Received: 22 May 2019 | Accepted: 6 July 2019

DOI: 10.1111/2041-210X.13291

RESEARCH ARTICLE

Methods in Ecology and Evolution BRITISH ECOLOGICAL SOCIETY

Matrix methods for stochastic dynamic programming in ecology and evolutionary biology

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

[1]Department of Biological Sciences, University of Alberta, Edmonton, AB, Canada

[2]Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, AB, Canada

[3]Department of Biology, University of Bergen, Bergen, Norway

[4]Institute of Marine Sciences and Department of Applied Mathematics, University of California, Santa Cruz, CA, USA

Correspondence
Jody R. Reimer
Email: jrreimer@ualberta.ca

Funding information
This work was supported by the Natural Sciences and Engineering Research Council of Canada, Alberta Innovates, and the Killam Trust through scholarships to J.R.R. M.M. acknowledges NSF grant DEB 1555729 and ONR Grant N00014-19-1-2494. A.E.D. 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). M.A.L. gratefully acknowledges an NSERC Discovery Grant and a Canada Research Chair.

Handling Editor: Torbjørn Ergon

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 fishery managers must make tradeoffs while striving for conservation or economic goals (e.g. costs vs. 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.

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

1 | INTRODUCTION

Tradeoffs are an unavoidable part of being alive. Tradeoffs may be physiological (e.g. how much energy to allocate to growth vs. reproduction; Rees, Sheppard, Briese, & Mangel, 1999), or behavioural (e.g. how to balance energy gain with predator avoidance; Mangel & Clark, 1986; McNamara & 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 & Johnson, 2002). The objective may be to control an invasive species (Bogich & 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 & Rieder, 2011) and artificial intelligence (Sigaud & 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.

Stochastic dynamic programming 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, Houston, and Collins (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, Metcalfe, & Mangel, 1996; Reimer, Mangel, Derocher, & Lewis, 2019a). 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 (Chan & Godfray, 1993; Mangel, 1989; Shea & 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 (Mangel & Bonsall, 2008; Venner et al., 2006), but are *infinite horizon problems*. For example, managers may wish to maximize the total number of animals that may be harvested indefinitely (Runge & 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 (Clark & Mangel, 2000; Puterman, 1994). 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 (Chadès, Chapron, Cros, Garcia, & Sabbadin, 2014; Lubow, 1995; Marescot et al., 2013).

Stochastic dynamic programming 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 analysing 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 (the 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 & 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 & Godfray, 1993).

2 | MATERIALS AND METHODS

2.1 | Stochastic dynamic programming

Stochastic dynamic programming models contain several key components (Clark & 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 $f(x,t) = \max \mathbb{E}$ [future reward, given state x 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; for example, 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 = {patch 1 or 2}:

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

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 Equations 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).

2.2 | 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 five 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 the directed graph (Figure 1). If an individual finds food in either patch, their reserves increase by 2 units

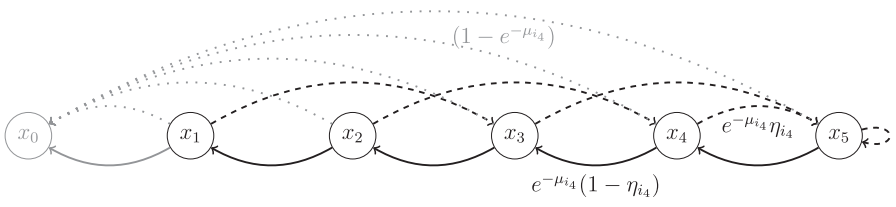

FIGURE 1 State and decision-dependent transition probabilities for the patch selection example. A living individual may be in 1 of 5 states (x_1, ..., x_5). State x_0 is the absorbing state of dead individuals. Due to space constraints, we have only written transition probabilities corresponding to each 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 $\hat{P}_\pi$)

($y_1 = y_2 = 3$; 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 $i_n \in \{\text{patch 1}, \text{patch 2}\}$ made by an individual in state x_n. We are interested in the stationary decision problem, that is, 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.

2.3 | Existing methods for obtaining stationary decisions

Backwards induction is typically used to solve stationary decision problems (see Clark & 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, that is, $T - t \gg 1$.

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 (The MathWorks Inc., Natick, MA, USA) and all code is available at https://doi.org/10.5281/zenodo.2547815. For those who prefer working in R, we have also included an overview of key R commands (Appendix S1, online Supplementary Material).

2.4 | Matrix notation

While applications of SDP in biology typically describe the fitness function component-wise for each state x, such as in Equation 1 or Equation 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. Chadès et al., 2014; Marescot et al., 2013), but do not discuss the rich theory of **nonnegative matrices** (bolded terms in Glossary, Appendix A) we use here.

We let $F(t) = \left[f(x_1, t), \ldots, f(x_k, t)\right]^\top$ 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 i-tuple of decisions, one for each state. Π denotes the set of all possible policies. In Equation 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 Equation 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 $G_\pi = \left[g_{\pi,1}, \ldots, g_{\pi,k}\right]^\top$ be a vector of immediate rewards, we can similarly rewrite Equation 2 as

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

2.5 | 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 $i_1, \ldots, i_5$ 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).

2.6 | 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 (For the classical Perron–Frobenius theorem in the context of matrix population models see Caswell (2001)) by McNamara (1991). We highlight relevant mathematical results and include full technical details in Appendix 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 $|\lambda_{\pi,1}| \geq \cdots \geq |\lambda_{\pi,k}|$. 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 Appendix 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

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

that is, $F(t)$ 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.

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 Appendix S4, online Supplementary Material.

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

2.7 | Analytic method for resource allocation problems

Using results from general SDP theory (Appendix 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,

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

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 (Appendix 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 $\pi \in \Pi$, that is,

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

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

2.8 | Host feeding behaviour of parasitic wasps

The evolution of insect parasitoid behaviour has been an especially fruitful area of SDP research (Charnov & Skinner, 1984; Clark &

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

Mangel, 2000; Mangel, 1989). 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; for example, 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 $e^{-\mu}$, 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 $T = 1{,}000$.

The resulting SDP equation is

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

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

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

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 Equation 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 Appendix 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.

2.9 | 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, $\prod$ 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 $k + 1$.

2.10 | 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 & 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 & 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(transitioning from state x_j to state x_k in one time step). Recall that $p_\pi(x_j, x_k)$ = Pr(transitioning from state x_j to state x_k 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(an optimally behaving individual is in state x at time t), with vector notation $Z(t)$. We obtain the probability of the individual being in each state using the forward recursion equation

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

TABLE 1 All possible policies π (i.e. the patch choice between patch 1 and 2 for an individual in each of the five 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 $\prod$								
	π_1	π_2	π_3	π_4	π_5	⋯	π^*	⋯	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	⋯	1	⋯	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	⋯	0.97	⋯	0.89

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 $z_0 \sim$ Poisson(40), and considered t = 1, ..., 15.

3 | RESULTS

3.1 | 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, π_1, ..., π_{32} (Table 1). Each of these policies corresponds to a matrix P_π, which takes the form of Equation 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 π^* = {patch 2, patch 2, patch 1, patch 1, patch 1}. The corresponding matrix is

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

By checking sequentially whether $(P_{\pi^*})^\xi$ is **positive** for ξ = 1, 2, ..., 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 $\lambda^t_{\pi^*,1}$, 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.

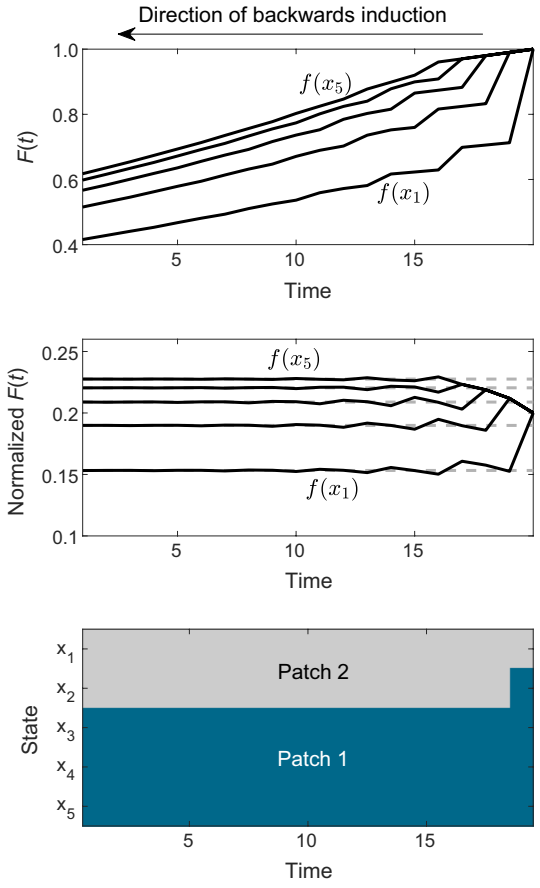

FIGURE 2 Solution (obtained using backwards induction; arrow at top) of the illustrative patch choice stochastic dynamic programming example. Top: Asymptotic exponential decay of the fitness vector $F(t)$ backwards in time, as t becomes further away from the terminal time. The bottom curve is $f(x_1, t)$ 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, π^* = {patch 2, patch 2, patch 1, patch 1, patch 1}

3.2 | Host feeding behaviour of parasitic wasps

Using the method outlined in Box 2, the optimal stationary policy π^* is to host feed if $x \le x_c$ = 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 4a), against which we compared the exact solutions obtained with the method

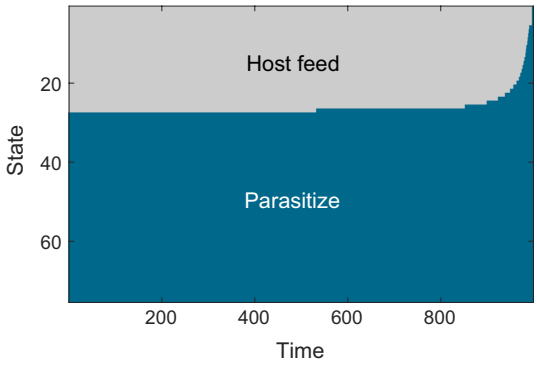

FIGURE 3 Optimal decisions of the parasitic wasp model of Chan and Godfrey (1993), obtained using backwards induction. The policy at time $t = 1$ is the stationary policy, which is the same as that obtained using our proposed matrix method

of Markov chains (Figure 4b). We also calculated the probability that the individual is in each state, conditional on the individual surviving to that time (Figure 4c).

4 | DISCUSSION

Formulating an SDP problem using matrices allowed 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 & 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

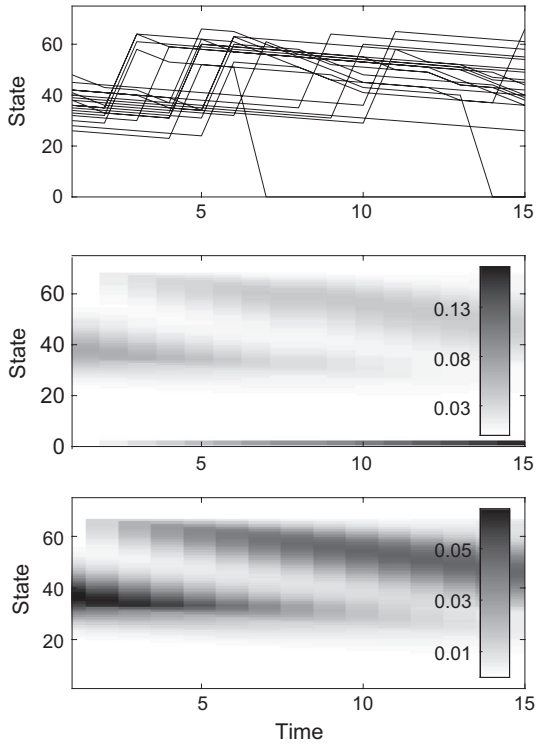

FIGURE 4 Changes in an optimally behaving individual's state in the parasitic wasp example. (a) 20 Monte Carlo simulations. If we continued to run more of these, and calculated the proportion of simulations in each state at a given time, we would end up with (b). (b) Heat map of the probability of being 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

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, Dearden, and Goldszmidt (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 s 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, Panchanathan, & Barto, 2018), or more heuristic methods (Nicol & 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 (Bulmer, 1994; Charnov & Schaffer, 1973). 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 & 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, Nilsson, & Nord, 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.

5 | 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.

AUTHORS' CONTRIBUTIONS

J.R.R. conducted all model analysis and wrote the manuscript. M.M., A.E.D. and M.A.L. provided substantial scientific direction and writing input.

DATA AVAILABILITY STATEMENT

All computations were performed in Matlab (The MathWorks Inc.) and all code is available at https://doi.org/10.5281/zenodo.2547815 (Reimer, Mangel, Derocher, & Lewis, 2019b) (https://zenodo.org/record/2547815).

ORCID

Jody R. Reimer https://orcid.org/0000-0001-7742-2728
Marc Mangel https://orcid.org/0000-0002-9406-697X
Andrew E. Derocher https://orcid.org/0000-0002-1104-7774

REFERENCES

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

Bellman, R. (1957). *Dynamic programming*. Princeton, NJ: Princeton University Press.

Bogich, T., & Shea, K. (2008). A state-dependent model for the optimal management of an invasive metapopulation. *Ecological Applications*, *18*(3), 748–761. https://doi.org/10.1890/07-0642.1

Boutilier, C., Dearden, R., & Goldszmidt, M. (2000). Stochastic dynamic programming with factored representations. *Artificial Intelligence*, *121*(1), 49–107. https://doi.org/10.1016/s0004-3702(00)00033-3

Brodin, A., Nilsson, J. Å., & Nord, A. (2017). Adaptive temperature regulation in the little bird in winter: Predictions from a stochastic dynamic programming model. *Oecologia*, *185*, 43–54. https://doi.org/10.1007/s00442-017-3923-3

Bull, C. D., Metcalfe, N. B., & Mangel, M. (1996). Seasonal matching of foraging to anticipated energy requirements in anorexic juvenile salmon. *Proceedings of the Royal Society B-Biological Sciences*, *263*(1366), 13–18. https://doi.org/10.1098/rspb.1996.0003

Bulmer, M. (1994). Life-history evolution. In *Theoretical evolution ecology* (pp. 70–101). Sunderland, MA: Sinauer.

Caswell, H. (2001). *Matrix population models* (2nd ed.). Sunderland, MA: Sinauer.

Caswell, H. (2011). Beyond R_0: Demographic models for variability of lifetime reproductive output. *PLoS ONE*, *6*(6). https://doi.org/10.1371/journal.pone.0020809

Chadès, I., Chapron, G., Cros, M. J., Garcia, F., & Sabbadin, R. (2014). MDPtoolbox: A multi-platform toolbox to solve stochastic dynamic programming problems. *Ecography*, *37*(9), 916–920. https://doi.org/10.1111/ecog.00888

Chan, M. S., & Godfray, H. C. (1993). Host-feeding strategies of parasitoid wasps. *Evolutionary Ecology*, *7*(6), 593–604. https://doi.org/10.1007/bf01237823

Charnov, E. L., & Schaffer, W. M. (1973). Life-history consequences of natural selection: Cole's result revisited. *American Naturalist*, *107*(958), 791–793. https://doi.org/10.1086/282877

Charnov, E. L., & Skinner, S. W. (1984). Evolution of host selection and clutch size in parasitoid wasps. *The Florida Entomologist*, *67*(1), 5–21. https://doi.org/10.2307/3494101

Clark, C. W., & Mangel, M. (2000). *Dynamic state variable models in ecology*. New York: Oxford University Press.

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

Frankenhuis, W. E., Panchanathan, K., & Barto, A. G. (2018). Enriching behavioral ecology with reinforcement learning methods. *Behavioural Processes*, (*161*), 94–100.

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

Lubow, B. C. (1995). Generalized software for solving stochastic dynamic optimization problems. *Wildlife Society Bulletin*, *23*(4), 738–742.

Mangel, M. (1987). Opposition site selection and clutch size in insects. *Journal of Mathematical Biology*, *25*(1), 1–22.

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

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

Mangel, M. (2015). Stochastic dynamic programming illuminates the link between environment, physiology, and evolution. *Bulletin of Mathematical Biology*, *77*(5), 857–877.

Mangel, M., & Bonsall, M. B. (2008). Phenotypic evolutionary models in stem cell biology: Replacement, quiescence, and variability. *PLoS ONE*, *3*(2), e1591. https://doi.org/10.1371/journal.pone.0001591

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

Mangel, M., & Clark, C. W. (1988). *Dynamic modeling in behavioral ecology*. Princeton, NJ: Princeton University Press.

Marescot, L., Chapron, G., Chadès, I., Fackler, P. L., Duchamp, C., Marboutin, E., & Gimenez, O. (2013). Complex decisions made simple: A primer on stochastic dynamic programming. *Methods in Ecology and Evolution*, *4*(9), 872–884. https://doi.org/10.1111/2041-210x.12082

McNamara, J. M. (1990). The policy which maximises long-term survival of an animal faced with the risks of starvation and predation. *Advances in Applied Probability*, *22*(2), 295–308. https://doi.org/10.2307/1427537

McNamara, J. M. (1991). Optimal life histories: A generalization of the Perron-Frobenius Theorem. *Theoretical Population Biology*, *40*, 230–245. https://doi.org/10.1016/0040-5809(91)90054-j

McNamara, J. M., & Houston, A. I. (1986). The common currency for behavioral decisions. *American Naturalist*, *127*(3), 358–378. https://doi.org/10.1086/284489

McNamara, J. M., Houston, A. I., & Collins, E. J. (2001). Optimality models in behavioral biology. *SIAM Review*, *43*(3), 413–466. https://doi.org/10.1137/s0036144500385263

Nicol, S., & Chadès, I. (2011). Beyond stochastic dynamic programming: A heuristic sampling method for optimizing conservation decisions in very large state spaces. *Methods in Ecology and Evolution*, *2*(2), 221–228. https://doi.org/10.1111/j.2041-210x.2010.00069.x

Parker, G., & Smith, J. M. (1990). Optimality theory in evolutionary biology. *Nature*, *348*, 27–33. https://doi.org/10.1038/348027a0

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

Puterman, M. L. (1994). *Markov decision processes; discrete stochastic dynamic programming*. Hoboken, NJ: John Wiley & Sons.

Rees, M., Sheppard, A., Briese, D., & Mangel, M. (1999). Evolution of size-dependent flowering in *Onopordum illyricum*: A quantitative assessment of the role of stochastic selection pressures. *American Naturalist*, *154*(154), 628–651. https://doi.org/10.1086/303268

Reimer, J., Mangel, M., Derocher, A. E., & Lewis, M. A. (2019a). Modeling optimal responses and fitness consequences in a changing arctic. *Global Change Biology*,*25*, 3450–3461. https://doi.org/10.1111/gcb.14681

Reimer, J., Mangel, M., Derocher, A. E., & Lewis, M. A. (2019b). Code from: Matrix methods for stochastic dynamic programming in ecology and evolutionary biology. *Zenodo digital repository*. https://zenodo.org/record/2547815

Runge, M. C., & Johnson, F. A. (2002). The importance of functional form in optimal control. *Ecology*, *83*(5), 1357–1371. https://doi.org/10.1890/0012-9658(2002)083[1357:tioffi]2.0.co;2

Shea, K., & Possingham, H. P. (2000). Optimal release strategies for biological control agents: An application of stochastic dynamic programming to population management. *Journal of Applied Ecology*, *37*(1), 77–86. https://doi.org/10.1046/j.1365-2664.2000.00467.x

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

Sigaud, O., & Buffet, O. (Eds.) (2010). *Markov decision processes in artificial intelligence*. Hoboken, NJ: Wiley.

van Daalen, S. F., & Caswell, H. (2017). Lifetime reproductive output: Individual stochasticity, variance, and sensitivity analysis. *Theoretical Ecology*, *10*(3), 355–374. https://doi.org/10.1007/s12080-017-0335-2

Venner, S., Chadès, I., Bel-Venner, M. C., Pasquet, A., Charpillet, F., & Leborgne, R. (2006). Dynamic optimization over infinite-time horizon: Web-building strategy in an orb-weaving spider as a case study. *Journal of Theoretical Biology*, *241*(4), 725–733. https://doi.org/10.1016/j.jtbi.2006.01.008

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

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Reimer JR, Mangel M, Derocher AE, Lewis MA. Matrix methods for stochastic dynamic programming in ecology and evolutionary biology. *Methods Ecol Evol*. 2019;00:1–10. https://doi.org/10.1111/2041-210X.13291

APPENDIX 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 **non-negative matrix** whose rows (or, equivalently, columns) sum to 1; also known as a stochastic matrix
- **non-negative 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 non-negative matrix whose rows sum to ≤1, with at least one row summing to < (or, equivalently, columns)

Supplementary materials for "Matrix methods for stochastic dynamic programming in ecology and evolutionary biology"

Jody R. Reimer†, Marc Mangel, ndrew E. Derocher, Mark . Lewis

† Corresponding author. [email] jrreimer@ualberta.ca.

S1 Helpful R commands

We have performed all of our computations in M TL B (accessible at doi:10.5281/zenodo.2547815). However, for readers who are more comfortable in R, we provide a quick overview of the matrix commands necessary to implement this method in R.

action	R command
create a square zero matrix P of size k	P $\leftarrow$ matrix(0,k,k)
create a zero vector G of length k	G $\leftarrow$ matrix(0,k,1)
assign $p(x_j, x_k) = C$ in matrix P	P[x_j, x_k] $\leftarrow$ C
multiply matrices and vectors	$\% * \%$ (e.g., P$\% * \%$G)
create a $k \times k$ identity matrix	diag(k)
transpose of matrix P	t(P)
inverse of matrix P (i.e., P^{-1})	solve(P)
eigenvalues of P	y $\leftarrow$ eigen(P); y$val are the eigenvalues
eigenvectors of P	y $\leftarrow$ eigen(P); y$vec are the eigenvectors
check sum of each row in matrix P	rowsums(P)

S2 Relevant theory

The Generalized Perron-Frobenius theorem (McNamara, 1991)

McNamara (1991) presented the theorem in the form we have used here, but this result relies heavily on results from (Sladky, 1976; Grey, 1984; McNamara, 1990).

Consider an equation of the form

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

and define the optimal policy π^* as that satisfying

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

If P_π is primitive (i.e., $P_\pi^\xi > 0$ for some $\xi > 0$), then the following are true: (i) the dominant eigenvalue $\lambda_{\pi,1}$ corresponding to P_π satisfies $\lambda_{\pi,1} = \max_\pi \lambda_{\pi,1}$, (ii) $\lambda_{\pi,1}$ is uniquely defined and $V_{\pi,1}$ is unique up to multiplication by a constant, (iii) $\lambda^* = \lambda_{\pi,1}$, and (iv) $\lim_{t\to\infty}(\lambda_{\pi,1})^{-t} F(t) \propto V_{\pi,1}$.

On the dominant eigenvalue of P

Many of these results rely on the fact that the magnitude of the dominant eigenvalue (i.e., the spectral radius) of P_π is < 1, for all π. We demonstrate this by observing that every term in matrix P_π will include a discount term (usually the survival probability of an individual) in biological applications of SDP. Let

$$m = \min_{i=1,\ldots,k} e$$

be the smallest of these discount terms. Provided there is a non-zero rate of mortality over each time step, so $\mu_i > 0$, then $m < 1$. If we factor out m, we can rewrite P_π as $P_\pi = m\tilde{P}_\pi$. Since $\tilde{P}_\pi$ is a sub-stochastic matrix, with each row summing to ≤ 1, its spectral radius is ≤ 1 (i.e., $\rho(\tilde{P}_\pi) \leq 1$). Then

$$\rho(P_\pi) = \rho(m\tilde{P}_\pi) = m\rho(\tilde{P}_\pi) < 1.$$

Existence, uniqueness, and structure of the optimal stationary solution for the second canonical equation

The following results and their proofs can be found in Puterman (1994). We restate the relevant theorems here for reference, with notation changed for consistency. The existence of a unique solution $\tilde{F}_\pi$ for any stationary policy π is guaranteed by Theorem 6.2.5. The form of this solution for any stationary policy is described in Theorem 6.1.1. The existence of an optimal stationary policy is guaranteed by Theorem 6.2.10 for an infinite horizon problem (the analagous theorem for a finite horizon problem can be found in Proposition 4.4.3 in Puterman (1994)). Theorem 6.2.7c states that this optimal stationary policy has the largest solution $\tilde{F}$ out of all possible policies.

Theorem 6.2.5 (Puterman (1994)) *Suppose P_π has a spectral radius < 1, the set of possible states χ is finite, and the immediate rewards G_π are bounded for all policies. Then there exists a unique solution $\tilde{F}_\pi$ satisfying $\tilde{F}_\pi = G_\pi + P_\pi \tilde{F}_\pi$.*

Theorem 6.1.1 (Puterman (1994)) *Suppose P_π has a spectral radius < 1. Then for any stationary policy π, $\tilde{F}_\pi$ is the unique solution of*

$$\tilde{F}_\pi = G_\pi + P_\pi \tilde{F}_\pi.$$

Further, $\tilde{F}_\pi$ may be written as

$$\tilde{F}_\pi = (I \quad P_\pi)^{\ 1} G_\pi.$$

Theorem 6.2.10 (Puterman (1994)) *ssume the set of possible states χ is discrete and that the set of possible actions is finite for an individual in each state $x \in \chi$. Then there exists an optimal stationary policy π^*.*

Theorem 6.2.7c (Puterman (1994)) *Let χ be discrete, then the solution of $\tilde{F}_\pi = G_\pi + P_\pi \tilde{F}_\pi$ satisfies*

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

S3 onditions of primitivity

nonnegative matrix P is primitive if $P^\xi > 0$ for some integer $\xi > 0$. The primitivity of a nonnegative matrix can be determined in several different ways (see Caswell (2001), Sec. 4.5.1.2 for a good overview). First, trial and error may yield a suitable ξ such that each of the entries in P^ξ is > 0. lternatively, primitivity can be assessed graphically by looking at the directed graph describing probabilistic state changes (e.g., Figure 1). directed graph (and associated matrix) is irreducible if it is strongly connected—i.e., a path exists from each node to every other node. n irreducible graph is primitive if the greatest common divisor of the lengths of those loops is 1 (Rosenblatt, 1957).

S4 Transient oscillating decisions in stochastic dynamic programming

Consider an SDP model with fitness functions (1), the first canonical equation of SDP models in biology. Under the conditions outlined in the main text, the stationary policy π^* is that corresponding to the matrix P_π with the largest dominant eigenvalue out of all possible policies $\pi \in$. For an SDP model with a finite time horizon, this is the policy which will be optimal as $t \to \infty$ (i.e., as we get further away from the terminal time).

We explore convergence to the stationary policy, using intuition from the theory of matrix

population models (Caswell, 2001). Matrix population models generally take the form $N(t+1) = A\,N(t)$. nalogously, we consider a model of the form

$$F(t) = P\,F(t+1), \tag{S4.1}$$

for some primitive matrix P with a spectral radius < 1 and nonnegative terminal condition $F(T)$ with at least one nonzero entry. The solution to (S4.1) is

$$F(T \quad \tau) = \sum_{j=1}^{k} c_j \lambda_j^{\tau} V_j$$

where c_j is a scalar, and λ_j and V_j are eigenvalue and corresponding right eigenvector pairs of P (Caswell, 2001). Thus the structure of F is influenced initially by the subdominant eigenvalues (i.e., eigenvalues smaller in magnitude than the dominant eigenvalue) and corresponding eigenvectors of P, as $\tau \to \infty$. If λ_j is positive, then the contribution of V_j is exponentially decreasing (since $|\lambda_j| < 1$, for all j). If $1 < \lambda_j < 0$, then this term contributes damped oscillations with period 2. If λ_j and λ_{j+1} are complex conjugates, $\lambda_j = a + bi$ and $\lambda_{j+1} = a \quad bi$, we may use polar coordinates, so $\lambda_j = |\lambda_j|(\cos\theta + i\sin\theta)$ and $\lambda_{j+1} = |\lambda_j|(\cos\theta \quad i\sin\theta)$. The contribution of this pair also oscillates, with period $2\pi/\theta$ (Caswell, 2001).

The damping ratio is defined as $\psi = \lambda_1/|\lambda_2|$ (Caswell, 2001). If ψ is close to 1, a significant influence from λ_2 and V_2 will persist for a long time before the dynamics are asymptotically governed only by λ_1 and V_1. For increasing values of ψ, the influence of λ_2 (and all subsequent eigenvalues) decays increasingly rapidly.

Returning to the SDP model (1), these concepts explain the convergence behaviour of $F(t)$ as $t \to \infty$. For example, if $\lambda_{\pi,2}$ is either negative or complex valued, we expect to see oscilla-

tions in the structure of $F(t)$ near the terminal time. If the damping ratio $\psi = \lambda_{\pi\ ,1}/|\lambda_{\pi\ ,2}|$ is close to 1, we expect these oscillations to be apparent for longer than if the damping ratio is very large. However, unlike in (S4.1), the matrix P is not fixed in time. If the oscillations in the structure of $F(t)$ are sufficiently large—or, analogously, if there exists another policy π' and matrix $P_{\pi'}$ with right eigenvector $V_{\pi',1}$ sufficiently similar to $V_{\pi\ ,1}$—the alternative policy π' may be optimal periodically, resulting in oscillating decision rules. These oscillations will continue until the influence of $\lambda_{\pi\ ,2}$ is sufficiently small compared to $\lambda_{\pi\ ,1}$ and the structure of $F(t)$ is very close to $V_{\pi\ ,1}$. These oscillations can be thought of as an artifact of not having any cost of switching strategies. We suspect that introducing a small cost for switching decisions, a kind of behavioral inertia (see, e.g., (Dukas and Clark, 1995; Boettiger et al., 2016)), would remove these oscillations. However, for models that do not include this cost, it may be reassuring to know that oscillating optimal policies may arise from the model structure, rather than being the result of a numerical error. We show below how these oscillations can arise even in a very simple model.

Revisiting the illustrative patch choice example

For the optimal policy $\pi^* = \{\text{patch } 2, \text{patch } 2, \text{patch } 1, \text{patch } 1, \text{patch } 1\}$, the matrix P_π has dominant eigenvalue $\lambda_1 = 0.97$, and subdominant eigenvalues $\lambda_2 = \quad 0.40 + 0.62i$, $\lambda_3 = \quad 0.40 \quad 0.62i$, so the damping ratio is $\psi = 0.97/|\quad 0.4 + 0.62i| = 1.32$. Thus we expect to see oscillations in $F(t)$ near the terminal time, but predict they should die out fairly quickly (Figure 2).

We now change one parameter, decreasing the probability of finding food in both patches, to $\eta_1 = 0.3$ and $\eta_2 = 0.6$. ll other parameters remain the same. Following the same steps as before, we find the same optimal policy, $\pi^* = \{\text{patch 2, patch 2, patch 1, patch 1, patch 1}\}$. However, now the dominant eigenvalue of P_π is $\lambda_1 = 0.94$, the subdominant eigenvalues are $\lambda_2 = \quad 0.42 + 0.66i$ and $\lambda_3 = \quad 0.42 \quad 0.66i$, resulting in $\psi = 1.21$. We again predict oscillations in $F(t)$, but these

oscillations will have a larger effect on the dynamics and will be evident further from the terminal time than for the previous parameter set. These oscillations are now sufficiently large that the policy $\pi' = \{\text{patch } 2, \text{patch } 2, \text{patch } 2, \text{patch } 1, \text{patch } 1\}$ is optimal periodically (Figure S1).

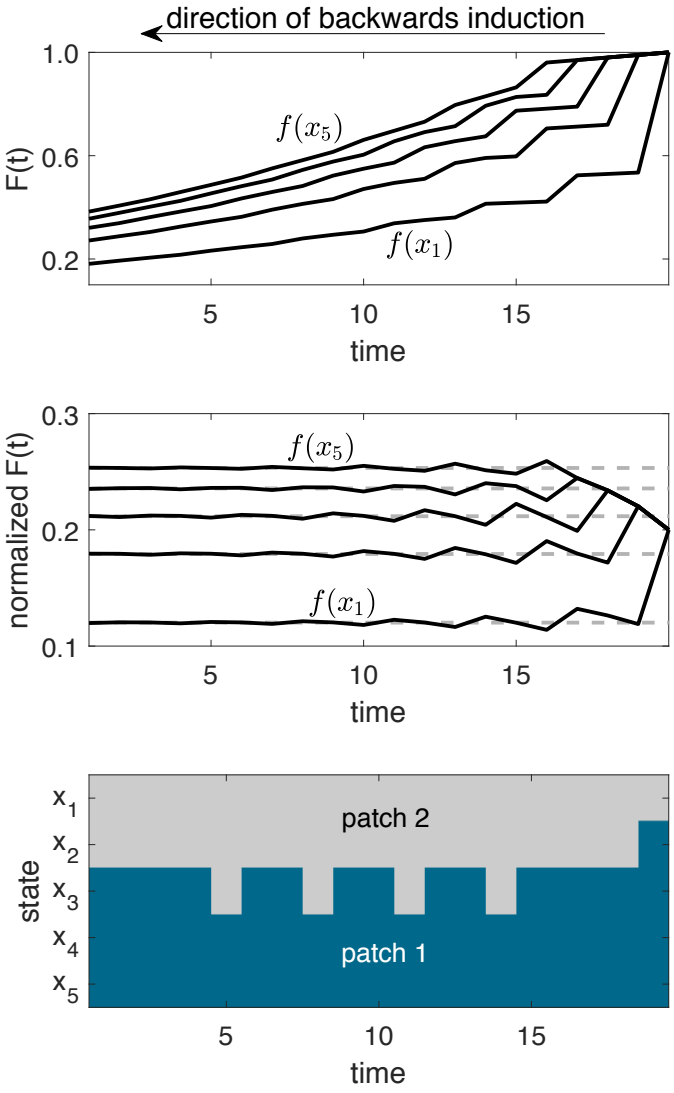

Figure S1: Solution (obtained using backwards induction; arrow at top) of the illustrative patch choice example, as described in Figure 2, but with a reduced probability of finding prey. For this parameter set, observe the oscillating decisions predicted in the bottom panel.

S5 Going from the biological parasitoid wasp problem to the corresponding matrix model

We here describe, in more detail, how to go from a biological understanding of the parasitoid wasp example to the matrix formulation of the model. We begin by constructing the directed graphs describing the state changes possible over one time step. There are four possible state changes that an individual in state x could experience from time t to $t+1$, which we address below:

(i) if the individual dies: $x \to 0$.

(ii) if no host is encountered: $x \to x \quad 1$

(iii) if a host is encountered and parasitized: $x \to x \quad 1 \quad \beta$

(iv) if a host is encountered and the individual host feeds: $x \to x \quad 1+$

(i) Individual dies: $x \to 0$

The individual has probability $(1 \quad e \quad)$ of dying over each time step, at which point we assume $x \to 0$. In a directed graph, this would be represented by an arrow from each state to 0 (Figure S2). Note that throughout this section, we do not label every arrow to maintain readability of the directed graphs, but each arrow of a similar type has the appropriate similar label.

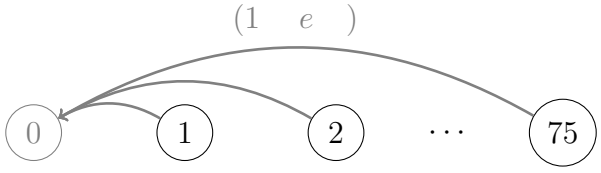

Figure S2

For the creation of matrix P, we ignore all processes associated with this absorbing state and use grey for them in all of our directed graphs to emphasize this point. However, when we wish to use the method of Markov chains later on, these processes are included in the Markov matrix.

(ii) No host encountered: $x \to x \quad 1$

Regardless of the individual's state, a host is not encountered with probability $(1 \quad \eta)$, conditional on the individual's survival (probability $e \quad$). We represent these probabilities with arrows going from each state x to state $x \quad 1$ in a directed graph (Figure S3).

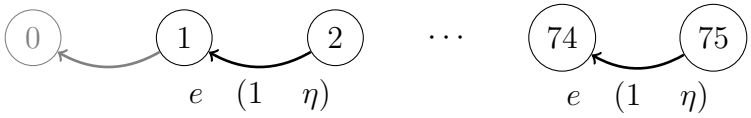

Figure S3

We now construct the corresponding transition matrix P. Begin with a square matrix of zeros, with dimensions 75×75. The row number corresponds to where the arrows are leaving "from" and the column number is where the arrows are going "to" in the directed graph. The transition probability assigned to each arrow in the directed graph going from state x to $x \quad 1$ now gets placed in location $p(x, x \quad 1)$ in matrix P (i.e., the entry in row x and column $x \quad 1$). Figure S3 thus corresponds to

$$P = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ e \quad (1 \quad \eta) & 0 & \cdots & \cdots & 0 \\ 0 & e \quad (1 \quad \eta) & & & \\ \vdots & & \ddots & & \\ 0 & \cdots & & e \quad (1 \quad \eta) & 0 \end{bmatrix}. \tag{S5.1}$$

(iii) Host encountered and parasitized: $x \to x \quad 1 \quad \beta$

If a host is encountered (probability η), the individual must make a decision whether to parasitize or host feed. Recall that decision $i = 1$ denotes parasitizing and $i = 2$ denotes host feeding. If the individual chooses to parasitize regardless of state (which we here denote policy $\pi_{(1)} = \{1, 1, \ldots, 1\}$), then state changes from $x \to x \quad 1 \quad \beta$. We here use the value $\beta = 4$, so $x \to x \quad 5$. Building from Figure S3, we add these arrows (in orange) to complete the directed graph for policy $\pi_{(\pi)}$ (Figure S4).

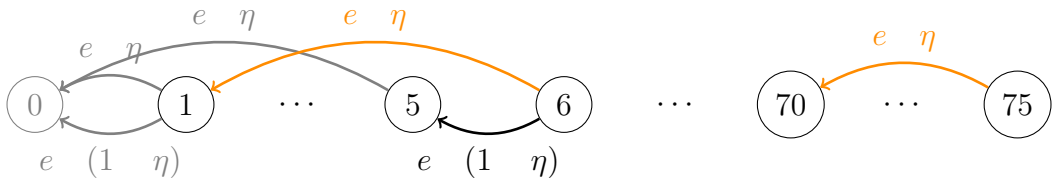

Figure S4

We now add the entries corresponding to the orange arrows to (S5.1), setting $p(x, x \quad 5) = \eta$ for all $x \geq 6$, resulting in

$$P_{\pi_{1)}} = \begin{bmatrix} 0 & 0 & \cdots & & & 0 \\ e & (1 & \eta) & 0 & \cdots & & & 0 \\ 0 & & e & (1 & \eta) & & & 0 \\ \vdots & & & & & \ddots & & \\ e & \eta & & 0 & & & & \\ 0 & & & e & \eta & & & \\ \vdots & & & & & \ddots & & \\ 0 & & & & & e & \eta & \cdots & e & (1 & \eta) & 0 \end{bmatrix}. \tag{S5.2}$$

(iv) Host encountered and used for host feeding: $x \to x \quad 1+$

If a host is encountered (probability η), and the individual always chooses to host feed, regardless of state (i.e., policy $\pi_{(2)} = \{2, 2, \ldots, 2\}$), the state changes from $x \to x \quad 1+ \quad$, again, conditional on survival (probability $e \quad$). Since $\quad = 30$, this means $x \to x + 29$ with probability η. $\quad$ gain building from Figure S3, we add these arrows (in green) to complete the directed graph for policy $\pi_{(2)}$ (Figure S5).

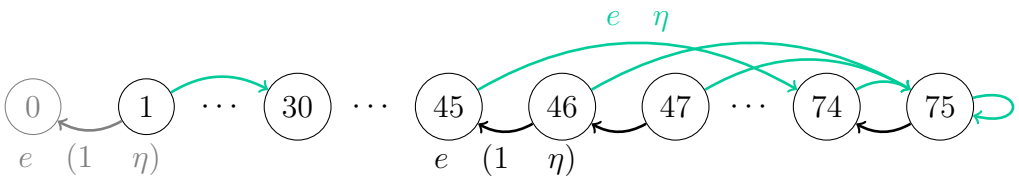

Figure S5

The arrows representing state changes caused by host feeding (green arrows) result in matrix entries $p(x, x + 29) = e \quad \eta$ for all $x \leq 46$. Note, however, what happens if an individual in state $x = 47$ host feeds; their state cannot increase to $x + 29 = 76$, as it then exceeds the maximum possible state of $x = 75$. We have assumed that an individual can increase their state to a maximum

of 75, so for $x \geq 47$, $x \to 75$, which corresponds to matrix entries $p(x, 75)$. This results in matrix

$$P_{\pi_{2)}} = \begin{bmatrix} 0 & 0 & \cdots & e & \eta & 0 & \cdots & & & 0 \\ e & (1 & \eta) & 0 & \cdots & 0 & e & \eta & \cdots & & & 0 \\ & 0 & e & (1 & \eta) & & & & \ddots & & & \\ & & & & & \ddots & & & & e & \eta & 0 \\ & & & & & & & & & 0 & e & \eta \\ & & & & & & & & & \vdots & e & \eta \\ & & & & & & & & \ddots & & & \vdots \\ & 0 & & \cdots & & & & 0 & e & (1 & \eta) & e & \eta \end{bmatrix}. \tag{S5.3}$$

reate the matrix for any policy π

We have created the directed graphs and matrices for policies consisting entirely of either parasitizing or host feeding ($\pi_{(1)} = \{1, \ldots, 1\}$ and $\pi_{(2)} = \{2, \ldots, 2\}$, respectively), regardless of the individual's state. From these two extremes, we can construct the matrix for any policy $\pi = \{i_1, \ldots, i_{75}\}$. Observe that a given row—say, row x_j—corresponds to all of the possible arrows leaving *from* state x_j in the associated directed graph. Thus, the decision i_{x_j} made by an individual in state x_j affects all of the entries in that row.

For a given policy $\pi = \{i_1, \ldots, i_{75}\}$, the corresponding matrix may be constructed from the appropriate rows from the corresponding matrices defined above. For example, consider $\pi' = \{2, 1, \ldots, 1\}$, where the individual parasitizes unless she is in the lowest state. The first row of the associated transition matrix will be the first row from $P_{\pi_{2)}}$ in (S5.3), while the rest of the rows will

come from $P_{\pi_{1)}}$ in (S5.2), i.e.,

$$P_{\pi'} = \begin{bmatrix} 0 & 0 & \cdots & e & \eta & \cdots & & 0 \\ e & (1 & \eta) & 0 & \cdots & & & 0 \\ 0 & e & (1 & \eta) & & & & 0 \\ \vdots & & & & \ddots & & & \\ e & \eta & 0 & & & & & \\ 0 & e & \eta & & & & & \\ \vdots & & & & \ddots & & & \\ 0 & & & & e & \eta & \cdots & e & (1 & \eta) & 0 \end{bmatrix}. \tag{S5.4}$$

Similarly, if considering policy $\pi'' = \{1, \ldots, 1, 2\}$, all rows would be as in $P_{\pi_{1)}}$ in (S5.2) except for the last row, which would be as in $P_{\pi_{2)}}$ in (S5.3). Each matrix P_π may be constructed in this way, once the structure of each row has been defined for each policy as we have done above.

Rewards vector G

Recall that if an individual chooses to parasitize a host, her fitness is immediately incremented by 1. If she chooses to host feed instead, their fitness does not see this immediate reward. The rewards vector G captures this, with $G_\pi = [g_{\pi,1}, \ldots, g_{\pi,75}]^\top$. For example, for the policy $\pi' = \{2, 1, \ldots, 1\}$, $G_{\pi'} = [0, 1, \ldots, 1]$, since $g_{\pi',x} = 1$ for all states x except state x_1, for which $g_{\pi',1} = 0$.

For each policy π, we have now constructed the corresponding P_π and G_π and can thus calculate $F_\pi = (I \quad P_\pi)^{\ 1}G_\pi$, where I is the identity matrix with dimensions 75×75. For an example of how to implement this using Matlab, see the code provided at doi:10.5281/zenodo.2547815.

For working in R, see Section S1 for a quick list of the necessary R commands.

Using P for Markov chains

One of the benefits of using matrix notation to formulate an SDP model is the ease with which one may then use Markov chains to predict the probability distribution of an optimally behaving individual's state. Only one further step remains: to convert matrix P from a substochastic matrix to the full Markov (stochastic) matrix $\hat{P}$. For the parasitic wasp example, this means including all transitions to absorbing state $x = 0$, so that all rows sum to 1 (i.e., the grey arrows in all of our directed graphs above). When these transitions are included in the associated matrix $\hat{P}$, it becomes a 76×76 matrix, with an additional column added on the left and an additional row added on the top (row and column "zero"), capturing all transitions to and from state $x = 0$. Once all of these possible transitions have been included, each row will sum to 1.

For example, consider again the policy of always parasitizing, i.e., $\pi_{(1)} = \{1, \ldots, 1\}$. When we consider all possible state changes, including those to the absorbing state, the substochastic matrix of (S5.2) becomes the stochastic matrix,

$$\hat{P}_{\pi_{1)}} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ (1\ e\ ) + \eta e & (1\ \eta)e & 0 & \cdots & 0 \\ (1\ e\ ) + \eta e & 0 & (1\ \eta)e & & 0 \\ \vdots & & \ddots & & \\ (1\ e\ ) & e\ \eta & 0 & & \\ (1\ e\ ) & 0 & \eta & & \\ \vdots & & \ddots & & \\ (1\ e\ ) & 0 & & \eta\ \cdots\ 1\ \eta & 0 \end{bmatrix}, \qquad \text{(S5.5)}$$

where all entries in grey are associated with the absorbing state. For a given policy π and corresponding Markov matrix $\hat{P}_\pi$, the Markov chain is described as in (9).

References

Boettiger, C., Bode, M., Sanchirico, J. N., Lariviere, J., Hastings, ., and rmsworth, P. R. (2016). Optimal management of a stochastically varying population when policy adjustment is costly. *Ecol. ppl.*, 26(3):808–817.

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

Dukas, R. and Clark, C. W. (1995). Searching for cryptic prey: a dynamic model. *Ecology*, 76(4):1320–1326.

Grey, D. R. (1984). Non-negative matrices, dynamic programming and a harvesting problem. *J. ppl. Probab.*, 21(4):685–694.

McNamara, J. M. (1990). The policy which maximises long-term survival of an animal faced with the risks of starvation and predation. *dv. ppl. 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.

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

Rosenblatt, D. (1957). On the graphs and asymptotic forms of finite boolean relation matrices and stochastic matrices. *Nav. Res. Logist. Q.*, 4:151–167.

Sladky, K. (1976). On dynamic programming recursions for multiplicative Markov decision chains. In *Math. Program. Study*, volume 6, pages 216–226. North-Holland Publishing Company.