Astin Bulletin (2023), **53**, pp. 233–257
doi:10.1017/asb.2023.13

International Actuarial Association
Association Actuarielle Internationale

RESEARCH ARTICLE

Premium control with reinforcement learning

Lina Palmborg* and Filip Lindskog

Department of Mathematics Stockholm University Stockholm 106 91, Sweden
*Corresponding author. E-mail: lina.palmborg@math.su.se

Received: 7 July 2022; **Revised:** 15 March 2023; **Accepted:** 15 March 2023; **First published online:** 11 April 2023

Keyword: Premium control; reinforcement learning; Markov decision process

JEL codes: C60; G22

Abstract

We consider a premium control problem in discrete time, formulated in terms of a Markov decision process. In a simplified setting, the optimal premium rule can be derived with dynamic programming methods. However, these classical methods are not feasible in a more realistic setting due to the dimension of the state space and lack of explicit expressions for transition probabilities. We explore reinforcement learning techniques, using function approximation, to solve the premium control problem for realistic stochastic models. We illustrate the appropriateness of the approximate optimal premium rule compared with the true optimal premium rule in a simplified setting and further demonstrate that the approximate optimal premium rule outperforms benchmark rules in more realistic settings where classical approaches fail.

1. Introduction

An insurance company's claim costs and investment earnings fluctuate randomly over time. The insurance company needs to determine the premiums before the coverage periods start, that is before knowing what claim costs will appear and without knowing how its invested capital will develop. Hence, the insurance company is facing a dynamic stochastic control problem. The problem is complicated because of delays and feedback effects: premiums are paid before claim costs materialise and premium levels affect whether the company attracts or loses customers.

An insurance company wants a steady high surplus. The optimal dividend problem introduced by de Finetti (1957) (and solved by Gerber, 1969) has the objective to maximise the expected present value of future dividends. Its solution takes into account that paying dividends too generously is suboptimal since a probability of default that is too high affects the expected present value of future dividends negatively. A practical problem with implementing the optimal premium rule, that is a rule that maps the state of the stochastic environment to a premium level, obtained from solving the optimal dividend problem is that the premiums would be fluctuating more than what would be feasible for a real insurance market with competition. A good premium rule needs to generate premiums that do not fluctuate wildly over time.

For a mutual insurance company, different from a private company owned by shareholders, maximising dividends is not the main objective. Instead the premiums should be low and suitably averaged over time, but also making sure that the surplus is sufficiently high to avoid a too high probability of default. Solving this multiple-objective optimisation problem is the focus of the present paper. Similar premium control problems have been studied by Martin-Löf (1983, 1994), and these papers have been a source of inspiration for our work.

Martin-Löf (1983) carefully sets up the balance equations for the key economic variables of relevance for the performance of the insurance company and studies the premium control problem as a linear

control problem under certain simplifying assumptions enabling application of linear control theory. The paper analyses the effects of delays in the insurance dynamical system on the linear control law with feedback and discusses designs of the premium control that ensure that the probability of default is small.

Martin-Löf (1994) considers an application of general optimal control theory in a setting similar to, but simpler than, the setting considered in Martin-Löf (1983). The paper derives and discusses the optimal premium rule that achieves low and averaged premiums and also targets sufficient solvency of the insurance company.

The literature on optimal control theory in insurance is vast, see for example the textbook treatment by Schmidli (2008) and references therein. Our aim is to provide solutions to realistic premium control problems in order to allow the optimal premium rule to be used with confidence by insurance companies. In particular, we avoid considering convenient stochastic models that may fit well with optimal control theory but fail to take key features of real dynamical insurance systems into account. Instead, we consider an insurance company that has enough data to suggest realistic models for the insurance environment, but the complexity of these models do not allow for explicit expressions for transition probabilities of the dynamical system. In this sense, the model of the environment is not fully known. However, the models can be used for simulating the behaviour of the environment.

Increased computing power and methodological advances during the recent decades make it possible to revisit the problems studied in Martin-Löf (1983, 1994) and in doing so allow for more complex and realistic dynamics of the insurance dynamical system. Allowing realistic complex dynamics means that optimal premium rules, if possible to be obtained, will allow insurance companies to not only be given guidance on how to set premiums but actually have premium rules that they can use with certain confidence. The methodological advances that we use in this work is reinforcement learning and in particular reinforcement learning combined with function approximation, see for example Bertsekas and Tsitsiklis (1996) and Sutton and Barto (2018) and references therein. In the present paper, we focus on the temporal difference control algorithms SARSA and Q-learning. SARSA was first proposed by Rummery and Niranjan (1994) and named by Sutton (1995). Q-learning was introduced by Watkins (1989). By using reinforcement learning methods combined with function approximation, we obtain premium rules in terms of Markovian controls for Markov decision processes whose state spaces are much larger/more realistic than what was considered in the premium control problem studied in Martin-Löf (1994).

There exist other methods for solving general stochastic control problems with a known model of the environment, see for example Han and E (2016) and Germain et al. (2021). However, the deep learning methods in these papers are developed to solve fixed finite-time horizon problems. The stochastic control problem considered in the present paper has an indefinite-time horizon, since the terminal time is random and unbounded. A random terminal time also causes problems for the computation of gradients in deep learning methods. There are reinforcement learning methods, such as policy gradient methods (see e.g., Williams, 1992; Sutton et al., 1999, or for an overview Sutton and Barto, 2018, Ch. 13), that enable direct approximation of the premium rule by neural networks (or other function approximators) when the terminal time is random. However, for problems where the terminal time can be quite large (as in the present paper) these methods likely require an additional approximation of the value function (so-called actor-critic methods).

In the mathematical finance literature, there has recently been significant interest in the use of reinforcement learning, in particular related to hedging combined with function approximation, for instance the influential paper by Buehler et al. (2019) on deep hedging. Carbonneau (2021) uses the methodology in Buehler et al. (2019) and studies approaches to risk management of long-term financial derivatives motivated by guarantees and options embedded in life-insurance products. Another approach to deep hedging based on reinforcement learning for managing risks stemming from long-term life-insurance products is presented in Chong et al. (2021). Dynamic pricing has been studied extensively in the operations research literature. For instance, the problem of finding the optimal balance between learning an unknown demand function and maximising revenue is related to reinforcement learning. We refer to den

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

Boer and Zwart (2014) and references therein. Reinforcement learning is used in Krasheninnikova et al. (2019) for determining a renewal pricing strategy in an insurance setting. However, the problem formulation and solution method are different from what is considered in the present paper. Krasheninnikova et al. (2019) considers retention of customers while maximising revenue and does not take claims costs into account. Furthermore, in Krasheninnikova et al. (2019) the state space is discretised in order to use standard Q-learning, while the present paper solves problems with a large or infinite state space by combining reinforcement learning with function approximation.

The paper is organised as follows. Section 2 describes the relevant insurance economics by presenting the involved cash flows, key economic quantities such as surplus, earned premium, reserves and how such quantities are connected to each other and their dynamics or balance equations. Section 2 also introduces stochastic models (simple, intermediate and realistic) giving a complete description of the stochastic environment in which the insurance company operates and aims to determine an optimal premium rule. The stochastic model will serve us by enabling simulation of data from which the reinforcement learning methods gradually learn the stochastic environment in the search for optimal premium rules. The models are necessarily somewhat complex since we want to take realistic insurance features into account, such as delays between accidents and payments and random fluctuations in the number of policyholders, partly due to varying premium levels.

Section 3 sets up the premium control problem we aim to solve in terms of a Markov decision process and standard elements of stochastic control theory such as the Bellman equation. Finding the optimal premium rule by directly solving the Bellman (optimality) equation numerically is not possible when considering state spaces for the Markov decision process matching a realistic model for the insurance dynamical system. Therefore, we introduce reinforcement learning methods in Section 4. In particular, we present basic theory for the temporal difference learning methods Q-learning and SARSA. We explain why these methods will not be able to provide us with reliable estimates of optimal premium rules unless we restrict ourselves to simplified versions of the insurance dynamical system. We argue that SARSA combined with function approximation of the so-called action-value function will allow us to determine optimal premium rules. We also highlight several pitfalls that the designer of the reinforcement learning method must be aware of and make sure to avoid.

Section 5 presents the necessary details in order to solve the premium control problem using SARSA with function approximation. We analyse the effects of different model/method choices on the performance of different reinforcement learning techniques and compare the performance of the optimal premium rule with those of simpler benchmark rules.

Finally, Section 6 concludes the paper. We emphasise that the premium control problem studied in the present paper is easily adjusted to fit the features of a particular insurance company and that the excellent performance of a carefully set up reinforcement learning method with function approximation provides the insurance company with an optimal premium rule that can be used in practice and communicated to stakeholders.

2. A stochastic model of the insurance company

The number of contracts written during year $t+1$ is denoted N_{t+1}, known at the end of year $t+1$. The premium per contract P_t during year $t+1$ is decided at the end of year t. Hence, P_t is $\mathcal{F}_t$-measurable, where $\mathcal{F}_t$ denotes the σ-algebra representing the available information at the end of year t. Contracts are assumed to be written uniformly in time over the year and provide insurance coverage for one year. Therefore, assuming that the premium income is earned linearly with time, the earned premium during year $t+1$ is

$$\mathrm{EP}_{t+1} = \frac{1}{2}\left(P_t N_{t+1} + P_{t-1} N_t\right),$$

that is for contracts written during year $t+1$, on average half of the premium income $P_t N_{t+1}$ will be earned during year $t+1$, and half during year $t+2$. Since only half of the premium income $P_t N_{t+1}$ is

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

Table 1. *Paid claim amounts from accidents during years $t-2,\dots,t+1$.*

	1	2	3	4
t–2	$I_{t-3,1}$	$I_{t-3,2}$	$I_{t-3,3}$	$I_{t-3,4}$
t–1	$I_{t-2,1}$	$I_{t-2,2}$	$I_{t-2,3}$	$I_{t-2,4}$
t	$I_{t-1,1}$	$I_{t-1,2}$	$I_{t-1,3}$	$I_{t-1,4}$
t+1	$I_{t,1}$	$I_{t,2}$	$I_{t,3}$	$I_{t,4}$

earned during year $t+1$, the other half, which should cover claims during year $t+2$, will be stored in the premium reserve. The balance equation for the premium reserve is $V_{t+1} = V_t + P_t N_{t+1} - \mathrm{EP}_{t+1}$. Note that when we add cash flows or reserves occurring at time $t+1$ to cash flows or reserves occurring at time t, the time t amounts should be interpreted as adjusted for the time value of money. We choose not to write this out explicitly in order to simplify notation.

That contracts are written uniformly in time over the year means that $I_{t,k}$, the incremental payment to policyholders during year $t+k$ for accidents during year $t+1$, will consist partly of payments to contracts written during year $t+1$ and partly of payments to contracts written during year t. Hence, we assume that $I_{t,k}$ depends on both N_{t+1} and N_t. Table 1 shows a claims triangle with entries $I_{j,k}$ representing incremental payments to policyholders during year $j+k$ for accidents during year $j+1$. For ease of presentation, other choices could of course be made, we will assume that the maximum delay between an accident and a resulting payment is four years. Entries $I_{j,k}$ with $j+k \le t$ are $\mathcal{F}_t$-measurable and coloured blue in Table 1. Let

$$\begin{aligned}
\mathrm{IC}_{t+1} &= I_{t,1} + \mathrm{E}\big[I_{t,2} + I_{t,3} + I_{t,4} \mid \mathcal{F}_{t+1}\big], \quad \mathrm{PC}_{t+1} = I_{t,1} + I_{t-1,2} + I_{t-2,3} + I_{t-3,4},\\
\mathrm{RP}_{t+1} &= \mathrm{E}\big[I_{t-3,4} + I_{t-2,3} + I_{t-2,4} + I_{t-1,2} + I_{t-1,3} + I_{t-1,4} \mid \mathcal{F}_t\big]\\
&\quad - \mathrm{E}\big[I_{t-3,4} + I_{t-2,3} + I_{t-2,4} + I_{t-1,2} + I_{t-1,3} + I_{t-1,4} \mid \mathcal{F}_{t+1}\big],
\end{aligned}$$

where IC, PC and RP denote, respectively, incurred claims, paid claims and runoff profit. The balance equation for the claims reserve is $E_{t+1} = E_t + \mathrm{IC}_{t+1} - \mathrm{RP}_{t+1} - \mathrm{PC}_{t+1}$, where

$$\begin{aligned}
\mathrm{IC}_{t+1} - \mathrm{RP}_{t+1} - \mathrm{PC}_{t+1} &= \mathrm{E}\big[I_{t,2} + I_{t,3} + I_{t,4} \mid \mathcal{F}_{t+1}\big] - \mathrm{E}\big[I_{t-1,2} + I_{t-2,3} + I_{t-3,4} \mid \mathcal{F}_t\big]\\
&\quad + \mathrm{E}\big[I_{t-1,3} + I_{t-2,4} + I_{t-1,4} \mid \mathcal{F}_{t+1}\big] - \mathrm{E}\big[I_{t-1,3} + I_{t-2,4} + I_{t-1,4} \mid \mathcal{F}_t\big].
\end{aligned} \tag{2.1}$$

The profit or loss during year $t+1$ depends on changes in the reserves:

$$\mathrm{P\&L}_{t+1} = P_t N_{t+1} - \mathrm{PC}_{t+1} + \mathrm{IE}_{t+1} - \mathrm{OE}_{t+1} - (E_{t+1} - E_t + V_{t+1} - V_t),$$

where IE denotes investment earnings and OE denotes operating expenses. The dynamics of the surplus fund is therefore

$$G_{t+1} = G_t + \mathrm{P\&L}_{t+1} = G_t + \mathrm{EP}_{t+1} + \mathrm{IE}_{t+1} - \mathrm{OE}_{t+1} - \mathrm{IC}_{t+1} + \mathrm{RP}_{t+1}. \tag{2.2}$$

We consider three models of increasing complexity. The simple model allows us to solve the premium control problem with classical methods. In this situation, we can compare the results obtained with classical methods with the results obtained with more flexible methods, allowing the assessment of the performance of a chosen flexible method. Classical solution methods are not feasible for the intermediate model. However, the similarity between the simple and intermediate model allows us to understand how increasing model complexity affects the optimal premium rule. Finally, we consider a realistic model, where the models for the claims payments and investment earnings align closer with common distributional assumptions for these quantities. Since the simple model is a simplified version of the intermediate model, we begin by defining the intermediate model in Section 2.1, followed by the simple model in Section 2.2. In Section 2.3, the more realistic models for claims payments and investment earnings are defined.

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

2.1. Intermediate model

We choose to model the key random quantities as integer-valued random variables with conditional distributions that are either Poisson or Negative Binomial distributions. Other choices of distributions on the integers are possible without any major effects on the analysis that follows. Let

$$\mathcal{L}(N_{t+1} \mid \mathcal{F}_t) = \mathcal{L}(N_{t+1} \mid P_t) = \mathsf{Pois}\big(aP_t^b\big)\,, \tag{2.3}$$

where $a > 0$ is a constant, and $b < 0$ is the price elasticity of demand. The notation says that the conditional distribution of the number of contracts written during year $t+1$ given the information at the end of year t depends on that information only through the premium decided at the end of year t for those contracts.

Let $\widetilde{N}_{t+1} = (N_{t+1} + N_t)/2$ denote the number of contracts during year $t+1$ that provide coverage for accidents during year $t+1$. Let

$$\mathrm{OE}_{t+1} = \beta_0 + \beta_1 \widetilde{N}_{t+1}, \tag{2.4}$$

saying that the operating expenses have both a fixed part and a variable part proportional to the number of active contracts. The appearance of $\widetilde{N}_{t+1}$ instead of N_{t+1} in the expressions above is due to the assumption that contracts are written uniformly in time over the year and that accidents occur uniformly in time over the year.

Let $\alpha_1, \ldots, \alpha_4 \in [0,1]$ with $\sum_{i=1}^4 \alpha_i = 1$. The constant α_k is, for a given accident year, the expected fraction of claim costs paid during development year k. Let

$$\mathcal{L}\big(I_{t,k} \mid \mathcal{F}_t, \widetilde{N}_{t+1}\big) = \mathcal{L}\big(I_{t,k} \mid \widetilde{N}_{t+1}\big) = \mathsf{Pois}\big(\alpha_k \mu \widetilde{N}_{t+1}\big)\,, \tag{2.5}$$

where μ denotes the expected claim cost per contract. We assume that different incremental claims payments $I_{j,k}$ are conditionally independent given information about the corresponding numbers of contracts written. Formally, the elements in the set

$$\big\{I_{j,k} : j \in \{t-l, \ldots, t\}, k \in \{1, \ldots, 4\}\big\}$$

are conditionally independent given $N_{t-l}, \ldots, N_{t+1}$. Therefore, using (2.1) and (2.5),

$$\begin{aligned}
\mathcal{L}\big(\mathrm{PC}_{t+1} \mid \mathcal{F}_t, \widetilde{N}_{t+1}\big) &= \mathcal{L}\big(\mathrm{PC}_{t+1} \mid \widetilde{N}_{t-2}, \ldots, \widetilde{N}_{t+1}\big) = \mathsf{Pois}\big(\alpha_1 \mu \widetilde{N}_{t+1} + \cdots + \alpha_4 \mu \widetilde{N}_{t-2}\big)\,, \\
\mathrm{IC}_{t+1} - \mathrm{RP}_{t+1} &= \mathrm{PC}_{t+1} + (\alpha_2 + \alpha_3 + \alpha_4)\,\mu \widetilde{N}_{t+1} - \alpha_2 \mu \widetilde{N}_t - \alpha_3 \mu \widetilde{N}_{t-1} - \alpha_4 \mu \widetilde{N}_{t-2}.
\end{aligned} \tag{2.6}$$

The model for the investment earnings IE_{t+1} is chosen so that $G_t \le 0$ implies $\mathrm{IE}_{t+1} = 0$ since $G_t \le 0$ means that nothing is invested. Moreover, we assume that

$$\mathcal{L}(\mathrm{IE}_{t+1} + G_t \mid \mathcal{F}_t, G_t > 0) = \mathcal{L}(\mathrm{IE}_{t+1} + G_t \mid G_t, G_t > 0) = \mathsf{NegBin}\left(\nu G_t, \frac{1+\xi}{1+\xi+\nu}\right), \tag{2.7}$$

where $\mathsf{NegBin}(r,p)$ denotes the negative binomial distribution with probability mass function

$$k \mapsto \binom{k+r-1}{k}(1-p)^r p^k$$

which corresponds to mean and variance

$$\mathrm{E}[\mathrm{IE}_{t+1} + G_t \mid G_t, G_t > 0] = \frac{p}{1-p} r = (1+\xi) G_t,$$

$$\mathrm{Var}\,(\mathrm{IE}_{t+1} + G_t \mid G_t, G_t > 0) = \frac{p}{(1-p)^2} r = \frac{1+\xi+\nu}{\nu}(1+\xi) G_t.$$

Given a premium rule π that given the state $S_t = (G_t, P_{t-1}, N_{t-3}, N_{t-2}, N_{t-1}, N_t)$ generates the premium P_t, the system (S_t) evolves in a Markovian manner according to the transition probabilities that follows from (2.3)–(2.7) and (2.2). Notice that if we consider a less long-tailed insurance product so that $\alpha_3 = \alpha_4 = 0$ (at most one year delay from occurrence of the accident to final payment), then the dimension of the state space reduces to four, that is $S_t = (G_t, P_{t-1}, N_{t-1}, N_t)$.

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

2.2 Simple model

Consider the situation where the insurer has a fixed number N of policyholders, who at some initial time point bought insurance policies with automatic contract renewal for the price P_t year $t+1$. The state at time t is $S_t = (G_t, P_{t-1})$. In this simplified setting, $\mathrm{OE}_{t+1} = \beta_0 + \beta_1 N$, all payments $I_{t,k}$ are independent, $\mathcal{L}(I_{t,k}) = \mathsf{Pois}(\alpha_k \mu N)$, $\mathrm{IC}_{t+1} - \mathrm{RP}_{t+1} = \mathrm{PC}_{t+1}$ and $\mathcal{L}(\mathrm{PC}_{t+1}) = \mathsf{Pois}(\mu N)$.

2.3 Realistic model

In this model, we change the distributional assumptions for both investment earnings and the incremental claims payments from the previously used integer-valued distributions. Let (Z_t) be a sequence of iid standard normals and let

$$\mathcal{L}(\mathrm{IE}_{t+1} + G_t \mid G_t, G_t > 0) = \mathcal{L}(G_t \exp\{\widetilde{\mu} + \widetilde{\sigma} Z_{t+1}\} \mid G_t, G_t > 0)\,.$$

Let $C_{t,j}$ denote the cumulative claims payments for accidents occurring during year $t+1$ up to and including development year j. Hence, $I_{t,1} = C_{t,1}$, and $I_{t,j} = C_{t,j} - C_{t,j-1}$ for $j > 1$. We use the following model for the cumulative claims payments:

$$\begin{aligned} C_{t,1} &= c_0 \widetilde{N}_{t+1} \exp\left\{\nu_0 Z_{t,1} - \nu_0^2/2\right\}, \\ C_{t,j+1} &= C_{t,j} \exp\left\{\mu_j + \nu_j Z_{t,j+1}\right\}, \quad j = 1 \ldots, J-1, \end{aligned} \tag{2.8}$$

where c_0 is interpreted as the average claims payment per policyholder during the first development year, and $Z_{t,j}$ are iid standard normals. Then,

$$\begin{aligned} \mathrm{E}\left[C_{t,j+1} \mid C_{t,j}\right] &= C_{t,j} \exp\left\{\mu_j + \nu_j^2/2\right\}, \\ \mathrm{Var}\left(C_{t,j+1} \mid C_{t,j}\right) &= C_{t,j}^2 \left(\exp\left\{\nu_j^2\right\} - 1\right) \exp\left\{2\mu_j + \nu_j^2\right\}. \end{aligned}$$

We do not impose restrictions on the parameters μ_j and ν_j^2 and as a consequence values $f_j = \exp\left\{\mu_j + \nu_j^2/2\right\} \in (0, 1)$ are allowed (allowing for negative incremental paid amounts $C_{t,j+1} - C_{t,j} < 0$). This is in line with the model assumption $\mathrm{E}\left[C_{t,j+1} \mid C_{t,1}, \ldots, C_{t,j}\right] = f_j C_{t,j}$ for some $f_j > 0$ of the classical distribution-free Chain Ladder model by Mack (1993). Moreover, with $\mu_{0,t} = \log\left(c_0 \widetilde{N}_{t+1}\right) - \nu_0^2/2$,

$$\mathcal{L}\left(\log C_{t,1}\right) = \mathsf{N}\left(\mu_{0,t}, \nu_0^2\right), \quad \mathcal{L}\left(\log \frac{C_{t,j+1}}{C_{t,j}}\right) = \mathsf{N}\left(\mu_j, \nu_j^2\right). \tag{2.9}$$

Given an (incremental) development pattern $(\alpha_1, \ldots, \alpha_J)$, $\sum_{j=1}^{J} \alpha_j = 1$, $\alpha_j \geq 0$,

$$\alpha_1 = \frac{1}{\prod_{k=1}^{J-1} f_k}, \quad \alpha_j = \frac{(f_{j-1} - 1) \prod_{k=1}^{j-2} f_k}{\prod_{k=1}^{J-1} f_k}, \quad j = 2, \ldots, J,$$

with the convention $\prod_{s=a}^{b} c_s = 1$ if $a > b$, where $f_j = \exp\{\mu_j + \nu_j^2/2\}$. We have

$$\begin{aligned} \mathrm{IC}_{t+1} &= C_{t,1} \prod_{k=1}^{J-1} f_k, \quad \mathrm{PC}_{t+1} = C_{t,1} + \sum_{k=1}^{J-1} \left(C_{t-k,k+1} - C_{t-k,k}\right), \\ \mathrm{RP}_{t+1} &= \sum_{k=1}^{J-1} \left(C_{t-k,k} f_k - C_{t-k,k+1}\right) \prod_{j=k+1}^{J-1} f_j. \end{aligned}$$

The state is $S_t = \left(G_t, P_{t-1}, N_{t-1}, N_t, C_{t-1,1}, \ldots, C_{t-J+1,J-1}\right)$.

3. The control problem

We consider a set of states $\mathcal{S}^+$, a set of non-terminal states $\mathcal{S} \subseteq \mathcal{S}^+$, and for each $s \in \mathcal{S}$ a set of actions $\mathcal{A}(s)$ available from state s, with $\mathcal{A} = \cup_{s \in \mathcal{S}} \mathcal{A}(s)$. We assume that $\mathcal{A}$ is discrete (finite or countable).

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

In order to simplify notation and limit the need for technical details, we will here and in Section 4 restrict our presentation to the case where $\mathcal{S}^+$ is also discrete. However, we emphasise that when using function approximation in Section 4.2 the update Equation (4.3) for the weight vector is still valid when the state space is uncountable, as is the case for the realistic model. For each $s \in \mathcal{S}$, $s' \in \mathcal{S}^+$, $a \in \mathcal{A}(s)$ we define the reward received after taking action a in state s and transitioning to s', $-f(a, s, s')$, and the probability of transitioning from state s to state s' after taking action a, $p(s'|s, a)$. We assume that rewards and transition probabilities are stationary (time-homogeneous). This defines a Markov decision process (MDP). A policy π specifies how to determine what action to take in each state. A stochastic policy describes, for each state, a probability distribution on the set of available actions. A deterministic policy is a special case of a stochastic policy, specifying a degenerate probability distribution, that is a one-point distribution.

Our objective is to find the premium policy that minimises the expected value of the premium payments over time, but that also results in (P_t) being more averaged over time, and further ensures that the surplus (G_t) is large enough so that the risk that the insurer cannot pay the claim costs and other expenses is small. By combining rewards with either constraints on the available actions from each state or the definition of terminal states, this will be accomplished with a single objective function, see further Sections 3.1–3.2. We formulate this in terms of a MDP, that is we want to solve the following optimisation problem:

$$\underset{\pi}{\text{minimise}}\ \mathrm{E}_\pi\left[\sum_{t=0}^{T} \gamma^t f(P_t, S_t, S_{t+1}) \mid S_0 = s\right], \tag{3.1}$$

where π is a policy generating the premium P_t given the state S_t, $\mathcal{A}(s)$ is the set of premium levels available from state s, γ is the discount factor, f is the cost function, and $\mathrm{E}_\pi[\cdot]$ denotes the expectation given that policy π is used. Note that the discount factor γ^t should not be interpreted as the price of a zero-coupon bond maturing at time t, since the cost that is discounted does not represent an economic cost. Instead γ reflects how much weight is put on costs that are immediate compared to costs further in the future. The transition probabilities are

$$p\big(s'|s, a\big) = \mathrm{P}(S_{t+1} = s' \mid S_t = s, P_t = a),$$

and we consider stationary policies, letting $\pi(a|s)$ denote the probability of taking action a in state s under policy π,

$$\pi(a|s) = \mathrm{P}_\pi(P_t = a \mid S_t = s).$$

If there are no terminal states, we have $T = \infty$, and $\mathcal{S}^+ = \mathcal{S}$. We want to choose $\mathcal{A}(s)$, $s \in \mathcal{S}$, f, and any terminal states such that the objective discussed above is achieved. We will do this in two ways, see Sections 3.1 and 3.2.

The value function of state s under a policy π generating the premium P_t is defined as

$$v_\pi(s) = \mathrm{E}_\pi\left[\sum_{t=0}^{T} \gamma^t(-f(P_t, S_t, S_{t+1})) \mid S_0 = s\right].$$

The Bellman equation for the value function is

$$v_\pi(s) = \sum_{a\in\mathcal{A}(s)} \pi(a|s) \sum_{s'\in\mathcal{S}} p\big(s'|s, a\big)\left(-f(a, s, s') + \gamma v_\pi(s')\right).$$

When the policy is deterministic, we let π be a mapping from $\mathcal{S}$ to $\mathcal{A}$, and

$$v_\pi(s) = \sum_{s'\in\mathcal{S}} p(s'|s, \pi(s))\big(-f(\pi(s), s, s') + \gamma v_\pi(s')\big).$$

The optimal value function is $v_*(s) = \sup_\pi v_\pi(s)$. When the action space is finite, the supremum is attained, which implies the existence of an optimal deterministic stationary policy (see Puterman, 2005,

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

Cor. 6.2.8, for other sufficient conditions for attainment of the supremum, see Puterman, 2005, Thm. 6.2.10). Hence, if the transition probabilities are known, we can use the Bellman optimality equation to find $v_*(s)$:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}} p\big(s'|s,a\big)\,\big(-f(a,s,s') + \gamma v_*(s')\big).$$

We use policy iteration in order to find the solution numerically. Let $k = 0$, and choose some initial deterministic policy $\pi_k(s)$ for all $s \in \mathcal{S}$. Then

(i) Determine $V_k(s)$ as the unique solution to the system of equations

$$V_k(s) = \sum_{s' \in \mathcal{S}} p\big(s'|s,\pi_k(s)\big)\,\Big(-f\big(\pi_k(s),s,s'\big) + \gamma V_k(s')\Big).$$

(ii) Determine an improved policy $\pi_{k+1}(s)$ by computing

$$\pi_{k+1}(s) = \operatorname*{argmax}_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}} p\big(s'|s,a\big)\,\Big(-f(a,s,s') + \gamma V_k(s')\Big).$$

(iii) If $\pi_{k+1}(s) \neq \pi_k(s)$ for some $s \in \mathcal{S}$, then increase k by 1 and return to step (i).

Note that if the state space is large enough, solving the system of equations in step (i) directly might be too time-consuming. In that case, this step can be solved by an additional iterative procedure, called iterative policy evaluation, see for example Sutton and Barto (2018, Ch. 4.1).

3.1. MDP with constraint on the action space

The premiums (P_t) will be averaged if we minimise $\sum_t c(P_t)$, where c is an increasing, strictly convex function. Thus for the first MDP, we let $f(a,s,s') = c(a)$. To ensure that the surplus (G_t) does not become negative too often, we combine this with the constraint saying that the premium needs to be chosen so that the expected value, given the current state, of the surplus stays nonnegative, that is

$$\mathcal{A}(S_t) = \{P_t : \mathrm{E}_\pi[G_{t+1} \mid S_t] \geq 0\}, \tag{3.2}$$

and the optimisation problem becomes

$$\underset{\pi}{\text{minimise}}\ \mathrm{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t c(P_t) \mid S_0 = s\right] \quad \text{subject to } \mathrm{E}_\pi[G_{t+1} \mid S_t] \geq 0 \text{ for all } t. \tag{3.3}$$

The choice of the convex function c, together with the constraint, will affect how quickly the premium can be lowered as the surplus or previous premium increases, and how quickly the premium must be increased as the surplus or previous premium decreases. Different choices of c affect how well different parts of the objective are achieved. Hence, one choice of c might put a higher emphasis on the premium being more averaged over time but slightly higher, while another choice might promote a lower premium level that is allowed to vary a bit more from one time point to another. Furthermore, it is not clear from the start what choice of c will lead to a specific result, thus designing the reward signal might require searching through trial and error for the cost function that achieves the desired result.

3.2. MDP with a terminal state

The constraint (3.2) requires a prediction of N_{t+1} according to (2.3). However, estimating the price elasticity in (2.3) is difficult task; hence, it would be desirable to solve the optimisation problem without having to rely on this prediction. To this end, we remove the constraint on the action space, that is we let $\mathcal{A}(s) = \mathcal{A}$ for all $s \in \mathcal{S}$, and instead introduce a terminal state which has a larger negative reward than all other states. This terminal state is reached when the surplus G_t is below some predefined level, and

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

it can be interpreted as the state where the insurer defaults and has to shut down. If we let $\mathcal{G}$ denote the set of non-terminal states for the first state variable (the surplus), then

$$f(P_t, S_t, S_{t+1}) = h(P_t, S_{t+1}) = \begin{cases} c(P_t), & \text{if } G_{t+1} \geq \min \mathcal{G}, \\ c(\max \mathcal{A})(1+\eta), & \text{if } G_{t+1} < \min \mathcal{G}, \end{cases}$$

where $\eta > 0$. The optimisation problem becomes

$$\underset{\pi}{\text{minimise}} \ \mathrm{E}_\pi\left[\sum_{t=0}^{T} \gamma^t h(P_t, S_{t+1}) \mid S_0 = s\right], \quad T = \min\{t : G_{t+1} < \min \mathcal{G}\}. \tag{3.4}$$

The reason for choosing $\eta > 0$ is to ensure that the reward when transitioning to the terminal state is lower than the reward when using action $\max \mathcal{A}$ (the maximal premium level), that is, it should be more costly to terminate and restart compared with attempting to increase the surplus when the surplus is low. The particular value of the parameter $\eta > 0$ together with the choice of the convex function c determines the reward signal, that is the compromise between minimising the premium, averaging the premium and ensuring that the risk of default is low. One way of choosing η is to set it high enough so that the reward when terminating is lower than the total reward using any other policy. Then, we require that

$$(1+\eta)c(\max \mathcal{A}) > \sum_{t=0}^{\infty} \gamma^t c(\max \mathcal{A}) = \frac{1}{1-\gamma} c(\max \mathcal{A}),$$

that is $\eta > \gamma/(1-\gamma)$. This choice of η will put a higher emphasis on ensuring that the risk of default is low, compared with using a lower value of η.

3.3. Choice of cost function

The function c is chosen to be an increasing, strictly convex function. That it is increasing captures the objective of a low premium. As discussed in Martin-Löf (1994), that it is convex means that the premiums will be more averaged, since

$$\frac{1}{T}\sum_{t=1}^{T} c(p_t) \geq c\left(\frac{1}{T}\sum_{t=1}^{T} p_t\right),$$

The more convex shape the function has, the more stable the premium will be over time. One could also force stability by adding a term related to the absolute value of the difference between successive premium levels to the cost function. We have chosen a slightly simpler cost function, defined by c, and for the case with terminal states, by the parameter η.

As for the specific choice of the function c used in Section 5, we have simply used the function suggested in Martin-Löf (1994), but with slightly adjusted parameter values. That the function c, together with the constraint or the choice of terminal states and the value of η, leads to the desired goal of a low, stable premium and a low probability of default needs to be determined on a case by case basis, since we have three competing objectives, and different insurers might put different weight on each of them. This is part of designing the reward function. Hence, adjusting c and η will change how much weight is put on each of the three objectives, and the results in Section 5 can be used as basis for adjustments.

4. Reinforcement learning

If the model of the environment is not fully known, or if the state space or action space are not finite, the control problem can no longer be solved by classical dynamic programming approaches. Instead, we can utilise different reinforcement learning algorithms.

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

4.1. Temporal-difference learning

Temporal-difference (TD) methods can learn directly from real or simulated experience of the environment. Given a specific policy π which determines the action taken in each state, and the sampled or observed state at time t, S_t, state at time $t+1$, S_{t+1}, and reward R_{t+1}, the iterative update for the value function, using the one-step TD method, is

$$V(S_t) \leftarrow V(S_t) + \alpha_t\big(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\big),$$

where α_t is a step size parameter. Hence, the target for the TD update is $R_{t+1} + \gamma V(S_{t+1})$. Thus, we update $V(S_t)$, which is an estimate of $v_\pi(S_t) = \mathrm{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t]$, based on another estimate, namely $V(S_{t+1})$. The intuition behind using $R_{t+1} + \gamma V(S_{t+1})$ as the target in the update is that this is a slightly better estimate of $v_\pi(S_t)$, since it consists of an actual (observed or sampled) reward at $t+1$ and an estimate of the value function at the next observed state.

It has been shown in for example Dayan (1992) that the value function (for a given policy π) computed using the one-step TD method converges to the true value function if the step size parameter $0 \le \alpha_t \le 1$ satisfies the following stochastic approximation conditions

$$\sum_{k=1}^{\infty} \alpha_{t^k(s)} = \infty, \quad \sum_{k=1}^{\infty} \alpha^2_{t^k(s)} < \infty, \quad \text{for all } s \in \mathcal{S},$$

where $t^k(s)$ is the time step when state s is visited for the kth time.

4.1.1. TD control algorithms

The one-step TD method described above gives us an estimate of the value function for a given policy π. To find the optimal policy using TD learning, a TD control algorithm, such as SARSA or Q-learning, can be used. The goal of these algorithms is to estimate the optimal action-value function $q_*(s,a) = \max_\pi q_\pi(s,a)$, where q_π is the action-value function for policy π,

$$q_\pi(s,a) = \mathrm{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s, A_0 = a\right].$$

To keep a more streamlined presentation, we will here focus on the algorithm SARSA. The main reason for this has to do with the topic of the next section, namely function approximation. While there are some convergence results for SARSA with function approximation, there are none for standard Q-learning with function approximation. In fact, there are examples of divergence when combining off-policy training (as is done in Q-learning) with function approximation, see for example Sutton and Barto (2018, Ch. 11). However, some numerical results for the simple model with standard Q-learning can be found in Section 5, and we do provide complete details on Q-learning in the Supplemental Material, Section 2.

The iterative update for the action-value function, using SARSA, is

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha_t(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)) .$$

Hence, we need to generate transitions from state-action pairs (S_t, A_t) to state-action pairs (S_{t+1}, A_{t+1}) and observe the rewards R_{t+1} obtained during each transition. To do this, we need a behaviour policy, that is a policy that determines which action is taken in the state we are currently in when the transitions are generated. Thus, SARSA gives an estimate of the action-value function q_π given the behaviour policy π. Under the condition that all state-action pairs continue to be updated, and that the behaviour policy is greedy in the limit, it has been shown in Singh et al. (2000) that SARSA converges to the true optimal action-value function if the step size parameter $0 \le \alpha_t \le 1$ satisfies the following stochastic approximation conditions

$$\sum_{k=1}^{\infty} \alpha_{t^k(s,a)} = \infty, \quad \sum_{k=1}^{\infty} \alpha^2_{t^k(s,a)} < \infty, \quad \text{for all } s \in \mathcal{S}, a \in \mathcal{A}(s), \tag{4.1}$$

where $t^k(s,a)$ is the time step when a visit in state s is followed by taking action a for the kth time.

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

To ensure that all state-action pairs continue to be updated, the behaviour policy needs to be exploratory. At the same time, we want to exploit what we have learned so far by choosing actions that we believe will give us large future rewards. A common choice of policy that compromises in this way between exploration and exploitation is the ε-greedy policy, which with probability $1-\varepsilon$ chooses the action that maximises the action-value function in the current state, and with probability ε chooses any other action uniformly at random:

$$\pi(a|s) = \begin{cases} 1-\varepsilon, & \text{if } a = \text{argmax}_a\, Q(s,a), \\ \dfrac{\varepsilon}{|\mathcal{A}|-1}, & \text{otherwise.} \end{cases}$$

Another example is the softmax policy

$$\pi(a|s) = \frac{e^{Q(s,a)/\tau}}{\sum_{a\in\mathcal{A}(s)} e^{Q(s,a)/\tau}}.$$

To ensure that the behaviour policy π is greedy in the limit, it needs to be changed over time towards the greedy policy that maximises the action-value function in each state. This can be accomplished by letting ε or τ slowly decay towards zero.

4.2. Function approximation

The methods discussed thus far are examples of tabular solution methods, where the value functions can be represented as tables. These methods are suitable when the state and action space are not too large, for example for the simple model in Section 2.2. However, when the state space and/or action space is very large, or even continuous, these methods are not feasible, due to not being able to fit tables of this size in memory, and/or due to the time required to visit all state-action pairs multiple times. This is the case for the intermediate and realistic models presented in Sections 2.1 and 2.3. In both models, we allow the number of contracts written per year to vary, which increases the dimension of the state space. For the intermediate model, it also has the effect that the surplus process, depending on the parameter values chosen, can take non-integer values. For the simple model $\mathcal{S} = \mathcal{G} \times \mathcal{A}$, and with the parameters chosen in Section 5, we have $|\mathcal{G}| = 171$ and $|\mathcal{A}| = 100$. For the intermediate model, if we let $\mathcal{N}$ denote the set of integer values that N_t is allowed to take values in, then $\mathcal{S} = \mathcal{G} \times \mathcal{A} \times \mathcal{N}^l$, where l denotes the maximum number of development years. With the parameters chosen in Section 5, the total number of states is approximately 10^8 for the intermediate model. For the realistic model, several of the state variables are continuous, that is the state space is no longer finite.

Thus, to solve the optimisation problem for the intermediate and the realistic model, we need approximate solution methods, in order to generalise from the states that have been experienced to other states. In approximate solution methods, the value function $v_\pi(s)$ (or action-value function $q_\pi(s,a)$) is approximated by a parameterised function, $\hat{v}(s;w)$ (or $\hat{q}(s,a;w)$). When the state space is discrete, it is common to minimise the following objective function,

$$J(w) = \sum_{s\in\mathcal{S}} \mu_\pi(s)\big(v_\pi(s) - \hat{v}(s;w)\big)^2, \tag{4.2}$$

where $\mu_\pi(s)$ is the fraction of time spent in state s. For the model without terminal states, μ_π is the stationary distribution under policy π. For the model with terminal states, to determine the fraction of time spent in each transient state, we need to compute the expected number of visits $\eta_{\lambda,\pi}(s)$ to each transient state $s \in \mathcal{S}$ before reaching a terminal (absorbing) state, where $\lambda(s) = \mathrm{P}(S_0 = s)$ is the initial distribution. For ease of notation, we omit λ from the subscript below, and write η_π and P_π instead of $\eta_{\lambda,\pi}$ and $\mathrm{P}_{\lambda,\pi}$. Let $p(s|s')$ be the probability of transitioning from state s' to state s under policy π, that is $p(s|s') = \mathrm{P}_\pi(S_t = s \mid S_{t-1} = s')$. Then,

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

$$\begin{aligned}\eta_\pi(s) &= \mathrm{E}_\pi\left[\sum_{t=0}^{\infty} \mathbf{1}_{\{S_t=s\}}\right] = \lambda(s) + \sum_{t=1}^{\infty} \mathrm{P}_\pi(S_t = s) \\ &= \lambda(s) + \sum_{t=1}^{\infty}\sum_{s'\in\mathcal{S}} p(s|s')\,\mathrm{P}_\pi(S_{t-1}=s') = \lambda(s) + \sum_{s'\in\mathcal{S}} p(s|s') \sum_{t=0}^{\infty} \mathrm{P}_\pi\,(S_t = s') \\ &= \lambda(s) + \sum_{s'\in\mathcal{S}} p(s|s')\eta_\pi(s'),\end{aligned}$$

or, in matrix form, $\eta_\pi = \lambda + P^\top \eta_\pi$, where P is the part of the transition matrix corresponding to transitions between transient states. If we label the states $0, 1, \ldots, |\mathcal{S}|$ (where state 0 represents all terminal states), then $P = (p_{ij} : i, j \in \{1, 2, \ldots, |\mathcal{S}|\})$, where $p_{ij} = p(j \mid i)$. After solving this system of equations, the fraction of time spent in each transient state under policy π can be computed according to

$$\mu_\pi(s) = \frac{\eta_\pi(s)}{\sum_{s'\in\mathcal{S}} \eta_\pi(s')}, \quad \text{for all } s \in \mathcal{S}.$$

This computation of μ_π relies on the model of the environment being fully known and the transition probabilities explicitly computable, as is the case for the simple model in Section 2.2. However, for the situation at hand, where we need to resort to function approximation and determine $\hat{v}(s; w)$ (or $\hat{q}(s, a; w)$) by minimising (4.2), we cannot explicitly compute μ_π. Instead, μ_π in (4.2) is captured by learning incrementally from real or simulated experience, as in semi-gradient TD learning. Using semi-gradient TD learning, the iterative update for the weight vector w becomes

$$w_{t+1} = w_t + \alpha_t\big(R_{t+1} + \gamma\,\hat{v}(S_{t+1}; w_t) - \hat{v}(S_t; w_t)\big)\,\nabla\hat{v}(S_t; w_t)\,.$$

This update can be used to estimate v_π for a given policy π, generating transitions from state to state by taking actions according to this policy. Similarly to standard TD learning (Section 4.1), the target $R_{t+1} + \gamma\,\hat{v}(S_{t+1}; w_t)$ is an estimate of the true (unknown) $v_\pi(S_{t+1})$. The name "semi-gradient" comes from the fact that the update is not based on the true gradient of $\big(R_{t+1} + \gamma\,\hat{v}(S_{t+1}; w_t) - \hat{v}(S_t; w_t)\big)^2$; instead, the target is seen as fixed when the gradient is computed, despite the fact that it depends on the weight vector w_t.

As in the previous section, estimating the value function given a specific policy is not our final goal – instead we want to find the optimal policy. Hence, we need a TD control algorithm with function approximation. One example of such an algorithm is semi-gradient SARSA, which estimates q_π. The iterative update for the weight vector is

$$w_{t+1} = w_t + \alpha_t\big(R_{t+1} + \gamma\,\hat{q}(S_{t+1}, A_{t+1}; w_t) - \hat{q}(S_t, A_t; w_t)\big)\,\nabla\hat{q}(S_t, A_t; w_t)\,. \tag{4.3}$$

As with standard SARSA, we need a behaviour policy that generates transitions from state-action pairs to state-action pairs, that both explores and exploits, for example an ε-greedy or softmax policy. Furthermore, for the algorithm to estimate q_* we need the behaviour policy to be changed over time towards the greedy policy. However, convergence guarantees only exist when using linear function approximation, see Section 4.2.1 below.

4.2.1. Linear function approximation

The simplest form of function approximation is linear function approximation. The value function is approximated by $\hat{v}(s; w) = w^\top x(s)$, where $x(s)$ are basis functions. Using the Fourier basis as defined in Konidaris et al. (2011), the ith basis function for the Fourier basis of order n is (here $\pi \approx 3.14$ is a number)

$$x_i(s) = \cos\big(\pi\, s^\top c^{(i)}\big)\,,$$

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

where $s = (s_1, s_2, \ldots, s_k)^\top$, $c^{(i)} = \left(c_1^{(i)}, \ldots, c_k^{(i)}\right)^\top$, and k is the dimension of the state space. The $c^{(i)}$'s are given by the k-tuples over the set $\{0, \ldots, n\}$, and hence, $i = 1, \ldots, (n+1)^k$. This means that $x(s) \in \mathbb{R}^{(n+1)^k}$. One-step semi-gradient TD learning with linear function approximation has been shown to converge to a weight vector w^*. However, w^* is not necessarily a minimiser of J. Tsitsiklis and Van Roy (1997) derive the upper bound

$$J(w^*) \leq \frac{1}{1-\gamma} \min_w J(w).$$

Since γ is often close to one, this bound can be quite large.

Using linear function approximation for estimating the action-value function, we have $\hat{q}(s, a; w) = w^\top x(s, a)$, and the ith basis function for the Fourier basis of order n is

$$x_i(s, a) = \cos\left(\pi \left(s^\top c_{1:k}^{(i)} + a c_{k+1}^{(i)}\right)\right),$$

where $s = (s_1, \ldots, s_k)^\top$, $c_{1:k}^{(i)} = \left(c_1^{(i)}, \ldots, c_k^{(i)}\right)^\top$, $c_j^{(i)} \in \{0, \ldots, n\}, j = 1, \ldots, k+1$, and $i = 1, \ldots, (n+1)^{k+1}$.

The convergence results for semi-gradient SARSA with linear function approximation depend on what type of policy is used in the algorithm. When using an ε-greedy policy, the weights have been shown to converge to a bounded region and might oscillate within that region, see Gordon (2001). Furthermore, Perkins and Precup (2003) have shown that if the policy improvement operator Γ is Lipschitz continuous with constant $L > 0$ and ε-soft, then SARSA will converge to a unique policy. The policy improvement operator maps every $q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ to a stochastic policy and gives the updated policy after iteration t as $\pi_{t+1} = \Gamma(q^{(t)})$, where $q^{(t)}$ corresponds to a vectorised version of the state-action values after iteration t, that is $q^{(t)} = xw_t$ for the case where we use linear function approximation, where $x \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times d}$ is a matrix with rows $x(s, a)^\top$, for each $s \in \mathcal{S}, a \in \mathcal{A}$, and d is the number of basis functions. That Γ is Lipschitz continuous with constant L means that $\|\Gamma(q) - \Gamma(q')\|_2 \leq L\|q - q'\|_2$, for all $q, q' \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$. That Γ is ε-soft means that it produces a policy $\pi = \Gamma(q)$ that is ε-soft, that is $\pi(a|s) \geq \varepsilon/|\mathcal{A}|$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. In both Gordon (2001) and Perkins and Precup (2003), the policy improvement operator was not applied at every time step; hence, it is not the online SARSA-algorithm considered in the present paper that was investigated. The convergence of online SARSA under the assumption that the policy improvement operator is Lipschitz continuous with a small enough constant L was later shown in Melo et al. (2008). The softmax policy is Lipschitz continuous, see further the Supplemental Material, Section 3.

However, the value of the Lipschitz constant L that ensures convergence depends on the problem at hand, and there is no guarantee that the policy the algorithm converges to is optimal. Furthermore, for SARSA to approximate the optimal action-value function, we need the policy to get closer to the greedy policy over time, for example by decreasing the temperature parameter when using a softmax policy. Thus, the Lipschitz constant L, which is inversely proportional to the temperature parameter, will increase as the algorithm progresses, making the convergence results in Perkins and Precup (2003) and Melo et al. (2008) less likely to hold. As discussed in Melo et al. (2008), this is not an issue specific to the softmax policy. Any Lipschitz continuous policy that over time gets closer to the greedy policy will in fact approach a discontinuous policy, and hence, the Lipschitz constant of the policy might eventually become too large for the convergence result to hold. Furthermore, the results in Perkins and Precup (2003) and Melo et al. (2008) are not derived for a Markov decision process with an absorbing state. Despite this, it is clear from the numerical results in Section 5 that a softmax policy performs substantially better compared to an ε-greedy policy, and for the simple model approximates the true optimal policy well.

The convergence results in Gordon (2001), Perkins and Precup (2003) and Melo et al. (2008) are based on the stochastic approximation conditions

$$\sum_{t=1}^{\infty} \alpha_t = \infty, \quad \sum_{t=1}^{\infty} \alpha_t^2 < \infty, \tag{4.4}$$

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

where α_t is the step size parameter used at time step t. Note that when using tabular methods (see Section 4.1), we had a vector of step sizes for each state-action pair. Here, this is not the case. This is a consequence of both that a vector of this size might not be possible to store in memory when the state space is large, and that we want to generalise from state-action pairs visited to other state-action pairs that are rarely/never visited, making the number of visits to each state-action pair less relevant.

5. Numerical illustrations

5.1. Simple model

We use the following parameter values: $\gamma = 0.9$, $N = 10$, $\mu = 5$, $\beta_0 = 10$, $\beta_1 = 1$, $\xi = 0.05$, and $\nu = 1$. This means that the expected yearly total cost for the insurer is 70 and the expected yearly cost per customer is 7. We emphasise that parameter values are meant to be interpreted in suitable units to fit the application in mind. The cost function is

$$c(p) = p + c_1 \left(c_2^p - 1\right), \tag{5.1}$$

with $c_1 = 1$ and $c_2 = 1.2$. For the model with a terminal state, we use $\eta = 10 > \gamma/(1-\gamma)$, as suggested in Section 3.2. The premium level is truncated and discretised according to $\mathcal{A} = \{0.2, 0.4, \ldots, 19.8, 20.0\}$. As the model for the MDP is formulated, $\mathrm{P}(G_t = g) > 0$ for all integers g. However, for actions/premiums that are considered with the aim of solving the optimisation problem, it will be sufficient to only consider a finite range of integer values for G_t since transitions to values outside this range will have negligible probability. Specifically, we will only consider $\{-20, -19, \ldots, 149, 150\}$ as possible surplus values. In order to ensure that transition probabilities sum to one, we must adjust the probabilities of transitions to the limiting surplus values according to the original probabilities of exiting the range of possible surplus values. For details on the transition probabilities and truncation for the simple model, see the Supplemental Material, Section 1.

Remark 5.1 (Cost function). The cost function (5.1) was suggested in Martin-Löf (1994), since it is an increasing, convex function, and thus will lead to the premium being more averaged over time. However, in Martin-Löf (1994), $c_2 = 1.5$ was used in the calculations. We have chosen a slightly lower value of c_2 due to that too extreme rewards can lead to numerical problems when using SARSA with linear function approximation.

Remark 5.2 (Truncation). Truncating the surplus process at 150 does not have a material effect on the optimal policy. However, the minimum value (here -20) will have an effect on the optimal policy for the MDP with a terminal state and should be seen as another parameter value that needs to be chosen to determine the reward signal, see Section 3.2.

5.1.1. Policy iteration

The top row in Figure 1 shows the optimal policy and the stationary distribution under the optimal policy, for the simple model with a constraint on the action space (Section 3.1) using policy iteration. The bottom row in Figure 1 shows the optimal policy and the fraction of time spent in each state under the optimal policy, for the simple model with terminal state (Section 3.2) using policy iteration. In both cases, the premium charged increases as the surplus or the previously charged premium decreases. Based on the fraction of time spent in each state under each of these two policies, we note that in both cases the average premium level is close to the expected cost per contract (7), but the average surplus level is slightly lower when using the policy for the model with a constraint on the action space compared to when using the policy for the model with the terminal state. However, the policies obtained for these two models are quite similar, and since (as discussed in Section 3.2) the model with the terminal state is more appropriate in more realistic settings, we focus the remainder of the analysis only on the model with the terminal state.

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

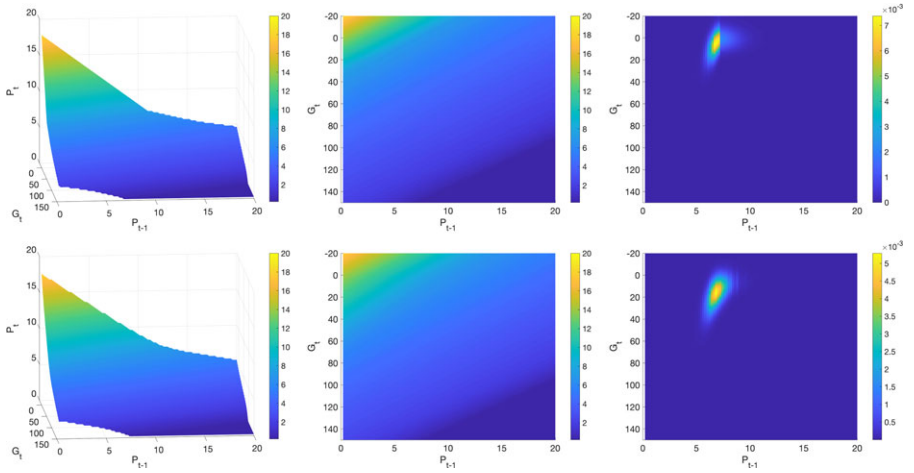

Figure 1. *Simple model using policy iteration. Top: with constraint. Bottom: with terminal state. First and second column: optimal policy. Third column: fraction of time spent in each state under the optimal policy.*

5.1.2 Linear function approximation

We have a 2-dimensional state space, and hence, $k+1=3$. When using the Fourier basis we should have $s \in [0,1]^k$, $a \in [0,1]$; hence, we rescale the inputs according to

$$\tilde{s}_1 = \frac{s_1 - \min \mathcal{G}}{\max \mathcal{G} - \min \mathcal{G}}, \quad \tilde{s}_2 = \frac{s_2 - \min \mathcal{A}}{\max \mathcal{A} - \min \mathcal{A}}, \quad \tilde{a} = \frac{a - \min \mathcal{A}}{\max \mathcal{A} - \min \mathcal{A}},$$

and use $(\tilde{s}_1, \tilde{s}_2, \tilde{a})^\top$ as input. We use a softmax policy, that is

$$\pi(a|s) = \frac{e^{\hat{q}(s,a;\boldsymbol{w})/\tau}}{\sum_{a \in \mathcal{A}(s)} e^{\hat{q}(s,a;\boldsymbol{w})/\tau}},$$

where τ is slowly decreased according to

$$\tau_t = \max\left\{\tau_{\min}, \tau_0 d^{t-1}\right\}, \quad \tau_0 = 2, \quad \tau_{\min} = 0.02, \quad d = 0.99999,$$

where τ_t is the parameter used during episode t. This schedule for decreasing the temperature parameter is somewhat arbitrary, and the parameters have not been tuned. The choice of a softmax policy is based on the results in Perkins and Precup (2003), Melo et al. (2008), discussed in Section 4.2.1. Since a softmax policy is Lipschitz continuous, convergence of SARSA to a unique policy is guaranteed, under the condition that the policy is also ε-soft and that the Lipschitz constant L is small enough. However, since the temperature parameter τ is slowly decreased, the policy chosen is not necessarily ε-soft for all states and time steps, and the Lipschitz constant increases as τ decreases. Despite this, our results show that the algorithm converges to a policy that approximates the optimal policy derived with policy iteration well when using a 3rd order Fourier basis, see the first column in Figure 2. The same cannot be said for an ε-greedy policy. In this case, the algorithm converges to a policy that in general charges a higher premium than the optimal policy derived with policy iteration, see the fourth column in Figure 2. For the ε-greedy policy, we decrease the parameter according to

$$\varepsilon_t = \max\left\{\varepsilon_{\min}, \varepsilon_0 d^{t-1}\right\}, \quad \varepsilon_0 = 0.2, \quad \varepsilon_{\min} = 0.01,$$

where ε_t is the parameter used during episode t.

The starting state is selected uniformly at random from $\mathcal{S}$. Furthermore, since discounting will lead to rewards after a large number of steps having a very limited effect on the total reward, we run each episode for at most 100 steps, before resetting to a starting state, again selected uniformly at random from $\mathcal{S}$.

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

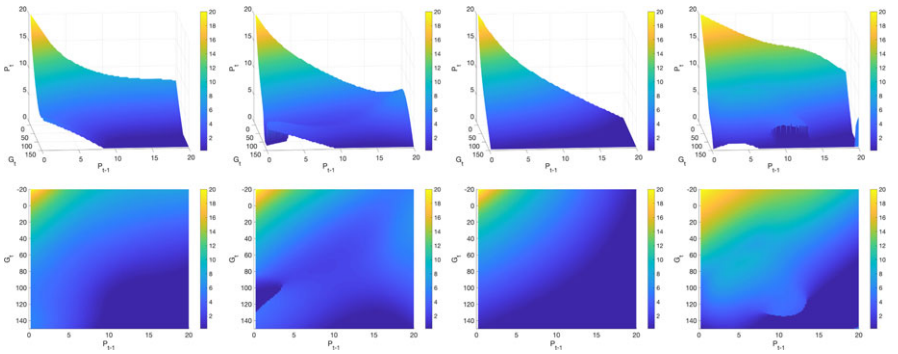

Figure 2. *Optimal policy for simple model with terminal state using linear function approximation. First column: 3rd order Fourier basis. Second column: 2nd order Fourier basis. Third column: 1st order Fourier basis. Fourth column: 3rd order Fourier basis with ε-greedy policy.*

This has the benefit of diversifying the states experienced, enabling us to achieve an approximate policy that is closer to the policy derived with dynamic programming as seen over the whole state space. The step size parameter used is

$$\alpha_t = \min\left\{\alpha_0, \frac{1}{t^{0.5+\theta}}\right\}, \tag{5.2}$$

where α_t is the step size parameter used during episode t, and $0 < \theta \leq 0.5$. The largest α_0 that ensures that the weights do not explode can be found via trial and error. However, the value of α_0 obtained in this way coincides with the "rule of thumb" for setting the step size parameter suggested in Sutton and Barto (2018, Ch. 9.6), namely

$$\alpha_0 = \frac{1}{\mathrm{E}_\pi\left[x^\top x\right]}, \quad \mathrm{E}_\pi\left[x^\top x\right] = \sum_{s,a} \mu_\pi(s)\pi(a|s)x(s,a)^\top x(s,a).$$

If $x(S_t, A_t)^\top x(S_t, A_t) \approx \mathrm{E}_\pi\left[x^\top x\right]$, then this step size ensures that the error (i.e., the difference between the updated estimate $w_{t+1}^\top x(S_t, A_t)$ and the target $R_{t+1} + \gamma w_t^\top x(S_{t+1}, A_{t+1})$) is reduced to zero after one update. Hence, using a step size larger than $\alpha_0 = \left(\mathrm{E}_\pi\left[x^\top x\right]\right)^{-1}$ risks overshooting the optimum, or even divergence of the algorithm. When using the Fourier basis of order n, this becomes for the examples studied here

$$\mathrm{E}_\pi\left[x^\top x\right] = \sum_{s,a} \mu_\pi(s)\pi(a|s) \sum_{i=1}^{(n+1)^{k+1}} \cos^2\left(\pi(sc_{1:k}^{(i)} + ac_{k+1}^{(i)})\right) \approx \frac{(n+1)^{k+1}}{2}.$$

For the simple model, we have used $\alpha_0 = 0.2$ for $n = 1$, $\alpha_0 = 0.07$ for $n = 2$, and $\alpha_0 = 0.03$ for $n = 3$. For the intermediate model, we used $\alpha_0 = 0.06$ for $n = 1$, $\alpha_0 = 0.008$ for $n = 2$, and $\alpha_0 = 0.002$ for $n = 3$. For the realistic model, we have used $\alpha_0 = 0.002$ for $n = 3$. In all cases $\alpha_0 \approx \left(\mathrm{E}_\pi\left[x^\top x\right]\right)^{-1}$. For θ, we tried values in the set $\{0.001, 0.1, 0.2, 0.3, 0.4, 0.5\}$. For the simple model, the best results were obtained with $\theta = 0.001$ irrespective of n. For the intermediate model, we used $\theta = 0.5$ for $n = 1$, $\theta = 0.2$ for $n = 2$, and $\theta = 0.3$ for $n = 3$. For the realistic model, we used $\theta = 0.2$ for $n = 3$.

Remark 5.3 (Step size). There are automatic methods for adapting the step size. One such method is the Autostep method from Mahmood et al. (2012), a tuning-free version of the Incremental Delta-Bar-Delta (IDBD) algorithm from Sutton (1992). When using this method, with parameters set as suggested by Mahmood et al. (2012), the algorithm performs marginally worse compared to the results below.

Figure 2 shows the optimal policy for the simple model with terminal state using linear function approximation with 3rd-, 2nd-, and 1st-order Fourier basis using a softmax policy, and with 3rd-order

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

Table 2. *Expected discounted total reward (uniformly distributed starting states) for simple model with terminal state. The right column shows the fraction of episodes that end in the terminal state, within 100 time steps.*

	Expected reward	Terminations
Policy iteration	−85.91	0.006
Q-learning	−86.50	0.002
Fourier 3 with softmax policy	−86.11	0.006
Fourier 2 with softmax policy	−86.30	0.007
Fourier 1 with softmax policy	−86.59	0.011
Fourier 3 with ε-greedy policy	−92.74	0.000
Best constant policy	−122.70	0.040
Myopic policy with terminal state, $p_{\min} = 0.2$	−97.06	0.133
Myopic policy with terminal state, $p_{\min} = 5.8$	−90.40	0.096
Myopic policy with constraint, $p_{\min} = 0.2$	−121.52	0.337
Myopic policy with constraint, $p_{\min} = 6.4$	−93.58	0.132

Fourier basis using an ε-greedy policy. Figure 1 shows that the approximate optimal policy using 3rd-order Fourier basis is close to the optimal policy derived with policy iteration. Using 1st- or 2nd-order Fourier basis also gives a reasonable approximation of the optimal policy, but worse performance. Combining 3rd-order Fourier basis and an ε-greedy policy gives considerably worse performance.

The same conclusions can be drawn from Table 2, where we see the expected total discounted reward per episode for these policies, together with the results for the optimal policy derived with policy iteration, the policy derived with Q-learning, and several benchmark policies (see Section 5.1.3). Clearly, the performance of 3rd-order Fourier basis is very close to the performance of the optimal policy derived with policy iteration, and hence, we conclude that the linear function approximation with 3rd-order Fourier basis using a softmax policy appears to converge to approximately the optimal policy. The policy derived with Q-learning shows worse performance than both the 3rd- and 2nd-order Fourier basis, while the number of episodes run for the Q-learning algorithm is approximately a factor 30 bigger than the number of episodes run before convergence of SARSA with linear function approximation. Hence, even for this simple model, the number of states is too large for the Q-learning algorithm to converge within a reasonable amount of time. Furthermore, we see that all policies derived with linear function approximation using a softmax policy outperform the benchmark policies. Note that the optimal policy derived with policy iteration, the best constant policy, and the myopic policy with the terminal state require full knowledge of the underlying model and the transition probabilities, and the myopic policy with the constraint requires an estimate of the expected surplus one time step ahead, while the policies derived with function approximation or Q-learning only require real or simulated experience of the environment.

To analyse the difference between some of the policies, we simulate 300 episodes for the policy with the 3rd-order Fourier basis, the best constant policy, and the myopic policy with terminal state, $p_{\min} = 5.8$, for a few different starting states, two of which can be seen in Figure 3. A star in the figure corresponds to one or more terminations. The total number of terminations (of 300 episodes) are as follows: $S_0 = (-10, 2)$: Fourier 3:1, best constant: 291, myopic $p_{\min} = 5.8$: 20. $S_0 = (50, 7)$: Fourier 3: 1, best constant: 0, myopic $p_{\min} = 5.8$: 20. For other starting states, the comparison is similar to that in Figure 3. We see that the policy with the 3rd order Fourier basis appears to outperform the myopic policy in all respects, that is on average the premium is lower, the premium is more stable over time, and we have very few defaults. The best constant policy naturally is the most stable, but leads to in general a higher premium compared to the other policies, and will for more strained starting states quickly lead to a large number of terminations.

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

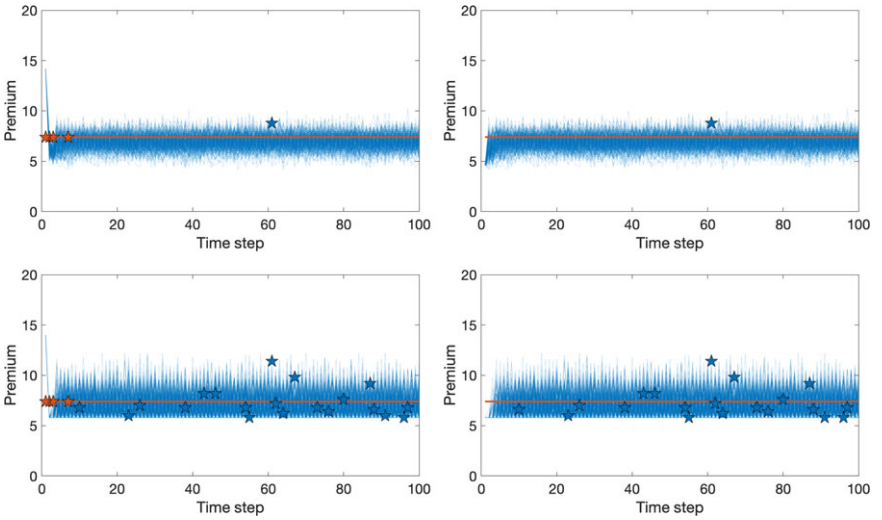

Figure 3. *Simple model. Top row: policy with 3rd order Fourier basis. Bottom row: myopic policy with terminal state, $p_{\min} = 5.8$. Left: starting state $S_0 = (-10, 2)$. Right: starting state $S_0 = (50, 7)$. The red line shows the best constant policy. A star indicates at least one termination.*

5.1.3. Benchmark policies

Best constant policy. The best constant policy is the solution to

$$\underset{p}{\text{minimise}} \ \mathrm{E}\left[\sum_{t=0}^{T} \gamma^t h(p, S_{t+1})\right].$$

For both the simple and intermediate models, $p = 7.4$ solves this optimisation problem. For details, see the Supplemental Material, Section 4.1.

Myopic policy for MDP with constraint. The myopic policy maximises immediate rewards. For the model with a constraint on the action space, the myopic policy solves

$$\underset{p}{\text{minimise}}\ c(p) \quad \text{subject to } \mathrm{E}\big[G_1 \mid S_0 = s, P_0 = p\big] \geq 0. \tag{5.3}$$

The myopic policy is given by the lowest premium level that satisfies the constraint. For details on how (5.3) is solved, see the Supplemental Material, Section 4.2.

For the simple, intermediate, and realistic model, the myopic policy charges the minimum premium level for a large number of states. Since this policy so quickly reduces the premium to the minimum level as the surplus or previously charged premium increases, it is not likely to work that well. Hence, we suggest an additional benchmark policy where we set the minimum premium level to a higher value, $p_{\min}$. Thus, this adjusted myopic policy is given by $\tilde{\pi}(s) = \max\{\pi(s), p_{\min}\}$, where π denotes the policy that solves (5.3). Based on simulations of the total expected discounted reward per episode for different values of $p_{\min}$, we conclude that $p_{\min} = 6.4$ achieves the best results for both the simple and intermediate model, and $p_{\min} = 10.5$ achieves the best result for the realistic model.

Myopic policy for MDP with terminal state. For the model with a terminal state, the myopic policy is the solution to the optimisation problem

$$\underset{p}{\text{minimise}}\ \mathrm{E}\big[h(p, S_1) \mid S_0 = s, P_0 = p\big]. \tag{5.4}$$

For details on how (5.4) is solved, see the Supplemental Material, Section 4.3. We also suggest an additional benchmark policy where the minimum premium level has been set to $p_{\min} = 5.8$, the level

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

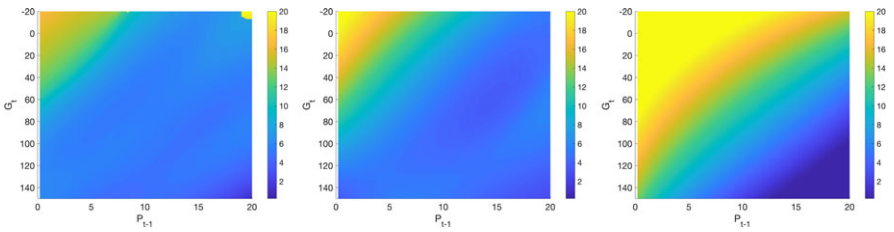

Figure 4. *Optimal policy for intermediate model with terminal state using linear function approximation, $N_t = N_{t-1} = 10$. Left: 3rd-order Fourier basis. Middle: 2nd-order Fourier basis. Right: 1st-order Fourier basis.*

that achieves the best results based on simulations of the total expected discounted reward per episode. For the intermediate and realistic model, this myopic policy is too complex and is therefore not a good benchmark.

5.2. Intermediate model

We use the following parameter values: $\gamma = 0.9$, $\mu = 5$, $\beta_0 = 10$, $\beta_1 = 1$, $\xi = 0.05$, $\nu = 1$, $\alpha_1 = 0.7$, $\alpha_2 = 0.3$, $\eta = 10$, $a = 18$, $b = -0.3$, and cost function (5.1) with $c_1 = 1$ and $c_2 = 1.2$. The premium level and number of contracts written are truncated and discretised according to $\mathcal{A} = \{0.2, 0.4, \ldots, 19.8, 20.0\}$ and $\mathcal{N} = \{0, 1, \ldots, 30\}$. The surplus process no longer only takes integer values (as in the simple model), instead the values that the surplus process can take are determined by the parameter values chosen. However, it is still truncated to lie between –20 and 150. For the parameter values above, we have $\mathcal{G} = \{-20.00, -19.95, \ldots, 149.95, 150.00\}$.

Figure 4 shows the optimal policy for the intermediate model using linear function approximation, with 3rd-, 2nd-, and 1st-order Fourier basis, for $N_t = N_{t-1} = 10$. Comparing the policy with the 3rd-order Fourier basis with the policy with the 2nd order Fourier basis, the former appears to require a slightly lower premium when the surplus or previously charged premium is very low. The policy with the 1st-order Fourier basis appears quite extreme compared to the other two policies. Comparing the policy with the 3rd-order Fourier basis for $N_t = N_{t-1} = 10$ with the optimal policy for the simple model (bottom row in Figure 1), we note that $\pi^i(g, p, 10, 10) \neq \pi^s(g, p)$, where π^s and π^i denotes, respectively, the policy for the simple and the intermediate model. There is a qualitative difference between these policies, since even given that we are in a state where $N_t = N_{t-1} = 10$ using the intermediate model, the policy from the simple model does not take into account the effect the premium charged will have on the number of contracts issued at time $t + 1$. The policy with 3rd-order Fourier basis for $N_t, N_{t-1} \in \{5, 10, 15\}$ can be seen in Figure 5.

To determine the performance of the policies for the intermediate model, we simulate the expected total discounted reward per episode for these policies. The results can be seen in Table 3. Here we clearly see that the policy with 3rd-order Fourier basis outperforms the other policies and that the policy with 1st-order Fourier basis performs quite badly since is not flexible enough to be used in this more realistic setting. We also compare the policies with the optimal policy derived with policy iteration from the simple model while simulating from the intermediate model. Though this policy performs worse compared to the policy with 3rd- and 2nd-order Fourier basis, it outperforms the policy with 1st-order Fourier basis. Note that the policies derived with function approximation only require real or simulated experience of the environment. The results for the myopic policy in Table 3 use the true parameters when computing the expected value of the surplus. Despite this, the policy derived with the 3rd order Fourier basis outperforms the myopic policy.

To analyse the difference between some of the policies, we simulate 300 episodes for the policy with the 3rd-order Fourier basis, the best constant policy and the policy from the simple model, for a few

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

Table 3. *Expected discounted total reward based on simulation, (uniformly distributed starting states). The right column shows the fraction of episodes that end in the terminal state, within 100 time steps.*

	Expected reward	Terminations
Fourier 3	−97.17	0.015
Fourier 2	−104.41	0.025
Fourier 1	−128.83	0.036
Policy from simple model	−116.70	0.075
Myopic policy with constraint, $p_{\min} = 0.2$	−360.69	0.988
Myopic policy with constraint, $p_{\min} = 6.4$	−100.92	0.169
Best constant policy	−131.85	0.060

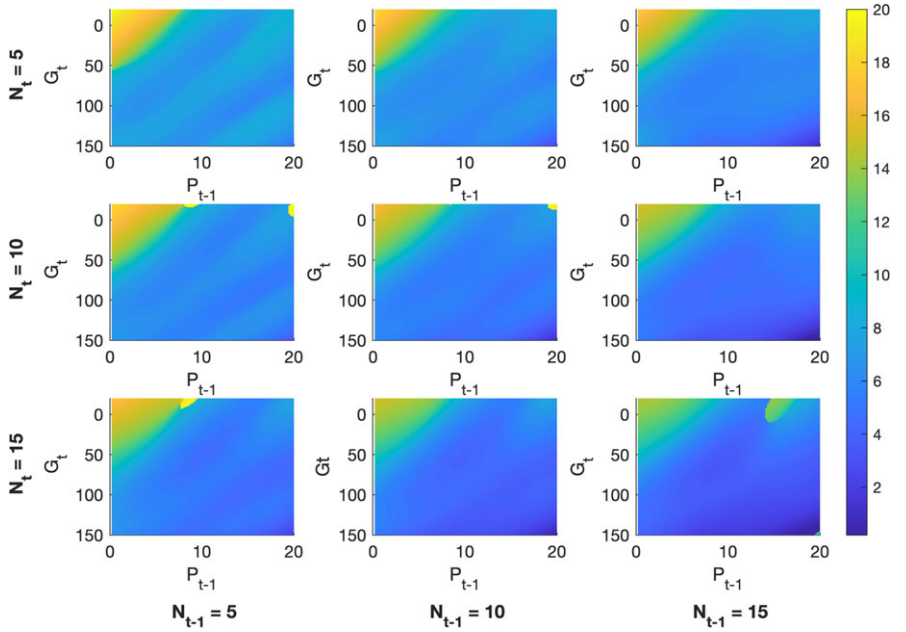

Figure 5. *Optimal policy for the intermediate model with terminal state using linear function approximation with 3rd-order Fourier basis, for $N_t, N_{t-1} \in \{5, 10, 15\}$.*

different starting states, two of which can be seen in Figure 6. Each star in the figures correspond to one or more terminations at that time point. The total number of terminations (of 300 episodes) are as follows: $S_0 = (0, 7, 10, 10)$: Fourier 3: 1, best constant: 13, simple: 5. $S_0 = (100, 15, 5, 5)$: Fourier 3: 0, best constant: 0, simple: 10. Comparing the policy with the 3rd-order Fourier basis with the policy from the simple model, we see that it tends to on average give a lower premium and leads to very few defaults, but is slightly more variable compared to the premium charged by the simple policy. This is not surprising, since the simple policy does not take the variation in the number of contracts issued into account. At the same time, this is to the detriment of the simple policy, since it cannot correctly take the risk of the varying number of contracts into account, hence leading to more defaults. For example, for the more strained starting state $S_0 = (-10, 2, 20, 20)$ (not shown in figure), the number of defaults for the policy with the 3rd order Fourier basis is 91 of 300, and for the simple policy it is 213 of 300. Similarly, for starting state $S_0 = (100, 15, 5, 5)$ (second column in Figure 6), the simple policy will tend set the premium much too low during the first time step, hence leading to more early defaults compared

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

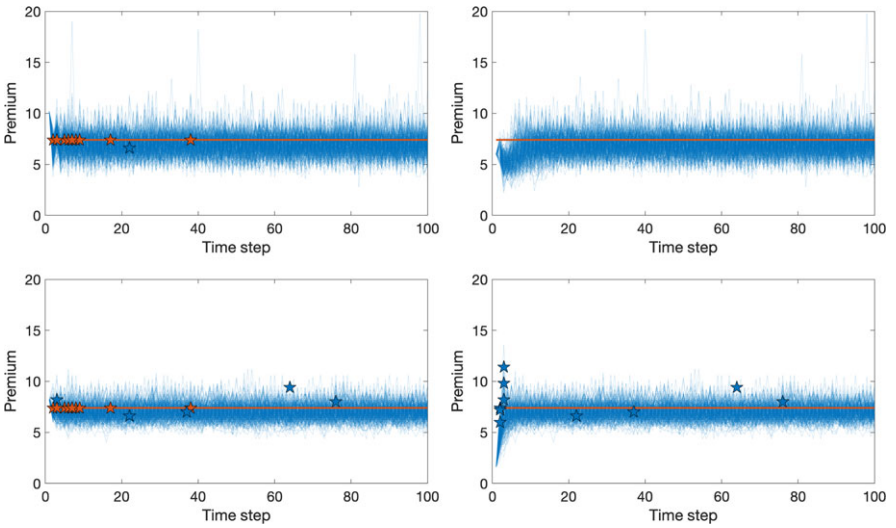

Figure 6. *Intermediate model. Top row: policy with 3rd Fourier basis. Bottom row: policy from the simple model. Left: starting state* $S_0 = (0, 7, 20, 20)$. *Right: starting state* $S_0 = (100, 15, 5, 5)$. *The red line shows the best constant policy. A star indicates at least one termination.*

to for example starting state $S_0 = (0, 7, 20, 20)$ (first column in Figure 6), despite the fact that the latter starting state has a much lower starting surplus.

5.3. Realistic model

To estimate the parameters of the model for the cumulative claims payments in (2.8), we use the motor third party liability data considered in Miranda et al. (2012). The data consist of incremental runoff triangles for number of reported accidents and incremental payments, with 10 development years. We have no information on the number of contracts. For parameter estimation only, we assume a constant number of contracts over the ten observed years, that is $\widetilde{N}_{t+1} = N$, and that the total number of claims for each accident year is approximately 5% of the number of contracts. We further assume that the total number of claims per accident year is well approximated by the number of reported claims over the first two development years. This leads to an estimate of the number of contracts $\widetilde{N}_{t+1}$ in (2.8) of $\widehat{N} = 2.17 \cdot 10^5$. For parameter estimation, we assume that $\mu_{0,t} = \mu_0$ in (2.9). Hence, μ_0 and ν_0^2 are estimated as the sample mean and variance of $\left(\log\left(C_{t,1}\right)\right)_{t=1}^{10}$. Similarly, μ_j and ν_j^2 are estimated as the sample mean and variance of $\left(\log\left(C_{t,j+1}/C_{t,j}\right)\right)_{t=1}^{10-j}$ for $j = 1, \ldots, 8$. Since we only have one observation for $j = 9$, we let $\widehat{\mu}_9 = \log\left(C_{1,10}/C_{1,9}\right)$ and $\widehat{\nu}_9 = 0$. c_0 is estimated by $\widehat{c}_0 = \exp\left\{\widehat{\mu}_0 + \widehat{\nu}_0^2/2\right\}/\widehat{N} \approx 2.64$. The parameter estimates can be seen in Table 1 in the Supplemental Material, Section 5.

For the model for investment earnings, the parameters are set to $\widetilde{\sigma} = 0.03$ and $\widetilde{\mu} = \log(1.05) - \widetilde{\sigma}^2/2$, which gives similar variation in investment earnings as in the intermediate model (2.7) when the surplus is approximately 50. The remaining parameters are $\gamma = 0.9$, $\beta_0 = 2 \cdot 10^5$, $\beta_1 = 1$, $\eta = 10$, and $b = -0.3$. The parameter a is set so that the expected number of contracts is $2 \cdot 10^5$ when the premium level corresponds to the expected total cost per contract, $\beta_0/(2 \cdot 10^5) + \beta_1 + c_0/\alpha_1 \approx 10.4$. Hence, $a \approx 4.03 \cdot 10^5$. The premium level is truncated and discretised according to $\mathcal{A} = \{0.5, 1.0, \ldots, 29.5, 30.0\}$ The cost function is as before (5.1), but now adjusted to give rewards of similar size as in the simple and intermediate setting. Hence, when computing the reward, the premium is adjusted to lie in [0.2, 20.0] according to

$$c(p) = \widetilde{p} + c_1\left(c_2^{\widetilde{p}} - 1\right), \quad \text{where } \widetilde{p} = 0.2 + \frac{p - 0.5}{30 - 0.5}(20 - 0.2).$$

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

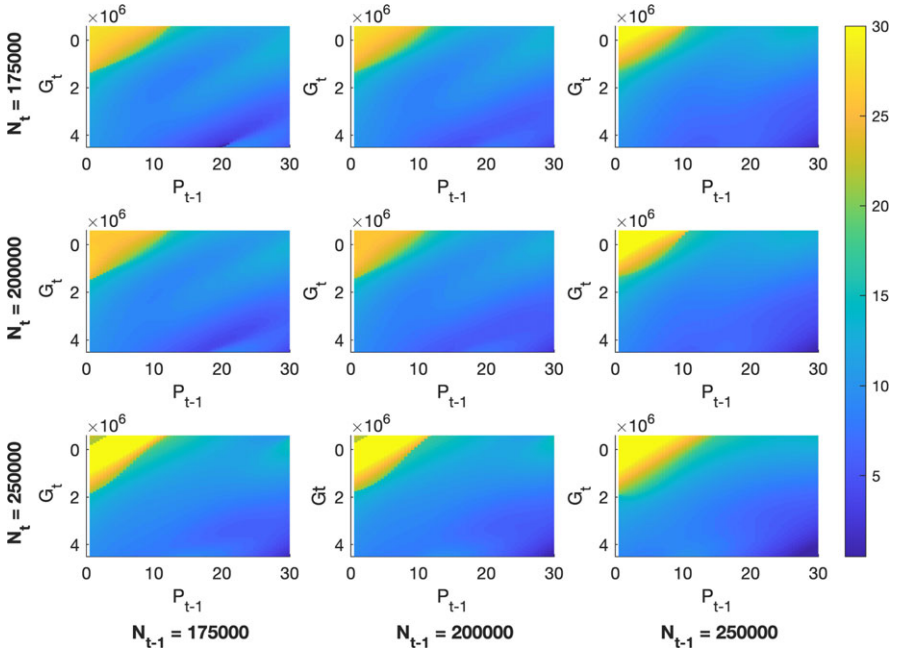

Figure 7. *Optimal policy for realistic model with terminal state using linear function approximation, for* $N_t, N_{t-1} \in \{1.75, 2.00, 2.50\} \cdot 10^5$, $C_{t-1,1} = c_0 \cdot 2 \cdot 10^5$, *and* $C_{t-j,j} = c_0 \cdot 2 \cdot 10^5 \prod_{k=1}^{j-1} f_k$.

The number of contracts is truncated according to $\mathcal{N} = \{144170, \ldots, 498601\}$. This is based on the 0.001-quantile of $\mathsf{Pois}(a(\max \mathcal{A})^b)$, and the 0.999-quantile of $\mathsf{Pois}(a(\min \mathcal{A})^b)$. The truncation of the cumulative claims payments $C_{t,j}$ is based on the same quantile levels and lognormal distributions with parameters $\mu = \log(c_0 \min \mathcal{N}) - \nu_0^2/2 + \sum_{k=1}^{j-1} \mu_k$ and $\sigma^2 = \sum_{k=0}^{j-1} \nu_k$, and with parameters $\mu = \log(c_0 \max \mathcal{N}) - \nu_0^2/2 + \sum_{k=1}^{j-1} \mu_k$ and $\sigma^2 = \sum_{k=0}^{j-1} \nu_k$, respectively, for $j = 1, \ldots, 9$. The truncation for the cumulative claims payments can be seen in Table 2 in the Supplemental Material, Section 5. The surplus is truncated to lie in $[-0.6, 4.5] \cdot 10^6$. Note that for the realistic model, with 10 development years the state space becomes 13-dimensional. Using the full 3rd-order Fourier basis is not possible since it consists of 4^{14} basis functions. We reduce the number of basis functions allowing for more flexibility where the model is likely to need it. Specifically,

$$\left\{\left(c_1^{(i)}, c_2^{(i)}, c_3^{(i)}, c_4^{(i)}, c_{14}^{(i)}\right) : i = 1, \ldots, 4^5\right\} = \{0, \ldots, 3\}^5,$$

$$c_1^{(i)} = c_2^{(i)} = c_3^{(i)} = c_4^{(i)} = c_{14}^{(i)} = 0 \quad \text{for } i = 4^5 + 1, \ldots, 4^5 + 9,$$

and, for $j = 1, \ldots, 9$,

$$c_{4+j}^{(i)} = \begin{cases} 1 \text{ for } i = 4^5 + j, \\ 0 \text{ for } i \neq 4^5 + j. \end{cases}$$

This means less flexibility for variables corresponding to the cumulative payments and no interaction terms between a variable corresponding to a cumulative payment and any other variable.

Figure 7 shows the optimal policy for the realistic model using linear function approximation for $N_t, N_{t-1} \in \{1.75, 2.00, 2.50\} \cdot 10^5$, $C_{t-1,1} = c_0 \cdot 2 \cdot 10^5$, and $C_{t-j,j} = c_0 \cdot 2 \cdot 10^5 \prod_{k=1}^{j-1} f_k$ for $j = 2, \ldots, 9$. To determine the performance of the approximate optimal policy for the realistic model, we simulate the expected total discounted reward per episode for this policy. The results can be seen in Table 4. The approximate optimal policy outperforms all benchmark policies. The best performing benchmark policy, the "interval policy," corresponds to choosing the premium level to be equal to the expected total cost per

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

Table 4. *Expected discounted total reward based on simulation, (uniformly distributed starting states). The right column shows the fraction of episodes that end in the terminal state, within 100 time steps.*

	Expected reward	Terminations
Fourier 3	−93.14	0.019
Interval policy ($p = 10.5$ when $G_t \in [1.2, 2.8] \cdot 10^6$)	−106.70	0.034
Myopic policy with constraint, $p_{\min} = 10.5$	−112.70	0.041
Best constant policy, $p = 11.5$	−136.60	0.061

contract, when the number of contracts is $2 \cdot 10^5$, as long as the surplus lies in the interval $[1.2, 2.8] \cdot 10^6$. This is based on a target surplus of $2 \cdot 10^6$ and choosing $\phi \in \{0.1, 0.2, \ldots, 1.0\}$ that results in the best expected total reward, where the interval for the surplus is given by $[1-\phi, 1+\phi] \cdot 2 \cdot 10^6$. When the surplus G_t is below (above) this interval, the premium is increased (decreased) in order to decrease (increase) the surplus. The premium for this benchmark policy is

$$P_t = \begin{cases} \min\left\{10.5 + \dfrac{(1-\phi)\cdot 2\cdot 10^6 - G_t}{2\cdot 10^5}, \max \mathcal{A}\right\}, & \text{if } G_t < (1-\phi)\cdot 2\cdot 10^6, \\ 10.5, & \text{if } (1-\phi)\cdot 2\cdot 10^6 \le G_t \le (1+\phi)\cdot 2\cdot 10^6, \\ \max\left\{10.5 + \dfrac{(1+\phi)\cdot 2\cdot 10^6 - G_t}{2\cdot 10^5}, \min \mathcal{A}\right\}, & \text{if } (1+\phi)\cdot 2\cdot 10^6 < G_t, \end{cases}$$

and rounded to the nearest half integer, to lie in $\mathcal{A}$. As before the approximate optimal policy outperforms the benchmark policies despite the fact that both the myopic and the interval policy use the true parameters when computing the expected surplus or the expected total cost per contract.

A comparison of the approximate optimal policy and the best benchmark policy, including a figure similar to Figure 6, is found in the Supplemental Material, Section 5.

6. Conclusion

Classical methods for solving premium control problems are suitable for simple dynamical insurance systems, and the model choice must to a large extent be based on how to make the problem solvable, rather than reflecting the real dynamics of the stochastic environment. For this reason, the practical use of the optimal premium rules derived with classical methods is often limited.

Reinforcement learning methods enable us to solve premium control problems in realistic settings that adequately capture the complex dynamics of the system. Since these techniques can learn directly from real or simulated experience of the stochastic environment, they do not require explicit expressions for transition probabilities. Further, these methods can be combined with function approximation in order to overcome the curse of dimensionality as the state space tends to be large in more realistic settings. This makes it possible to take key features of real dynamical insurance systems into account, for example payment delays and how the number of contracts issued in the future will vary depending on the premium rule. Hence, the optimal policies derived with these techniques can be used as a basis for decisions on how to set the premium for insurance companies.

We have illustrated strengths and weaknesses of different methods for solving the premium control problem for a mutual insurer and demonstrated that given a complex dynamical system, the approximate policy derived with SARSA using function approximation outperforms several benchmark policies. In particular, it clearly outperforms the policy derived with classical methods based on a more simplistic model of the stochastic environment, which fails to take important aspects of a real dynamical insurance system into account. Furthermore, the use of these methods is not specific to the model choices

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press

made in Section 2. The present paper provides guidance on how to carefully design a reinforcement learning method with function approximation for the purpose of obtaining an optimal premium rule, which together with models that fit the experience of the specific insurance company allows for optimal premium rules that can be used in practice.

The models may be extended to include dependence on covariates. However, it should be noted that if we want to model substantial heterogeneity among policyholders and consider a large covariate set, then the action space becomes much larger and function approximation also for the policy may become necessary.

Acknowledgment. We thank the anonymous reviewers for comments and suggestions that substantially improved the paper. We would also like to thank Mathias Lindholm for valuable discussions. Filip Lindskog acknowledges financial support from the Swedish Research Council, Project 2020-05065, and from Länsförsäkringars Forskningsfond, Project P9.20.

Conflicts of interest. The authors declare none.

Supplementary material. To view supplementary material for this article, please visit https://doi.org/10.1017/asb.2023.13.

References

Bertsekas, D.P. and Tsitsiklis, J.N. (1996) *Neuro-dynamic Programming*. Belmont, Massachusetts: Athena Scientific.
Buehler, H., Gonon, L., Teichmann, J. and Wood, B. (2019) Deep hedging. *Quantitative Finance*, **19**(8), 1271–1291.
Carbonneau, A. (2021) Deep hedging of long-term financial derivatives. *Insurance: Mathematics and Economics*, **99**, 327–340.
Chong, W. F., Cui, H. and Li, Y. (2021) Pseudo-model-free hedging for variable annuities via deep reinforcement learning. arXiv preprint arXiv:2107.03340.
Dayan, P. (1992) The convergence of TD(λ) for general λ. *Machine Learning*, **8**(3–4), 341–362.
De Finetti, B. (1957) Su un' impostazione alternativa della teoria collettivadel rischio. *Transactions of the XVth International Congress of Actuaries*, vol. 2, pp. 433–443.
Den Boer, A.V. and Zwart, B. (2014) Simultaneously learning and optimizing using controlled variance pricing. *Management Science*, **60**(3), 770–783.
Gerber, H.U. (1969) *Entscheidungskriterien für den zusammengesetzten Poisson-Prozess*. Ph.D. Thesis, ETH Zurich.
Germain, M., Pham, H. and Warin, X. (2021) Neural networks-based algorithms for stochastic control and PDEs in finance. arXiv preprint arXiv:2101.08068.
Gordon, G.J. (2001) Reinforcement learning with function approximation converges to a region. In *Advances in Neural Information Processing Systems*, pp. 1040–1046.
Han, J. and E, W. (2016) Deep learning approximation for stochastic control problems. *Advances in Neural Information Processing Systems, Deep Reinforcement Learning Workshop*.
Konidaris, G., Osentoski, S. and Thomas, P. (2011) Value function approximation in reinforcement learning using the Fourier basis. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.
Krasheninnikova, E., García, J., Maestre, R. and Fernández, F. (2019) Reinforcement learning for pricing strategy optimization in the insurance industry. *Engineering Applications of Artificial Intelligence*, **80**, 8–19.
Mack, T. (1993). Distribution-free calculation of the standard error of chain ladder reserve estimates. *ASTIN Bulletin: The Journal of the IAA*, **23**(2), 213–225.
Mahmood, A.R., Sutton, R.S., Degris, T. and Pilarski, P.M. (2012) Tuning-free step-size adaptation. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2121–2124. IEEE.
Martin-Löf, A. (1983) Premium control in an insurance system, an approach using linear control theory. *Scandinavian Actuarial Journal*, **1983**(1), 1–27.
Martin-Löf, A. (1994) Lectures on the use of control theory in insurance. *Scandinavian Actuarial Journal*, **1994**(1), 1–25.
Melo, F.S., Meyn, S.P. and Ribeiro, M.I. (2008) An analysis of reinforcement learning with function approximation. *Proceedings of the 25th International Conference on Machine Learning*, pp. 664–671.
Miranda, M.D.M., Nielsen, J.P. and Verrall, R. (2012) Double chain ladder. *ASTIN Bulletin: The Journal of the IAA*, **42**(1), 59–76.
Perkins, T.J. and Precup, D. (2003) A convergent form of approximate policy iteration. In *Advances in Neural Information Processing Systems*, pp. 1627–1634.
Puterman, M.L. (2005) *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, New Jersey: John Wiley & Sons.
Rummery, G.A. and Niranjan, M. (1994) On-line Q-learning using connectionist systems. Cued/f-infeng/tr, Cambridge University Engineering Department.
Schmidli, H. (2008) *Stochastic Control in Insurance*. London: Springer.
Singh, S., Jaakkola, T., Littman, M.L. and Szepesvári, C. (2000) Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, **38**(3), 287–308.

Sutton, R.S. (1992) Adapting bias by gradient descent: An incremental version of Delta-Bar-Delta. *AAAI*, San Jose, CA, pp. 171–176.
Sutton, R.S. (1995) Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in Neural Information Processing Systems 8*.
Sutton, R.S. and Barto, A.G. (2018) *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: MIT Press.
Sutton, R.S., McAllester, D., Singh, S. and Mansour, Y. (1999) Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems 12*.
Tsitsiklis, J.N. and Van Roy, B. (1997) An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, **42**(5), 674–690.
Watkins, C.J.C.H. (1989) *Learning from delayed rewards*. Ph.D. Thesis, University of Cambridge, UK.
Williams, R.J. (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, **8**, 229–256.

https://doi.org/10.1017/asb.2023.13 Published online by Cambridge University Press