K-N-MOMDPs: Towards Interpretable Solutions for Adaptive Management

Jonathan Ferrer-Mestres,[1] Thomas G. Dietterich,[2] Olivier Buffet,[3] Iadine Chadès[1]
[1] CSIRO
[2] Oregon State University
[3] INRIA
{jonathan.ferrermestres,iadine.chades}@csiro.au, tgd@cs.orst.edu, olivier.buffet@loria.fr

Abstract

In biodiversity conservation, adaptive management (AM) is the principal tool for decision making under uncertainty. AM problems are planning problems that can be modelled using Mixed Observability MDPs (MOMDPs). MOMDPs tackle decision problems where state variables are completely or partially observable. Unfortunately, MOMDP solutions (policy graphs) are too complex to be interpreted by human decision-makers. Here, we provide algorithms to solve K-N-MOMDPs, where K represents the maximum number of fully observable states and N represents the maximum number of α-vectors. Our algorithms calculate compact and more interpretable policy graphs from existing MOMDP models and solutions. We apply these algorithms to two computational sustainability applications: optimal release of biocontrol agents to prevent dengue epidemics and conservation of the threatened bird species *Gouldian finch*. The methods dramatically reduce the number of states and α-vectors in MOMDP problems without significantly reducing their quality. The resulting policies have small policy graphs (4-6 nodes) that can be easily interpreted by human decision-makers.

Introduction

Determining the best management actions is challenging when critical information is missing. However, urgency and limited access to data require that decisions must be made despite this uncertainty. In conservation and natural resource management, the "best practice" method for managing uncertain systems is adaptive management (AM) or learning by doing (Walters and Hilborn 1976; Keith et al. 2011). Applications of AM include translocation of threatened species (McDonald-Madden et al. 2011), management of migratory shorebirds under climate change (Nicol et al. 2015) or response to epidemics (Shea et al. 2014). AM requires thinking ahead and calculating the consequences of all possible values of the unknown information before deciding the optimal action. AM problems can be modeled and solved as a *planning* problem using Mixed Observable MDP (MOMDP) (Chadès et al. 2012; Nicol et al. 2013) as it is usually assumed that the state of the system (e.g. abundance of a population) is completely observable but the dynamics of a population is unknown under management. We can cast this unknown information on dynamics as a hidden state variable. These potential dynamics are often derived from ecological theory and expert opinions (Williams 2011). In an MOMDP designed to solve an AM problem, the state space is expanded to account for the hidden model state variable. Chadès et al. (2012) have shown that existing MOMDP solvers can benefit by modelling an AM problem as a restricted Mixed Observability problem called a hidden model MDP (hmMDP). Péron et al. (2017) demonstrated that further computational gain could be achieved by initializing any α-vector MOMDP solver to a lower bound of the value function.

Being able to understand and explain optimal policies is a *critical* step in ensuring uptake of decision models in human operated systems such as conservation, environmental management and health. Explainable artificial intelligence (XAI) is an emerging research area that focuses on both interpretability and explainability to assist humans in their decision-making process (Petrik and Luss 2016; Chakraborti et al. 2019). Building XAI is a challenging and open problem (Miller 2019). XAI application domains have included robotics (Miller, Pearce, and Sonenberg 2018), health (Payrovnaziri et al. 2020; Ahmad, Eckert, and Teredesai 2018), marketing (Rai 2020) and criminal justice (Rudin and Ustun 2018; Lakkaraju and Rudin 2017). Some work has addressed interpretability through data visualization (Walsh et al. 2020; Chung et al. 2020) to better understand models and solutions. Other authors emphasize the importance of building interpretable models instead of explaining black box models (Rudin 2019). Here, we contribute to the emerging XAI research by increasing interpretability of MOMDPs motivated by our computational sustainability domains. One way to improve interpretability for decision-making is to simplify models and solutions. To date, algorithms have simplified the solution of POMDPs to policy graph with at most N α-vectors (N-POMDPs) (Dujardin, Dietterich, and Chadès 2015, 2017). More recently, Ferrer-Mestres et al. (2020) propose to solve K-MDPs providing algorithms to abstract the state space for MDPs to at most K states. However, none of these approaches can be directly applied to MOMDPs because MOMDPs are defined over both the completely and partially observable state variables.

Our challenge is two-fold: increase the interpretability of MOMDP models and increase the interpretability of MOMDP solutions. Building on previous work, we define and solve the problem of computing easy-to-interpret MOMDPs as K-N-MOMDP problems, where K represents the maximum number of fully observable states and N represents the maximum number of α-vectors. In doing so, a K-N-MOMDP solution can be represented with a policy graph with at most N vertices and K edges per vertex. We will see that solving K-N-MOMDPs is not as straightforward as applying previously proposed approaches.

We first review relevant concepts for MDPs. We define and solve K-MOMDPs, N-MOMDPs and finally K-N-MOMDPs. We assess our approaches on two computational sustainability case studies: a novel application to release bio-control agents to prevent dengue epidemics and a previously-published application to conserve the threatened Australian *Gouldian finch* (Chadès et al. 2012). We hope that our approach will increase the uptake of AI solutions for real-world AM problems and inspire future research.

MDPs, POMDPs and MOMDPs

MDPs and K-MDPs

Markov Decision Processes (MDPs) represent sequential decision-making problems assuming complete observation of the system state and knowledge of the stochastic dynamics (Puterman 2014). A finite MDP is a tuple $M = \langle S, A, T, r, \gamma\rangle$, where S is a set of states, A is a set of actions, T is a probabilistic transition function, r is a reward function and γ is a discount factor. The solution to an MDP is a function $\pi : S \to A$, called a policy, that maps states to actions. We can evaluate policies based on their expected values given an optimization criterion–hereafter the expected sum of discounted rewards. $V^{\pi}(s)$ is the expected value of executing policy π starting in state s. We denote π^* an optimal policy and V^* the corresponding optimal value function. Solving an MDP is polynomial in time (Papadimitriou and Tsitsiklis 1987).

Definition 1. *(Ferrer-Mestres et al. 2020) Given an MDP M, a K-MDP $M_K = \langle S_K, A, T_K, r_K, \gamma, \phi\rangle$ is an MDP where S_K is a reduced state set of size at most K, A is the original set of actions, T_K is the probability transition function, r_K is the reward function, γ is the discount factor and ϕ is a mapping function from the original MDP state space S to the K-MDP state space S_K.*

An optimal solution for a K-MDP is a policy $\pi^*_K : S_K \to A$ that maximizes the expected sum of discounted rewards. The problem of finding the best reduced state space ($|S_K| \leq K$) is a gap minimization problem

$$gap^* = \min_{S_K \in \mathcal{P}(S),\, |S_K| \leq K} \max_{s \in S} [V^{\pi^*}(s) - V^{\pi^*_K}_{\phi}(s)], \quad (1)$$

between the original optimal MDP policy and the reduced optimal K-MDP policy where $\mathcal{P}(S)$ is the power set of S. Ferrer-Mestres et al. (2020) solve K-MDPs using state abstraction functions and binary search-based algorithms that give a performance guarantee. We propose a new state abstraction function and algorithm to solve K-MOMDPs.

POMDPs and N-POMDPs

Partially Observable MDPs (POMDPs) model sequential decision-making problems when the state of system is partially observable (Sigaud and Buffet 2013). A discrete POMDP is a tuple $\langle S, A, O, T, Z, r, b_0, \gamma\rangle$, where S, A, T, r and γ are defined as in MDPs, O is the set of observations, Z is the observation function and b_0 is an initial probability distribution over states. Belief states, i.e., probability distributions over states, serve as sufficient statistics to summarize the action-observation history (Åström 1965). Solving a POMDP means finding a policy $\pi : B \to A$ mapping belief states ($b \in B$) to actions ($a \in A$). An optimal policy π^* maximizes the expected sum of discounted rewards over an infinite time horizon. For a given belief state b and a given policy π this expected sum is also referred to as the value function $V_\pi(b)$. The optimal value function V^* can be computed using the dynamic programming operator for a POMDP represented as a belief MDP (Bellman 1957), i.e., $\forall b \in B,\ V^*(b) = \max_{a \in A} \left[\sum_{s \in S} r(s,a) b(s) + \gamma \sum_{o'} p(o'|b,a) V^*(b^{ao'})\right]$, where $b^{ao'}$ is the updated belief given that action a was performed and o' is observed.

The infinite time horizon optimal value function can be approximated arbitrarily closely by a PWLC function (Sondik 1971). This value function can be written as the upper envelope of finitely many $|S|$-dimensional hyperplanes as $V(b) = \max_{\alpha \in \Gamma} \alpha \cdot b$, where Γ is a finite set of vectors called α-vectors, one per hyperplane, with each α-vector being associated with an action, and b is the belief represented as a finite vector of size $|S|$. Exact resolution of POMDPs is intractable (Papadimitriou and Tsitsiklis 1987; Madani, Hanks, and Condon 2003).

Definition 2. *(Dujardin, Dietterich, and Chadès 2017) An N-POMDP is a POMDP with an additional parameter N that defines the maximum size of any admissible policy represented by a set of α-vectors at each time step.*

Solving N-POMDPs is a gap minimization problem

$$g^* = \min_{\mathbf{\Gamma}_N \subseteq \Gamma, |\mathbf{\Gamma}_N| \leq N} \max_{\boldsymbol{b} \in B} [V(\boldsymbol{b}) - V_{\mathbf{\Gamma}_N}(\boldsymbol{b})], \quad (2)$$

between the initial value function and the new value function. Solving N-POMDPs is NP-hard (Dujardin, Dietterich, and Chadès 2015, 2017). Our N-MOMDP algorithms build on α-min2-fast and α-min2-p (Dujardin, Dietterich, and Chadès 2017), [two post-processing algorithms] which take as input a policy Γ provided by a third-party POMDP solver (Kurniawati, Hsu, and Lee 2008) and select the best combination of N α-vectors from Γ.

MOMDPs

Mixed Observability MDP (MOMDP) (Ong et al. 2010) are represented as a tuple $\langle X, Y, A, O, T_x, T_y, Z, r, b_0, \gamma\rangle$:

- $S = X \times Y$ is the factored set of states with X representing the completely observable components and Y representing the partially observable components.
- A is the finite set of actions;

- $O = O_x \times O_y$ is set of observations with $O_x = X$ the completely observable component, and O_y the set of observations of the partially observable variables;
- $T_x(x, y, a, x')$ and $T_y(x, y, a, x', y')$ are the probabilistic transition functions for X and Y respectively;
- Z is the observation function with $p(o'_x, o'_y|a, x', y')$ the probability of observing o'_x, o'_y after performing action a;
- $r(x, y, a)$ defines the immediate reward received from implementing action a in state (x, y);
- γ is a discount factor and b_0 is an initial belief.

A MOMDP policy $\pi : X \times B_Y \rightarrow A$ maps a system state (x, b) to an action. A policy can be assessed through its value function, $\forall x, b \in X \times B_Y$, $V_\pi(x, b) = E[\sum_{t=0}^{\infty} \gamma^t R(x_t, b_t, \pi(x_t, b_t))|x, b]$. We then have $\pi^* = \text{argmax}_\pi V_\pi(x_0, b_0)$. The PWLC property applies to $V_\pi(x, \cdot)$, i.e., there exists a finite set Γ_x of $|Y|$-tuples (α-vectors) to represent V as $V_\pi(x, b) = \max_{\alpha \in \Gamma_x} b \cdot \alpha$, where $b \cdot \alpha = \sum_{y \in Y} b(y)\alpha(y)$. In conservation, AM problems can be modelled as a *hidden model MDP* (hmMDP), i.e., an MOMDP where the partially observable state variable corresponds to the hidden model (Ong et al. 2010; Chadès et al. 2012). In this setting, managers can perfectly observe the state of the studied system but are uncertain about the dynamics of the system, which will not change over time. T_y becomes the identity matrix, i.e., $p(y'|y) = 1$ if $y = y'$ and 0 otherwise. This formulation assumes that the real but unknown MDP model y_r is one of a finite set Y of known models. Y is also independent from X, i.e., $p(y'|x, y, a, x') = p(y'|y)$. While we propose to solve K-N-MOMDPs, our motivation is to make adaptive management (hmMDPs) interpretable. Our approach applies to general MOMDPs.

Solving K-MOMDPs

Reducing the size of the state space increases the interpretability of both MOMDP models and solutions. Our approach requires an initial MOMDP model and policy from which it builds a K-MOMDP.

Definition 3. *A K-MOMDP is an MOMDP with an additional parameter K that constrains the number of fully observable states $x \in X$ to be at most K.*

Given an MOMDP $\mathcal{M}$, let us define the K-MOMDP $\mathcal{M}_\mathcal{K} = \langle X_K, Y, A, O, T_{x_K}, T_y, Z, r_K, b_0, \gamma, \hat{\phi} \rangle$ where:

- $X_K = \{\hat{\phi}(x)|x \in X\}$ the abstract fully observable state space component with $|X_K| \leq K$ and $\hat{\phi}$ a function that maps elements of X to X_K. The inverse function $\hat{\phi}^{-1}(x_K)$ maps elements of X_K to its constituent states in the original MOMDP; Y defines the partially observable components;
- A is the same set of actions as in the original MOMDP;
- $O = O_{x_K} \times O_y$ is the set of observations with $O_{x_K} = X_K$ the completely observable component, and O_y the set of observations of the hidden variables;
- $T_{x_K}(x_K, y, a, x'_K)$ gives the probability that the fully observable state variable takes the value x'_K if action a is performed in state (x_K, y); $T_y(x_K, y, a, x'_K, y')$ gives the probability that the value of the partially observable state variable changes from y to y' given that action a is performed in state (x_K, y) and x_K transitions to x'_K; $T_{x_K} = \sum_{x \in \hat{\phi}^{-1}(x_K)} \sum_{x' \in \hat{\phi}^{-1}(x'_K)} T_x(x, y, a, x')\omega(x)$, and $T_y = \sum_{x \in \hat{\phi}^{-1}(x_K)} \sum_{x' \in \hat{\phi}^{-1}(x'_K)} T_x(x, y, a, x', y')\omega(x)$, where $\omega(s) \in [0, 1]$ is a probability distribution over the original fully observable variables that aggregate to x_K:
$$\forall x_K \in X_K, \Big(\sum_{x \in \hat{\phi}^{-1}(x_K)} \omega(x) \Big) = 1;$$
- Z is the observation function with $p(o_{x'_K}, o_{y'}|a, x'_K, y')$ being the probability of observing $o_{x'_K}, o_{y'}$ after performing action a and transitioning to (x'_K, y'); in MOMDPs, we assume the variable X' is perfectly observable, so we have $p(o_{x'_K}|a, x'_K, o_{y'})$ = 1 if $o_{x'_K} = x'_K$ and 0 otherwise;
- $r_K(x_K, y, a)$: $X_K \times Y \times A \rightarrow [0, R_{max}]$ is the reward function defined as a weighted sum over X, i.e., $r_K(x_K, y, a) = \sum_{x \in \hat{\phi}^{-1}(x_K)} r(x, y, a)\omega(x)$;
- b_0 and γ are defined as for MOMDPs.

The optimal solution for a K-MOMDP is a policy $\pi^*_{X_K}$: $X_K \times B_Y \rightarrow A$ that maximizes the expected sum of discounted rewards. The optimal policy can be applied to the original MOMDP using the function $\hat{\phi}$ and its performance can be evaluated through:

$$V_{\hat{\phi}}^{\pi^*_K}(x_K, b) = E[\sum_{t=0}^{\infty} \gamma^t r(x_{K_t}, b_t, \pi^*_K(\hat{\phi}(x_{K_t}), b_t))|x_K, b]. \tag{3}$$

Solving K-MOMDPs with state space $|X_K| \leq K$ is a gap minimization problem between the optimal MOMDP policy π^* and the optimal K-MOMDP policy π^*_K:

$$g^*_K = \min_{\substack{X_K \in \mathcal{P}(X), \\ |X_K| \leq K}} \max_{x \in X} [V^{\pi^*}(x, b) - V_{\hat{\phi}}^{\pi^*_K}(x_K, b)], \tag{4}$$

where $\mathcal{P}(X)$ is the power set of X. We now propose an algorithm based on a state abstraction function and binary search to solve K-MOMDPs. Our K-MOMDP algorithm calls a *BUILD-K-MOMDP* procedure once X is reduced (Alg. 1). Given an MOMDP $\mathcal{M}$, an abstract state space $S = X_K \times Y$ and $\hat{\phi}$, Alg. 1 computes the weights $\omega(s)$, the reward r_K and transition function T_{x_K}.

Algorithm 1 *BUILD-K-MOMDP*

```
Require: $\mathcal{M} = \langle X, Y, A, O, T_x, T_y, Z, r, H, b_0, \gamma \rangle, X_K, \hat{\phi}$
$\forall x \in \hat{\phi}^{-1}(x_K), \omega(x) \leftarrow$ computeWeights$(\hat{\phi}, x_K)$
$r_K \leftarrow \sum_{x \in \hat{\phi}-1(x_K)} r(x, y, a)\omega(x)$
$T_{x_K} \leftarrow \sum_{x \in \hat{\phi}^{-1}(x_K)} \sum_{x' \in \hat{\phi}^{-1}(x'_K)} T_{x_K}(x_K, y, a, x'_K)\omega(x)$
return $M_K \leftarrow \langle X_K, Y, A, O, T_{x_K}, T_y, Z, r_K, H, b_0, \gamma, \hat{\phi} \rangle$
```

The K-MOMDP algorithm

We propose a new approximate transitive state abstraction function defined for MOMDPs, that, given the optimal value function $V^*(x, b)$, solves the K-MOMDP problem by sorting states into bins (i.e. groups). Approximate state abstractions allow greater degrees of aggregation between states while exact state abstraction functions aggregate states that are exactly equal given a metric (Li, Walsh, and Littman 2006; Dean and Givan 1997). A transitive function means that given a predicate $p(x, x')$ (a binary relation between fully observable state components), then $p(x_1, x_2) \wedge p(x_2, x_3) \implies p(x_1, x_3)$. Transitivity is a desirable property to write efficient state abstraction algorithms because we can reduce many calculations. (Ferrer-Mestres et al. 2020).

For any pair $i, j \in X$, our approximate transitive MOMDP state abstraction function $\hat{\phi}_{a^*_d}$ satisfies:

$$\begin{gathered}\hat{\phi}_{a^*_d}(i) = \hat{\phi}_{a^*_d}(j) \Leftrightarrow \\ \forall b \in \hat{B}_Y \left(a^*_{(i,b)} = a^*_{(j,b)} \wedge \left\lceil \frac{V^*(i,b)}{d} \right\rceil = \left\lceil \frac{V^*(j,b)}{d} \right\rceil \right), \end{gathered} \tag{5}$$

for $0 < d \leq \text{VMAX}$, where $a^*_{(i,b)}$ and $a^*_{(j,b)}$ are the optimal actions to implement in states (i, b) and (j, b) respectively, and $\hat{B}_Y$ is a finite sample set of the continuous belief space B_Y. According to Eq. 5, two fully observable state components i and j can be aggregated if they belong to the same bin and their optimal actions are the same for all beliefs $b \in \hat{B}_Y$. This state abstraction function is the MOMDP formulation of the MDP state abstraction function proposed by (Abel et al. 2018; Ferrer-Mestres et al. 2020).

Similar to the MDP case, our MOMDP state abstraction function $\hat{\phi}_{a^*_d}$ has a value loss that scales in accordance with d (and R_{max}):

$$\max_{x\in X} \max_{b\in B_Y} V^{\pi^*}(x, b) - V^{\pi^*_K}_{\hat{\phi}_{a^*_d}}(x, b) \leq \frac{2dR_{max}}{(1-\gamma)^2}. \tag{6}$$

We propose the K-MOMDP algorithm to find the minimum value of d that returns an abstract set X_K, where $|X_K| \leq K$, given a policy $\mathbf{\Gamma}$, a sampled set of beliefs $\hat{B}_Y$, a precision parameter p_t and the abstraction function $\hat{\phi}_{a^*_d}$. Alg. 2 performs a binary search on d by setting the upper and lower bounds d^+ and d^- to VMAX and 0 respectively. For all pairs (x, b), *bindings* contains the *ceil* values of their optimal values and optimal actions (Line 3). *unique* returns an abstracted fully observable state space component X_K by grouping those states that belong to the same bin. Finally, bounds are updated. Alg. 2 has time complexity $O(|X||\hat{B}_Y| \log(\frac{\text{VMAX}}{p_t}))$ due to the binary search and the *unique* procedure (linear in $|X||\hat{B}_Y|$).

Proposition 1. *Alg. 2 solves the K-MOMDP problem within precision p_t in a finite number of iterations. The optimal value function derived from the solution $\mathcal{M}_K$ has suboptimality bounded polynomially in d.*

Proof. This follows from the binary search and Eq. 6. □

Algorithm 2 K-MOMDP algorithm

```
Require: $\mathcal{M}, \mathbf{\Gamma}, \hat{B}_Y, K, \hat{\phi}, p_t, d^+ = \text{VMAX}, d^- = 0$
1: while $p > p_t$ do
2:     $p \leftarrow (p^+ - p^-)$ ; $d \leftarrow (d^- + \frac{d^+ - d^-}{2})$
3:     $bindings \leftarrow [\lceil \frac{V^*(x,b)}{d} \rceil, \pi^*(x, b)]_{(x,b)\in X\times\hat{B}_Y}$
4:     $X_K \leftarrow unique(bindings)$
5:     if $|X_K| \leq K$ then $d^+ \leftarrow d$
6:     else $d^- \leftarrow d$
7: return $\mathcal{M}_K \leftarrow$ BUILD-K-MOMDP$(\mathcal{M}, X_K, \hat{\phi})$
```

Solving N-MOMDPs

Definition 4. *An N-MOMDP is an MOMDP with an additional parameter $N \geq |X|$ that constrains admissible policies to at most N $|Y|$-dimensional α-vectors.*

Proposition 2. *The N-MOMDP problem is NP-hard.*

Proof. Any N-MOMDP can be reduced to solving an N-POMDP. N-POMDPs are NP-hard (Dujardin, Dietterich, and Chadès 2017). Solving an N-MOMDP requires solving an N-POMDP. N-MOMDPs are NP-hard. □

While it can be tempting to directly apply algorithms for solving N-POMDPs to solve N-MOMDPs, we have to consider the factored representation of MOMDPs and the additional constraint that at least one α-vector is required per visible state x. Solving an N-MOMDP is a gap minimization problem between the original policy $\mathbf{\Gamma}$ and a new policy $\mathbf{\Gamma}_N$ composed of only N α-vectors, where $\mathbf{\Gamma} = \{\Gamma_x\}_{x\in X}$ and $\mathbf{\Gamma}_N = \{\Gamma_{N,x}\}_{x\in X}$ with i) for each x, $\Gamma_{N,x} \subseteq \Gamma_x$ and $|\Gamma_{N,x}| \geq 1$, and ii) $\sum_x |\Gamma_{N,x}| \leq N$. Formally,

$$g^*_N = \min_{\substack{\mathbf{\Gamma}_N \subseteq \Gamma \\ |X| \leq |\mathbf{\Gamma}_N| \leq N}} \max_{\substack{x\in X \\ \boldsymbol{b}\in B_Y}} [V(x, \boldsymbol{b}) - V_{\mathbf{\Gamma}_N}(x, \boldsymbol{b})], \tag{7}$$

with $V(x, \boldsymbol{b})$=$\max_{\alpha\in\Gamma_x} \alpha \cdot \boldsymbol{b}$ and $V_{\mathbf{\Gamma}_N}(x, \boldsymbol{b})$=$\max_{\alpha\in\Gamma_{N,x}} \alpha \cdot \boldsymbol{b}$, where $\Gamma_x \subseteq \mathbf{\Gamma}$ and $\Gamma_{N,x} \subseteq \mathbf{\Gamma}_N$.

We now propose three new algorithms ($\hat{\alpha}$-min-fast, $\hat{\alpha}$-min-p and $\hat{\alpha}$-min-pruning) to solve N-MOMDPs, that accommodate the differences between N-MOMDPs and N-POMDPs.

The $\hat{\alpha}$-min-fast N-MOMDP algorithm

Given a policy $\mathbf{\Gamma}$, let $\hat{s}$ be a positive semi-definite function such that $\hat{s}(\alpha, \tilde{\alpha}) = \max_{b\in\hat{B}_Y(\alpha)}(\alpha \cdot b - \tilde{\alpha} \cdot b)$, where $\alpha, \tilde{\alpha} \in \Gamma_x$ and $\hat{B}_Y(\alpha)$ is the set of vertices of the polytope subspace of B_Y where α dominates the other α-vectors. Solving the following problem provides an upper bound for Eq. 7:

$$\min_{\substack{\mathbf{\Gamma}_N \subseteq \Gamma \\ |X| \leq |\mathbf{\Gamma}_N| \leq N}} \max_{\alpha\in\Gamma_x} \min_{\tilde{\alpha}\in\Gamma_{N,x}} \hat{s}(\alpha, \tilde{\alpha}). \tag{8}$$

To solve Problem 8 we propose a pure integer linear program with $|\mathbf{\Gamma}|$ 0-1 decision variables and $|\mathbf{\Gamma}|+|X|+1$ constraints.

$$
\begin{aligned}
f: \quad & \min \sum_{x\in X}\sum_{\alpha\in\Gamma_x} z_{x\alpha} \\
\text{s.t.} \quad & \sum_{x\in X}\sum_{\alpha\in\Gamma_x} c_{\alpha,\alpha'} z_{x\alpha} \geq 1, \alpha' \in \Gamma_x \\
& \sum_{x\in X}\sum_{\alpha\in\Gamma_x} z_{x\alpha} \leq N \\
& \forall x \in X, \sum_{\alpha\in\Gamma_x} z_{x\alpha} \geq 1 \qquad (ILP_{MIN})
\end{aligned}
\tag{9}
$$

ILP_{MIN} encodes a set of α-vectors of size at most N where i) $z_{x\alpha}$ takes value 1 if both $\alpha' \in \Gamma_x$ and $\alpha \in \mathbf{\Gamma}_{N_x}$ hold; ii) $z_{x\alpha}$ can take value 1 if, for any given $\alpha' \in \Gamma_x$, $\hat{c}_{\alpha,\alpha'} = 1$, so α can be added to $\mathbf{\Gamma}_N$; iii) There are at most N α-vectors in $\mathbf{\Gamma}_N$ and iv) There is at least one α-vector per each fully observable value $x \in X$. ILP_{MIN} involves more constraints than the ILP devised to solve N-POMDPs to account for the factored representation of MOMDPs and the need to have one α-vector per visible state variable. Alg. 3 performs a binary search on the decision version of the N-MOMDP problem, using the function $\hat{s}(\alpha, \tilde{\alpha})$ and the linear program ILP_{MIN}. Alg. 3 searches for the smallest ϵ value that allows a valid N-MOMDP solution. For each $\Gamma_x \subseteq \mathbf{\Gamma}$ and for each pair $\alpha, \alpha' \in \Gamma_x$, if $\hat{s}(\alpha, \alpha') \leq \epsilon$, then α is a possible candidate to be included in $\mathbf{\Gamma}_N$ (i.e. $c_{\alpha,\alpha'} = 1$). Given the adjacency matrix C of $c_{\alpha,\alpha'}$ and N, ILP_{MIN} returns a set of $\alpha \in \mathbf{\Gamma}_N \iff z_{x\alpha} = 1$.

Algorithm 3 $\hat{\alpha}$-min-fast N-MOMDP

```
Require: $\mathbf{\Gamma}, \hat{B}_Y, p_t, |X| \leq N \leq 1, \epsilon^+ = \epsilon_{ub}, \epsilon^- = 0$
while $\delta \geq p_t$ do
    $\delta \leftarrow (\epsilon^+ - \epsilon^-)$ ; $\epsilon \leftarrow \frac{\epsilon^+ + \epsilon^-}{2}$
    $\forall \Gamma_x, \forall(\alpha, \alpha') \in \Gamma_x,\ C(\alpha, \alpha') \leftarrow \mathbb{1}_{\hat{s}(\alpha,\alpha')\leq\epsilon}$
    if $\mathbf{\Gamma}_N \leftarrow ILP_{MIN(C,N)}$ has solution then $\epsilon^+ \leftarrow \epsilon$
    else $\epsilon^- \leftarrow \epsilon$
return $\mathbf{\Gamma}_N, \epsilon$
```

Proposition 3. *Alg. 3 solves the N-MOMDP problem (8) within precision p_t in a finite number of iterations.*

Proof. We are improving on *α-min-2-fast* by defining $\hat{s}$ instead of s because $\alpha - \tilde{\alpha}$ is an α-vector, i.e., an hyperplane, and will thus have at least one maximum in a corner of $\hat{B}_Y(\alpha)$ (convex polytope). The binary search algorithm applied to a continuous variable ϵ using a given precision parameter p_t requires $\log(\frac{\epsilon_{ub}}{p_t})$ iterations and the ILP_{MIN} can be solved using Branch and Bound for a 0-1 pure integer linear program. There are $|\mathbf{\Gamma}|$ variables an $|\mathbf{\Gamma}|+|X|+1$ constraints. Therefore, the complexity of ILP_{MIN} is $O(2^{|\mathbf{\Gamma}|})$. Thus, Alg. 3 has time complexity in $O(\log(\frac{\epsilon_{ub}}{p_t})2^{|\mathbf{\Gamma}|})$, due to the binary search and the Branch&Bound algorithm employed to solve the 0-1 integer linear program ILP_{MIN}. □

The $\hat{\alpha}$-min-p N-MOMDP algorithm

Using Alg. 3, the real gap between V and $V_{\mathbf{\Gamma}_N}$ is approximated by an upper bound (ϵ). To increase performance of our solution, a logical first step is to adapt the α-min-2-p algorithm (Dujardin, Dietterich, and Chadès 2017) designed to solve N-POMDPs with a better approximation of the optimal gap (eq. 10). However, this also requires careful consideration of the factored properties of N-MOMDPs. Let us represent the value function V by both $\mathbf{\Gamma}$ and a finite set $\mathbf{\Delta}$ of β-points, where $\mathbf{\Delta} = \{\Delta_x\}_{x\in X}$ and a β-point is of the form $\langle b, V(x,b)\rangle$, where $b \in B_{\mathbf{\Delta}}$, with $B_{\mathbf{\Delta}}$ being a finite subset of B_Y, and $V(x,b) = \max_{\alpha\in\Gamma_x} \alpha \cdot b$ being the optimal value at (x,b). Let us define $\hat{s}'$ to be a function such that $\hat{s}'(\tilde{\alpha}, \beta) = \hat{s}'(\tilde{\alpha}, \langle b, V(x,b)\rangle) = \tilde{\alpha} \cdot b - V(x,b)$, where $\tilde{\alpha} \in \Gamma_x$. We decrease the optimal gap (eq. 7) by adding β-points to $\mathbf{\Delta}$ as necessary:

$$
g_N^*(\mathbf{\Delta}) = \min_{\substack{\mathbf{\Gamma}_N\subseteq\mathbf{\Gamma}\\ |X|\leq|\mathbf{\Gamma}_N|\leq N}} \max_{\beta\in\mathbf{\Delta}} \min_{\tilde{\alpha}\in\mathbf{\Gamma}_N} \hat{s}'(\tilde{\alpha}, \beta), \tag{10}
$$

where $1 \leq N$. The main challenge is to build $\mathbf{\Delta}$ such that the real gap between V and $V_{\mathbf{\Gamma}_N}$ is minimized. Alg. 4 iteratively adds β-points to $\mathbf{\Delta}$ corresponding to the biggest gap between the original value function V and the reduced value function $V_{\mathbf{\Gamma}_N}$ until the current gap $g_r(V, V_{\mathbf{\Gamma}_N})$ and the optimal gap $g_N^*(\mathbf{\Delta})$ are close enough given a precision parameter p_t. Alg. 4 initializes $\mathbf{\Delta} \leftarrow \{\Delta_x\}_{x\in X}$, with the β-points corresponding to the corners of the belief simplexes $\hat{B}_Y$: $\hat{b}_1$=$(1,\ldots,0),\ldots,\hat{b}_{|Y|}$=$(0,\ldots,1)$, so that, $\forall_{x\in X}$, $\Delta_x = \{\beta_1 = \langle\hat{b}_1, V(x,\hat{b}_1)\rangle, \ldots, \beta_{|Y|} = \langle\hat{b}_{|Y|}, V(x,\hat{b}_{|Y|})\rangle\}$, where $\Delta_x \subseteq \mathbf{\Delta}$. The algorithm runs until a precision criterion is reached. Line 2 performs α-min-β, which is the $\hat{\alpha}$-min-fast algorithm with three differences: i) pairs $\alpha, \alpha' \in \Gamma_x$ are now pairs (α, β) where $\alpha \in \Gamma_x$ and $\beta \in \Delta_x$, ii) $\hat{s}'(\tilde{\alpha}, \beta)$ replaces $\hat{s}(\alpha, \tilde{\alpha})$, and iii) $\mathbf{\Delta}$ is required. Notice that the returned value $g_{\mathbf{\Delta}}$ is ϵ in $\hat{\alpha}$-min-fast N-MOMDP. Alg. 4 computes, for all $x \in X$, the beliefs b_x^* that maximize the gap between $V(x,b)$ and $V_{\mathbf{\Gamma}_N}(x,b)$ (line 3). Finally, b_x^* is added to Δ_x as a β-point (line 6).

Algorithm 4 $\hat{\alpha}$-min-p N-MOMDP algorithm

```
Require: $\Gamma, \hat{B}_Y, p_t, N, \mathbf{\Delta}, \delta = \infty, g_{ub} = \infty$
1: while $\delta \geq \frac{p_t}{2}$ do
2:     $(\mathbf{\Gamma}_N, g_{\mathbf{\Delta}}) \leftarrow \alpha$-min-$\beta(\Gamma, \mathbf{\Delta}, \frac{p_t}{2}, N)$
3:     $\forall_{x\in X},\ b_x^* \leftarrow \mathrm{argmax}_{b\in\hat{B}_Y}(V(x,b) - V_{\mathbf{\Gamma}_N}(x,b))$
4:     $g_{ub} \leftarrow \min(g_{ub}, \max_{x\in X}(V(x,b_x^*) - V_{\mathbf{\Gamma}_N}(x,b_x^*)))$
5:     $\delta \leftarrow g_{ub} - g_{\mathbf{\Delta}}$
6:     $\forall_{\Delta_x},\ \Delta_x \leftarrow \Delta_x \cup \{\beta = \langle b_x^*, V(x,b_x^*)\rangle\}$
7: return $\mathbf{\Gamma}_N, g_{ub}$
```

Proposition 4. *Alg. 4 solves the N-MOMDP Problem (8) within precision p_t in a finite number of iterations.*

Proof. We adapt the proof from (Dujardin, Dietterich, and Chadès 2017). Alg. 4 provides a better approximation of the optimal gap g_N^* by solving $g_N^*(\mathbf{\Delta})$, so $g_N^*(\mathbf{\Delta}) \leq g_N^*$ because $\mathbf{\Delta} \subseteq \{(b, V(x,b)), b \in \hat{B}_Y\}$ and beliefs in Δ are a subset of $\hat{B}_Y$. Procedure α-min-$\beta(\Gamma, \mathbf{\Delta}, \frac{p_t}{2}, N)$ provides an optimal solution to problem (8) with $g_{\mathbf{\Delta}} \leq g^*_N(\mathbf{\Delta}) + \frac{p_t}{2}$, and, by definition, this leads to $g_{\mathbf{\Delta}} \leq g^*_N + \frac{p_t}{2}$. Optimal gap is $g^*_N \leq g_r(V, V_{\mathbf{\Gamma}_N})$ and in line 4 g_{ub} is set to the real gap $\max_{x \in X} V(x, b^*) - V_{\mathbf{\Gamma}_N}(x, b^*)$, then, $g^*_N \leq g_{ub}$. At last iteration, $\delta \leq \frac{p_t}{2}$ and $\delta = g_{ub} - g_{\mathbf{\Delta}}$, so $g_{ub} - g_{\mathbf{\Delta}} \leq \frac{p_t}{2}$. Then, $g_{ub} \leq g_{\mathbf{\Delta}} + \frac{p_t}{2} \leq (g^*_N + \frac{p_t}{2}) + \frac{p_t}{2}$. Finally we have $g^*_N \leq g_{ub} \leq g^*_N + p_t$. At each iteration, Alg. 4 provides a reduced policy $\mathbf{\Gamma}_N$. If $\mathbf{\Gamma}_N$ is the same as a previously-generated $\mathbf{\Gamma}_N$, then $\delta \leftarrow 0$. So, in the worst case, $\mathbf{\Gamma}_N$ will be equal at each iteration to every possible combination of N α-vectors before $\delta < \frac{p_t}{2}$. Then, the number of iterations in Alg. 4 is bounded by $|\mathbf{\Gamma}|^N$. α-min-β has same time complexity as Alg. 3 as detailed in the proof of Proposition 3, being the time complexity $O(\log(\frac{\epsilon_{ub}}{p_t})2^{|\mathbf{\Gamma}|})$. Overall, the complexity of Alg. 4 is $O(|\mathbf{\Gamma}|^N \log(\frac{\epsilon_{ub}}{p_t})2^{|\mathbf{\Gamma}|})$. □

The $\hat{\alpha}$-pruning N-MOMDP algorithm

Algs. 3 and 4 require solving a pure integer linear program. We relax this requirement and propose a new algorithm (Alg. 5) to iteratively prune α-vectors that are equivalent given a metric. Any pair of α-vectors $\alpha, \alpha' \in \Gamma_x$ are equivalent if the following predicate is satisfied:

$$\hat{\phi}_{\alpha_d}(\alpha) = \hat{\phi}_{\alpha_d}(\alpha') \implies \left\lceil \frac{\alpha}{d} \right\rceil = \left\lceil \frac{\alpha'}{d} \right\rceil, \qquad (11)$$

for $0 < d \leq \text{VMAX}$, where VMAX represents the maximum optimal value. This transitive predicate offers a new unique minimal policy $\mathbf{\Gamma}_N$ for which all α-vector pairs that satisfy the predicate belong to the same group. A set of α-vectors belong to the same group if they belong to the same bin as defined by Eq. 11. Alg. 5 performs a binary search on d and it computes the bin indexes of all α-vectors given d, (line 3) and store them in *bindings*. Function *unique* returns the α-vectors that belong to the same bin υ in the same set Γ_x (line 4). There is a set Υ_x of bins υ per each Γ_x. Each bin υ in Υ_x contains one or more α-vectors. Let's call this set the *alphas* of υ: $alphas(\upsilon)$. The α-vectors in $\mathbf{\Gamma}_N$ are selected from each bin υ (line 5) such that $\tilde{\alpha} \leftarrow \text{argmin}_{\alpha_\upsilon \in alphas(\upsilon)} \max_{b \in \hat{B}_Y}(V(x,b) - \alpha_\upsilon \cdot b)$, where $\tilde{\alpha} \in alphas(\upsilon)$, $\tilde{\alpha}, \alpha_\upsilon \in \Gamma_x$ and $\upsilon \in \Upsilon_x$. Remaining α-vectors are removed from $\mathbf{\Gamma}_N$. Finally, bounds are updated and the algorithm continues until precision criterion is reached.

Algorithm 5 $\hat{\alpha}$-pruning N-MOMDP algorithm

```
Require: $\mathbf{\Gamma}, \hat{B}_Y, p_t, N, d^+ = \text{VMAX}, d^- = 0$
1: while $p \geq p_t$ do
2:     $p \leftarrow (d^+ - d^-)$ ; $d \leftarrow (d^- + \frac{d^+ - d^-}{2})$
3:     $\forall \Gamma_x, \forall \alpha \in \Gamma_x,\ bindings(x, \alpha) \leftarrow \lceil \alpha / d \rceil$
4:     $\forall_{x \in X}, \Upsilon_x \leftarrow \Upsilon_x \cup \upsilon = unique(bindings(x, \cdot))$
5:     $\forall_{\Upsilon_x}, \forall_{\upsilon \in \Upsilon_x},\ \mathbf{\Gamma}_N \leftarrow$
   $\mathbf{\Gamma}_N \cup \text{argmin}_{\alpha_\upsilon \in alphas(\upsilon)} \max_{b \in \hat{B}_Y}(V(x,b) - \alpha_\upsilon \cdot b)$
6:     if $|\mathbf{\Gamma}_N| \leq N$ then $d^+ \leftarrow d$
7:     else $d^- \leftarrow d$
8: return $\mathbf{\Gamma}_N$
```

Proposition 5. *Alg. 5 solves the N-MOMDP problem within precision p_t in a finite number of iterations.*

Proof. This proposition is a direct consequence of the binary search algorithm, Eqs. (7) and (11). Alg. 5 has time complexity $O(|\mathbf{\Gamma}| \log(\frac{\text{VMAX}}{p_t}))$ due to the binary search and the *unique* procedure, which is linear in $|\mathbf{\Gamma}|$. □

Solving K-N-MOMDPs

Definition 5. *A K-N-MOMDP is an MOMDP with two additional parameters, K and N, that constrain admissible policies to be defined (i) over at most K abstract visible states and (ii) with at most N α-vectors.*

Proposition 6. *The K-N-MOMDP problem is NP-hard.*

Proof. Solving a K-N-MOMDP requires solving an N-MOMDP. An N-MOMDP can be reduced to solving an N-POMDP. N-POMDPs are NP-hard. K-N-MOMDPs are NP-hard. □

Solving a K-N-MOMDP is a gap minimization problem between the original policy and a new policy composed of only K visible states and N α-vectors:

$$g^*_{K-N} = \min_{\substack{\mathbf{\Gamma}_N \subseteq \mathbf{\Gamma}, |\mathbf{\Gamma}_N| \leq N \\ X_K \in \mathcal{P}(X), |X_K| \leq K}} \max_{\substack{x \in X \\ \boldsymbol{b} \in B_Y}} [V(x, \boldsymbol{b}) - V_{X_K, \mathbf{\Gamma}_N}(x, \boldsymbol{b})]. \qquad (12)$$

Minimizing over both the $\mathbf{\Gamma}_N \subseteq \mathbf{\Gamma}$ and $X_K \in X$ at the same time is challenging. Our approach aims to first minimize over the set of visible states $X_K \in X$ and then over the set of α vectors, $\mathbf{\Gamma}_N \subseteq \mathbf{\Gamma}$, solving Problems 4 and 7.

Algorithm to solve K-N-MOMDPs

Our proposed algorithm builds a K-MOMDP given an MOMDP $\mathcal{M}$ (line 1). Then we use an MOMDP solver (e.g. SARSOP (Kurniawati, Hsu, and Lee 2008)) to produce a good quality policy (line 2). Finally the policy is reduced to N α-vectors by using any of our proposed N-MOMDP algorithms (line 3) and returns a K-MOMDP $\mathcal{M}_{\mathcal{K}}$ and a reduced policy $\mathbf{\Gamma}_N$.

Algorithm 6 K-N-MOMDP algorithm

```
Require: $\mathcal{M}, \mathbf{\Gamma}, \hat{B}_Y, K, N, p_{K_t}, p_{N_t}$
1: $\mathcal{M}_{\mathcal{K}} \leftarrow$ solve-K-MOMDP($\mathcal{M}, \mathbf{\Gamma}, \hat{B}_Y, K, \hat{\phi}, p_{K_t}$)
2: $\mathbf{\Gamma} \leftarrow$ MOMDP-solver($\mathcal{M}_{\mathcal{K}}$)
3: $\mathbf{\Gamma}_N \leftarrow$ solve-N-MOMDP($\mathbf{\Gamma}, \hat{B}_Y, p_{N_t}, N$)
4: return $\mathcal{M}_{\mathcal{K}}, \mathbf{\Gamma}_N$
```

Proposition 7. *Alg. 6 solves the K-N-MOMDP problem in a finite number of iterations.*

Proof. This follows from proofs for algorithms $\hat{\alpha}$-min-fast, $\hat{\alpha}$-min-p and $\hat{\alpha}$-pruning N-MOMDP (Algs. 3, 4, 5) and the proof for K-MOMDP algorithm (Alg. 2). □

Experimental Results

To assess the effectiveness and practicality of our algorithms, we evaluated them on two adaptive management problems that arise in conservation and biosecurity. We ran our experiments on an i7-8650U, 1.90 GHz, with 16 GB and Ubuntu 18.04. Experiments were conducted using the *MDP-Toolbox* (Chadès et al. 2014), MATLAB (R2020a), *MATLAB Optimization Toolbox* and MO-SARSOP solver (Kurniawati, Hsu, and Lee 2008), a version of SARSOP algorithm adapted for MOMDPs, which computes a policy $\mathbf{\Gamma}$ composed of $|Y|$-dimensional α-vectors, with each α-vector associated with an action. For simplicity, in the state abstractions, we assumed that, for any given abstract fully observable state component x_K, if the number of original fully observable state components aggregated to x_K is $|\phi^{-1}(x_K)|$, then the weight of each $x \in \phi^{-1}(x_K)$ is uniformly distributed: $\omega(x) = 1/|\phi^{-1}(x_K)|$. We set up a precision target p_t of 0.01 for all our algorithms and no time limit.

Computational Sustainability Case Studies

Management of a threatened bird The objective of our first case study is to maximize the persistence probability of the *Gouldian finch* (Chadès et al. 2012). The population state is fully observable but we are uncertain about the response to the management actions. The problem was originally modeled as as a factored state space representation where $|X|$=2 represents the probability of persistence (Low or High) and $|Y|$=4 represents expert-elicited models (Expert1, Expert2, Expert3, Expert4) that predict the state dynamics when the management actions are applied. The set of actions A that managers can choose are: do nothing (DN), improve fire and grazing management (FG), control feral cats (C), and provide nesting boxes (N). We define the set of observations O as the set of probability of persistence of the *Gouldian finch* population. The observation function Z is the probability of observing o given a state pair $s = (x, b_y)$ and equals 1 if $o'_x = x$ and 0 otherwise. Finally we define the reward function $r(x, a)$ so that $r(Low, DN) = 0$, $r(High, DN) = 20$, $r(Low, FG) = r(Low, C) = r(Low, N) = -5$, and $r(High, FG) = r(High, C) = r(High, N) = 15$. As initial belief b_0, we assume that each expert is as likely to be correct at the beginning of our adaptive management program. First, we solve the problem using MO-SARSOP ($|\mathbf{\Gamma}|$= 11644 α-vectors, Table 1). Our N-MOMDP algorithms for N=6 performed well with $V_{\mathbf{\Gamma}_N}(\cdot, b_0) \geq 0.98V(\cdot, b_0)$ for all algorithms. This is a reduction of 99.95% in the number of α-vectors while retaining 98% of the policy value. Fig. 2A shows the resulting policy graph (6 nodes representing α-vectors and a branching factor of at most $|X| = 2$). We learn that under a low probability of persistence and no prior information, managers should invest in improving fire and grazing management. If the response is positive (x = High), the policy recommends continuing unless observation 'Low' occurs. In that case, providing nesting boxes is recommended. Similarly, controlling feral cats is recommended if implementing management of fire and grazing in state 'Low' remains in a 'Low' state. If controlling feral cats does not improve the status of the species, the recommended action is to go back to fire and grazing management. The original policy could not be efficiently represented as a policy graph given the number of nodes ($|\mathbf{\Gamma}|$ >11000 α-vectors). Chadès et al. (2012) describe similar recommendations to ours. Theirs were derived through explorations of simulations for given scenarios but a complete understanding was missing given the impossibility of representing the original policy graph. We also ran our N-MOMDP algorithms on a version of the *Gouldian finch* problem with $|X|$=81, representing the population state of four different species (*Gouldian finch*, long-tailed finch, dingo and cats), $|Y|$=2 representing the dynamics of the system given by two different experts (Expert1, Expert2), and the same four management actions. For a value of N=$|X|$=81, we obtained $V_{\mathbf{\Gamma}_N}(\cdot, b_0) \geq 0.98V(\cdot, b_0)$. A policy of $|\mathbf{\Gamma}_N|$=81 α-vectors represents a reduction of 97.9% of the original policy size. The computing times are less than 2 seconds for all algorithms (precision of 0.01).

Release of biocontrol agents In our second case study, we seek to optimally release biocontrol agents to control epidemics of dengue fever. Dengue viruses can be transmitted between humans by *Aedes Albopictus* mosquitoes. The number of dengue infections per year is estimated to be 390 million (Bhatt et al. 2013). Genetically modified biological control agents (SIT) can be used to control the population of mosquitoes under some conditions. Deciding how much biocontrol agent to release is difficult due to the uncertainty surrounding key predictive population parameters such as the density parameter (Alphey, Alphey, and Bonsall 2011). We modeled this problem as a hmMDP with a factored state space, $S = \langle X, Y \rangle$, with X representing the discretized abundance of female mosquitoes ranging from 0 to 10^8 ($|X| = 100$), and Y = {0.302; 0.84; 0.94; 1.04} ($|Y|$=4) the set of discretized density dependent parameters that lead to very different responses to the release of biocontrol agents. We define the finite set of actions that managers can choose as a set A = {0, 5, 10, 20}*10^6 ($|A|$ = 4), that correspond to the amount of biocontrol agent to be released each day. We define the set of observations O as the set of abundance states of female mosquitoes X. The observation function Z is the probability of observing o given a state pair $s = (x, b_y)$ and equals 1 if $o'_x = x$ and 0 otherwise. We define the reward function $r(x, a)$ as the cost of being in abundance state x and implementing action a. We define the cost of implementing an action a as the production cost of the quantity of biocontrol agents to release over the management period (\$813/10^6 insects, (Alphey, Alphey, and Bonsall 2011)). We define the cost of being in state x as the expected cost of a dengue epidemics occurring for a given abundance of female mosquitoes. As initial belief b_0, we assume that each discretized density parameter is as likely to be correct at the beginning of our adaptive management program. We evaluate our K-MOMDP algorithm for all values of K from K = 2 to K = 100. Fig. 1 shows the evolution of the lower bound $V_0(b_0)$. For values of $K < 8$, the lower bounds decrease, which indicates poor performance. However, for values of $K \geq 8$, the resulting K-MOMDP has a value of $V_{X_K}(\cdot, b_0) \geq 0.93V(\cdot, b_0)$. We then evaluate our N-MOMDP algorithms with $|\mathbf{\Gamma}|$ = 789 α-vectors. For $N = |X|$ = 100, we obtain a good performance with $V_{\mathbf{\Gamma}_N}(\cdot, b_0) \geq 0.98V(\cdot, b_0)$ for all our N-MOMDP algorithms. Finally, we run our K-N-MOMDP algorithm for K=8 and N=8. The resulting policy performs well, with $V_{X_K, \mathbf{\Gamma}_N}(\cdot, b_0) \geq 0.91V(\cdot, b_0)$. Fig. 2B represents its policy graph. Study of the policy graph reveals that some states were not reachable from our initial state (x_0, b_0) and the policy graph can be represented with only 4 α-vectors. Completely observable abstract state variables $x_{K_i} \in X_K$ represent the aggregated fully observable state variables $x_j \in X$ from the original MOMDP problem. Each aggregated x_{K_i} defines a more compact representation of the populations of female mosquitoes. The computing times for K=8 and N=8 are less than 4.5 seconds for all algorithms (precision target is 0.01).

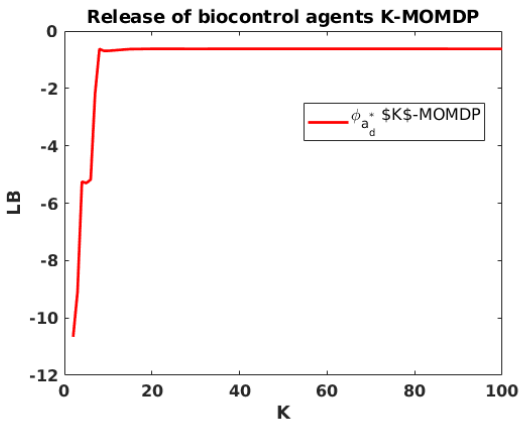

Figure 1: Lower bound values for the *bio-control agents* problem.

Problem $(\lvert X\rvert, \lvert Y\rvert, \lvert A\rvert)$	Algorithm	N	$V_{\mathbf{\Gamma}_N}$	Time(sec.)
Gouldian4Exp (2,4,4)	sarsop	11644	85.6	NA
	$\hat{\alpha}$-min-fast	6	84.58	6.03
	$\hat{\alpha}$-min-p	6	**85.04**	45.83
	$\hat{\alpha}$-pruning	6	84.32	8.23
Gouldian2Exp (81,2,4)	sarsop	3861	75.74	NA
	$\hat{\alpha}$-min-fast	81	**74.23**	0.88
	$\hat{\alpha}$-min-p	81	75.33	0.97
	$\hat{\alpha}$-pruning	81	74.5	1.37
Bio-control agents (100,4,4)	sarsop	789	-0.59	NA
	$\hat{\alpha}$-min-fast	100	**-0.61**	0.65
	$\hat{\alpha}$-min-p	100	**-0.61**	0.92
	$\hat{\alpha}$-pruning	100	-0.63	0.56

Table 1: Comparison of $\hat{\alpha}$-min N-MOMDP algorithms. Bold values represent the best $V_{\mathbf{\Gamma}_N}$.

Conclusion

Our experiments show that our algorithms achieve dramatic reduction in the number of states and α-vectors in MOMDPs while producing policies that achieve similar value and are highly interpretable. Motivated by the need to provide more interpretable models and policies for human operated systems, we proposed a new problem: solving K-N-MOMDPs.

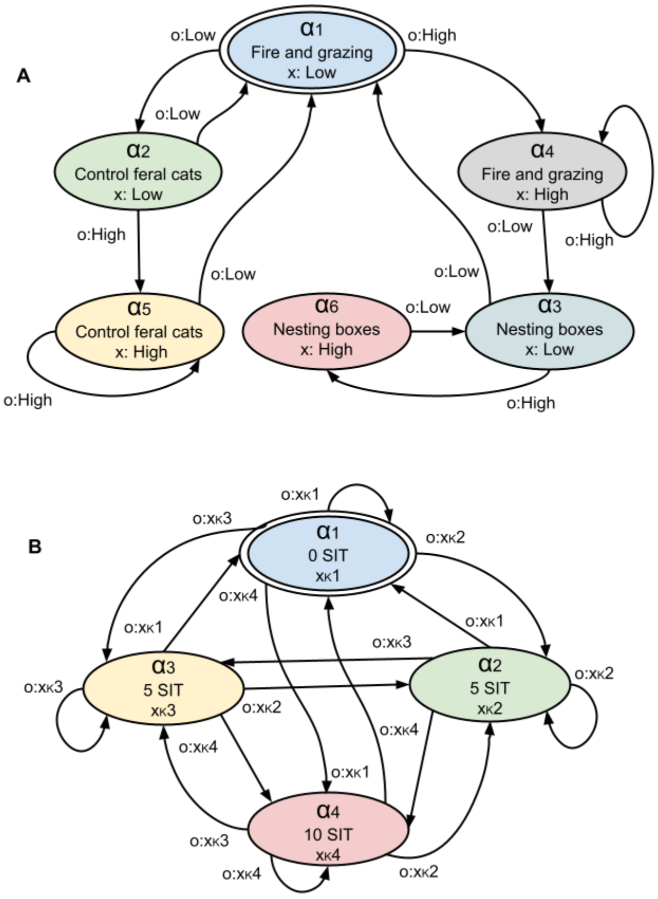

Figure 2: Policy graph for (A) the *Gouldian finch* with 4 experts and N = 6, (B) *bio-control agents* K-N-MOMDP problem with 4 α-vectors. Nodes represent α-vectors and edges are observations. Sterile Insect Technology (SIT) represents the amount of bio-control agents to be released.

Our approach finds the best possible policy of size K visible states and N α-vectors. First, for solving K-MOMDPs, we developed an algorithm that make use of previously proposed state abstraction algorithms for MDPs and we adapted them for MOMDPs. Second, for solving N-MOMDPs, we developed two algorithms based on the α-min principle, and a new algorithm that performs a pruning on the set of α-vectors $\mathbf{\Gamma}$. The three N-MOMDP algorithms perform a binary search on a parameter that minimizes the gap between the original and the reduced policy given N. Finally, we provided an algorithm to solve K-N-MOMDPs. We assessed our algorithms on two adaptive management problems. The resulting K-N-policy graphs provided precious insights for managers that will hopefully results in higher uptake of AI approaches. The main drawback of our approach is that it will only work for relatively small problems of low dimension. Future work will investigate how to deal with more dimensions and assessing interpretability of proposed K-N-MOMDP solutions with domain experts and behavioral scientists.

References

Abel, D.; Arumugam, D.; Lehnert, L.; and Littman, M. 2018. State abstractions for lifelong reinforcement learning. In *ICML*.

Ahmad, M. A.; Eckert, C.; and Teredesai, A. 2018. Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, 559–560.

Alphey, N.; Alphey, L.; and Bonsall, M. B. 2011. A model framework to estimate impact and cost of genetics-based sterile insect methods for dengue vector control. *PloS one* 6(10): e25384.

Åström, K. J. 1965. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications* 10(1): 174–205.

Bellman, R. 1957. *Dynamic programming*. Princeton University Press, John Wiley & Sons.

Bhatt, S.; Gething, P. W.; Brady, O. J.; Messina, J. P.; Farlow, A. W.; Moyes, C. L.; Drake, J. M.; Brownstein, J. S.; Hoen, A. G.; Sankoh, O.; et al. 2013. The global distribution and burden of dengue. *Nature* 496(7446): 504–507.

Chadès, I.; Carwardine, J.; Martin, T. G.; Nicol, S.; Sabbadin, R.; Buffet, O.; et al. 2012. MOMDPs: A Solution for Modelling Adaptive Management Problems. In *AAAI*.

Chadès, I.; Chapron, G.; Cros, M.-J.; Garcia, F.; and Sabbadin, R. 2014. MDPtoolbox: a multi-platform toolbox to solve stochastic dynamic programming problems. *Ecography* 37(9): 916–920.

Chakraborti, T.; Sreedharan, S.; Grover, S.; and Kambhampati, S. 2019. Plan explanations as model reconciliation. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 258–266. IEEE.

Chung, Y.; Bagheri, N.; Salinas-Perez, J. A.; Smurthwaite, K.; Walsh, E.; Furst, M.; Rosenberg, S.; and Salvador-Carulla, L. 2020. Role of visual analytics in supporting mental healthcare systems research and policy: A systematic scoping review. *International Journal of Information Management* 50: 17–27.

Dean, T.; and Givan, R. 1997. Model minimization in Markov decision processes. In *AAAI/IAAI*, 106–111.

Dujardin, Y.; Dietterich, T.; and Chadès, I. 2015. α-min: A Compact Approximate Solver For Finite-Horizon POMDPs. In *IJCAI*, 2582–2588.

Dujardin, Y.; Dietterich, T.; and Chadès, I. 2017. Three New Algorithms to Solve N-POMDPs. In *AAAI*, 4495–4501.

Ferrer-Mestres, J.; Dietterich, T. G.; Buffet, O.; and Chadès, I. 2020. Solving K-MDPs. In *ICAPS*, volume 30, 110–118.

Keith, D. A.; Martin, T. G.; McDonald-Madden, E.; and Walters, C. 2011. Uncertainty and adaptive management for biodiversity conservation.

Kurniawati, H.; Hsu, D.; and Lee, W. S. 2008. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *RSS*.

Lakkaraju, H.; and Rudin, C. 2017. Learning cost-effective and interpretable treatment regimes. In *AISTATS*.

Li, L.; Walsh, T. J.; and Littman, M. L. 2006. Towards a Unified Theory of State Abstraction for MDPs. In *ISAIM*.

Madani, O.; Hanks, S.; and Condon, A. 2003. On the Undecidability of Probabilistic Planning and Related Stochastic Optimization Problems. *Artificial Intelligence* 147(1-2): 5–34.

McDonald-Madden, E.; Runge, M. C.; Possingham, H. P.; and Martin, T. G. 2011. Optimal timing for managed relocation of species faced with climate change. *Nature Climate Change* 1(5): 261–265.

Miller, T. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267: 1–38.

Miller, T.; Pearce, A. R.; and Sonenberg, L. 2018. Social planning for trusted autonomy. In *Foundations of trusted autonomy*, 67–86. Springer, Cham.

Nicol, S.; Buffet, O.; Iwamura, T.; and Chadès, I. 2013. Adaptive Management of Migratory Birds Under Sea Level Rise. In *IJCAI*.

Nicol, S.; Fuller, R. A.; Iwamura, T.; and Chadès, I. 2015. Adapting environmental management to uncertain but inevitable change. *Proceedings of the Royal Society of London B: Biological Sciences* 282(1808): 20142984.

Ong, S. C.; Png, S. W.; Hsu, D.; and Lee, W. S. 2010. Planning under uncertainty for robotic tasks with mixed observability. *The International Journal of Robotics Research* 29(8): 1053–1068.

Papadimitriou, C. H.; and Tsitsiklis, J. N. 1987. The complexity of Markov decision processes. *Mathematics of operations research* 12(3): 441–450.

Payrovnaziri, S. N.; Chen, Z.; Rengifo-Moreno, P.; Miller, T.; Bian, J.; Chen, J. H.; Liu, X.; and He, Z. 2020. Explainable artificial intelligence models using real-world electronic health record data: a systematic scoping review. *Journal of the American Medical Informatics Association* 27(7): 1173–1185.

Péron, M.; Becker, K. H.; Bartlett, P.; and Chadès, I. 2017. Fast-tracking Stationary MOMDPs for Adaptive Management Problems. In *AAAI*.

Petrik, M.; and Luss, R. 2016. Interpretable Policies for Dynamic Product Recommendations. In *UAI*.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rai, A. 2020. Explainable AI: From black box to glass box. *Journal of the Academy of Marketing Science* 48(1): 137–141.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1(5): 206–215.

Rudin, C.; and Ustun, B. 2018. Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice. *Interfaces* 48(5): 449–466.

Shea, K.; Tildesley, M. J.; Runge, M. C.; Fonnesbeck, C. J.; and Ferrari, M. J. 2014. Adaptive management and the value of information: learning via intervention in epidemiology. *PLoS Biol* 12(10): e1001970.

Sigaud, O.; and Buffet, O. 2013. *Markov decision processes in artificial intelligence*. John Wiley & Sons.

Sondik, E. 1971. *The Optimal Control of Partially Observable Markov Decision Processes*. Ph.D. thesis, Stanford University, California.

Walsh, E. I.; Chung, Y.; Cherbuin, N.; and Salvador-Carulla, L. 2020. Experts' perceptions on the use of visual analytics for complex mental healthcare planning: an exploratory study. *BMC medical research methodology* 20: 1–9.

Walters, C. J.; and Hilborn, R. 1976. Adaptive control of fishing systems. *Journal of the Fisheries Board of Canada* 33(1): 145–159.

Williams, B. K. 2011. Adaptive management of natural resources—framework and issues. *Journal of environmental management* 92(5): 1346–1353.