Matrix methods for stochastic dynamic programming in ecology and evolutionary biology

Jody R. Reimer[1,2*], Marc Mangel[3,4], Andrew E. Derocher[1] and Mark A. Lewis[1,2]

[1]Department of Biological Sciences, University of Alberta, Canada
[2]Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, Canada
[3]Department of Biology, University of Bergen, Norway
[4]Institute of Marine Sciences and Department of Applied Mathematics, University of California, Santa Cruz, USA
[*]Corresponding author. jrreimer@ualberta.ca

Abstract

1. Organisms are constantly making tradeoffs. These tradeoffs may be behavioural (e.g., whether to focus on foraging or predator avoidance) or physiological (e.g., whether to allocate energy to reproduction or growth). Similarly, wildlife and fisheries managers must make tradeoffs while striving for conservation or economic goals (e.g., costs versus rewards). Stochastic dynamic programming (SDP) provides a powerful and flexible framework within which to explore these tradeoffs. A rich body of mathematical results on SDP exist but have received little attention in ecology and evolution.

2. Using directed graphs—an intuitive visual model representation—we reformulated SDP models into matrix form. We synthesized relevant existing theoretical results which we then applied to two canonical SDP models in ecology and evolution. We applied these matrix methods to a simple illustrative patch choice example and an existing SDP model of parasitoid wasp behaviour.

3. The proposed analytical matrix methods provide the same results as standard numerical methods as well as additional insights into the nature and quantity of other, nearly optimal, strategies, which we may also expect to observe in nature. The mathematical results highlighted in this work also explain qualitative aspects of model convergence. An added benefit of the proposed matrix notation is the resulting ease of implementation of Markov chain analysis (an exact solution for the realized states of an individual) rather than Monte Carlo simulations (the standard, approximate method). It also provides an independent validation method for other numerical methods, even in applications focused on short-term, non-stationary dynamics.

4. These methods are useful for obtaining, interpreting, and further analysing model convergence to the optimal time-independent (i.e., stationary) decisions predicted by an SDP model. SDP is a powerful tool both for theoretical and applied ecology, and an understanding

of the mathematical structure underlying SDP models can increase our ability to apply and interpret these models.

key words: backwards induction, Markov chain, Markov decision process, optimality models, stochastic dynamic programming, stationary decisions, value iteration

Introduction

Tradeoffs are an unavoidable part of being alive. Tradeoffs may be physiological (e.g., how much energy to allocate to growth versus reproduction (Rees et al., 1999)), or behavioural (e.g, how to balance energy gain with predator avoidance (Mangel and Clark, 1986; McNamara and Houston, 1986). What constitutes a successful strategy is ultimately influenced by natural selection, as strategies that increase population mean fitness will tend to spread in the population if they have a heritable component.

Similarly, conservation ecologists and wildlife or fisheries managers must also make tradeoffs while striving to achieve conservation or management goals. In this context, tradeoffs are often between immediate and future rewards (e.g., how much to harvest now while maintaining a sufficient population to harvest later (Runge and Johnson, 2002)). The objective may be to control an invasive species (Bogich and Shea, 2008) or ensure the long term viability of a population.

Optimal control theory predicts how an individual should navigate a series of risks and rewards to achieve an objective, subject to relevant constraints. Often, the rewards may be probabilistic (e.g., the probability of individual finding food), and the optimal control may depend on both the state of the individual (e.g., an animal's physiological state) and a temporal

component (e.g., how many days remain in a season). We use the word decision (rather than control) to describe the action taken by an individual whenever there is more than one possible action. These decisions include events beyond cognition such as the decision by an animal to abort a pregnancy based on their level of energy reserves. An optimal decision question may be framed as a state-dependent Markov decision process.

Stochastic dynamic programming (SDP) is a common method to deal with state-dependent Markov decision processes. It is common in both ecology and resource management to refer to both the model and the method of solving the model as SDP (Marescot et al., 2013) and we follow this convention. SDP has a rich history of application and theoretical developments in a wide array of disciplines (Puterman, 1994), including engineering (Sheshkin, 2010), finance (Bäuerle and Rieder, 2011), and artificial intelligence (Sigaud and Buffet, 2010). However, many of these theoretical advances have not been popularized in the biological literature, despite their powerful implications both for model analysis and biological interpretation.

SDP has been used in many areas of biology, including behavioural biology, evolutionary biology, and conservation and resource management (for reviews in each of these areas, see McNamara et al. (2001) and Mangel (2015), Parker and Smith (1990), and Marescot et al. (2013), respectively).

In some applications of SDP, one is interested in the temporal aspects of the optimal decisions, especially near some terminal time; these are *finite time horizon problems*. For example, we may expect an individual to make riskier foraging decisions near the end of a feeding season (Bull et al., 1996; Reimer et al., 2019). In many cases, the optimal decisions are stationary (i.e., not varying from one time step to the next) when they are sufficiently far away from the terminal time. In some applications of SDP, these stationary decisions are used for prediction (Mangel, 1989; Chan and Godfray, 1993; Shea and Possingham, 2000), rather than the

transient dynamics near the end of the optimization period; we refer to these as *stationary decision problems*. Finally, other questions do not concern a finite time period at all (Venner et al., 2006; Mangel and Bonsall, 2008), but are *infinite horizon problems*. For example, managers may wish to maximize the total number of animals that may be harvested indefinitely (Runge and Johnson, 2002).

Stationary decision problems and infinite horizon problems in biology are often solved using essentially the same numerical, iterative method, though it appears in the literature under different names: backwards induction or value iteration (Puterman, 1994; Clark and Mangel, 2000). Several software packages have been created to run these, and other (e.g., policy iteration) numerical routines for a wide range of applications in biology (Lubow, 1995; Chadès et al., 2014; Marescot et al., 2013).

SDP models are typically constructed component-wise, separately considering an individual in each possible state at each time. This component-wise model formulation hides the elegant mathematical structure underlying SDP. The theoretical results in the SDP literature outside of ecology (Puterman, 1994) depend on this mathematical structure. In this paper, we promote the use of vector and matrix notation for SDP applications, allowing for consideration of an individual in all possible states at each time. A few examples of this approach in ecology do exist (McNamara, 1990, 1991; McNamara et al., 2001). For example, McNamara (1990) analyzed tradeoffs in the context of risk-sensitive foraging by formulating an SDP model in the language of matrices and analyzing the eigenvalue equation, which led to one of the main results we use here—a generalization of the Perron-Frobenius theorem for the SDP operator (McNamara, 1991). We build on this foundation, applying results from general SDP theory to another broad class of SDP models in ecology (so-called "resource allocation models"). We demonstrate how formulating an SDP model in the language of matrices leads to analytic methods for obtaining

optimal decisions for both stationary decision and infinite horizon problems. We provide step-by-step instructions for implementing these analytic methods for two canonical equations of SDP in ecology (Mangel, 2015) and illustrate key steps with a simple example.

These analytic matrix methods have several notable additional benefits. A byproduct of obtaining the optimal decisions in this way is a comprehensive picture of all other possible decisions. This provides a sense of which other, nearly optimal, decisions we could also expect to observe in nature, or a range of possible management options with comparable outcomes. The intuition behind these analytic results also allows us to explain non-intuitive transient oscillating decisions. Further, ecologists interested in how an optimally behaving individual's state changes over time typically run thousands of Monte Carlo simulations (an approximate method). Alternatively, Markov chains provide an exact method for determining the probability distribution of an individual's realized state at each time (Mangel and Clark, 1988). We illustrate how the Markov transition matrix is conveniently constructed as a by-product of formulating an SDP model using matrices.

We apply these matrix methods to an existing study of host feeding behaviour in parasitic wasps (Chan and Godfray, 1993).

Methods

Stochastic dynamic programming

SDP models contain several key components (Clark and Mangel, 2000). These include discrete time steps t and a time horizon, which may either be finite with a terminal time T, or infinite. The set of possible state variables $x \in \chi = \{x_1, \ldots, x_k\}$ must be defined, and any relevant constraints on the states included. The actions available to an individual in a given state at each time must be

made explicit. We assume a finite number of actions available to an individual. The probabilistic state dynamics (e.g., the probability of survival or reproduction), which may vary depending on the individual's decision, must be defined. The fitness function $f(x,t)$, also known as the reward or value function, describes the expected future reward for an optimally behaving individual in state x at time t. Its value is determined by specifying the dynamic programming equation, so that $f(x,t) = \max \mathbb{E}[\text{future reward, given state } x \text{ at time } t]$, where the maximum is taken over all possible decisions and the expectation is taken over all possible future rewards. For finite horizon problems, with $T < \infty$, a terminal fitness function $f(x,T) = \Phi(x)$ must be specified. Relevant boundary conditions (i.e., critical levels of the state variable) must also be specified; e.g., if $x = 0$ implies mortality, then $f(0,t) = 0$ for all t, as there can be no further future fitness gains. Note that we used lowercase f to describe the fitness function for an individual in a given state. When we later consider all states simultaneously, we will use capital F to denote the fitness vector. We follow this convention throughout, using lowercase letters to denote scalar quantities and capital letters to denote vectors and matrices.

Most applications of SDP in biology find their roots in one of two canonical equations (Mangel, 2015). Both have an individual's energy stores x as the state variable, μ is the mortality rate (excluding starvation), η is the probability of finding food, and y is the energy gained if the individual finds food. In the first canonical equation, c is the daily energetic cost. This equation describes a model of activity choice, with an individual choosing between two possible foraging patches, so the decision is $i = \{\text{patch 1 or 2}\}$:

$$f(x,t) = \max_{i=1,2} \underbrace{e^{-\mu_i}}_{\text{survival}} \left[\underbrace{\eta_i f(x - c_i + y_i, t+1)}_{\text{obtain food}} + \underbrace{(1-\eta_i) f(x - c_i, t+1)}_{\text{do not obtain food}} \right]. \tag{1}$$

Here the probability of survival, the probability of finding food, the energetic costs, and the

energetic gains from finding food all vary depending on patch choice, so are subscripted by i.

The second canonical equation describes a model of resource allocation, such as how much energy to devote to reproduction at a given time, so the decision is the amount of energy r to allocate to immediate reward:

$$f(x,t) = \max_r \left(\underbrace{g(r)}_{\substack{\text{immediate} \\ \text{rewards}}} + \underbrace{e^{-\mu}}_{\text{survival}} \underbrace{\left[\underbrace{\eta\, f(x-r+y, t+1)}_{\text{obtain food}} + \underbrace{(1-\eta) f(x-r, t+1)}_{\text{do not obtain food}} \right]}_{\text{future rewards}} \right). \quad (2)$$

Here the probabilities of survival and finding food do not vary with the individual's choice. Rather, the individual must balance the immediate rewards $g(r)$ of spending r resources against any possible future rewards. In both (1) and (2), survival acts as a discount factor on future rewards. Applications in resource management also tend to be structured like this second canonical equation (Marescot et al., 2013).

Illustrative example

We illustrate key concepts using a simple patch choice example. Consider an individual in a non-breeding season of length T who may take one of 5 states $x \in \chi = \{x_1, \ldots, x_5\}$ corresponding to their level of energy reserves (i.e., $x_1 < \ldots < x_5$). Each day, $t = 1, 2, \ldots, T-1$, the individual chooses one of two foraging patches, with the objective of maximizing survival to time T. Patch 1 is low risk and low reward ($\eta_1 = 0.4$, $e^{-\mu_1} = 0.99$) and Patch 2 is high risk and high reward ($\eta_2 = 0.8$, $e^{-\mu_2} = 0.891$). Probabilistic state changes may be represented by arrows in directed graph (Figure 1). If an individual finds food in either patch, their reserves increase by 2 units ($y_1 = y_2 = 3$; dashed arrows). If an individual does not find food, their reserves decrease by 1 unit ($c_1 = c_2 = 1$; solid arrows). An individual in state x_1 who

does not find food that day dies (i.e., transitions to state x_0, the absorbing death state). An individual survives each of these transitions with probability $e^{-\mu_{i_n}}$; an individual in any state dies with probability $1 - e^{-\mu_{i_n}}$ (dotted arrows). These probabilities all depend on the patch decision $i_n \in \{\text{patch 1, patch 2}\}$ made by an individual in state x_n. We are interested in the stationary decision problem, i.e., predicting the patch an individual in state x at time t uses, away from any transient effects of the terminal time. To answer this question, we use an SDP model with the first canonical equation (1) as the fitness function.

Existing methods for obtaining stationary decisions

Backwards induction is typically used to solve stationary decision problems (see Clark and Mangel (2000) for an overview). This is a numerical routine that exploits the recurrence relation between $f(x,t)$ and $f(x',t+1)$, for each x and some $x' \in \chi$. Backwards induction starts by defining the terminal fitness function, $f(x,T) = \Phi(x)$, for all x. One then calculates $f(x,T-1)$ for all x, using the values of $f(\cdot,T)$. After $f(x,T-1)$ is calculated, one goes on to calculate $f(x,T-2)$, and continues in this way until $f(x,1)$ is computed for all x. For large T, the optimal decisions are often stationary from one time step to the next, depending only on state, for t far from T, i.e., $T - t >> 1$.

In a similar fashion, one may solve infinite horizon problems using the method of value iteration, which is analogous to backwards induction applied repeatedly from a zero terminal rewards function $\phi(x) = 0$ for all x, until some convergence criterion for $f(x,t)$ is reached (see Marescot et al. (2013) for an overview). We compare results obtained using these numerical methods with the proposed matrix methods. All computations were performed in Matlab (2018b) and all code is available at doi:10.5281/zenodo.2547815. For those who prefer working in R, we have also included an overview of key R commands (S1, online Supplementary Material).

Matrix notation

While applications of SDP in biology typically describe the fitness function component-wise for each state x, such as in (1) or (2), mathematical results follow more readily if these equations are reformulated in matrix notation. A few papers and software programs use the language of matrices (e.g., Marescot et al. (2013); Chadès et al. (2014)) but do not discuss the rich theory of **nonnegative matrices** (bolded terms in Glossary, Appendix A) we use here.

We let $F(t) = [f(x_1, t), \cdots, f(x_k, t)]^\top$ denote a column vector of fitness functions for each state at time t. We do not here explicitly consider death, the absorbing state x_0 (grey arrows in Figure 1). This exclusion of death is necessary for the **primitivity** of P_π, a condition required for the results described below. Further, each matrix P_π is **substochastic** due to the discounting effect of survival, which ensures convergence in the mathematical results that follow.

We create a square $k \times k$ matrix of state transition probabilities P_π, where each entry $p_\pi(x_j, x_k)$ describes the probability of transitioning from state x_j to state x_k. A policy π is a k-tuple of decisions, one for each state. Π denotes the set of all possible policies. In (1), each entry in π may take one of two values, patch 1 or patch 2, and so Π contains 2^k possible policies (i.e., (number of possible actions)^(number of states in χ)). Each policy has a corresponding matrix P_π, so there are 2^k possible matrices P_π.

We rewrite (1) using matrix notation as

$$F(t) = \max_{\pi \in \Pi} P_\pi F(t+1), \tag{3}$$

where the maximum is taken over each of the independent vector components. Letting

$G_\pi = [g_{\pi,1}, \ldots, g_{\pi,k}]^{\mathrm{T}}$ be a vector of immediate rewards, we can similarly rewrite (2) as

$$F(t) = \max_{\pi \in \Pi} \left[G_\pi + P_\pi \, F(t+1) \right]. \tag{4}$$

Matrix notation for illustrative example

For our illustrative patch choice example,

$$P_\pi = \begin{bmatrix} 0 & 0 & e^{-\mu_{i_1}} \eta_{i_1} & 0 & 0 \\ e^{-\mu_{i_2}} (1 - \eta_{i_2}) & 0 & 0 & e^{-\mu_{i_2}} \eta_{i_2} & 0 \\ 0 & e^{-\mu_{i_3}} (1 - \eta_{i_3}) & 0 & 0 & e^{-\mu_{i_3}} \eta_{i_3} \\ 0 & 0 & e^{-\mu_{i_4}} (1 - \eta_{i_4}) & 0 & e^{-\mu_{i_4}} \eta_{i_4} \\ 0 & 0 & 0 & e^{-\mu_{i_5}} (1 - \eta_{i_5}) & e^{-\mu_{i_5}} \eta_{i_5} \end{bmatrix}, \tag{5}$$

and $\pi = \{i_1, \ldots, i_5\}$ describes the patch choices for individuals in states x_1 through x_5. Intuition may be gained by comparing P_π with Figure 1, where a black arrow from state x_j to x_k correspond to entry $p_\pi(x_j, x_k)$ in P_π. In our example, each patch choice i_1, …, i_5 is equal to patch 1 or patch 2, giving rise to values of μ_1 or μ_2, and η_1 or η_2. Thus there are 2^5 possible matrices P_π.

Note that in this example, the locations of the nonzero entries in P_π are the same for all $\pi \in \Pi$. In other applications, this need not be the case. A nonzero entry of P_π will change location between different policies if the corresponding arrow in the directed graph changes the nodes that it connects, rather than just changing the probability associated with that arrow (e.g., the parasitoid wasp example below).

Analytic method for activity choice problems

We now describe a method for obtaining the stationary policy for SDP models of form (3) using a generalization of the Perron-Frobenius theorem[1] by McNamara (1991). We highlight relevant mathematical results and include full technical details in S2, online Supplementary Material. Each matrix P_π has k **eigenvalues** $\lambda_{\pi,j}$, which we order according to their magnitude with subscripts $j = 1, \ldots, k$ so that $|\lambda_{\pi,1}| \geq \ldots \geq |\lambda_{\pi,k}|$. Each eigenvalue $\lambda_{\pi,j}$ has a corresponding right **eigenvector** $V_{\pi,j}$. The optimal policy π^* is defined as the policy satisfying

$$P_{\pi^*}V^* = \max_\pi P_\pi V^*,$$

for V^* satisfying $P_\pi V^* = \lambda^* V^*$. If P_{π^*} is **primitive** (see S3, online Supplementary Material for details), the generalized Perron-Frobenius states that P_{π^*} has a uniquely defined dominant eigenvalue $\lambda_{\pi^*,1}$ and corresponding right eigenvector $V_{\pi^*,1}$, which determine the asymptotic behaviour of $F(t)$ according to

$$\lim_{t\to-\infty} (\lambda_{\pi^*,1})^{-t} F(t) \propto V_{\pi^*,1},$$

i.e., $F(t)$ decays exponentially according to $(\lambda_{\pi^*,1})^{-t}$ and converges in structure to $V_{\pi^*,1}$ as $t \to -\infty$. This dominant eigenvalue satisfies $\lambda_{\pi^*,1} = \max_\pi \lambda_{\pi,1}$ (McNamara, 1991). If we are interested in obtaining the stationary policy analytically, without using backward induction or value iteration, we may thus follow the steps in Box 1.

[1]For the classical Perron-Frobenius theorem in the context of matrix population models see Caswell (2001).

Box 1. Stationary policy for activity choice problems

1. Determine the set of all possible policies $\pi \in \Pi$ and construct the corresponding matrices P_π
2. Calculate the dominant eigenvalue $\lambda_{\pi,1}$ of each matrix P_π
3. Find the largest of these dominant eigenvalues: $\lambda_{\pi^*,1} = \max_{\pi\in\Pi} \lambda_{\pi,1}$
4. Confirm that the corresponding matrix P_{π^*} is primitive, and if so, π^* is the stationary policy

Note that primitivity is a sufficient but not necessary condition for π^* to be the optimal stationary strategy. The assumption of primitivity can usually be satisfied by omitting any absorbing, or otherwise redundant, states (McNamara et al., 2001). If there truly are multiple optimal strategies (i.e., step 3 in Box 1 does not have a unique answer), this method will identify all of them.

What is more likely than multiple truly optimal policies is that there are several policies which are nearly optimal, with corresponding dominant eigenvalues just slightly smaller than $\lambda_{\pi^*,1}$ (Mangel, 1991). This is one of the strengths of this type of approach; by calculating the asymptotic properties of the SDP model explicitly for each possible policy, we not only find the optimal policy, but also obtain information about which other policies are nearly optimal.

We applied the steps in Box 1 to the illustrative patch choice example to obtain the stationary decisions. We also found policies which are nearly optimal by looking at which matrices P_π have dominant eigenvalues within 1% of $\lambda_{\pi^*,1}$. The properties of P_{π^*} are not only relevant as $t \to \infty$, but also for understanding transient behaviour during convergence. For an example illustrating how the other eigenvalues of P_{π^*} may lead to surprising oscillations, see S4,

online Supplementary Material.

Analytic method for resource allocation problems

Using results from general SDP theory (S2, online Supplementary Material), we know that an optimal stationary policy π^* exists for equations of form (4) and that for any policy π there exists a unique solution $\tilde{F}$ satisfying $\tilde{F}_\pi = G_\pi + P_\pi \tilde{F}_\pi$. This solution has the form $\tilde{F}_\pi = (I - P_\pi)^{-1} G_\pi$, which can be seen using the recursive nature of this equation. For a given stationary policy π,

$$\begin{aligned}
F(T-1) &= G_\pi + P_\pi F(T) \\
F(T-2) &= G_\pi + P_\pi \left[G_\pi + P_\pi F(T)\right] \\
&= G_\pi + P_\pi G_\pi + P_\pi P_\pi F(T) \\
&\vdots \\
F(T-\tau) &= \underbrace{\sum_{q=0}^{\tau-1} (P_\pi)^q G_\pi}_{\text{A}} + \underbrace{(P_\pi)^\tau F(T)}_{\text{B}}.
\end{aligned}$$

If we increase T, the number of time steps under consideration increases. Alternatively, we may fix T and look increasingly far back in time (i.e., letting $\tau \to \infty$). Mathematically, these are equivalent; we are making the time period under consideration very large, whether by changing the initial time or the terminal time. As $\tau \to \infty$, Part B $\to 0$, since $|\lambda_{\pi,1}| < 1$ for **substochastic** matrices such as these (S2, online Supplementary Material). Part A is a matrix geometric series with $|\lambda_{\pi,1}| < 1$, so

$$\sum_{q=0}^{\tau-1} (P_\pi)^q G_\pi \to (I - P_\pi)^{-1} G_\pi \tag{6}$$

as $\tau \to \infty$, where I is the $k \times k$ identity matrix. The solution corresponding to π^* is the largest of the solutions corresponding to all $\pi \in \Pi$, i.e.,

$$\tilde{F}_{\pi^*} = \max_{\pi \in \Pi} \tilde{F}_\pi.$$

Thus for SDP problems following the second canonical equation, the steps in Box 2 determine the optimal stationary policy.

Box 2. Stationary policy for resource allocation problems

1. Determine the set of all possible policies $\pi \in \Pi$ and construct the corresponding P_π and G_π
2. Calculate $\tilde{F}_\pi = (I - P_\pi)^{-1} G_\pi$ for each policy
3. Determine which policy π^* yields the largest $\tilde{F}_\pi$; π^* is the optimal stationary policy

Host feeding behaviour of parasitic wasps

The evolution of insect parasitoid behaviour has been an especially fruitful area of SDP research (Charnov and Skinner, 1984; Mangel, 1989; Clark and Mangel, 2000). We apply our method to Chan and Godfray's (1993) resource pool model of host feeding behaviour in parasitoid wasps, where an adult female wasp requires host resources both for maintenance as well as the maturation of eggs. Upon encountering a host, she must choose whether to use it for host feeding or for oviposition. If she uses the host for food, she forgoes immediate fitness rewards but gains energy with which she may obtain future rewards. Chan and Godfray's goal was to predict the optimal state-dependent feeding strategy of such parasitic wasps, specifically the stationary energetic threshold x_c below which an adult female wasp is predicted to host feed rather than

oviposit, provided she was neither close to some terminal time nor running out of eggs.

Chan and Godfray described an individual's physiological state with a single variable x. Time was scaled so that each time step corresponds to the amount of time it takes to lose one unit of energy; e.g., if an individual's state is $x = 10$, that individual can survive 10 time steps without feeding before death by starvation occurs.

The probability of finding a host over one time step is η. If a host is not encountered, the wasp's state decreases by 1 for daily maintenance. If a host is encountered and the wasp decides to host feed, her state decreases by 1 for daily maintenance but increases by α, the energy gained from host feeding. If instead she parasitizes the host, her state decreases by 1 for daily maintenance and then further decreases by β, the cost of egg maturation. However, she receives an immediate fitness gain of 1 unit. Her daily survival probability is $\mathrm{e}^{-\mu}$, where μ is the instantaneous risk of mortality. If $x = 0$, the wasp dies of starvation. Chan and Godfray used parameters $\eta = 0.2$, $\alpha = 30$, and $\mu = 0.0125$. They considered two values for the cost of egg maturation, $\beta = 4$ and 16, but we consider only $\beta = 4$. The largest possible x value and the terminal time T were chosen to be large enough that they did not affect the threshold value between host feeding and parasitizing. As they did not state these values explicitly, we used 75 as an upper bound for x and $T = 1000$.

The resulting SDP equation is ,

$$f(x,t) = \max \overbrace{\left\{ \eta \underbrace{\left[1 + \mathrm{e}^{-\mu} f(x-1-\beta, t+1)\right]}_{\text{parasitize}}, \underbrace{\mathrm{e}^{-\mu} f(x-1+\alpha, t+1)}_{\text{host feed}} \right\}}^{\text{encounter host}} + \underbrace{(1-\eta)\mathrm{e}^{-\mu} f(x-1, t+1)}_{\text{no host encountered}}, \tag{7}$$

with boundary conditions $f(x,T)=0$ and $f(0,t)=0$ for all x and t. We rewrite (7) as

$$f(x,t)=\max_{i\in\{1,2\}}\eta\Big[g_i+\mathrm{e}^{-\mu}f(x-1+c_i,t+1)\Big]+(1-\eta)\mathrm{e}^{-\mu}f(x-1,t+1) \tag{8}$$

where $i=1$ denotes parasitizing and $i=2$ denotes host feeding, $g_1=1$, $g_2=0$, $c_1=-\beta$, and $c_2=\alpha$. This now resembles the second canonical equation (2) and can thus be written as (4), where each $\pi\in\Pi$ is a k-tuple of ones and twos. Each π has a corresponding P_π and G_π (for more details, see S5, online Supplementary Material). For each $\pi\in\Pi$, we calculated $\tilde{F}_\pi=(I-P_\pi)^{-1}G_\pi$ and then determined which was largest. The corresponding policy π^* is the optimal stationary policy.

A computational note

The number of policies π which need to be explored grows exponentially as the number of states k increases. In both of our examples, Π contained 2^k possible policies (= (number of possible actions)^(number of states in χ)). It quickly becomes computationally unwieldy to explore each of these options. Fortunately, this is not necessary because the decision made in each state is independent of the optimal decision of any other state; observe that $f(x,t)$ does not depend on $f(x',t)$ for any other state x'. For example, in the parasitic wasp problem, we first considered $\pi=\{1,1,\ldots,1\}$. We then checked whether $\tilde{F}_\pi$ increased if $\pi=\{2,1,\ldots,1\}$. If so, we left 2 in that location, if not, we returned it to 1. We then checked whether $\tilde{F}_\pi$ was greater when the second entry of π was 2, again retaining 2 in that location if so, and discarding it if not. Continuing in this way reduced the number of policies considered from 2^k to $k+1$.

Forward iteration using Markov chains

Monte Carlo simulations are often used to study the realized states of an optimally behaving individual over time (see Clark and Mangel (2000) for details). Many such simulations are required to get an approximation of the probability distribution of the individual's state over time. One way to obtain the exact solution, rather than these approximations, is through the use of Markov chains (Mangel and Clark, 1988). Component wise formulation of SDP models, however, means that this approach is often not considered. We suspect this is because it appears far removed from the paradigm of component wise backwards induction already in use, and may seem less intuitive than Monte Carlo simulations. However, it may be simpler to obtain exact Markov chain results than the approximate Monte Carlo results, provided the problem is already formulated using matrices.

To see this, let M denote a **Markov matrix**, where $m(x_k, x_j) = \Pr(\text{transitioning from state } x_j \text{ to state } x_k \text{ in one time step})$. Recall that $p_\pi(x_j, x_k) = \Pr(\text{transitioning from state } x_j \text{ to state } x_k \text{ in one time step})$ under policy π and that P_π is a **substochastic matrix**. This can easily be modified to be a true stochastic matrix $\hat{P}_\pi$, with rows summing to 1, by adding the appropriate column and row for any absorbing states such as death (grey arrows in Figure 1). The Markov matrix corresponding to the SDP model for a given policy π is then $M = \hat{P}_\pi^\top$, the transpose of matrix $\hat{P}_\pi$. Let $z(x,t) = \Pr(\text{an optimally behaving individual is in state } x \text{ at time } t)$, with vector notation $Z(t)$. We obtain the probability of the individual being in each state using the forward recursion equation

$$Z(t+1) = M(t)\, Z(t) = (\hat{P}_{\pi(t)})^\top\, Z(t), \qquad Z(0) = z_0 \tag{9}$$

where z_0 is a probability mass function for the individual's initial state.

We calculated the probability that an individual is in state x at time t for the parasitic wasp example using this method of Markov chains. We assumed $z_0 \sim \text{Poisson}(40)$, and considered $t = 1, \ldots, 15$.

Results

Illustrative example

In the patch choice example, an individual in each of the 5 states has the same 2 available patch choices, so there are $2^5 = 32$ possible policies, $\pi_1, \ldots, \pi_{32}$ (Table 1). Each of these policies corresponds to a matrix P_π, which takes the form of (5). We calculated the dominant eigenvalue of each of these 32 matrices (Table 1) and found the largest of these dominant eigenvalues was $\lambda_{\pi^*,1} = 0.97$, corresponding to policy $\pi^* = \{\text{patch 2, patch 2, patch 1, patch 1, patch 1}\}$. The corresponding matrix is

$$P_{\pi^*} = \begin{bmatrix} 0 & 0 & 0.71 & 0 & 0 \\ 0.18 & 0 & 0 & 0.71 & 0 \\ 0 & 0.59 & 0 & 0 & 0.40 \\ 0 & 0 & 0.59 & 0 & 0.40 \\ 0 & 0 & 0 & 0.59 & 0.40 \end{bmatrix}. \tag{10}$$

By checking sequentially whether $(P_{\pi^*})^\xi$ is **positive** for $\xi = 1, 2, \ldots$, we found that $(P_{\pi^*})^6$ is positive, so P_{π^*} is primitive. Thus the conditions of the generalized Perron-Frobenius theorem are satisfied and we know that the rewards vector $F(t)$ will asymptotically decay exponentially according to $\lambda_{\pi^*,1}^t$, its structure will tend towards that of the corresponding right eigenvector $V_{\pi^*,1}$,

and policy π^* is the stationary policy. We confirmed this using the typical method of backwards induction (Figure 2).

We determined which of the dominant eigenvalues $\lambda_{\pi,1}$ of P_π for each policy π (Table 1), were within 1% of $\lambda_{\pi^*,1}$ and found five such policies: $\{1,2,1,1,1\}$, $\{1,2,2,1,1\}$, $\{2,1,1,1,1\}$, $\{2,1,2,1,1\}$, and $\{2,2,2,1,1\}$, where 1's and 2's denote patches 1 and 2, respectively.

Host feeding behaviour of parasitic wasps

Using the method outlined in Box 2, the optimal stationary policy π^* is to host feed if $x \leq x_c = 27$, the stationary threshold, and to parasitize otherwise. This stationary policy was the same as that found using backwards induction (Figure 3).

We performed Monte Carlo simulations (Figure 4 (a)), against which we compared the exact solutions obtained with the method of Markov chains (Figure 4 (b)). We also calculated the probability that the individual is in each state, conditional on the individual surviving to that time (Figure 4 (c)).

Discussion

Formulating an SDP problem using matrices alowed us to analytically determine optimal stationary policies and interpret the nature of convergence to these stationary policies. One of the most notable benefits of applying matrix tools to SDP analysis is a better understanding of the relative performance of other stationary policies. Numerical methods result in a single, optimal stationary policy. However, there may be several stationary policies which perform nearly as well so as to be indistinguishable in light of the uncertainty in parameter estimates and model structure (Mangel, 1991). Gaining a better picture of all policies with comparable fitness values can

provide a range of good options for managers, or help interpret field observations. For example, two distinct colour morphs of the desert flower *Linanthus Parryae* coexist in many areas (Epling and Dobzhansky, 1942; Wright, 1943), and multiple life history strategies—annual, biennial, and iteroparous—also coexist within a single population of *Streptanthus tortuosus*, a Californian wildflower (Gremer et al., in review). Stable coexistence suggests similar lifetime fitness between distinct strategies.

The matrix of state transition probabilities P_π is useful not only for finding stationary decisions but also for studying the evolution of an optimally behaving individual's state over time using Markov chains as the Markov transition matrix $M(t)$ is constructed as a by-product of constructing P_π.

In stationary decision and infinite horizon problems, numerical iterative methods require the user to specify a suitable stopping time criterion. This may be the number of time steps over which the optimal policy does not change or a requirement that the max norm, $||\cdot||_\infty$ (or, alternatively, the span seminorm (Puterman, 1994)) between successive iterations of the fitness function be very small (Marescot et al., 2013). For example, if we set a stopping criterion for backwards induction of $||F(:, T-(t+1)) - F(:, T-t)||_\infty < \epsilon = 0.001$, in the model for the parasitic wasp, we would stop at time $T - 391$. However, we can see in Figure 3, that this terminates the iterative method before the stationary policy is achieved. If, instead, we used the stopping criterion of Boutilier et al. (2000), which requires $||F(:, T-(t+1)) - F(:, T-t)||_\infty < \epsilon(1-e^{-\mu})/(2e^{-\mu})$, where $e^{-\mu}$ is the discount factor in this example, then we would stop at time $T - 791$, by which time the stationary policy has been reached. Analytic computation using matrix analytic methods can confirm that convergence to the true optimal solution has been reached by the stopping time.

For applications with a level of complexity similar to those discussed here, computational

constraints will likely be minor. For example, all of the code required in our examples using any of the methods considered (i.e., backwards induction or matrix methods) ran in less than 20 seconds on a modern laptop PC (Intel(R) Core(TM) i7 CPU, 32 GB of RAM, and a 64-bit operating system). We suspect that the numerical iterative methods will tend to find solutions faster than the matrix analytic methods in most cases, though we have not given this a thorough treatment here. For both matrix and numerical methods, computational complexity increases exponentially with the addition of more state variables (e.g., simultaneous consideration of an individual's age, reproductive state, energetic state, etc.), leading to the "curse of dimensionality" (Bellman, 1957). If multiple state variables must be considered, other methods may become more appropriate, requiring approximate dynamic programming methods (Powell, 2007) such as reinforcement learning (Frankenhuis et al., 2018), or more heuristic methods (Nicol and Chadès, 2011).

There are similarities between the mathematical SDP results described here and other areas of ecological theory. For example, analytical eigenvalue equations have been used to study the evolution of optimal life history strategies (Charnov and Schaffer, 1973; Bulmer, 1994). Selection on life history strategies has also been considered in the context of matrix population models, where sensitivity analysis on expected lifetime reproduction (R_0) indicates the strength of selection acting on a given life history parameter (see Caswell (2001) for an overview). Theoretical results on Markov chains with rewards initially developed in the context of stochastic dynamic programming (Howard, 1960) have recently been applied to studies in demography (Caswell, 2011; van Daalen and Caswell, 2017).

We do not propose that these matrix methods replace backwards induction or value iteration, but rather that they are additional tools. The two approaches are complementary, and, ideally, will be used in concert. Even if one is interested in transient dynamics near the terminal

time, running that same model until it reaches its stationary decision state and then confirming that it has reached the correct state with our proposed matrix methods would be an excellent check for errors in the numerical code.

The examples we have considered here were chosen for their simplicity and general applicability. One of the benefits of SDP, however, is model flexibility. For example, some SDP applications include variable time increments; e.g., $f(x,t)$ is a function of both $f(x,t+\tau)$ and $f(x,t+1)$ for some integer τ (Mangel, 1987). Others require more than one state variable (Brodin et al., 2017), which would need to be dealt with using either tensors or matrices incorporating multiple states. These modifications will need to be dealt with on a case-by-case basis, building from the foundations of the two canonical equations.

Conclusion

We have illustrated an alternative formulation of SDP models in biology, using the language of matrices, as well as highlighted useful applications of relevant mathematical results. For two canonical equations of SDP in ecology, we used these mathematical results to analytically obtain the optimal stationary decisions. This resulted in additional insights into the existence and nature of alternate, nearly optimal policies, as well as novel insight into the nature of convergence. The transition matrices required for this method also allowed for straightforward implementation of Markov chains to study the probability distribution of an individual's state. We hope this will encourage the incorporation of further results from SDP theory outside ecology and expand the standard toolkit used to analyse SDP models in ecology, evolutionary biology, conservation, and resource management.

Data Availability

All computations were performed in Matlab (2018b) and all code is available at doi:10.5281/zenodo.2547815.

Author Contributions

JRR conducted all model analysis and wrote the manuscript. MM, AED, and MAL provided substantial scientific direction and writing input.

Funding

This was work was supported by the Natural Sciences and Engineering Research Council of Canada, Alberta Innovates, and the Killam Trust through scholarships to JRR. MM acknowledges NSF grant DEB 1555729 and ONR Grant N00014-19-1-2494. AED acknowledges support from ArcticNet, Environment and Climate Change Canada, Hauser Bears, Natural Sciences and Engineering Research Council of Canada, Polar Bears International, Polar Continental Shelf Project, Quark Expeditions, and World Wildlife Fund (Canada). MAL gratefully acknowledges an NSERC Discovery Grant and a Canada Research Chair.

References

Bäuerle, N. and Rieder, U. (2011). *Markov decision processes with applications to finance*. Springer-Verlag, Heidelberg.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton.

Bogich, T. and Shea, K. (2008). A state-dependent model for the optimal management of an invasive metapopulation. *Ecol. Appl.*, 18(3):748–761.

Boutilier, C., Dearden, R., and Goldszmidt, M. (2000). Stochastic dynamic programming with factored representations. *Artif. Intell.*, 121(1):49–107.

Brodin, A., Nilsson, J. Å., and Nord, A. (2017). Adaptive temperature regulation in the little bird in winter: predictions from a stochastic dynamic programming model. *Oecologia*, 185:43–54.

Bull, C. D., Metcalfe, N. B., and Mangel, M. (1996). Seasonal matching of foraging to anticipated energy requirements in anorexic juvenile salmon. *Proc. R. Soc. B Biol. Sci.*, 263(1366):13–18.

Bulmer, M. (1994). Life-history evolution. In *Theor. Evol. Ecol.*, pages 70–101. Sinauer, Sunderland, MA.

Caswell, H. (2001). *Matrix Population Models*. Sinauer, Sunderland, MA, second edition.

Caswell, H. (2011). Beyond R_0: Demographic models for variability of lifetime reproductive output. *PLoS One*, 6(6).

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

Chan, M. S. and Godfray, H. C. (1993). Host-feeding strategies of parasitoid wasps. *Evol. Ecol.*, 7(6):593–604.

Charnov, E. L. and Schaffer, W. M. (20191973). Life-history consequences of natural selection: Cole's result revisited. *Am. Nat.*, 107(958):791–793.

Charnov, E. L. and Skinner, S. W. (1984). Evolution of host selection and clutch size in parasitoid wasps. *Florida Entomol.*, 67(1):5–21.

Clark, C. W. and Mangel, M. (2000). *Dynamic State Variable Models in Ecology*. Oxford University Press, New York.

Epling, C. and Dobzhansky, T. (1942). Genetics of natural populations. VI. Microgeographic races in *Linanthus Parryae*. *Genetics*, 27:317–332.

Frankenhuis, W. E., Panchanathan, K., and Barto, A. G. (2018). Enriching behavioral ecology with reinforcement learning methods. *Behav. Processes*, (January):0–1.

Howard, R. A. (1960). *Dynamic programming and Markov processes*. MIT Press, Cambridge, MA.

Lubow, B. C. (1995). Generalized software for solving stochastic dynamic optimization problems. *Wildl. Soc. Bull.*, 23(4):738–742.

Mangel, M. (1987). Opposition site selection and clutch size in insects. *J. Math. Biol.*, 25(1):1–22.

Mangel, M. (1989). Evolution of host selection in parasitoids: does the state of the parasitoid matter? *Am. Nat.*, 133(5):688–705.

Mangel, M. (1991). Adaptive walks on behavioural landscapes and the evolution of optimal behaviour by natural selection. *Evol. Ecol.*, 5:30–39.

Mangel, M. (2015). Stochastic dynamic programming illuminates the link between environment, physiology, and evolution. *Bull. Math. Biol.*, 77(5):857–877.

Mangel, M. and Bonsall, M. B. (2008). Phenotypic evolutionary models in stem cell biology: replacement, quiescence, and variability. *PLoS One*, 3(2).

Mangel, M. and Clark, C. (1986). Towards a unified foraging theory. *Ecology*, 67(5):1127–1138.

Mangel, M. and Clark, C. W. (1988). *Dynamic Modeling in Behavioral Ecology*. Princeton University Press, Princeton, NJ.

Marescot, L., Chapron, G., Chadès, I., Fackler, P. L., Duchamp, C., Marboutin, E., and Gimenez, O. (2013). Complex decisions made simple: a primer on stochastic dynamic programming. *Methods Ecol. Evol.*, 4(9):872–884.

McNamara, J. M. (1990). The policy which maximises long-term survival of an animal faced with the risks of starvation and predation. *Adv. Appl. Probab.*, 22(2):295–308.

McNamara, J. M. (1991). Optimal life histories: a generalization of the Perron-Frobenius Theorem. *Theor. Popul. Biol.*, 40:230–245.

McNamara, J. M. and Houston, A. I. (1986). The common currency for behavioral decisions. *Am. Nat.*, 127(3):358–378.

McNamara, J. M., Houston, A. I., and Collins, E. J. (2001). Optimality models in behavioral biology. *SIAM Rev.*, 43(3):413–466.

Nicol, S. and Chadès, I. (2011). Beyond stochastic dynamic programming: A heuristic sampling method for optimizing conservation decisions in very large state spaces. *Methods Ecol. Evol.*, 2(2):221–228.

Parker, G. and Smith, J. M. (1990). Optimality theory in evolutionary biology. *Nature*, 348:27–33.

Powell, W. B. (2007). *Approximate dynamic programming: Solving the curses of dimensionality*. John Wiley & Sons, Hoboken, NJ.

Puterman, M. L. (1994). *Markov Decision Processes; Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Hoboken, New Jersey.

Rees, M., Sheppard, A., Briese, D., and Mangel, M. (1999). Evolution of size-dependent flowering in *Onopordum illyricum*: a quantitative assessment of the role of stochastic selection pressures. *Am. Nat*, 154(154):628–651.

Reimer, J., Mangel, M., Derocher, A. E., and Lewis, M. A. (2019). Modeling optimal responses and fitness consequences in a changing arctic. *Global Change Biology*, doi: 10.1111/gcb.14681.

Runge, M. C. and Johnson, F. A. (2002). The importance of functional form in optimal control. *Ecology*, 83(5):1357–1371.

Shea, K. and Possingham, H. P. (2000). Optimal release strategies for biological control agents: an application of stochastic dynamic programming to population management. *J. Appl. Ecol.*, 37(1):77–86.

Sheshkin, T. J. (2010). *Markov chains and decision processes for engineers and managers*. CRC Press, Boca Raton, FL.

Sigaud, O. and Buffet, O., editors (2010). *Markov decision processes in artificial intelligence*. Wiley, Hoboken, NJ.

van Daalen, S. F. and Caswell, H. (2017). Lifetime reproductive output: individual stochasticity, variance, and sensitivity analysis. *Theor. Ecol.*, 10(3):355–374.

Venner, S., Chadès, I., Bel-Venner, M. C., Pasquet, A., Charpillet, F., and Leborgne, R. (2006). Dynamic optimization over infinite-time horizon: web-building strategy in an orb-weaving spider as a case study. *J. Theor. Biol.*, 241(4):725–733.

Wright, S. (1943). An analysis of local variability of flower color in *Linanthus Parryae*. *Genetics*, 28(March):139–156.

Tables

Table 1: All possible policies π (i.e., the patch choice between patch 1 and 2 for an individual in each of the 5 possible states) and the dominant eigenvalue $\lambda_{\pi,1}$ of each policy's associated matrix P_π. The stationary policy π^* is the one with the largest dominant eigenvalue, in grey.

		policies Π								
		π_1	π_2	π_3	π_4	π_5	$\cdots$	π^*	$\cdots$	π_{32}
patch choice	i_1	1	1	1	1	1		2		2
	i_2	1	1	1	1	1		2		2
	i_3	1	1	1	1	2	$\cdots$	1	$\cdots$	2
	i_4	1	1	2	2	1		1		2
	i_5	1	2	1	2	1		1		2
	$\lambda_{\pi,1}$	0.94	0.90	0.94	0.89	0.96	$\cdots$	0.97	$\cdots$	0.89

Figures

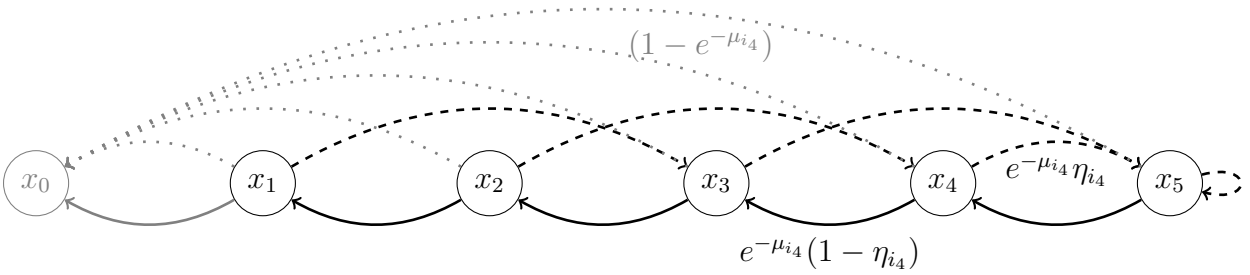

Figure 1: State and decision dependent transition probabilities for the patch selection example. A living individual may be in 1 of 5 states ($x_1, \ldots, x_5$). State x_0 is the absorbing state of dead individuals. Due to space constraints, we have only written transition probabilities corresponding to each arrow for an individual in state x_4. All arrows in grey are associated with the absorbing state and not included in the matrix P_π (but are included in the Markov matrix $\hat{P}_\pi$).

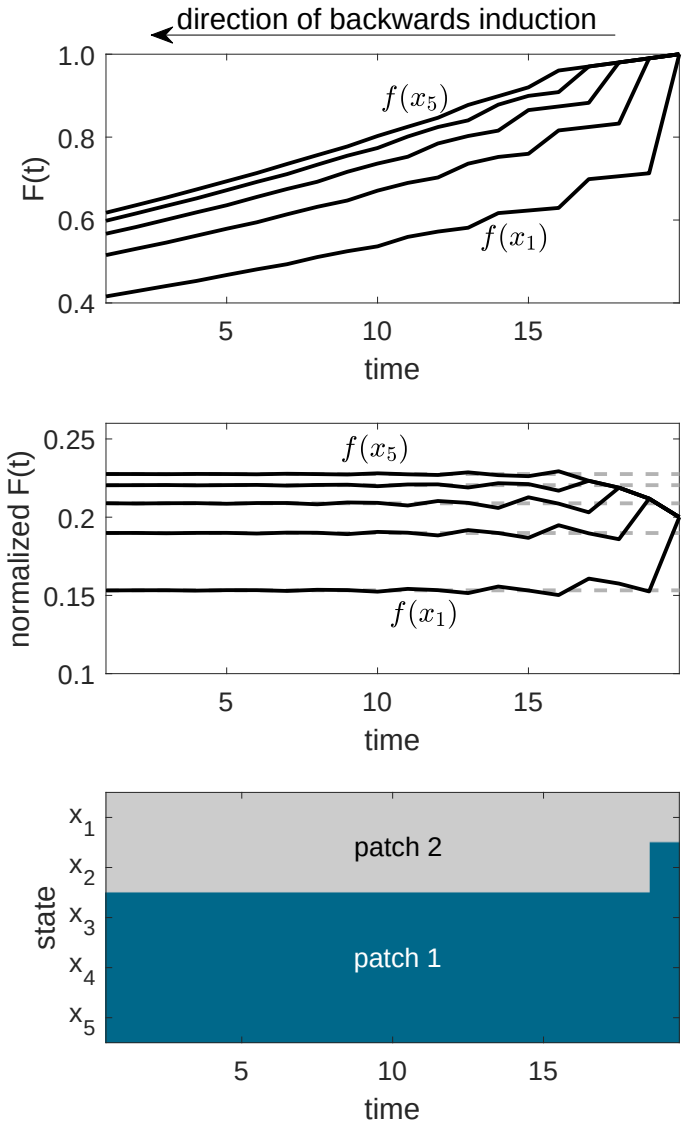

Figure 2: Solution (obtained using backwards induction; arrow at top) of the illustrative patch choice stochastic dynamic programming example. Top: Asymptotic exponential decay of the fitness vector $F(t)$ backwards in time, as t becomes further away from the terminal time. The bottom curve is $f(x_1, t)$ and the top curve is $f(x_5, t)$, with the fitness curves for states x_2 to x_4 in between. Middle: Normalized solution of $F(t)$ converging backwards in time to the right eigenvector $V_{\pi^*,1}$ (grey dashed lines) corresponding to the stationary policy π^*. Bottom: Convergence backwards in time to the stationary policy, $\pi^* = \{\text{patch } 2, \text{patch } 2, \text{patch } 1, \text{patch } 1, \text{patch } 1\}$.

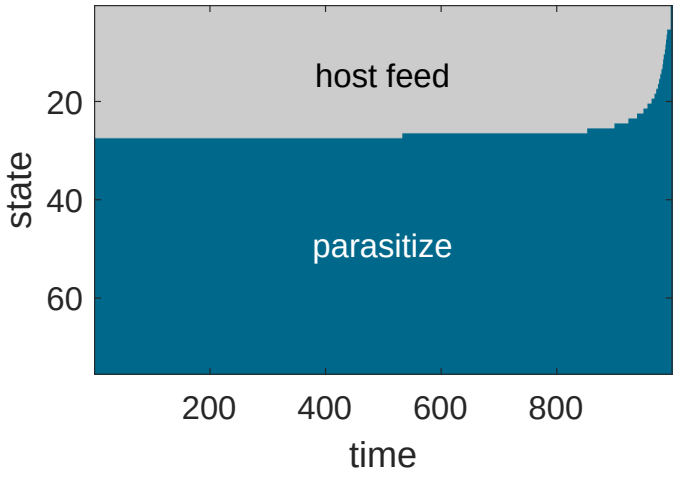

Figure 3: Optimal decisions of the parasitic wasp model of Chan and Godfrey (1993), obtained using backwards induction. The policy at time $t = 1$ is the stationary policy, which is the same as that obtained using our proposed matrix method.

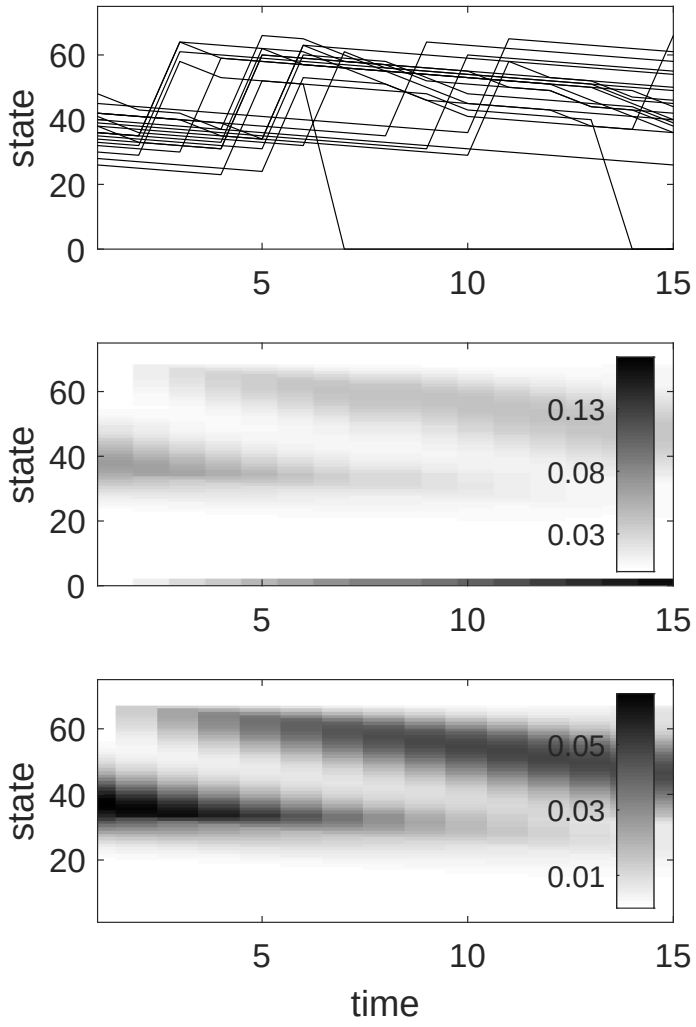

Figure 4: Changes in an optimally behaving individual's state in the parasitic wasp example. (a) 20 Monte Carlo simulations. If we continued to run more of these, and calculated the proportion of simulations in each state at a given time, we would end up with (b). (b) Heat map of the probability of being in a given state at a given time, obtained using Markov chains. (c) Heat map of the probability of being in a given state at a given time, conditional on surviving to that time, obtained using Markov chains.

A Glossary of matrix terminology

For a square matrix P, of size $k \times k$, we remind the reader of the following definitions:

- **dominant eigenvalue** of P: the largest (in magnitude) of all eigenvalues of P
- **eigenvector** of P: a vector of length k which, when multiplied by P, changes only by multiplication with a scalar, i.e., $PV = \lambda V$, where λ is the associated eigenvalue
- **eigenvalue** of P: a scalar (real or complex number) λ with the property that $PV = \lambda V$, where V is the eigenvector corresponding to λ
- **Markov matrix**: a **nonnegative matrix** whose rows (or, equivalently, columns) sum to 1; also known as a stochstic matrix
- **nonnegative matrix**: a matrix where each of the entries is ≥ 0
- **positive matrix**: a matrix where each of the entries is > 0
- **primitive matrix**: a matrix for which P^{ξ} is **positive** for some integer ξ
- **substochastic matrix**: a nonnegative matrix whose rows sum to ≤ 1, with at least one row summing to $<$ (or, equivalently, columns)