

## Conservation decision-making in large state spaces

Nicol, S.C.<sup>1</sup>, I. Chadès<sup>1</sup>, S. Linke<sup>1</sup> and H.P. Possingham<sup>1</sup>

<sup>1</sup>University of Queensland, Brisbane, Australia  
Email: [snicol@maths.uq.edu.au](mailto:snicol@maths.uq.edu.au)

**Abstract:** For metapopulation management problems with small state spaces, it is typically possible to model the problem as a Markov decision process (MDP), and find an optimal control policy using stochastic dynamic programming (SDP). SDP is an iterative procedure that seeks to optimise a value function at each timestep by trying each of the actions defined in the MDP.

Although SDP gives the optimal solution to conservation management questions in a stochastic world, its applicability has always been limited by the so-called curse of dimensionality. The curse of dimensionality is the problem that adding new state variables inevitably results in much larger (often exponential) increases in the size of the state space, which can make solving superficially small problems impossible. A large state space makes optimal SDP solutions computationally expensive to compute because optimal SDP techniques require the value function to be updated for the entire state space for every time step. The high computational requirements of large SDP problems means that only simple population management problems can be analysed.

In this paper we present an application of the on-line sparse sampling algorithm proposed by Kearns, Mansour & Ng (2002), which can be used to approximate the optimal solution of a MDP for a given starting state. The algorithm is particularly attractive for problems with large state spaces as it has a running time that is independent of the size of the state space of the problem.

We apply the algorithm of Kearns et al. (2002) to a hypothetical fish metapopulation where we define the management objective to be to maximise the number of occupied patches during the management horizon. Our model has multiple management options to combat the threats of water abstraction and waterhole sedimentation. We compare the performance of the optimal solution to the results of the on-line sparse sampling algorithm for a simple 3-waterhole case. We find that the approximation algorithm out-performs a random strategy for almost all combinations tested. In this case, the on-line algorithm gives the best approximation to the optimal SDP solution with three look-ahead steps and just two samples per approximation step. For this simple problem there does not appear to be much benefit in increasing the number of samples, although deeper simulation is required to confirm this. If only limited computing power is available, these results suggest that the best approximation will be obtained using three look-ahead steps and two samples per approximation step.

General results for management are hard to obtain from this theoretical model, as the results are highly dependent on the arbitrary parameters used to generate the results. However the results obtained suggest that the continued presence of water in the system is critical to the survival of the metapopulation, and that for this parameter set, prevention of water abstraction is more important than catchment management for this system. Further work needs to be done to properly parameterize this model before general results can be drawn with more confidence.

This paper introduces a new algorithm to conservation management that provides a way to avoid the effects of the curse of dimensionality. The work has the potential to allow us to approximate solutions to much more complex metapopulation management problems in the future.

**Keywords:** *Markov decision processes; Stochastic dynamic programming; Metapopulation; Optimal management; On-line sparse sampling algorithm*

## 1. INTRODUCTION

**Markov Decision Processes.** Many conservation problems can be thought of as sequentially trying to make ‘good’ decisions when the dynamics of the system are stochastic, where ‘good’ refers to maximising or minimising a specified conservation objective (Possingham et al. 2001, Shea et al. 1998). Thinking of conservation problems this way makes them amenable to solution using Markov Decision Processes (MDPs). A MDP provides a formal framework for making sequential decisions under uncertainty and has five components: i) the state space (we assume that this is finite and discrete), ii) the set of management actions that can influence the system, iii) the time frame of interest, iv) the transition probability matrix, and v) the immediate costs (or rewards) associated with being in a given state and taking a given action. The solution to a MDP is an optimal policy. The policy associates each state with an action so that if the policy is followed, the objective will be achieved with maximum probability. For more in depth discussion on using Markov decision theory in ecology and conservation management, see Mangel & Clark (2000), or Possingham et al. (2001).

### **Generating the optimal solution: stochastic dynamic programming.**

For Markov Decision Processes (MDP) with small state spaces, it is possible to use stochastic dynamic programming (SDP) (Puterman 1994) to compute an optimal control solution. SDP is an iterative procedure that optimises a value function at each timestep by trying each of the actions  $a$  defined in the MDP. The set of all possible actions for a state  $s$  is defined as  $A(s)$ . The value function represents the global value of being in a state relative to the objective. The optimal value function  $V^*(s)$  for a state  $s \in S$  at any time  $t$  can be computed iteratively using the infinite-time version of Bellman’s optimality equation (Puterman 1994) and the estimate of the value function from the previous time step:

$$V_t(s) = \min_{a \in A(s)} [r_t(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) \cdot V_{t-1}(s')], \quad (1)$$

where  $r_t(s, a)$  is the local cost paid for being in state  $s$  and taking action  $a$  at time  $t$ ,  $\gamma$  is a discount factor ( $0 < \gamma \leq 1$ ) that discounts the importance of future states because they are uncertain, and  $P(s'|s, a)$  is the probability of transition from state  $s$  to state  $s'$  after taking action  $a$ . The discount factor can be thought of as how much we value future rewards compared to immediate benefits. Systems with higher values of  $\gamma$  place a greater value on future rewards, whereas systems with lower values of  $\gamma$  place a greater emphasis on obtaining immediate rewards. For discount factors less than unity ( $\gamma < 1$ ), equation 1 converges to the optimal value function  $V^*(s)$ .

Once the value function has converged, the optimal policy  $\pi^*(s)$  for state  $s$  is the action that minimises the value function:

$$\pi^*(s) = \arg \min_{a \in A(s)} [r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) \cdot V^*(s')]. \quad (2)$$

The applicability of SDP is limited by the curse of dimensionality, which is the problem that adding new state variables results in exponential increases in the size of the state space. This ‘curse’ makes computing even a simple optimal SDP solution computationally expensive for many realistic models. The reason that the SDP algorithm is time-consuming for large state spaces is that optimal SDP techniques require the value function to be updated for the entire state space for every time step. The resulting high costs mean that only very simple models can be analysed.

### **Dealing with large state spaces : an on-line sparse sampling algorithm.**

When the state space for a conservation management problem is large, true optimal SDP solutions become simply too difficult to calculate. In this section we describe an on-line sparse sampling algorithm proposed by Kearns et al. (2002), which can be used to approximate the optimal solution for a given starting state. The term ‘on-line’ means that the policy is evaluated one step at a time based on the current state of the system. The algorithm is particularly attractive for problems with large state spaces as it has a running time that is independent of the size of the state space of the problem.

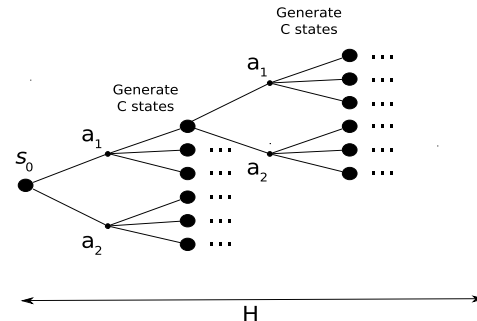
The Kearns et al. (2002) algorithm requires both an input state and a generative model which is able to predict transition probabilities from the current state (for this paper the generative model is equivalent to the transition probability matrix because the underlying dynamic is a discrete-time Markov chain). The algorithm calculates a near optimal policy for the input state in the next timestep using the generative model to sample states in the vicinity of the starting state. The algorithm “looks ahead” a defined number of time steps and carries out a smaller sub-MDP calculation on the states that are likely to be visited in the future. Because the algorithm only looks at states in the neighbourhood of the current state, the policy obtained will only approximate the optimal

solution for the current state. This means we sacrifice the computationally expensive global optimal solution for a faster local approximation that can be computed for any sized state space.

In practice, the Kearns et al. (2002) algorithm means trying every action  $C$  times from each state  $s$ . For each new state  $s'$ , the generative model is used to again try each action  $C$  times. The recursion is repeated for a defined number ( $H$ ) of look-ahead steps. The process is best visualised as a tree diagram of depth  $H$  and branch width  $C$  (Figure 1). The total size of the tree is  $O(|A|C)^H$ .

Using the states generated by the model, the value function for the current state can be updated according to (Péret & Garcia 2004):

$$V_{C,h}(s) = \begin{cases} \tilde{V}(s) & \text{if } h = 0 \\ \min_{a \in A(s)} Q_h(s, a) & \text{otherwise} \\ \text{where } Q_h(s, a) = \\ [r(s, a) + \frac{\gamma}{C} \sum_{s' \in S(s, a, C)} V_{C, h-1}(s')] \end{cases} \quad (3)$$



**Figure 1.** The Kearns et al. sparse look-ahead tree for an initial state  $s_0$ , with  $C=3$  samples and two actions  $a_1, a_2$  (modified from Péret & Garcia (2004)).

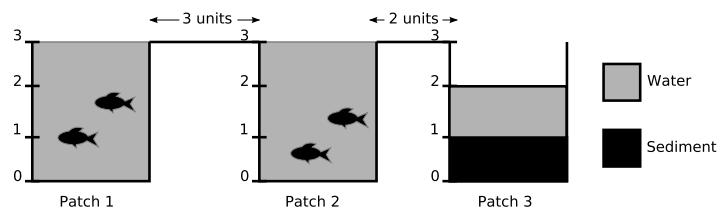
where  $h \in \{0, 1, \dots, H\}$  is the iteration index and  $S(s, a, C)$  is the set of  $C$  states that are sampled from initial state  $s$  after carrying out action  $a$ . Equation 3 is an iterative equation where each step is denoted using the index  $h$ . The approximately optimal value function  $\tilde{V}(s)$  is computed for the current state ( $h = 0$ ) by simulating  $C$  daughter states  $s'$  and computing the value function for each daughter state. The Q-function is zero for all actions  $a$  when  $h = 0$  (i.e.  $Q_0(s, a) = 0$ ), and so all future values can be derived iteratively from this. The best action for the current time step is the action that results in the minimum value of  $Q_H(s, a)$ . For further explanation of the algorithm, refer to Kearns et al. (2002).

In this paper we follow the work of Péret & Garcia (2004) and demonstrate the use of this algorithm in metapopulation management. A metapopulation is a collection of spatially separated populations of a single species with some degree of movement between populations (Levins 1969, Hanski & Gaggiotti 2004). We investigate the performance of the algorithm by computing the optimal solution for a spatially explicit metapopulation management problem with three patches and comparing it to the predictions of the Kearns et al. (2002) approximation for different values of  $H$  and  $C$ .

## 2. METHODS

**An application: an arid-zone fish metapopulation.** To illustrate how the sparse sampling approximation can be used to solve more complex conservation problems, a spatially explicit metapopulation model (Day & Possingham 1995, Ovaskainen & Hanski 2004) was constructed to solve an optimal management problem for fish populations in a series of waterholes in a hypothetical semi-arid zone river. For most of the year the river does not flow, restricting fish species in the river to remnant waterholes. Infrequent large rainfall events cause the river to flow, and allow the fish to disperse to other sites in the river. We assume that there are two major threats to fish in the river: sedimentation of the waterholes due to poor land practices causing erosion in the riparian zone; and abstraction of water from the waterholes by local land owners for use on agricultural land. To test the algorithm, we modelled a simplified 3-patch system. Because this system is relatively small, we can calculate an optimal solution using SDP which can be compared to the results of the on-line solution. The initial state of the system is shown in figure 2.

**The metapopulation model.** Each patch in the metapopulation is allowed to have three state variables: patch occupancy ( $n = 1$  or  $0$  if occupied or empty respectively), current water depth of the pond ( $d \in \{0, 1, 2, \dots, x\}$ ), and total depth (i.e. top of bank to deepest point) of the pond ( $x \in \{0, 1, 2, \dots, x_{max}\}$ ). In this paper we set  $x_{max} = 3$ . The units of depth are not specified but can be thought of as relative units (e.g. empty, low, medium and high water levels). For a



**Figure 2.** Initial state of the hypothetical river system. The water capacity of each patch is 3 units. Patches 1 and 2 are occupied and at full water capacity. Patch 3 is unoccupied, partly sedimented, and has 1 unit of water.

system with  $M$  patches, the size of the state space will be  $[2(x_{max} + 1)(\frac{x_{max}+2}{2})]^M$ . The system is modelled as a discrete-time Markov chain (e.g. Akcakaya & Ginzburg (1991), Day & Possingham (1995), Possingham (1996)). This section describes the model and records how to define the transition probability for the Markov chain.

In the absence of management, there are five independent processes that may occur: sedimentation of one or more of the waterholes, water abstraction from the waterholes, a rainfall event, and fish colonisation and extinction events. It is assumed that evaporation reduces the water level in the ponds on a much slower timescale than sedimentation and abstraction, and so we neglect evaporation from the model.

Sedimentation of a waterhole is assumed to occur with probability  $\delta$  and, if it occurs, has the result that the total depth of pond  $i$  is reduced by one unit ( $x_i \rightarrow x_i - 1$ , where  $x_i$  is the pond depth of the  $i^{th}$  pond). To define the transition probability after sedimentation for a system of  $M$  ponds, we first define a vector  $\tilde{X} = (x_1, x_2, \dots, x_M)$ . The effect of sedimentation in the  $i^{th}$  patch in the next time step is determined by the random variable  $\tilde{X}'(i)$ , which has observed value  $x_i \in \{0, 1, 2, \dots, x_{max}\}, \forall i \in \{1, 2, \dots, M\}$ . The probability distribution of a transition in the  $i^{th}$  pond from state  $\tilde{X}(i)$  to a new state determined by  $\tilde{X}'(i)$  is:  $P(\tilde{X}'(i) = \tilde{X}(i) - 1 | \tilde{X}(i), \tilde{X}'(i) \geq 0) = \delta$ ,  $P(\tilde{X}'(i) = \tilde{X}(i) | \tilde{X}(i)) = 1 - \delta$ , and otherwise the probability of transition is zero. Each pond may undergo a transition independently of the others, so the probability of transition from  $\tilde{X}$  to a state  $\tilde{X}'$  is:

$$P(\tilde{X}' | \tilde{X}) = \prod_{i=1}^M P(\tilde{X}'(i) | \tilde{X}(i)). \quad (4)$$

Similarly, abstraction may occur with probability  $\kappa$ , in which case it reduces the depth of water in pond  $i$  by one unit ( $d_i \rightarrow d_i - 1$ , where  $d_i$  is the depth of water in the  $i^{th}$  pond). We define a vector of depths  $\tilde{D} = (d_1, d_2, \dots, d_M)$ . The effect of abstraction in the  $i^{th}$  patch in the next time step is determined by the random variable  $\tilde{D}'(i)$ , which has observed value  $d_i \in \{0, 1, 2, \dots, x_i\}, \forall i \in \{1, 2, \dots, M\}$ . The probability distribution of a transition due to abstraction is given by:  $P(\tilde{D}'(i) = \tilde{D}(i) - 1 | \tilde{D}(i), \tilde{D}'(i) \geq 0) = \kappa$ ,  $P(\tilde{D}'(i) = \tilde{D}(i) | \tilde{D}(i)) = 1 - \kappa$ , otherwise the probability of transition is zero. Analogous to the sedimentation case, the probability of transition from  $\tilde{D}$  to  $\tilde{D}'$  is given by the product of the per-patch conditional probabilities:  $P(\tilde{D}' | \tilde{D}) = \prod_{i=1}^M P(\tilde{D}'(i) | \tilde{D}(i))$ .

The system is assumed to be driven by occasional very large rainfall events that completely fill the ponds, so we neglect small rainfall events. Rainfall can occur with probability  $p$  and is assumed to fill all ponds to their maximum water depth ( $x_i$ ) simultaneously. The transition probability to state  $\tilde{D}''$  resulting from a rainfall event using the vector of water depths after sedimentation and abstraction  $\tilde{D}'$  is  $p$ , if  $\tilde{D}'' = \tilde{X}'$ , and  $1 - p$  if  $\tilde{D}'' = \tilde{D}'$ . If neither condition is satisfied, the transition probability is zero.

We can combine the probabilities of abstraction and rainfall to get the probability that the water depth changes in one time step:

$$P(\tilde{D}'' | \tilde{D}) = P(\tilde{D}'' | \tilde{D}')P(\tilde{D}' | \tilde{D}). \quad (5)$$

The processes of colonisation and extinction are obtained using a modified incidence function model (Hanski 1994). For a waterhole to be colonised, there needs to be a sufficient depth of water in the river for neighbouring waterholes to be connected by a continuous stream of water. We denote the waterhole depth at which connection occurs with the parameter  $d^*$ , and we let the nearest upstream waterhole from a waterhole  $i$  that has a depth less than  $d^*$  be at position  $v$  ( $v \in \{i, i+1, \dots, M\}$ ), and the nearest downstream waterhole with depth less than  $d^*$  be at position  $u$  ( $u \in \{1, \dots, i\}$ ). If there are no waterholes downstream or upstream with depth less than  $d^*$ ,  $u$  and  $v$  respectively are not defined by this notation, and we define the connected region explicitly. If there are waterholes with depth less than  $d^*$  above and below patch  $i$ , then  $u$  and  $v$  are already defined, and the stream around waterhole  $i$  is connected from waterholes  $u+1$  up to  $v-1$ . We define this connected stream segment using the index  $j$ , such that  $j \in \{u+1, \dots, i-1, i, i+1, \dots, v-1\}$ . When all upstream patches have depth greater than  $d^*$  (i.e. when  $v$  is not defined), the connected section is from  $j \in \{u+1, \dots, M-1, M\}$ , and when all downstream patches have depth greater than  $d^*$  (i.e. when  $u$  is not defined), the connected segment is from  $j \in \{1, 2, \dots, v-1\}$ .

For a state  $\tilde{N} = (n_1, n_2, \dots, n_M)$ , we define a random variable  $\tilde{N}'(i)$  representing the occupancy state of the  $i^{th}$  waterhole, which has observed value  $n_i \in \{0, 1\}, \forall i \in \{1, 2, \dots, M\}$ . The probability of colonisation of the  $i^{th}$  waterhole is given by:  $P(\tilde{N}'(i) = 1 | \tilde{N}(i) = 0) = c_i$ ,  $P(\tilde{N}'(i) = 0 | \tilde{N}(i) = 0) = 1 - c_i$ , and  $P(\tilde{N}'(i) | \tilde{N}(i)) = 1$  if neither of these conditions are satisfied. We define  $c_i = \frac{1}{1+(\frac{y_i}{S_i})^2}$ , if  $S_i > 0$ , and  $c_i = 0$

otherwise. The parameter  $y'$  is a fitting parameter that describes the colonisation ability of the species (smaller  $y'$  values correspond to better dispersal (Hanski 1994)) and  $S_i = \sum_{j \neq i} n_j \exp(-\alpha \rho_{ij})(d_j - d^*)$ . The depth of the  $j^{\text{th}}$  waterhole  $d_j$  serves as a measure of patch quality and causes deeper waterholes to have a higher probability of sending colonists to other waterholes. The distance between waterholes  $i$  and  $j$  is  $\rho_{ij}$ ,  $n_j$  is the occupancy of the  $j^{\text{th}}$  waterhole, and  $\alpha$  is a constant that represents the survival rate of migrants. For the illustrative case study in this paper, we make the simplifying assumption that the waterholes are connected whenever there is water in the system ( $d^* = 0$ ).

The probability of extinction of the  $i^{\text{th}}$  waterhole from the new occupancy state  $\tilde{N}'(i)$  is determined by a random variable  $\tilde{N}''(i)$ , which has observed value  $n_i \in \{0, 1\}, \forall i \in \{1, 2, \dots, M\}$ . The probability of extinction is given by  $P(\tilde{N}''(i) = 0 | \tilde{N}'(i) = 1) = e_i$ ,  $P(\tilde{N}''(i) = 1 | \tilde{N}'(i) = 1) = 1 - e_i$ , and  $P(\tilde{N}''(i) | \tilde{N}'(i)) = 1$  if neither of these two conditions are satisfied. We define  $e_i = \min[\frac{\theta}{d_i^\beta}, 1]$ , where  $d_i$  is the depth of the  $i^{\text{th}}$  pond, and  $\theta$  and  $\beta$  are two constants that scale the extinction probability to the depth of water in the waterhole.

The colonisation and extinction probabilities can be used to define the probability that the waterhole occupancy changes in one time step:

$$P(\tilde{N}'' | \tilde{N}) = \prod_{i=1}^M P(\tilde{N}''(i) | \tilde{N}'(i)) P(\tilde{N}'(i) | \tilde{N}(i)). \quad (6)$$

We are now ready to define the transition probability. We can represent a state  $(\tilde{N}, \tilde{D}, \tilde{X})$  as a vector of the vectors  $\tilde{N}$ ,  $\tilde{D}$  and  $\tilde{X}$  defined above. Thus the probability of transition to a new state  $(\tilde{N}'', \tilde{D}'', \tilde{X}')$  is  $P(\tilde{N}'', \tilde{D}'', \tilde{X}' | \tilde{N}, \tilde{D}, \tilde{X})$ , which we can simplify to:

$$\begin{aligned} &= P(\tilde{N}, \tilde{D}, \tilde{X}' | \tilde{N}, \tilde{D}, \tilde{X}) P(\tilde{N}, \tilde{D}'', \tilde{X}' | \tilde{N}, \tilde{D}, \tilde{X}') P(\tilde{N}'', \tilde{D}'', \tilde{X}' | (\tilde{N}, \tilde{D}'', \tilde{X}')) \\ &= P(\tilde{N}'' | \tilde{N}) P(\tilde{D}'' | \tilde{D}) P(\tilde{X}' | \tilde{X}). \end{aligned} \quad (7)$$

Using equations 4, 5, and 6, the value of equation 7 can be determined. The order of events in a timestep is sedimentation-abstraction-rainfall-colonisation-extinction.

**Problem formulation as a Markov Decision Process.** We define the management objective to be to minimise the number of waterholes that do not have fish present each year throughout the management horizon. We have already defined the state space of the system in the metapopulation model section above. For a system of  $M$  patches, we define  $M + 3$  management actions. The metapopulation model allows managers to reduce sedimentation by improving catchment management practices (i.e. decrease  $\delta \rightarrow \delta'$ ), or by direct intervention by dredging one of the  $M$  waterholes to deepen it sufficiently for fish to re-colonise it during the next rainfall event (i.e.  $x_i \rightarrow x_{max}$ ). To manage abstraction, managers may directly police waterholes to prevent illegal water removal (reduce  $\kappa \rightarrow \kappa'$ ). Finally, if no management is required, managers may choose to take no action. It is assumed that managers can carry out only one action per time step, and we do not consider the cost of management actions. The transition probabilities for the MDP assuming no management are defined by the metapopulation model of section 2. When a management action is selected, the transition probabilities have the same form, but the parameter corresponding to the selected action is changed as described earlier in this paragraph.

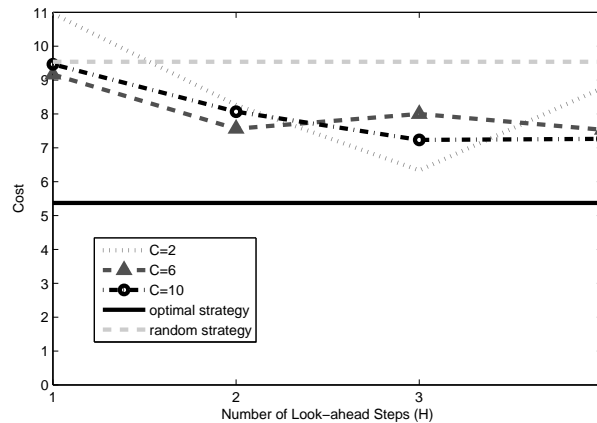
The cost function  $r(s)$  for being in a state  $s$  is simply equal to the total number of waterholes minus the number of occupied waterholes in that state (i.e.  $M - \sum_{i=1}^M n_i, n_i \in \{0, 1\}$ ). The management objective is equivalent to minimising the cost function over the management period. For this paper, we set an infinite time horizon for management, and evaluate the optimal solution until it reaches an equilibrium strategy.

**Algorithm evaluation.** The optimal solution for the problem is given by evaluating equations 1 and 2 using the transition probability from equation 7, where the state  $s = (\tilde{N}, \tilde{D}, \tilde{X})$ , and  $s' = (\tilde{N}'', \tilde{D}'', \tilde{X}')$ . The cost is given by  $r(s, a, t) = r(s)$ , and the action is chosen from the set  $A(s) = 1, 2, \dots, M + 3$ , where each number in  $A(s)$  refers to one of the management options.

To evaluate the Kearns et al. (2002) algorithm, we use equation 3 and a set value of  $H$  and  $C$  to find the optimal action for the current time step. The approximation is used to generate a trajectory of states over the management period by carrying out the algorithm to find the best action, simulating the next state by applying the best action to the current state, then repeating the approximation process to find the best action for the new state. This is repeated for the desired trajectory length.

### 3. RESULTS AND DISCUSSION

**Performance of the approximation algorithm.** The results of carrying out the on-line sampling algorithm are compared against the optimal solution for the 3-patch ( $M = 3$ ) metapopulation in figure 3. The total cost plotted in figure 3 is the total number of instances that a waterhole was unoccupied over the full management period. The solid black line gives the cost of carrying out the optimal solution until an equilibrium policy is reached (11 timesteps). The random solution is the result of randomly choosing a management option for each timestep. Both of these scenarios are evaluated 500 times to obtain the average result in the figure. The remaining three lines on the figure show the results of the on-line approximation algorithm for  $C=2$ , 6, and 10 samples respectively, plotted against the depth of the look-ahead algorithm (horizontal axis). Each approximation was run for the same length of time as the optimal solution (11 timesteps) and the process was repeated 30 times to give the average values plotted in the figure.



**Figure 3.** Performance of the Kearns et al. sparse look-ahead algorithm against the optimal solution and a random management strategy. The trajectories are run until equilibrium is reached (11 steps). The vertical axis shows the average cost over the full 11-step trajectory after multiple trials (see text). Parameters used for this simulation were:  $\delta = 0.05$ ;  $\delta' = 0.0005$ ;  $\kappa = 0.2$ ;  $\kappa' = 0.02$ ;  $p = 0.3$ ;  $\alpha = 0.01$ ;  $y' = 0.1$ ;  $\theta = 0.1$ ;  $\beta = 0.3$ ;  $\gamma = 0.8$ .

The approximation algorithm out-performs a random strategy except when  $C=2$ ,  $H=1$ . In this case the algorithm only chooses one more new state from the next-state distribution than the random strategy and looks no further ahead than the random strategy, so the result should not differ greatly from the random strategy. The approximation was simulated just 30 times (the random strategy is simulated 500 times) so there is likely to be uncertainty in the approximation since so few samples are taken at each step. It is likely that the cost of approximating with just one look-ahead step should in fact be roughly equal to a random strategy (as is the case when  $C=6$  or 10).

Péret & Garcia (2004) found the existence of an optimal number of look-ahead steps when the number of samples  $C$  is fixed. In figure 3, there appears to be a minimum cost at three look-ahead steps in the case where  $C=2$ , although deeper simulation should be carried out to lend this claim more credence. However where computer power is limited, best results for the approximation will be obtained by using 3 look-ahead steps. Interestingly there does not seem to be much gain in using more than two samples to look-ahead, since the results for  $C=6$  and  $C=10$  do not result in average costs that are noticeably lower than the  $C=2$  case. This is probably caused by this particular model, which has a high probability of visiting only a few states in any one time step. It would be very interesting to see if this applies more generally to metapopulation models, as removing the need to sample a large number of states in each timestep will increase the speed of the approximation.

**Lessons for River Management.** The system is highly dependent on the presence of adequate water in the system. This means that whenever a state is reached where the remaining occupied patches have low water, the optimal action is always to prevent water removal from the stream. Where only one patch was occupied and had sufficient water, deepening the nearest patch was the best option. In general, catchment management was only a good option if there was no advantage to preventing abstraction (i.e. remaining waterholes full/empty). However this finding reflects the parameters used in this model (the probability of abstraction is much greater than the probability of sedimentation). The results intuitively make sense, however since the parameters were not based on empirical data, it is hard to generalise management lessons from this hypothetical model. More work needs to be done to properly parameterise the model.

#### 4. CONCLUSIONS

The on-line approximation algorithm of Kearns et al. (2002) provides a relatively efficient way to solve Markov decision problems with very large state spaces. In this paper, we demonstrated that the algorithm performs well against the optimal solution to a small but complex metapopulation problem. We also showed that the approximation out-performs a random strategy. This is the first time this method has been applied to a metapopulation problem and provides a powerful tool for solving complex conservation management problems in the future.

The greatest limitation to the algorithm is that although the algorithm is state space independent, it still has a state space that increases exponentially with the number of look-ahead steps (Kearns et al. 2002). Consequently the algorithm still has a long running time. An interesting question for further research will be to consider at what sized state space it is worth moving from a conventional SDP solution to an approximation algorithm such as this.

Other than being state space independent, the other benefit of this algorithm is that it is calculated on-line. This means that each update is calculated based on the current state of the system and is based on the current generative model. This means that the algorithm could potentially be used to evaluate optimal solutions when the generative model changes with time. This adaptive approach differs from off-line approaches like SDP, which need to evaluate the full optimal solution assuming a static generative model. Another problem for future research is to attempt to use an on-line approach to solve partially observable MDPs in conservation.

This paper introduces a new algorithm to conservation management. This method theoretically enables us to solve optimal management problems with infinitely large state spaces. Greater complexity in decision making models means we can better represent the real world, leading to more realistic decisions and improved conservation management.

#### REFERENCES

- Akcakaya, H. & Ginzburg, L. (1991), Ecological risk analysis for single and multiple populations, *in* A. Seitz & V. Loeschke, eds, 'Species Conservation: A Population-Biological Approach', Birkhauser Verlag, Basel, pp. 73–87.
- Day, J. R. & Possingham, H. P. (1995), 'A stochastic metapopulation model with variability in patch size and position', *Theoretical Population Biology* **48**(3), 333–360.
- Hanski, I. (1994), 'A practical model of metapopulation dynamics', *The Journal of Animal Ecology* **63**(1), 151–162.
- Hanski, I. & Gaggiotti, O. E. (2004), *Ecology, Genetics and Evolution of Metapopulations*, Elsevier Academic Press, Burlington, USA.
- Kearns, M., Mansour, Y. & Ng, A. Y. (2002), 'A sparse sampling algorithm for near-optimal planning in large Markov decision processes', *Machine Learning* **49**, 193–208.
- Levins, R. (1969), 'Some demographic and genetic consequences of environmental heterogeneity for biological control', *Bulletin of the Entomological Society of America* **15**, 237–240.
- Mangel, M. & Clark, C. W. (2000), *Dynamic State Variable Models in Ecology: Methods and Applications*, Oxford series in ecology and evolution, Oxford University Press, New York.
- Ovaskainen, O. & Hanski, I. (2004), Metapopulation dynamics in highly fragmented landscapes, *in* I. Hanski & O. E. Gaggiotti, eds, 'Ecology, Genetics and Evolution of Metapopulations', Elsevier Academic Press, Burlington, USA, pp. 73–103.
- Péret, L. & Garcia, F. (2004), On-line search for solving Markov decision processes via heuristic sampling, *in* R. L. de Mntaras & L. Saitta, eds, '16th European Conference on Artificial Intelligence', IOS press, Valencia, Spain, pp. 530–535.
- Possingham, H., Andelman, S., Noon, B., Trombulak, S. & Pulliam, H. (2001), Making smart conservation decisions, *in* M. E. Soule & G. H. Orians, eds, 'Conservation Biology: Research Priorities for the Next Decade', Island Press, Washington.
- Possingham, H. P. (1996), Decision theory and biodiversity management: how to manage a metapopulation, *in* R. B. Floyd, A. W. Sheppard & P. J. D. Barro, eds, 'Frontiers of Population Ecology', CSIRO, Melbourne, pp. 391–398.
- Puterman, M., ed. (1994), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley and Sons, New York.
- Shea, K., Amarasekara, P., Mangel, M., Moore, J., Murdoch, W., Noonburg, E., Parma, A., Pascual, M., Possingham, H., Wilcox, W. & Yu, D. (1998), 'Management of populations in conservation, harvesting and control', *Trends in Ecology and Evolution* **13**, 371–374.