MOOR: Model-based Offline Reinforcement Learning for Sustainable Fishery Management

Jun Ju [a], **Hanna Kurniawati** [b], **Dirk Kroese** [a] **and Nan Ye** [a]

[a] *School of Mathematics and Physics, The University of Queensland, St Lucia, QLD 4072, Australia*
[b] *Research School of Computer Science, Australian National University, Canberra ACT 0200, Australia*
Email: jun.ju@uq.net.au

Abstract: Fisheries play multi-faceted roles in our society, economy, and environment, and the management decisions often involve competing driving forces — for example, higher harvest is desirable for maximizing economical benefits, but this may cause fishery collapse and is thus detrimental to the environment. The need to account for multiple and possibly conflicting objectives make sustainable fishery management a highly challenging task. This is further compounded by the large amount of uncertainties present in the problem: in particular, our knowledge of the fishery system is limited, and the state of the fishery system is not directly observable.

The Partially Observable Markov Decision Processes (POMDPs) — a general principled framework for sequential decision making for partially observable environments — is well-suited for sustainable fishery management: it is able to account for the long-term effect of actions, and it can conveniently take uncertainties into account. A few recent works have explored the potential of using POMDPs for sustainable fishery management. In this paper, we leverage recent advances in two sub-fields of machine learning, namely, deep learning and reinforcement learning, to develop a novel approach for sustainable fishery management using POMDPs.

We first propose an offline reinforcement learning approach for sustainable fishery management. While typical reinforcement learning approaches learn an optimal policy by directly interacting with the environment, offline reinforcement learning approaches learn an optimal policy using a dataset of past interactions with the environment. This is a highly desirable feature for fishery management, in which the effect of a management decision is only observed after many years, and poor decisions may lead to fisheries' collapse. We believe this perspective will allow us to tap into recent advances in offline reinforcement learning.

Our second contribution is a new algorithm, MOOR, which stands for MOdel-based Offline Reinforcement learning algorithm for sustainable fishery management. MOOR first learns a POMDP fishery dynamics model using catch and effort data, and then solves the POMDP using a state-of-the-art solver. In the model learning step, we view the POMDP fishery dynamics model as a recurrent neural net (RNN), and leverage RNN learning techniques to learn the model. This presents some new challenges, but we show that these can be overcome with a few tricks to yield a very effective learning algorithm.

Finally, MOOR demonstrates strong performance in preliminary simulation studies. The learned models are generally very similar to the true models. In addition, the management policies obtained using the learned models perform similarly as the optimal management policies for the true models. While previous POMDP studies for fishery management evaluate policy performance in the learned model, we evaluate the policy in the true model, thus our results suggest that it is possible to develop a POMDP approach that can be robust against mild model learning error. Moreover, although this paper focuses on fisheries applications, the approach is general enough for other problems where the dynamics are nonlinear, though further research are needed to understand the extent and efficiency of the method on other domains. Our source code will be made available after the publication of the work.

Keywords: *Offline reinforcement learning, fishery management, POMDPs*

1 INTRODUCTION

Sustainable fishery management aims to use fisheries in a way that is beneficial to our society, economy, and environment, for both the present and long-turn future. The importance and urgency of this problem is reflected by one of the United Nations Sustainable Development Goals, which seeks to *conserve and sustainably use the oceans, seas and marine resources for sustainable development*, in the recognition that *the sustainability of our oceans is under severe threat*. Sustainable fishery management poses multiple challenges. One is the need to balance between the conflicting objectives of harvest and conservation: while we want to catch more fish to maximize economical benefits, we also want to avoid over-fishing so that the stock is sustainable. Another challenge is the need to deal with the large amount of uncertainties present in fishery systems [Charles, 1998]. Two notable uncertainties are the uncertainty on the stock sizes as they are not directly observable, and the lack of knowledge of the fishery system dynamics.

Recently, several works have explored the potential of Partially Observable Markov Decision Processes (POMDPs) for sustainable fishery management [Memarzadeh et al., 2019; Filar et al., 2019], with the help of scalable POMDP solvers [Kurniawati et al., 2008; Silver and Veness, 2010; Ye et al., 2017]. They demonstrated that POMDP is capable of generating effective adaptive management strategies by taking partial observability and other uncertainties into account. It is noteworthy that Lane [1989] applied POMDP to the related problem of decision making by fishermen, using a simplified small model due to the lack of a solver for large POMDPs at that time.

As a model is rarely the same as the real world, it is important to develop effective model learning algorithm and evaluate the sensitivity of the POMDP policy with respect to the POMDP model used. This issue has not been considered by previous studies mentioned above, and we address it in this paper.

Our first contribution is to propose an offline reinforcement learning approach for sustainable fishery management. This allows learning a management strategy from past fishery data without the need to wait for years to observe the effect of a management decision, and eliminates the possibility of dangerous actions during the learning phase. This perspective allows us to leverage recent advances in offline reinforcement learning for sustainable fishery management. Secondly, we develop an instantiation of the approach, MOOR (MOdel-based Offline Reinforcement learning for sustainable fishery management), which first learns a POMDP fishery dynamics model using catch and effort data, and then solves the POMDP using a state-of-the-art solver. The learning algorithm is developed for the Beverton–Holt model, but in principle, our approach can be applied to other fishery dynamics models. Finally, our simulation study demonstrates that our model learning algorithm is fairly accurate, and the POMDP approach can be robust against mild model learning errors.

2 POMDP AND OFFLINE REINFORCEMENT LEARNING

We briefly describe POMDP and offline reinforcement learning in this section. These serve as building blocks in our approach: our fishery model is a POMDP, and our solution approach is an offline reinforcement learning algorithm.

POMDP. Partially observable Markov decision processes (POMDP) provide a general framework for sequential decision making for partially observable and uncertain environments. Formally, a POMDP is a 6-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{Z}, T, R, O)$. $\mathcal{S}$ is a set of states that the environment can be in, and $\mathcal{A}$ is a set of actions that the agent can take. When the agent takes an action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, the environment transitions to a new state $s' \in \mathcal{S}$ with transition probability $T(s, a, s') = p(s'|s, a)$. At the same time, it receives a reward $R(s, a)$, and an observation $z \in \mathcal{Z}$ with observation probability $O(s', a, z) = p(z|s', a)$. Since the current environment state is not fully observable, the agent maintains a belief b, which is a probability distribution on the states. If we receive an observation z after executing action a, then we can use Bayes' rule to update the current belief b to a new belief:

$$\tau(b, a, z) = p(s'|b, a, z) = \frac{O(s', a, z)p(s'|a, b)}{p(z|a, b)} \tag{1}$$

where $p(s'|a, b) = \sum_{s \in S} T(s, a, s')b(s)$, $p(z|a, b) = \sum_{s' \in S} O(s', a, z)p(s'|a, b)$.

An agent acts according to a policy π, which is a mapping from beliefs to actions. The value function $V_\pi(b)$ computes the total expected discounted reward for the policy π when starting from a belief b:

$$V_\pi(b) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(S_t, A_t)|b, \pi], \tag{2}$$

where S_t, A_t refer to the state and the action at time step t, and $\gamma \in (0, 1)$ is the discount factor.

An optimal policy π^* is a policy that achieves the maximum value for each belief, that is, $V_{\pi^*}(b) \geq V_\pi(b)$ for any policy π and any belief b.

Offline Reinforcement Learning. Reinforcement learning aims to learn an optimal policy by interacting with the environment. Most reinforcement learning algorithms require direct interactions with the environment, but in domains such as healthcare it is expensive to obtain data via interactions and poor decisions made in the learning phase may have dangerous consequences. Offline reinforcement learning aims to find an optimal policy using previous experiences instead. It is well-suited to domains in which collecting online data is expensive and dangerous, and has found applications in healthcare [Shortreed et al., 2011] and robotics [Ebert et al., 2018].

3 MOOR FOR SUSTAINABLE FISHERY MANAGEMENT

In fishery management, it takes years to assess the effect of management decisions, and poor management decisions may lead to fishery collapse. In other words, collecting experiences by directly interacting with the environment is costly and potentially dangerous. We thus take the offline reinforcement learning approach.

We focus on model-based offline reinforcement learning (e.g., see Kidambi et al. [2020]), which first learns an approximate environment model using the dataset, and then computes a near-optimal policy using the learned environment model. To the best of our knowledge, existing works on model-based offline reinforcement learning consider fully observable environments. However, in our case, the environment is partially observable and is modelled as a POMDP, which is difficult to learn in general. A contribution of this paper is an algorithm that can successfully learn high-quality POMDP fishery models.

Our MOOR algorithm consists of three steps as shown in Algorithm 1. It first learns the parameters of a continuous POMDP model using given catch and effort data, where in practice, catch can be measured in various ways such as the number or the mass of caught fish, and effort is typically measured as the number of the total fishing hours scaled to take fishing power of the gears into account. A discretized version $\mathcal{M}$ of the continuous POMDP is then obtained using a Monte Carlo method described in [Filar et al., 2019]. Details of the parameters used in the method are provided in Section 4. Finally, we make use of efficient solvers for large discrete POMDPs to find an optimal policy π^* for $\mathcal{M}$. We used the DESPOT algorithm [Ye et al., 2017] in our experiments.

Algorithm 1: MOOR

Input: Catch data $c_{1:T}$, effort data $e_{1:T}$

1 $(\rho, K, B_0, q) \leftarrow \texttt{OfflineModelLearning}(c_{1:T}, e_{1:T})$

2 $\mathcal{M} \leftarrow \texttt{ModelDiscretization}(\rho, K, B_0, q)$

3 Find an optimal policy π^* for $\mathcal{M}$

Below, we provide details on the continuous POMDP model and our model learning algorithm.

3.1 Fishery POMDPs

In our POMDP model, the state is the population biomass, an action is an amount of fishing effort, the observation after an action is the amount of catch, and the reward is simply defined as the catch, which serves as a measure of the profit. The state is not directly observable, but partially observable via the amount of catch.

Our transition dynamics and observation model are based on a combination of the Beverton–Holt model [Beverton and Holt, 2012] and a popular catch model. The Beverton–Holt model is a deterministic model that relates the current population biomass B_t to the population biomass B_{t+1} at the next time step via

$$B_{t+1} = f(B_t; K, \rho) = \frac{\rho K B_t}{(\rho - 1)B_t + K}, \tag{3}$$

where the parameters ρ and K are known as the proliferation rate and the carrying capacity respectively. We used the catch model below:

$$C_t = g(B_t, e_t; q) + \epsilon, \text{ with } g(B_t, e_t; q) = q e_t B_t, \tag{4}$$

where C_t is the amount of catch, e_t is the amount of effort, q is a constant called the catchability constant, and ϵ is a random noise drawn from a fixed Gaussian distribution.

Our transition model builds on the Beverton–Holt model and the catch model described above, and has the following form:

$$B_{t+1} = f(B_t; K, \rho) - g(B_t, e_t; q). \tag{5}$$

Our observation model is simply the catch model given in Eq. (4).

To sum up, our continuous fishery POMDP model has four parameters: initial biomass B_0, the carrying capacity K, the proliferation rate ρ and the catchability constant q. We describe how to learn these parameters in next few paragraphs.

3.2 Model learning

We describe how to learn the POMDP model parameters ρ, K, B_0, q using given catch data $c_{1:T} = (c_1, c_2, \ldots, c_T)$ and efforts data $e_{1:T} = (e_1, \ldots, e_T)$.

We learn the parameters by minimizing the quadratic error

$$L(\rho, K, B_0, q) = \sum_{t=1}^{T} (g(B_t, e_t; q) - c_t)^2. \tag{6}$$

where B_t evolves according Eq. (5) with g defined in Eq. (4). Minimizing L is the same as maximizing the likelihood under the Gaussian noise assumption in Eq. (4).

The loss function L is a complex function of the parameters, though there are only four parameters. We consider gradient-based optimization algorithms for minimizing L. While in principle the gradient of L can be calculated in a closed-form, this is a tedious task. We address this issue by leveraging on recent development in deep learning.

The key observation behind our learning algorithm is that the fishery POMDP is in fact an RNN. This is shown schematically in Figure 1. At time step t, the hidden state of the RNN is the biomass B_t, the input is the effort e_t, and the hidden state is updated using the recurrence Eq. (5). The output is the predicted catch $\hat{c}_t = g(B_t, e_t; q)$. This perspective suggests that we can exploit existing tools for learning general RNNs. In particular, we can make use of the automatic differentiation tools in major deep learning platforms such as PyTorch [Paszke et al., 2019] and Tensorflow [Abadi et al., 2016] to obtain the gradient automatically from an implementation of the loss function L.

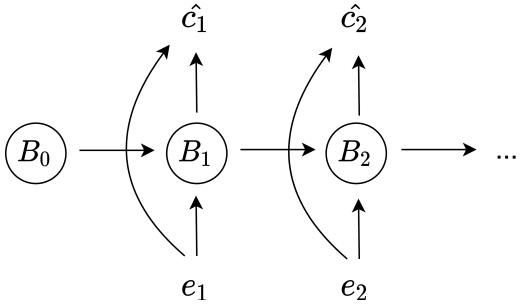

Figure 1. The fishery POMDP as an RNN.

For the choice of the optimization algorithm, we cannot use the stochastic gradient descent (SGD) algorithm which is commonly used to train RNNs in the deep learning literature. This is because we only have one training sequence, thus we do not have stochastic gradients calculated on randomly sampled batches of training sequences. We also note that gradient descent generally fails to learn the true parameters despite extensive tuning of the step size. This is mainly due to two difficulties: (a) The scale of the parameters are very different, and their derivatives can be of very different magnitudes. When using a gradient-based method, the parameters with a large derivative can easily dominate in the training process in the sense that updates in the other variables often become too small. (b) The final solution obtained using gradient descent depends heavily on the initial solution, suggesting that the loss function L can easily trap gradient descent in poor local optima.

To address the two observed difficulties, our final learning algorithm is the Limited-memory BFGS (L-BFGS) algorithm [Liu and Nocedal, 1989], combined with a few tricks. L-BFGS is a quasi-Newton method and is better at exploiting the surface geometry to find an optimal solution as compared to gradient descent, but does not require Hessian computation as in Newton's method. To address the first issue of different parameter scales, we scale the effort and catch data to the range of [0, 1], by dividing them with their maximums respectively. Specifically, let $c_{\max}$ and $e_{\max}$ be the maximum catch and the maximum effort. We will first convert $c_{1:t}$ to $\tilde{c}_{1:t}$ with $\tilde{c}_i = c_i / c_{\max}$, and convert $e_{1:t}$ to $\tilde{e}_{1:t}$ with $\tilde{e}_i = e_i / e_{\max}$. Now we need to minimize

$$\tilde{L}(\rho, K, B_0, q) = \sum_{t=1}^{T} (g(B_t, \tilde{e}_t; q) - \tilde{c}_t)^2. \tag{7}$$

If $\tilde{\rho}^*$, $\tilde{K}^*$, $\tilde{B}_0^*$, $\tilde{q}^*$ are the minimizers of $\tilde{L}(\rho, K, B_0, q)$, then $\rho^* = \tilde{\rho}^*$, $K^* = \tilde{K}^* c_{\max}$, $B_0^* = \tilde{B}_0^* c_{\max}$, $q^* = \tilde{q}^*/e_{\max}$ are the minimizers of $L(\rho, K, B_0, q)$. To address the second issue of poor local minimizers, the initial parameters, for the normalized objective function $\tilde{L}$, are all set to 1 added with a small amount of random noise. The initial values of 1 are chosen because with these values, the biomass will never become negative in Eq. (5) as $f(B_t; K, \rho) = B_t$ and $g(B_t, e_t; q) \leq B_t$. In addition, we also use the commonly used trick of running the optimization process multiple times, until $L(\rho, K, B_0, q)$ is less than a certain threshold or when a maximum number of runs is reached.

4 Simulation Study

We evaluate the performance of MOOR by performing a simulation study to answer the following questions:

1. How do the model learning algorithm perform? We compare the learned parameters and the learned population dynamics model with the true ones.
2. How do the final learned policies perform? In particular, how good are the final learned policies in the presence of model learning error – i.e. how do the value of the learned policy compared with the optimal policy computed for the true model?

We consider fishery systems with different proliferation rates in our simulation study. We briefly describe our evaluation process as shown in Figure 2 before presenting the details. We first generate some catch and effort data, then use MOOR to learn a policy π^*. The value of the policy π^* is then estimated by simulating it multiple times. To assess the effect of model learning error, we compare the estimated value of π^* with the estimated value of the optimal policy found by DESPOT for the true model.

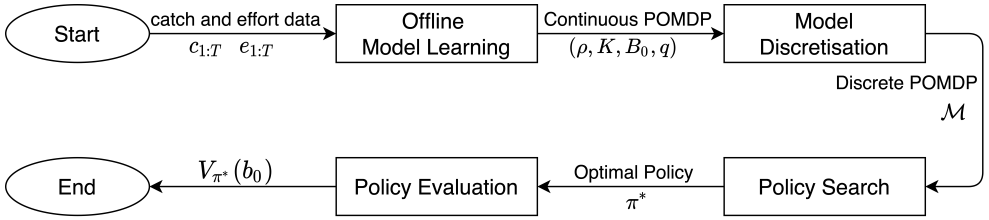

Figure 2. Performance evaluation for MOOR.

Groundtruth model and data generation. The groundtruth model for the environment is assumed to be the model of the form in Eq. (5). For the model parameters (ρ, K, B_0, q), we choose $K = 1000$, $B_0 = 5000$, $q = 0.005$, and consider three proliferation rates $\rho = 1.3, 2.0$ and 3.0, representing low, medium, and high proliferation rates respectively.

For data generation, we first randomly generate 50 effort values drawn from $N(10, 3)$, then we generate the corresponding catches using Eq. (4) with the noise set to 0.

Model learning. We repeated optimization using L-BFGS multiple times. We stop when the quadratic error (sum of squared errors) is less than 1 (which makes the root mean squared error less than 1% of the minimum true catch), or when the number of runs reaches 30. In each trial, L-BFGS is run for 15 iterations with a learning rate of 0.5.

Model discretization. The continuous versions of the learned POMDP model and the groundtruth POMDP models are generally not the same, but their discrete versions need to have shared actions and observations, so that we can evaluate a policy computed using the discretized learned model on the discretized groundtruth model. To ensure this, a continuous state space $[0, K]$ is discretized into $\lceil \frac{K}{L_s} \rceil$ states, with the i-th state representing $[(i-1)L_s, \min(iL_s, K))$, where $L_s = 1,000$ in our experiments, leading to around 10 states in the discrete models. We used the same intervals to represent the discretized observations for the observation space. For a continuous model, we only need to consider action (i.e., effort) not more than $a_{\max} = \frac{1}{q}$, because the catch cannot exceed the biomass in $C = qeB$. We discretize the action spaces for the learned model and the groundtruth model into intervals of length L_a (except that the last may be truncated), then choose their midpoints as the discretized actions. We chose $L_a = 15$ in our experiments, leading to around 15 actions in

Table 1. Results of model learning and policy learning.

	learned parameters				policy values	
ρ	$\hat{\rho}$	$\hat{K}$	$\hat{B}_0$	$\hat{q}$	learned models	groundtruth models
1.3	1.3001	9985.77	4991.60	0.005	12949 $\pm$ 168.28	12818 $\pm$ 139.08
2.0	1.9902	10914.07	5501.46	0.0045	33667 $\pm$ 191.51	34860 $\pm$ 136.70
3.0	3.0009	9986.70	4991.65	0.0050	56201 $\pm$ 134.01	56178 $\pm$ 173.58

the discrete models. Finally, we follow Filar et al. [2019] to obtain the discretized transition dynamics and observation model, by drawing 10,000 randomly sampled (s', o) for each (s, a).

Policy search. We used the DESPOT [Ye et al., 2017] algorithm to solve the discretized POMDP model.

Policy evaluation. Figure 3 illustrates how we estimate a policy π's value. We start with an initial belief b_0. At time step t, we compute the action $\pi(b_t)$. The world (i.e., the groundtruth model) executes the action, and generates the resulting reward and observation. The simulator (i.e., the learned POMDP model) then updates the belief b_t using Eq. (1). We run the policy 500 times, with 100 time steps for each run. The average total discounted reward with a discount factor $\gamma = 0.95$ is then calculated as an estimate of the policy value.

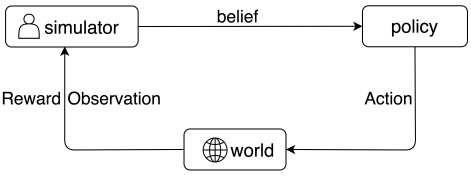

Figure 3. Evaluation for learned models.

4.1 Quality of learned models

The learned parameters $(\hat{\rho}, \hat{K}, \hat{B}_0, \hat{q})$ for the three groundtruth models are shown in the left subtable of Table 1. Note that for the true parameters, we only show the value of ρ as other parameter values are the same. The learned parameters are generally very accurate, with less than 1% error, except that for $\rho = 2$, the values of $\hat{K}$ and $\hat{B}_0$ are around 10% larger, while that of $\hat{q}$ is around 10% smaller. The larger errors for $\rho = 2$ suggests that a mild error in $\hat{q}$ can be compensated by small errors of opposite signs in $\hat{r}$ and $\hat{K}$, in the sense that the predicted catches (qe_tB_t values) are kept about the same as those obtained using the true model. Thus recovering the true parameters appears more difficult than learning a functionally similar model. Figure 4 shows the plots for the learned population models and the groundtruth population models. For $\rho = 1$ and $\rho = 3$, the two curves are nearly the same, which is expected as the learned parameters are all very accurate. For $\rho = 2$, despite an error of around 10% for $\hat{K}$, $\hat{B}_0$ and $\hat{q}$, the two curves still appear to be very close to each other.

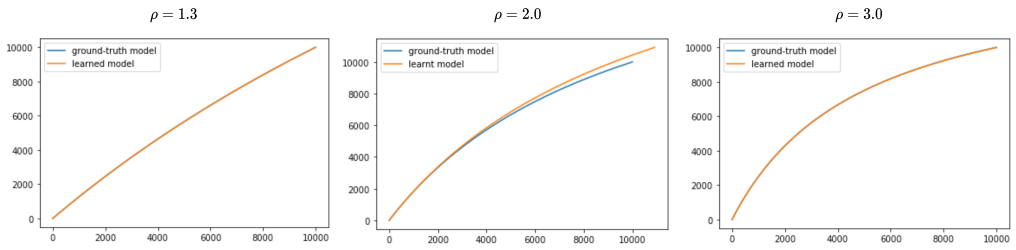

Figure 4. Plots of $B' = f(B; K, \rho)$ against B for the groundtruth and learned Beverton–Holt models.

4.2 Quality of learned policies

The right subtable of Table 1 shows the average total discounted rewards of the policies computed for the learned POMDP models and the corresponding groundtruth models, together with their standard errors. To take the randomness in model discretization into account, we generated three discretized models for each continuous fishery POMDP model, performed 500 runs for the policy computed for each of them, and then used all the 1,500 runs to calculate the average total discounted rewards and the standard error.

The results show that learned policies achieves the same level of performance as the policies computed for the groundtruth models. This is not surprising for the case of $\rho = 1$ and $\rho = 3$, because the learned models are

nearly the same as the groundtruth models. Interestingly, for the case of $\rho = 2$, despite around 10% errors in three of the four parameters ($\hat{K}$, $\hat{B}_0$ and $\hat{q}$), the value of the learned policy is only around 3.5% smaller than the policy computed for the groundtruth model. Thus MOOR is relatively robust against model learning error in this case.

5 Conclusion

We proposed MOOR, a model-based offline reinforcement learning algorithm for sustainable fishery management. MOOR learns a management strategy using past catch and effort data. Our simulation study shows that MOOR learns high-quality models and policies.

Our results suggest the potential of offline reinforcement learning for sustainable fishery management, and will be interesting to further explore recent advances in offline reinforcement learning. In addition, inspired by recent developments in deep learning, we developed a gradient-based algorithm for learning a fishery POMDP model. We focused on the deterministic Beverton–Holt dynamics model as an initial investigation, but our technique can be applied to models with alternative population dynamics, and to other domains with nonlinear dynamics.

Acknowledgement

We would like to thank Jerzy Filar for many helpful discussions. This work is partially supported by the Australian Research Coucil (ARC) Discovery Project 200101049 and the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS, grant number CE140100049).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. et al. [2016], Tensorflow: A system for large-scale machine learning, *in* '12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)', pp. 265–283.

Beverton, R. J. and Holt, S. J. [2012], *On the dynamics of exploited fish populations*, Vol. 11, Springer Science & Business Media.

Charles, A. T. [1998], 'Living with uncertainty in fisheries: analytical methods, management priorities and the Canadian groundfishery experience', *Fisheries Research* **37**(1-3), 37–50.

Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A. and Levine, S. [2018], 'Visual foresight: Model-based deep reinforcement learning for vision-based robotic control', *arXiv preprint arXiv:1812.00568* .

Filar, J. A., Qiao, Z. and Ye, N. [2019], POMDPs for sustainable fishery management, *in* '23rd International Congress on Modelling and Simulation-Supporting Evidence-Based Decision Making: The Role of Modelling and Simulation, MODSIM 2019', Modelling and Simulation Society of Australia and New Zealand, pp. 645–651.

Kidambi, R., Rajeswaran, A., Netrapalli, P. and Joachims, T. [2020], 'Morel: Model-based offline reinforcement learning', *arXiv preprint arXiv:2005.05951* .

Kurniawati, H., Hsu, D. and Lee, W. S. [2008], SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces, *in* 'Proc. Robotics: Science and Systems', Vol. 62.

Lane, D. E. [1989], 'A partially observable model of decision making by fishermen', *Operations Research* **37**(2), 240–254.

Liu, D. C. and Nocedal, J. [1989], 'On the limited memory BFGS method for large scale optimization', *Mathematical programming* **45**(1), 503–528.

Memarzadeh, M., Britten, G. L., Worm, B. and Boettiger, C. [2019], 'Rebuilding global fisheries under uncertainty', *Proceedings of the National Academy of Sciences* **116**(32), 15985–15990.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. et al. [2019], 'Pytorch: An imperative style, high-performance deep learning library', *Advances in neural information processing systems* **32**, 8026–8037.

Shortreed, S. M., Laber, E., Lizotte, D. J., Stroup, T. S., Pineau, J. and Murphy, S. A. [2011], 'Informing sequential clinical decision-making through reinforcement learning: an empirical study', *Machine learning* **84**(1-2), 109–136.

Silver, D. and Veness, J. [2010], Monte-carlo planning in large pomdps, *in* 'Proc. Neural Information Processing Systems (NeurIPS)'.

Ye, N., Somani, A., Hsu, D. and Lee, W. S. [2017], 'DESPOT: Online POMDP planning with regularization', *Journal of Artificial Intelligence Research* **58**, 231–266.