is explored more extensively below.

The Maze Case

We investigated exploratory behavior in a variety of two-dimensional mazes involving barriers, with a simple model of environmental change. At state $x \in \mathcal{X}$, the agent can choose one of four geographical actions $a \in \{\text{N,S,E,W}\}$. The agent is required to minimize the number of actions (steps) taken until it reaches a goal, which is the only absorbing state. It knows from the outset the coordinates of the goal, which also does not move over time. All actions are admissible at all states, however, there are barriers in the maze. If there is a barrier between two states, then those actions that would cross the barrier are termed *ineffective*, and if the agent attempts to take such an action, it costs one step but leaves the agent in the state where it started. Barriers are always bidirectional. The efficacies of the actions taking during trial t, $e_x^t(a)$, are defined in the obvious way:

$$e_x^t(a) = \begin{cases} 1 & \text{if action } a \text{ is effective at state } x \text{ in trial } t, \\ 0 & \text{otherwise.} \end{cases}$$

These efficacies are related to the $p_{xy}^t(a)$ in the general description above. There are no rewards in the maze, and each action costs one step.

As in Moore and Atkeson (1994), the information the agent is given at the start consists of the dimensions of the maze (*ie* the layout showing all the possible transitions, some of which may, of course, be barred), the coordinates of the goal, and the fact that every action costs one step and there are no other rewards or punishments. During a trial, it is also told its exact location in the maze. It does not know the $e_x^t(a)$, *ie* it does not know the locations of the barriers in the maze. The agent finds the barriers by trying actions at states to test if they cause state changes. The agent can use the same methods for manipulating uncertainty if it is unsure about its own location as well as the transition structure of the maze, but we do not treat that case here. Knowing where the goal is in the maze is a substantial simplification for the agent – it does not need to have an efficient strategy for finding the goal in the first place. We have not included uncertainty about the reward structure of the environment, although this would be straightforward.

Such uncertainty, like uncertainty about the location of the goal, would lead to different patterns of exploration.

Under our scheme, the agent requires a probabilistic model of the efficacy of the transitions, and a model of how this changes over time. Specifically, let $q_x^{t,n}(a)$ be the agent's estimate of the efficacy of action a at x at timestep n during trial t. In this case, $q_x^{t,n}(a)$ is just the probability that $e_x^t(a) = 1$.[4] The agent assumes that between each trial with some small probability κ, each $e_x^t(a)$ gets set to a new value, independent of its previous value and the efficacies of all the other transitions. The new value is drawn from a prior distribution, with probability ϕ of being 1.

After trial $t-1$, the agent's actual posterior distribution is updated to $q_x^{t,0}(a)$ as follows:

$$q_x^{t,0}(a) = \begin{cases} \kappa\phi + (1-\kappa)q_x^{t-1,0}(a) & \text{if } a \text{ was not tried at } x \text{ during the trial,} \\ 1-\kappa(1-\phi) & \text{if } a \text{ was tried at } x \text{ and was successful,} \\ \kappa\phi & \text{if } a \text{ was tried at } x \text{ and was unsuccessful.} \end{cases} \quad (2)$$

During the course of a single trial $q_x^{t,n}(a)$ is reset when the agent tries a at x to whatever actually happened. For actions that were not attempted, $q_x^{t,n}(a)$ relaxes towards ϕ at a rate governed by the update probability κ. At the start of learning, the agent postulates that every transition has probability ϕ of being effective, *ie* $q_x^{1,0}(a) = \phi \; \forall x, a$.

This model of uncertainty was designed for simplicity. The full distribution $\mathcal{T}^{t,n}[p_x(a)]$ over the transition probabilities (discussed in the previous section) is *factorial*, which means that the efficacy of action a at state x (which is a random variable) is independent of the efficacies of all the other actions at state x and of all actions at all other states. Furthermore, each efficacy has a binomial distribution, which is characterized just by its mean. The update rule for $q_x^{t,n}(a)$ therefore captures the entire distribution for the efficacy. The approximation lies in the use of just the mean values $q_x^{t,n}(a)$ in equation 1 rather than taking correct account of the distribution. This is the same problem we encountered when treating a coin as being fair between tails and heads when, in fact, with probability 0.5 it will always produce tails, and with probability 0.5 it will always produce heads.

Figure 1 illustrates the technique in the absorbing case where there is no discount factor. The agent is at state i and is trying to get to state k. The expensive route, going S around the obstacle, is certainly available, but barriers (a) between i and j and (b) between j and k are each present with probability 0.1 (dropping the time and trial indices, the probabilities are: $q_i(E) = q_j(W) = q_j(E) = q_k(W) = 0.9$). The true expected return for trying action E at state i (namely the cost of performing that action plus the cost of getting to the goal

[4] This is a slight modification of the previous notation, but the meaning is clear since there are only two possible transitions for each action.

optimally from the state this leads to) is:

$$
\begin{aligned}
\mathcal{Q}(i,E) &= 0.9 \times 0.9 \times 2 && \text{in case a and b are both absent} \\
&+ 0.9 \times 0.1 \times 9 && \text{in case a is absent and b is present} \\
& && \text{(the agent will try action } E \text{ at } j\text{)} \\
&+ 0.1 \times 7 && \text{if a is present} \\
&= 3.13
\end{aligned}
$$

Under our approximation, the agent models the value of i as if the barriers a and b are present with probability 0.1 *every time it tries to cross them*. Performing value iteration as in equation 1, this makes the estimated value of j, $\hat{V}(j) = 1.11$ and the estimated value of i using action E, $\hat{\mathcal{Q}}(i,E) = 2.22$. This is quite close to the real value, although the agreement gets worse as the cost of doing action S at state i, or the probabilities that the barriers are there, go up. As mentioned above, the resulting control could be disastrous if, for instance, there were states that the agent could explore but could not escape. So if the barriers were not bidirectional, and the agent could move E from i to j but could not move either E or W from j, then it could fail to get to the goal. These cases were ruled out in the experiments – each point in the mazes always had some possible path to the goal.

Sutton (1990) originally developed exploration bonuses in the context of mazes with moveable barriers, and indeed the equivalent here has a particularly clear form. Consider action a at state x which, if effective, would take the agent to state y. If the agent believes y to be much closer to the goal than x on the basis of the other possible routes from x, *ie* $\hat{V}_{\alpha}^{t,n}(x) \gg \hat{V}_{\alpha}^{t,n}(y)$, then the less sure the agent is that a is ineffective, *ie* the larger $q_x^{t,n}(a)$, the lower the cost of the right hand side of equation 1 for action a. This potentially lowers the cost of $\hat{V}_{\alpha+1}^{t,n}(x)$. Thus the system is encouraged to explore actions to the extent that they are adaptive in its task of getting to the goal. The subjective probabilities $q_x^{t,n}(a)$ relax towards ϕ as the time increases since the action was last tried. If ϕ is large, this becomes the equivalent of the exploration bonus in DYNA (Sutton, 1990), for which the value of a function of the length of time since the action was last tried is explicitly added to the value of a state-action pair. In figure 1, imagine that barrier b is absent and the agent knows this. Then the $\mathcal{Q}$-value of action E at state i is $(1 + q_i(E))/q_i(E)$ where $q_i(E)$ is the probability that going East at i is effective. Under the relaxation scheme in equation 2, $q_i(E) = \phi\,(1 - (1-\kappa)^{s+1})$ if it has been s trials since the barrier was last detected and the action has not been tried since. Thus $\mathcal{Q}(i,E)$ decreases as s increases, just as if it had been given an exploration bonus. If $s > \log(1 - 1/(5\phi))/\log(1-\kappa) - 1$, then $\mathcal{Q}(i,E) < \mathcal{Q}(i,S)$, and the agent will attempt to take action E and will find out whether the barrier has now disappeared.

Our strategy is actually a slightly more specialized and directed form of exploration bonus than that in DYNA (Sutton, 1990). In DYNA, every state-action pair is made more

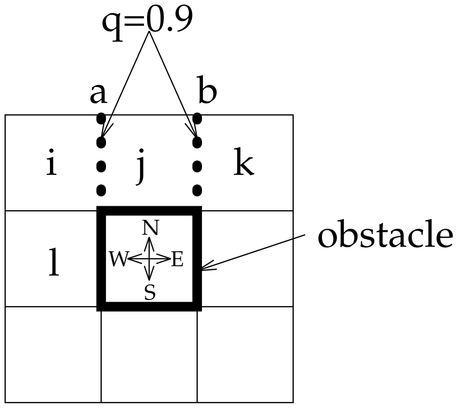

Figure 1: The certainty equivalence approximation. The agent is at state i and has to get to state k. Every action has a unit cost, and the solid thick lines are barriers that cannot be crossed. The dotted thick lines are the potential barriers a and b whose probabilities of being present are 0.1. All barriers are bidirectional. The costs of different ways of getting to k are given by:

State/Action Sequence	Cost	Existence of: a	b
$i \xrightarrow{E} j \xrightarrow{E} k$	2	absent	absent
$i \xrightarrow{E} j \xrightarrow{E} j \xrightarrow{W} i \xrightarrow{S} l \xrightarrow{S} \ldots \xrightarrow{N} k$	9	absent	present
$i \xrightarrow{E} i \xrightarrow{S} l \xrightarrow{S} \ldots \xrightarrow{N} k$	7	present	*
$i \xrightarrow{S} l \xrightarrow{S} \ldots \xrightarrow{N} k$	6	*	*

where the '*'s indicate that the existences of the barriers at a or b are irrelevant since the agent does not attempt the actions that would cross them.

---

attractive according to the length of time since it was last tried. This includes states which, as far as the system knows, are not on short paths from the start to the goal. Exploration bonuses more like Sutton's would be produced by a transition model in which every state has some probability of getting to every other state, were the relevant transition effective. However this obviously implies a different topology for the maze. Also the exploration bonuses here are bounded, whereas in Sutton's scheme they are potentially unbounded.

The parameter ϕ can be used to tilt the balance between exploration and exploitation. If ϕ is 1, then the system will always eventually explore to find paths that might be shorter than its current one. In a fixed maze, it will therefore find the shortest path available (although it will continue to explore if there could be advantage in doing so). As ϕ gets smaller, the system becomes more pessimistic. This restricts the range of excursions it will try about its currently optimal path – *ie* it will not make large detours to try possibly better paths. The lower ϕ, the more restricted the excursions. Using $\phi \neq 1$ can lead to the permanent use of suboptimal paths – it can only be justified if the costs of the extra exploration for better ones are prohibitive.

The other parameter κ also has an effect on the relative amounts of exploration and exploitation, but in a more indirect way. Whereas ϕ determines the ultimate amount of exploration about the current best-known path, κ controls how soon that exploration happens, by controlling how long it takes for information from the past to decay.

If mazes were really generated by the process that the agent assumes and the agent had the computational resources to work out the optimal adaptive control, then the best strategy would be to use the actual values for ϕ and κ. Under our approximation, it may not be best to use the correct values. The experiments below partially address this issue.

3 Experiments on Mazes

This section illustrates how exploration bonuses arise in the new algorithm, and how the amount of exploration depends on the parameters ϕ and κ. We tested the algorithm on three sorts of mazes, two regular ones and one constructed using the random procedure that the algorithm assumes.

The agent's experience was divided into a number of trials. In each trial, the agent starts at some point in the maze (which is fixed for the regular mazes and selected at random for the random maze) and takes actions until it reaches the goal, at which point the trial ends. In all cases complete synchronous value iteration was performed before each step in each trial using equation 1 to convert the current model of the world into a suggested action. The value function produced at the previous step was used to initialize value iteration, so convergence took only a few iterations unless the agent found out in the previous step that the world was substantially different from what it expected. The model was updated during the course of a trial as the agent found that transitions were effective or ineffective. At the end of a trial, all the expected efficacies $q_x^{t,n}(a)$ were updated to take account of the chance (under the model) that each transition might have been changed. Barriers were bidirectional, which helped prevent the agent from getting stuck. All moves in the maze cost 1 step, and the discount factor was also 1. The algorithm takes actions greedily according to the value function, and if multiple actions share the same optimal value, the tie was resolved in the order W, N, E, and then S. The overall exploration algorithm is deterministic in a fixed maze. The results in the first two mazes were averaged over 2000 trials merely to erase the effect of the agent's initial uncertainty about the structure of the maze it is in and to focus on measures of longer-term performance.

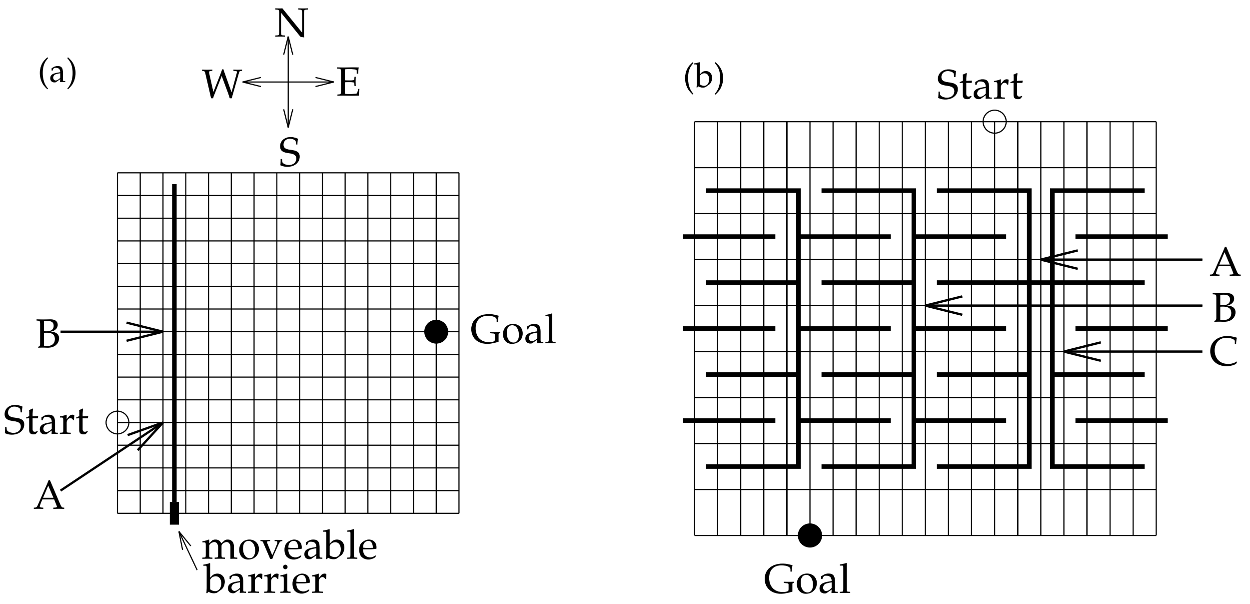

Figure 2: Two deterministic mazes. (a) The movable barrier maze. The lowest segment of the barrier (shown with a thick line) could be removed after some number of trials opening up a shorter path for the agent from the start to the goal. Points A and B are shown in the graphs in figure 3. (b) The zig-zag maze. We tested how often the agent attempted to go South at A, on the blocked shorter path to the goal, at B on one of the two better longer paths, and at C on one of the two worse longer paths.

Moveable Barrier Maze

The first task, shown in figure 2(a), is patterned after one used by Sutton (1990) to illustrate the workings of DYNA. The agent moves in a 16x16 grid, getting from a start point which is asymmetrically located at one end, to a goal which is centrally located at the other end. There is a single barrier near to the starting point which impedes the agent's passage. In Sutton's task, the lowest section of the barrier was removed after some number of trials. Since the agent was provided incentive to explore, the shorter path would eventually be discovered and then mostly exploited.

Good performance on this task requires the agent to plan to visit the various segments of the barrier that would permit it faster paths to the goal, if the transition East at those places were to become effective. Figure 3 shows the performance of the algorithm for various values of ϕ and κ. The abscissa shows position along the barrier starting at the South-most point. Performance here is given by the average number of times per trial that the agent attempted to make an Eastward transition across each segment of the barrier.[5]

[5]The reciprocal of this quantity is the number of trials between attempts.

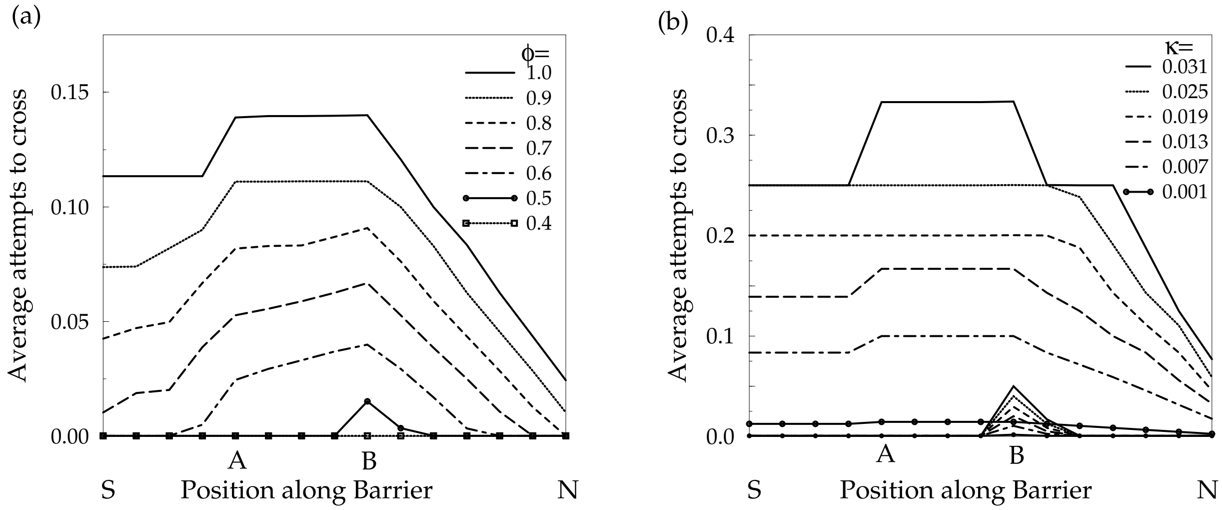

Figure 3: Performance in the moveable barrier task. a) the average number of eastwards attempts for all elements of the barrier for various values of ϕ and $\kappa = 0.01$. b) the same for various values of κ and $\phi = 1.0$ (flatter lines) and $\phi = 0.5$ (small peaked lines around B). Point A has the same y coordinate as start point and point B the same y coordinate as the goal. Positions are counted from the bottom, from South (S) to North (N).

The lowest segment was *never* opened, so that the maze remained as shown in figure 2(a).

One optimal path for the agent, given the state of the maze, is to go to point A on the West side of the barrier, then move North until the top, and finally cross at the North-most point (not shown on the graph). Therefore visiting points on the barrier south of A requires the agent to go substantially out of its way. Further, for points along the barrier North of B, the paths to the goal look suboptimal from the perspective of the maze without barriers.

For very low values of ϕ, the agent was so pessimistic about the steps it had yet to try on either the West or the East side of the barrier, that it barely explored at all, and certainly not anywhere off its path. However unless $\phi = 0$, it would find a path to the goal in the first place if one existed. For $\phi = 0.5$ it explored only North of point B because it never attempted getting to the barrier at points South of B. It was hindered for Southerly points by the uncertainty of the untested steps on the East side of the barrier. For larger values of ϕ, the agent attempted to cross the barrier where shorter paths were most likely.

One might think that all the points on the barrier between A and B would appear the same to the agent, since they are all on a shortest possible route to the goal. However the agent explored more at the ones nearer B, at least for moderate values of ϕ, because these were closer to the long path to the goal which was known to be open. The points nearer B were less affected by the uncertainty about the transitions to the East of the barrier on the way to the goal.

The two sets of lines in figure 3(b) show the effect of changing κ. It is hard to understand the precise shapes of the curves for different κ. It is clear, however, that increasing κ increased the amount of exploration – but it could not tempt the agent to explore paths that were too far from the one it currently preferred. Thus κ acts to modulate the effect of ϕ.

The Zig-Zag Maze

We next tested the algorithm on the zig-zag maze shown in figure 2(b). Here there are a number of long paths to the goal which go via the zig-zags, but there is also the possibility of a shorter path which is prevented by the single barrier under point A. One would expect exploration to be directed preferentially at this shorter path since it could save 21 of the 42 steps that it takes to get to the goal along either of the two long routes.

Figure 4 shows the number of times per trial that the agent attempted to go South at points A, B and C in the maze as a function of ϕ ($\kappa = 0.01$). If the transitions at B and C were open, this would save 8 steps along their respective paths. One would expect the transition at B to be tested more frequently than that at C for two reasons. First, B is on an optimal path, and therefore testing it does not take the agent out of its way. Second, the transition at B is more valuable than the transition at C since it would overall make for a shorter path. The more interesting question is how often the transition at A is tried compared to B and C. The graph shows that A was indeed tried substantially more often than either B or C, even though it was not on a currently optimal path to the goal. It would not be straightforward to predict how changing ϕ changes the amount of exploration. Even though ϕ affects $q_x^{t,n}(a)$ in a highly non-linear way, figure 4 shows that the number of attempts to cross varies almost linearly with ϕ.

Random Mazes

Finally, we tested the algorithm on 16×16 mazes that were produced according to the generative scheme embodied in the model. The goal was always at $(15, 13)$. On each trial the agent started at a random position in the maze, and transition probabilities were changed independently between each trial, with probability κ. If a transition probability was changed, then the probability that the associated action was set to be effective was ϕ (barriers were always bidirectional). Transitions around the edges of the maze were never blocked (a fact not known to the agent), since this tended to make a high proportion of such random mazes unsolvable (in the sense that there would be some states for which there is no path to the goal). Exact dynamic programming with full knowledge of the

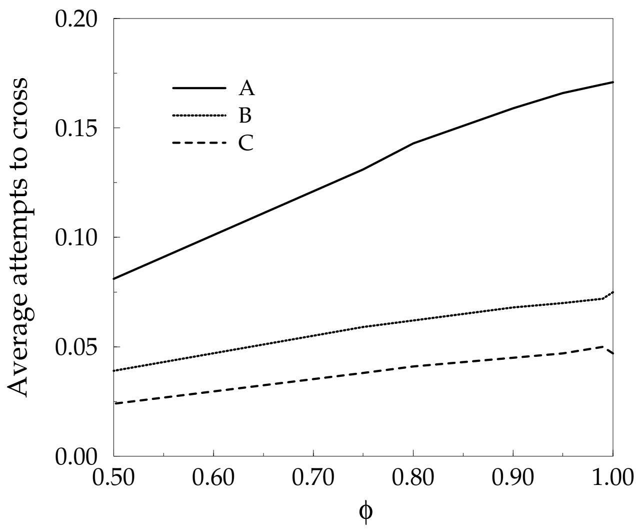

Figure 4: The average number of attempts per trial to move South at A, B and C in the zig-zag maze as a function of ϕ.

---

barriers was also performed, to find out the minimal length of the path from the start to the goal, and this allowed us to discover mazes that were indeed unsolvable. If this happened, a new trial was started, so that the barriers in the maze changed randomly according to the process described above.

Figure 5(a) shows the average number of extra steps over the optimal[6] the agent took for various values of the real ϕ against the assumed value of ϕ actually used when executing its dynamic programming (κ was fixed at 0.015 and was known to the agent). The actual average length of the shortest paths also decreased (from 15.33 steps at the real value of $\phi = 0.55$ to 11.93 steps at the real value of $\phi = 0.95$). None of the differences within a single line was significant, although there was a slight trend that suggests that the agent does better when it uses the real value of ϕ. Although this would clearly be true of the optimal algorithm, one might reasonably have expected that this suboptimal algorithm would tend to under-explore, and therefore overestimating ϕ would be good. It is apparently safe to use $\phi = 1$ at least for these random mazes. The overall performance was substantially degraded for a real value of $\phi = 0.55$. Unfortunately it is too computationally intensive to work out how well the optimal adaptive exploration system would have performed in this task.

Figure 5(b) shows average number of extra steps over the optimal for various assumed and actual values of κ and for fixed, known $\phi = 0.85$. There is a weak preference for using

[6]Optimal, assuming knowledge of the transitions that are effective.

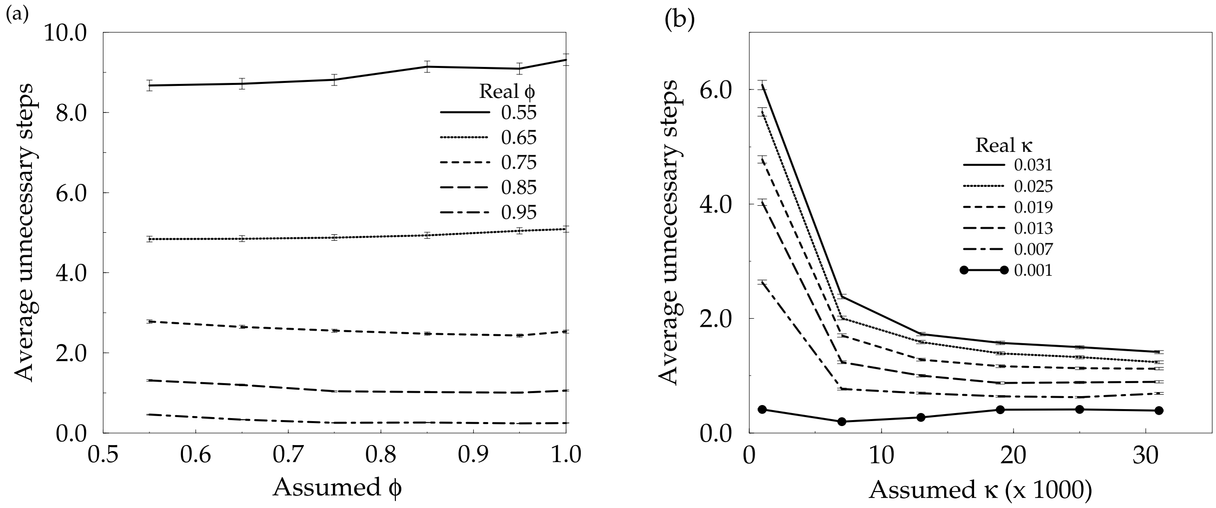

Figure 5: The average number of excess steps over the optimal for random mazes. All points are averages over $10,000$ trials, and standard error bars are shown (these only estimate the true variance since there are long range correlations in the numbers of excess steps arising from the slow change in the mazes and the agent's knowledge about them). (a) Unnecessary steps for various real and assumed values of ϕ (the agent uses the true value of $\kappa = 0.015$). (b) Unnecessary steps for various real and assumed values of κ (the agent uses the true value of $\phi = 0.85$).

something close to the actual value of κ. However, the two main points from the graph are: i) that using very small values of κ (which reduces exploration) leads to substantially more unnecessary steps, and ii), there is only a weak dependence on κ for values larger than this. These results are encouraging, since they suggest that the algorithm is somewhat robust against the choice of both of its parameters.

We also tested a version of DYNA in the random mazes to assess whether our exploration strategy was beneficial. We adapted DYNA to make it fit into our synchronous dynamic programming framework.[7] Rather than have the agent model the efficacy of transitions, we had it directly add to the immediate cost for a move an exploration bonus of $\alpha\sqrt{n_x(a)}$ for trying action a at state x, if that action had not been tried for $n_x(a)$ trials. The parameter $\alpha > 0$ controls the amount of exploration. Expected efficacies were set to 1 if the transition was last observed to be effective, and 0.01 if it was last seen to be ineffective.[8] The net immediate cost for any action was forced to be greater than 0.01 – if this could become negative, then the agent might believe it to be better to stay at a particular state rather

[7]This should improve the performance of DYNA over its original formulation, at a somewhat greater computational expense.

[8]This cannot be 0, otherwise the agent can come to believe that it is trapped when it is not.

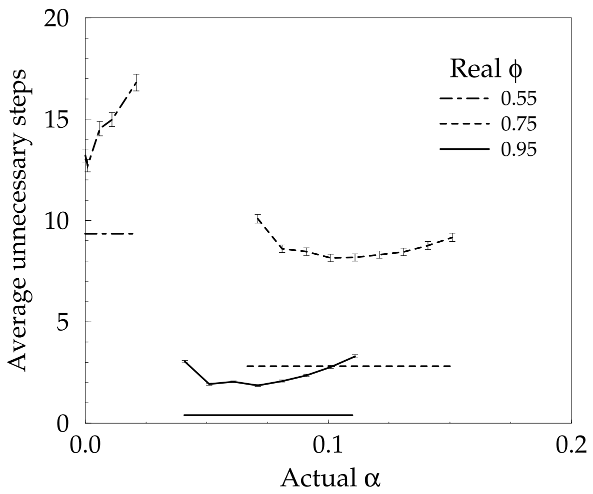

Figure 6: Average number of excess steps over the optimal for random mazes for modified DYNA. The three lines show results for three given values of ϕ as a function of α. Averages are over 5000 trials. The straight lines are taken from figure 5a and show the *worst* performance of our algorithm across all assumed values of ϕ. See the text for details of the algorithm.

---

than head for the goal and the value iteration in equation 1 would not converge.

Figure 6 shows how the DYNA version compared with our approach (the straight lines are taken from figure 5a and are the *worst* performances of our system). The ranges of α for the different values of ϕ were chosen to capture the best performance of the system (as seen by the 'U' shaped performance curves). It is apparent that the DYNA version was significantly worse for all values of ϕ, and was also much more sensitive to the choice of α than our algorithm was to the choice of ϕ. This is not surprising since for α high, DYNA is encouraged to attempt *all* transitions, irrespective of their utility in getting to the goal. For α low, DYNA is too cautious.

4 Discussion

This paper has systematized exploration bonuses and has presented results for agents traversing simple mazes showing that bonuses emerge naturally if the agent has a model of its uncertainty about the transition structure of the world. The ultimate algorithm has to make a form of certainty equivalence approximation to avoid the exponential computational cost of determining the optimal trade-off between exploration and exploitation. If the agent's model suggests that transitions that have been blocked can become available, then the agent has an incentive to explore those that might permit shorter paths to the goal. In our approximation, although the barriers are really either present or absent, the agent uses dynamic programming to compute policies that would be correct in a stochastic maze in which the barriers are probabilistically present or absent on each attempt at traversal, with the probability set by the agent's expectations. There is a clear distinction between the agent's model of the world, which reflects its assumptions about how things change, and the sup-optimal way in which it uses this model, which is based on a certainty equivalence approximation.

This certainty-equivalence approximation to performing DP in the full information state of the agent was first suggested by Cozzolino *et al* (1965). They developed it in the context of initial uncertainty about the transition matrices in a decision problem, and showed empirically that it performed quite well in a class of problems. Their system had to explore to find a good policy that it could then exploit for long periods. They pointed out that the algorithm is asymmetric in the sense that actions that seem poor, given the model of the environment (*eg* if ϕ is low) are never even tested, and so cannot be discovered to be good; whereas actions that seem excellent will tend to be tested and so can be proved bad.

Our experiments demonstrated that the combination of algorithm and model generate appropriate exploration in a variety of mazes. The agent strays off the path it currently uses to get to the goal in order to test transitions that would allow it to get to the goal faster. States only gain value if the agent has some expectation that they are on short paths to the goal. The algorithm for generating the exploration bonuses has no free parameters, but the model of how the world changes has several parameters and performance depends on how they are set. The model assumes that independently between each trial in the maze and with some small probability κ, each transition is reset from its previous efficacy to be open (with probability ϕ) or closed (with probability $1-\phi$). Qualitatively, ϕ governs the overall balance between exploration and exploitation, in the sense that it determines how far out of its way the agent will go to test a potential path of a given quality. κ, by determining how long it takes for knowledge about previous trials to decay, determines how often the agent will retry transitions that it has found blocked. The results appear reasonably robust against incorrect choices of ϕ and κ, at least in the problems we tried.

In most conventional approximations to dual control methods in standard control theory, agents are uncertain about their exact state (equivalent to their position in the maze) rather than the transitions. This is only a practical rather than a theoretical difference. The current algorithm was inspired by a technique called open loop feedback (or receding horizon) control (Dreyfus, 1965; Barto, Bradtke & Singh, 1995), which uses current uncertainty to determine current control. Completely open loop or feedforward control in this context would calculate optimally all the moves the agent would make from its current position to the goal, allowing for the uncertainty about the efficacies. This would clearly not work here, since the agent could try to cross a barrier of which it is unaware. It would then not update its control in the light of this failure. Open loop feedback control recalculates the optimal control at every step in the light of the information it has received in each of the preceding steps. This allows the agent to reach the goal, at least if there is some path from every state to the goal. Optimal open loop feedback control is intractable in the present context, since the agent would again have to consider the power set of all effective transitions, and our method is therefore an approximation to this. Note that the choice of open loop feedback control does not take account of the fact that the agent can find out more about the world in succeeding steps, as the overall optimal controller would. Open-loop feedback control and the present algorithm also share the unfortunate characteristic that the agent does not take account of the possibility of future use of the knowledge acquired. This possibility, which is at the heart of the tradeoff between exploration and exploitation, is incorporated into a more sophisticated and complicated method called wide-sense dual adaptive control (Tse, Bar-Shalom & Meier, 1973; Tse & Bar-Shalom, 1973) which also has a counterpart in the sort of problems we are considering here. We are currently testing an algorithm in which the agent knows that it is will perform repeated trials in the same maze, and knows that if it finds a transition to be open on one trial then it has the expectation that it will be able to use the transition in succeeding trials, which gives the agent an incentive to learn.

Although all these approaches have used complete DP (we used value iteration for the simulations), it could equally well have been implemented using some form of model based $\mathcal{Q}$-learning, as in DYNA (Sutton, 1990). Furthermore, as in Moore and Atkeson's Prioritized Sweeping (1993) and Peng and Williams' DYNA-$\mathcal{Q}$-queue (1992) variants, a computationally more efficient form of DP could have been used. Also a less specific exploration bonus, more like the one in DYNA, would have emerged had there been some possibility of a path from every state to the goal which might or might not be blocked. Further, the agent could be unsure about the rewards available from the environment instead of, or indeed as well as, the transitions. This would also lead to DYNA-like exploration, if the model of how rewards might appear had them being uniformly distributed across the states.

The theory developed here is entirely for indirect methods of control, in which the agent uses models of the world to perform its control. The extension to direct methods, which avoid the use of a model, is not at all obvious. One possibility is that the agent could have a model of how certainty changes over time and space, which is nevertheless too coarse to be useful for indirect control. The agent could use these uncertainties in an analogue of the systems of Schmidhuber (1991) or Thrun & Möller (1992). The latter make the agent learn the areas of the state space in which it makes errors in its predictions (of the value or transition functions of the environment); however, they again lack a statistical model of the agent's uncertainty, or a systematic way of turning this into exploratory behavior.

We have described our theory for finite state environments, and tested it exclusively in the context of mazes. However, two parts of our technique — using uncertainty about the environment to govern exploration, and employing certainty equivalence or other approximations in order to take tractable advantage of this information — are also applicable in other contexts, although different simplifying approximations may be appropriate in other environments.

For example, consider the quadratic regulator problem of Tse, Bar-Shalom and Meier (1973) which has imperfect state information and a nonlinear transition function. Their system used a second order model that maintained only the mean and variance, which were updated using an extended Kalman filter in the light of information from the world. This is the equivalent of maintaining the model of the maze using Bayes rule. Some approximation must then be made to take this uncertainty and generate from it controls that balance exploration and exploitation. Exploration entails taking actions that are expected to reduce the variance in the knowledge of the state; exploitation entails forcing the system towards its regulation point. If the regulator were linear, then straightforward certainty equivalence would hold and the controls could be correctly determined just on the basis of the mean value of the state, ignoring the variance. This is not true in the nonlinear case. Tse, Bar-Shalom and Meier's method involves choosing a nominal set of controls in the future, linearizing about them and using a second-order perturbation method to work out an approximate and sub-optimal cost of there being a particular mean and variance in the system's model of its state after one more time step. The system can then optimally choose a control in the current time step. As in the maze example, the stochasticity of the system was simplified to make control tractable; in this case, by treating only the mean and variance in the knowledge of the state.

There is a dual problem to this quadratic regulator problem that is even closer to the case we treated. Imagine that the system sees imperfect state information about the world, as before, but the transition function is also subject to random drift (Tse & Bar-Shalom, 1973; see Dersin, Athans & Kendrick, 1981 for adverse analysis of a special case). Again, the agent can use an extended Kalman filter to model the uncertainty about the transitions, and can make second-order approximations to determine controls that balance exploration (actions designed to be revealing about the transitions) and exploitation (actions that force the system towards its regulation point). Unlike the maze task, there are no exploration bonuses as such; instead, uncertainty always costs. However, this is just a function of the simplicity of the model of how the transitions change over time, and bonuses can occur in other models of environmental change.

Exploration was treated here as a potential benefit, but it can also be dangerous — an agent can easily fritter away valuable time in the fruitless investigation of its surroundings without helping itself find the goal faster. This makes it important for the agent to have a model of how the world changes and for it to be clear how this model is used to determine a (suboptimal) balance between exploration and exploitation. Even a poor model, such as the random independent update studied here may be better than none.

Acknowledgements

We are very grateful to Olivier Coenen, David Cohn, Andrew Moore, Satinder Singh, Sebastian Thrun, three anonymous reviewers, and particularly Andy Barto for their very careful reading of and helpful comments on earlier versions of this paper. The maze in figure 2b was suggested by one of the reviewers, as was the comment about modeling uncertainty in the rewards rather than the transitions. We especially thank Mike Duff for showing us the prior use of the approximation by Cozzolino *et al* (1965). This work was funded by the Howard Hughes Medical Institute, the UK SERC and the Canadian NSERC.

References

Barto, AG, Bradtke, SJ & Singh, SP (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence,* **72**, 81-138.

Barto, AG, Sutton, RS & Watkins, CJCH (1989). Learning and sequential decision making. In M Gabriel & J Moore, editors, *Learning and Computational Neuroscience: Foundations of Adaptive Networks.* Cambridge, MA: MIT Press, Bradford Books.

Bertsekas, D & Shreve, SE (1978). *Stochastic Optimal Control: The Discrete Time Case.* New York, NY: Academic Press.

Cohn, DA (1994). Neural network exploration using optimal experiment design. In JD Cowan, G Tesauro & J Allspector, editors, *Advances in Neural Information Processing Systems, 6.* San Mateo, CA: Morgan Kaufmann, 679-686.

Cozzolino, JM, Gonzalez-Zubieta, R & Miller, R (1965). *Markov Decision Processes with Uncertain Transition Probabilities.* Techical Report 11, Operations Research Center, MIT, March 1965.

Dersin, PL, Athans, M & Kendrick, DA (1981). Some properties of the dual adaptive stochastic control algorithm. *IEEE Transactions on Automatic Control,* **26**, 1001-1008.

Dreyfus, SE (1965). *Dynamic Programming and the Calculus of Variations.* New York, NY: Academic Press.

Fedorov, V (1972). *Theory of Optimal Experiments.* New York: Academic Press.

Fe'ldbaum, AA (1965). *Optimal Control Systems.* New York, NY: Academic Press.

Howard, RA (1960). *Dynamic Programming and Markov Processes.* New York, NY: Technology Press & Wiley.

Kumar, PR (1985). A survey of some results in stochastic adaptive control. *SIAM Journal on Control and Optimization,* **23**, 329-380.

Lovejoy, WS (1991). A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research,* **28**, 47-66.

Meier, L, IIIrd (1965). Combined optimal control and estimation. *Proceedings of the Third Annual Allerton Conference on Circuit and System Theory.*

Monahan, GE (1982). A survey of partially observable Markov decision processes: Theory, models and algorithms. *Management Science,* **28**, 1-16.

Moore, AW & Atkeson, CG (1993). Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning,* **13**, 103-130.

Moore, AW & Atkeson CG (1994). The Parti-Game algorithm. In G Tesauro, JD Cowan & J Alspector, editors, *Advances in Neural Information Processing Systems, 6.* San Mateo, CA: Morgan Kaufmann.

Peng, J & Williams, RJ (1992). *Efficient search control in DYNA.* College of Computer Science, Northeastern University.

Rishel, RW (1970). Necessary and sufficient dynamic programming conditions for continuous time stochastic optimal control. *SIAM Journal of Control,* **8**, 559-571.

Sato, M, Abe, K & Takeda, H (1982). Learning control of finite Markov chains with unknown transition probabilities. *IEEE Transactions on Automatic Control,* **27**, 502-505.

Schmidhuber, JH (1991). *Adaptive Confidence and Adaptive Curiosity.* TR FKI-149-91, Technische Universität München, Germany.

Striebel, CT (1965). Sufficient statistics in the optimal control of stochastic systems. *Journal of Mathematical Analysis and Applications,* **12**, 576-592.

Sutton, RS (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *Machine Learning: Proceedings of the Seventh International Conference,* 216-224.

Thrun, SB (1992). The role of exploration in learning control. In DA White & DA Sofge, editors, *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches.* New York, NY: Van Nostrand Reinhold.

Thrun, SB & Möller, K (1992). Active exploration in dynamic environments. In JE Moody, SJ Hanson & RP Lippmann, editors *Advances in Neural Information Processing Systems, 4,* 531-538. San Mateo, CA: Morgan Kaufmann.

Tse, E & Bar-Shalom, Y (1973). An actively adaptive control for linear systems with random parameters via the dual control approach. *IEEE Transactions on Automatic Control,* **18**, 109-117.

Tse, E, Bar-Shalom, Y & Meier, L, IIIrd (1973). Wide-sense adaptive dual control for nonlinear stochastic systems. *IEEE Transactions on Automatic Control,* **18**, 98-108.

Watkins, CJCH (1989). *Learning from Delayed Rewards.* PhD Thesis. University of Cambridge, England.