Foraging in Stochastic Environments

by

Nikhil Krishnan

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Applied Mathematics

2019

This thesis entitled:
Foraging in Stochastic Environments
written by Nikhil Krishnan
has been approved for the Department of Applied Mathematics

Prof. Zachary Kilpatrick

Prof. Nancy Rodriguez

Prof. Juan Restrepo

Date ________________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Krishnan, Nikhil (M.S., Applied Mathematics)

Foraging in Stochastic Environments

Thesis directed by Prof. Zachary Kilpatrick

For many organisms, foraging for food and resources is integral to survival. Mathematical models of foraging can provide insight into the benefits and drawbacks of different foraging strategies. We begin by considering the movement of a memoryless starving forager on a one-dimensional periodic lattice, where each location contains one unit of food. As the forager lands on sites with food, it consumes the food leaving the sites empty. If the forager lands consecutively on a certain number of empty sites, then it starves. The forager has two modes of movement: it can either diffuse by moving with equal probability to adjacent lattice sites, or it can jump uniformly randomly amongst the lattice sites. The lifetime of the forager can be approximated in either paradigm by the sum of the cover time plus the number of empty sites it can visit before starving. The lifetime of the forager varies nonmontonically according to the probability of jumping. The tradeoff between jumps and diffusion is explored using simpler systems as well as numerical simulation, and we demonstrate that the best strategy is one that incorporates both jumps and diffusion. When long range jumps are time-penalized, counterintuitively, this shifts the optimal strategy to pure jumping. We next consider optimal strategies for a group of foragers to search for a target (such as food in an environment where it is sparsely located). There is a single target in one of several patches, with a greater penalty if the foragers decide to switch their positions among the patches. Both in the case of a single searcher, and in the case of a group of searchers, efficient deterministic strategies can be found to locate the target.

Contents

Chapter

1 Introduction **1**

2 A Jump-Diffusion Model of Optimal Foraging **6**

2.1 Introduction 6

2.2 The optimal jump rate 6

2.3 Cover times at extremes 9

2.3.1 Diffusion 9

2.3.2 Jumping 10

2.4 Forager lifetime at extremes 11

2.4.1 Diffusion 13

2.4.2 Jumping 16

2.5 Tractable Models and Approximations of Jump-Diffusion 17

2.5.1 Small System 18

2.5.2 A Jump-Wait Model 19

2.6 Jump-Penalty Models 22

2.6.1 Constant Penalty 22

2.6.2 Distance-Based Penalty 25

2.7 Conclusions 28

3 Collective Search **30**

3.1 Introduction 30

3.2 Partially Observable Markov Decision Processes 30

3.3 Single Searcher 33

3.4 Multiple Searchers 35

3.5 Conclusions 40

4 Conclusions **41**

Bibliography **42**

Appendix

A Calculating Small System Transitions **45**

B Transition matrix for small system **46**

Figures

Figure

1.1 Jump-diffusion foraging model parametrized by p_j the probability of jumping. (a) The forager moves to non-adjacent sites with probability $\frac{p_j}{n}$ and to adjacent sites with probability $\frac{p_j}{n}\frac{1-p_j}{2}$, accounting for the possibility of diffusion. (b)-(d) Example forager paths (blue lines/dots) for $p_j = 0, 1, 0.03$. Green represents sites with food, while white represents empty sites. 3

2.1 (a) Mean survival time T of a starving forager obeying jump-diffusion with jump rate p_j. Forager lifetime T varied nonmonotonically with p_j for $s = 320, 160, 80, 40, 20$ (top-to-bottom). The maximal lifetime is marked in black. Environment size $n = 40$. Means at each value of p_j are generated using 10^6 Monte Carlo simulations. (b) The jump rate p_j that maximizes the forager lifetime primarily decreases as a function of s. Shown for $n = 400, 80, 20$ (top-to-bottom). Maxima are found using golden-section search [21] using 10^6 simulations per point. 8

2.2 (a),(b) Forager lifetime computed from Eq. (2.9) in the case of pure diffusion (a) and pure jumping (b). Cover time approximations (dashed lines) computed from Eq. (2.1) agree in the limit of large s: $n = 15, 10, 5$ for black, dark grey, and grey. Red dots are means computed using 10^6 numerical simulations. (c),(d) Ratio of forager lifetime for pure diffusion to forager lifetime for pure jumping: $T_{\text{diff}}/T_{\text{jump}}$. The contour on (d) marks where the ratio is one, marking the boundary between where jumping vs. diffusion is the better strategy. 12

2.3 Jump-diffusion model of foraging for small environment size and survival time ($n = 4$, $s = 2$). (a) Enumeration of food/forager geometries for $n = 4$ and $s = 2$ system. The forager is always in the left (bold) site after a rotation of the system. The arrows denote admissible transitions between geometries. Note that since $s = 2$, every geometry can return to itself once. (b) Expected forager lifetime T computed explicitly as a function of p_j, Eq. (2.17), has an interior maximum (black dot). . . . 18

2.4 Jump-wait model of foraging. (a) Forager lifetime as a function of p_j. The maximal lifetime is marked in black: $n = 40$ and $s = 320, 160, 80, 40, 20$ top-to-bottom as in Fig. 2.1(a). Means computed using 10^6 numerical simulations per point are given by grey dots. (b) The value of p_j that maximizes the forager lifetime as a function of s: $n = 400, 80, 20$ top-to-bottom as in Fig. 2.1(b). Maxima computed using golden-section search [21] using 10^6 simulations per point are given by grey dots. . . 20

2.5 Jump-diffusion model of foraging with constant penalty c where the forager can [(a),(c),(e)] and cannot [(b),(d),(f)] starve during a jump. (a),(b) The expected lifetime of a forager as a function of p_j with $s = 40$. Means are computed with 10^6 numerical simulations per point. (c),(d) The optimal value of p_j as a function of s. Maxima are computed using golden-section search [21] with 10^7 simulations per point. (e),(f) The expected lifetime of a forager moving with the optimal p_j. Means are computed using 10^6 numerical simulations per point. $n = 40$ and blue, red, yellow, and purple correspond to $c(s) = 1, 2, s/2,$ and s. 23

2.6 Jump-diffusion model of foraging with distance based penalty $c \cdot d$ where the forager can [(a),(c),(e)] and cannot [(b),(d),(f)] starve during a jump. (a),(b) The expected lifetime of a forager as a function of p_j with $s = 40$. Means are computed with 10^6 numerical simulations per point. (c),(d) The optimal value of p_j as a function of s. Maxima are computed using golden-section search [21] with 10^7 simulations per point. (e),(f) The expected lifetime of a forager moving with the optimal p_j. Means are computed using 10^6 numerical simulations per point. $n = 40$ and blue, red, yellow, and purple correspond to $c(s) = 1, 2, s/2,$ and s. 26

3.1 (a),(c) Time to find the target. Solid lines given by optimizing Eq. (3.4), dots given by applying POMCP [26] to the POMDP formulation. (b),(d) Final patch dwell time n_{z-1}, given from the arguments of optimizing Eq. (3.4). One-searcher, two-patch case with $\tau = 5, 10, 15$ corresponds to red, blue, and green. One-searcher with $\tau = 10$ and two, three, and four patches corresponds to blue, magenta, and cyan. . . 36

3.2 Time to find the target for different configurations of patches and searchers and $\tau = 10$. Blue denotes the thresholding strategy found by optimizing Eq. (3.6), while the dots denote the results of applying POMCP [26] to the POMDP formulation. (a) Two-searcher two-patch, red denotes Eq. (3.8). (b) Two-searcher three-patch, red denotes optimizing Eq. (3.9). (c) Two-searcher four-patch, red denotes optimizing Eq. (3.4). (d) Three-searcher two-patch, red denotes Eq. (3.10) 39

Chapter 1

Introduction

Virtually all motile organisms must forage for resources such as food, habitats, or mates. Optimal foraging theory typically examines what strategies best balance search cost with reward [2]. An integral component of foraging is the balance between exploiting the known and/or nearby resources versus exploring one's broader environment for new resources [13]. Organisms typically deplete resources in their immediate vicinity over time [27], unless depletion is slow and resources are renewable [10]. Thus, organisms often invoke strategies in which they compare the known yield at their current location with distribution of yields from distant sites [28].

The predictions of theoretical models of foraging strongly depend on the information available to the forager. If foragers have partial knowledge of the statistical distribution of resources, optimal foraging strategies are usually straightforward to identify and typically balance an explore/exploit tradeoff [2, 9]. In contrast, foragers may possess no knowledge of their environment and may be incapable or unwilling to learn based on their foraging history [5]. Recent models along these lines study the dynamics of foragers moving in environments organized on a lattice, according to a random walk. Previous work has examined the effect of making the forager more or less likely to pursue food [6], making the forager wait before consuming food [3], and giving the forager a chance not to consume encountered food [22]. In particular, this recent work has studied the added constraint of starvation, whereby the forager cannot go longer than s steps without food. Exploration/exploitation tradeoffs are then determined by how search strategy parameters shape the lifetime of the forager, corresponding to the number of steps until it starves.

In Chapter 2, we explore a model of a starving forager executing a random walk similar to that developed in [5, 6]. We consider the movement of a forager on a one-dimensional periodic lattice with n sites, where each location contains one unit of food. If the forager lands on a site with food, the forager consumes the food, leaving the site empty. After the forager lands on s consecutive empty sites, it starves. Since the food is depleted and never regenerated, the forager will eventually starve, and can survive at most $s \cdot n$ steps, though the mean lifetime T is typically much less than this upper limit.

Recent analyses have focused on cases in which foragers only move locally, according to biased or unbiased random walks [3,5,6]. In contrast here, we explore the effects of allowing the forager to make large jumps. Food is typically distributed heterogeneously in an environment, and animals can adapt their foraging strategy as such [2]. For example, penguins alternate between foraging locally on patches of krill and moving ballistically between them [30]. One foraging strategy for this situation is a Lévy-type movement, where animals combine small-scale movements with long-distance displacements [1, 17, 19, 23]. Our model will emulate this type of movement as follows.

Our forager has two modes of movement [Fig. 1.1(a)]: it can either diffuse, by moving with equal probability to adjacent points on the lattice [Fig. 1.1(b)], or it can jump to a uniformly randomly chosen site on the lattice [Fig. 1.1(c)]. In particular, we examine a hybridized approach, where the forager jumps with probability p_j, or diffuses with probability $1 - p_j$ [Fig. 1.1(d)]. Providing our forager with both types of movement allows us to consider how much time the forager should spend exploiting a given location, and how frequently the forager should move to other locations. We demonstrate that the mean lifetime T of the forager varies non-monotonically with respect to p_j, and the forager's lifetime is maximized through a mixture of jumping and diffusion.

This work extends the recent studies of [3, 5, 6] by incorporating long-range motion into the dynamics of a starving forager. Those previous studies were primarily concerned with how additional considerations like greed or frugality could affect the lifetime of a starving forager that moves via local diffusion. Note that in the limit of $p_j \to 0$ our model reduces to the basic form of

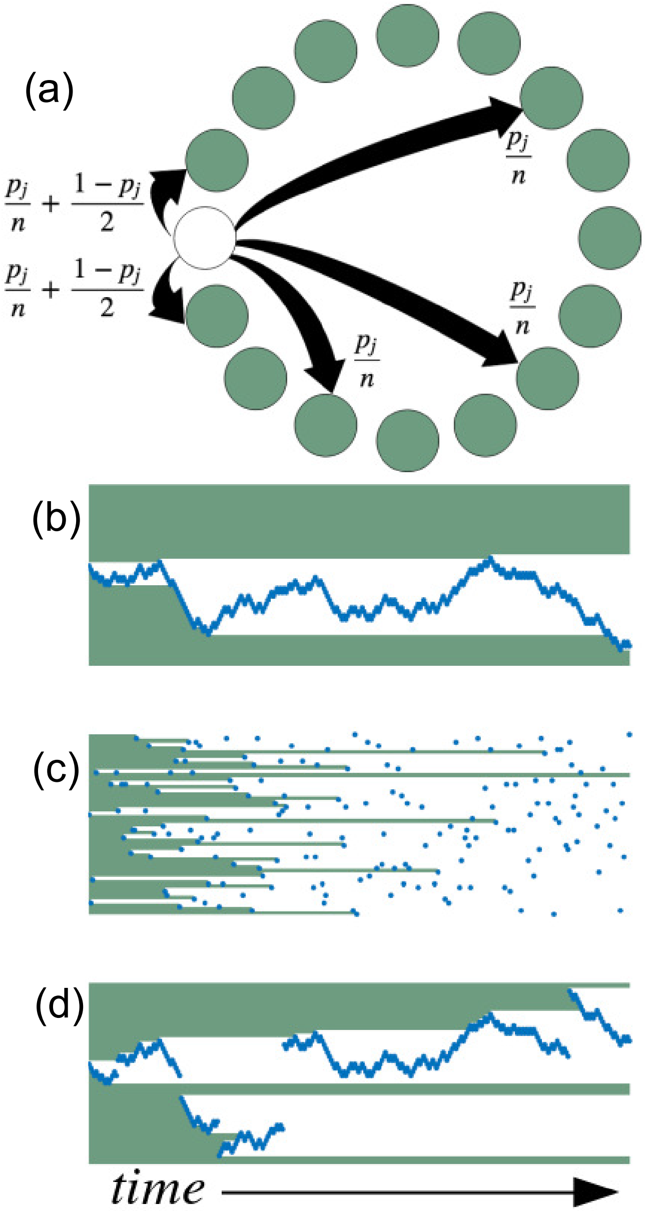
(a)
$\frac{p_j}{n} + \frac{1-p_j}{2}$
$\frac{p_j}{n} + \frac{1-p_j}{2}$
$\frac{p_j}{n}$
$\frac{p_j}{n}$
$\frac{p_j}{n}$
(b)
(c)
(d)
time

Figure 1.1: Jump-diffusion foraging model parametrized by p_j the probability of jumping. (a) The forager moves to non-adjacent sites with probability $\frac{p_j}{n}$ and to adjacent sites with probability $\frac{p_j}{n}\frac{1-p_j}{2}$, accounting for the possibility of diffusion. (b)-(d) Example forager paths (blue lines/dots) for $p_j = 0, 1, 0.03$. Green represents sites with food, while white represents empty sites.

those previous models.

Obtaining an explicit formula for the forager lifetime proves difficult, perhaps even intractable. Thus, we employ a number of alternative methods for gaining insight into how the mean forager lifetime T depends on model parameters. First, we study separately the two boundary cases of pure diffusion and pure jumping. In both cases, we can determine an upper bound for the forager lifetime as the sum of the cover time and survival time, and explicitly derive formulas for the forager lifetime. This reveals that a diffusive strategy is more advantageous when the survival time s is longer, whereas a jumping strategy is better for short survival times. Next, we analyze the jump-diffusion model in a very small environment (with $n = 4$ food sites) and short survival time ($s = 2$), showing mean lifetime is optimized by using a mix of jumping and diffusion. Finally, we analyze a jump-wait model, where we replace the diffusive behavior with waiting behavior where the forager remains in the same location until jumping. The qualitative performance of this model is similar to the jump-diffusion model, suggesting that foragers extend their lifetime by simply not consuming food when they have recently fed.

We conclude by considering several extensions of our model in which long-range jumps require more time than diffusion. In this extended model, the non-monotonicity of the forager lifetime in p_j mostly disappears. Typically, the forager does best when enacting a strategy of pure jumping in these cases. Although, when time penalties scale linearly with distance, pure diffusion becomes optimal, as most jumps result in death. In either case, a forager's lifetime can be lengthened by allowing a mixed jump-diffusion strategy whereby the forager only makes jumps that do not kill them and diffuse otherwise.

In Chapter 3, we consider the problem of foraging in a 'patchy' environment where food is difficult to come by. This type of search for a hidden target can be applied outside of foraging (for example in finding a mate or shelter) and is faced by a variety of organisms [4]. A common metric of the efficiency of a search strategy is the amount of time it takes to find the target [2], so that the best strategy is the one which minimizes the time spent searching. Search processes frequently occur in patchy environments where the target can be in specific regions, but is guaranteed not to be in the intermediate regions [15,29]. While individual organisms can, and do, search for resources, efficient search is a problem that is also important to coordinated groups of organisms [20].

Collective decision making is frequently seen in nature as a way of transmitting the information of individuals to the collective. Bees can choose the best of multiple nectar sources [24], and ants can communicate to the rest of their colony the quality of new nesting sites [20]. In both cases, information collected by an individual is passed on to the group as a whole. We can utilize these insights from ecology by applying this type of decentralized collective decision making to finding targets, allowing us to exploit the patchy nature of the domain by having searchers explore different patches and communicate their findings to one another.

Our model describes the search for a target in a ‘patchy’ domain. We consider the movement of y foragers among z patches searching for a single target. If a subset of the foragers are in the same patch as the target, then on each time step, each of these foragers independently has probability α of finding the target. If the target is found, the process terminates. In order to find the target more efficiently, the foragers can decide how to distribute themselves among the patches. However, if the foragers need to change how they are distributed, then this reorganization take τ time steps. Given these constraints, we wish to describe how the foragers should move in the patches so that the total time to find the target is minimized.

We begin by detailing how our model can be formulated as a Partially Observable Markov Decision Process (POMDP). We then develop several heuristics in the cases of a single searcher, and of multiple searchers, by restricting the set of policies to choose from. In doing so, we are able to demonstrate that in the case of a single searcher, a thresholding strategy has good performance, where the searcher rotates between the patches in order. For multiple searchers, we show that spreading the searchers out across the patches is a more efficient strategy than clustering them all together. We validate the results from these heuristics by comparing them to numerical solutions to the POMDP.

Chapter 2

A Jump-Diffusion Model of Optimal Foraging

2.1 Introduction

In this chapter, we discuss a jump-diffusion model of optimal foraging. This model permits us to examine the exploit-explore balance which arises in a wide variety of fields [2]. This chapter begins with Sec. 2.2 where we numerically explore the effect of the jump rate p_j on the forager lifetime. Specifically, a mixture of jumping and diffusion tends to be optimal. To further explore this, in Sec. 2.3, we use cover times to analyze the boundary cases of pure diffusion and pure jumping. The large n behaviour of the cover times demonstrates that jumping consumes resources more rapidly, while diffusion increases the risk of starving between feedings. We follow up by explicitly calculating the forager lifetime in these two boundary cases in Sec. 2.4. Sec. 2.5 describes several tractable approximations to the full model, giving more insight into why a hybrid of jumping and diffusion is optimal. An extension of the model to the cases where jumps take additional time is presented in Sec. 2.6. This modification generally makes jumping far more beneficial. We conclude by discussing the results of our analysis and their significance for foraging in Sec. 2.7. This chapter is adapted from [14].

2.2 The optimal jump rate

To begin, we consider the full hybrid model, where the forager can both jump and diffuse. We will numerically determine the effect of p_j on the mean forager lifetime $T(n, s, p_j)$, while varying the environment size n and survival time s. Across a wide range of parameters, a mixture of jumping and diffusion ($0 < p_j < 1$) leads to higher values of T. For larger s relative to n, the value of p_j that maximizes T becomes smaller. This trend will be studied in detail by analyzing related models in subsequent sections. Numerical results are shown in Fig. 2.1. As shown in Fig. 2.1(a), T is non-monotonic in p_j for different values of s, so there is an interior p_j that maximizes T. As we demonstrate in subsequent sections, a larger p_j (more jumping) causes the forager to consume food more rapidly, lowering the odds of starving between feedings, but depleting the resources more rapidly. Thus, the optimal p_j balances the tradeoff of slowing the rate of food consumption (decreasing p_j) with decreasing the probability of starving early on (increasing p_j). For lower values of s, there is a broad range of p_j values over which T is relatively unchanged. This suggests that the advantage gained by slowing the rate of food consumption is roughly counteracted by the increased probability of starvation. As the survival time s is increased, the optimal value of p_j decreases, since the forager becomes less likely to die between feedings [Fig. 2.1(b)]. Utilizing diffusive motion (lower p_j) more often limits that rate at which food is consumed. On the other hand, as the size of the environment is increased (larger n), the optimal p_j increases. This is because there is more food initially available, so the forager can afford to increase the rate of food consumption to decrease their probability of starving.

Our interpretations of the mean lifetime T dependence on p_j, s, and n can be analyzed in further detail by considering a few different limiting cases and approximations of the jump-diffusion model. We begin by studying the behavior of the model at the two extremes of pure diffusion [$p_j = 0$: Fig. 1.1(b)] and pure jumping [$p_j = 1$: Fig. 1.1(c)]. Our two main findings in this analysis are that (a) a diffusive forager covers the environment more slowly, decreasing the rate of food consumption as discussed above; and (b) jumping is a better strategy in large environments (large n) with lower survival times (small s). Indeed this is consistent with our numerical results above. We conclude with an analysis of two simpler models that demonstrate the same nonmonotonicity of T in p_j as shown in Fig. 2.1.

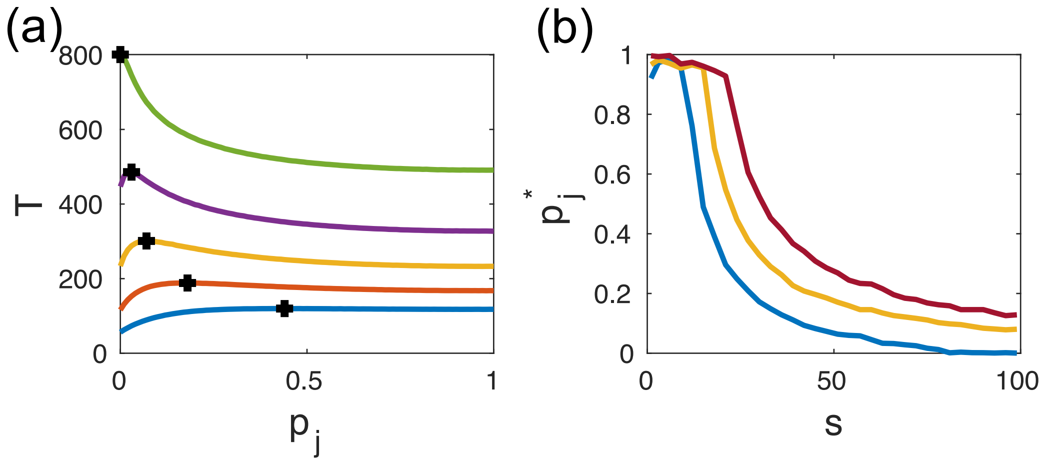

Figure 2.1: (a) Mean survival time T of a starving forager obeying jump-diffusion with jump rate p_j. Forager lifetime T varied nonmonotonically with p_j for $s = 320, 160, 80, 40, 20$ (top-to-bottom). The maximal lifetime is marked in black. Environment size $n = 40$. Means at each value of p_j are generated using 10^6 Monte Carlo simulations. (b) The jump rate p_j that maximizes the forager lifetime primarily decreases as a function of s. Shown for $n = 400, 80, 20$ (top-to-bottom). Maxima are found using golden-section search [21] using 10^6 simulations per point.

2.3 Cover times at extremes

Considering the boundary cases of pure diffusion $p_j = 0$ and pure jumping $p_j = 1$ allows us to derive explicit formulas for how model parameters, such as the environment size n and starvation time s impact the mean lifetime T of the forager. This can be approximated first by calculating the mean cover time $\mathbb{E}(\tau_{cover})$ of the forager: the time it takes the forager to reach all of the food sites in the environment. This quantity plus the starvation time s constitutes an upper bound on the lifetime in general, but for large s it provides a reasonable approximation of

$$T(n, s, p_j) \approx s + \mathbb{E}(\tau_{cover}(n, p_j)). \tag{2.1}$$

This is because, when s is large, the forager generally consumes almost all of the food in the domain before dying, since it will typically have enough time between feeding to locate remaining food in the environment.

The mean cover time $\mathbb{E}(\tau_{cover})$ can be computed explicitly. If t_k denotes the time the k^{th} piece of food is eaten, then $\tau_{cover} = t_n$, $t_1 = 0$, and by the linearity of expectation, we have:

$$\mathbb{E}(\tau_{cover}) = \sum_{k=2}^{n} \mathbb{E}(t_k - t_{k-1}). \tag{2.2}$$

In the case of both pure diffusion ($p_j = 0$) and pure jumping ($p_j = 1$), $\mathbb{E}(t_k - t_{k-1})$ can be explicitly calculated.

2.3.1 Diffusion

We first consider the case where $p_j = 0$, so the forager moves only to adjacent sites. Following along the lines of [16], to calculate the cover time, we first consider the time between eating the k^{th} piece of food and the $k-1^{\text{th}}$ piece of food. The k^{th} piece of food here refers to the time-ordering of food consumption in a single foraging realization. Since the forager can only move to adjacent locations, after eating $k-1$ pieces of food, it must be on the boundary of a contiguous region of $k-1$ sites with no food – a desert [6]. If we label the current location of the forager as site 1, and the opposite end of the desert as site $k-1$, then the time to consume the k^{th} piece of food is simply the hitting time of either site 0 or site k. We let f_i be the average time to hit either state 0 or state k starting at state i, as described by the recursion relation

$$f_i = \frac{1}{2}(f_{i-1}+1) + \frac{1}{2}(f_{i+1}+1) \tag{2.3}$$

with $f_0 = f_k = 0$. A detailed analysis of the time for a biased random walk to escape a finite interval is given in [7], where the lifetime of a starving greedy forager is studied. In that work, a parameter p determines the probability of moving towards a site containing food. Their results (given in Appendix B of [7]) reduce to ours for the unbiased random walk when $p = 1/2$. We can solve Eq. (2.3) for $f_i = i(k-i)$, and note that [11]

$$\mathbb{E}(t_k - t_{k-1}) = f_1 = k - 1, \tag{2.4}$$

so by plugging into Eqs. (2.1) and (2.2), we find

$$T(n, s, p_j = 0) \approx s + \frac{n(n-1)}{2}. \tag{2.5}$$

Note, this approximation is linear in s and quadratic in n, the size of the environment. Fig. 2.2(a) demonstrates that as s increases, Eq. (2.5) becomes more accurate, as the forager generally consumes almost all of the food in the environment. For this to be true, s must be nearly an order of magnitude larger than n. When s is too small, the forager will typically die before it can consume all of the food, so the cover time approximation breaks down.

2.3.2 Jumping

We next study the case in which the forager always jumps to a uniformly randomly chosen site on each timestep ($p_j = 1$). The cover time is then precisely the solution to the 'coupon collecting problem' [16]. Assume the forager has eaten $k-1$ pieces of food. There are then $n-(k-1)$ pieces of food remaining, and the time it takes to eat the k^{th} piece of food is geometrically distributed:

$$(t_k - t_{k-1}) \sim \frac{(k-1)^{t-1}(n-k+1)}{n^t}.$$

Plugging this result into Eqs. (2.1) and (2.2) yields

$$T(n,s,p_j=1)\approx s+n\sum_{k=1}^{n-1}\frac{1}{k}. \tag{2.6}$$

Fig. 2.2(b) shows the exact lifetime converges to this approximation as s is increased. Eq. (2.6) is again linear in s, but now scales much more slowly in n than in the case of pure diffusion. In the limit of large n, we can estimate the scaling in n as follows:

$$n\sum_{k=1}^{n-1}\frac{1}{k}\leq n\sum_{k=1}^{n}\frac{1}{k}\leq 2n\int_{1}^{n}\frac{dx}{x}=2n\log(n).$$

In particular, if we compare the cover times of the two boundary cases, we see that $\tau_{cover}(p_j=0)=\mathcal{O}(n^2)$ while $\tau_{cover}(p_j=1)=\mathcal{O}(n\,\log(n))$. This shows that if the forager can consume almost all of the food, then for large n, it will live longer by diffusing rather than jumping. This suggests that as s increases, the optimal value of p_j goes to 0, and this is indeed the case.

2.4 Forager lifetime at extremes

We now determine the exact formula for the lifetime of the forager. While the formula we derive actually applies to all values of $p_j\in[0,1]$, we can only compute its constituent parts explicitly in the boundary cases $p_j\in\{0,1\}$. Let X_k denote the time between eating the k^{th} piece of food and the $k-1^{th}$ piece of food, where $X_1=0$, since the forager immediately consumes food at their initial position. The probability the forager, with starvation time s, consumes k pieces of food before starving thus equals

$$\mathbb{P}(k^*=k)=\mathbb{P}(X_1,...,X_k\leq s,X_{k+1}>s), \tag{2.7}$$

so $k^*\in\{1,...,n\}$ is a random variable arising from the stochastic movement and death of the forager. We can determine the distribution of k^* by first computing the cumulative distribution for each X_k:

$$F_k(s)=\mathbb{P}(X_k\leq s)=\sum_{j=1}^{s}\mathbb{P}(X_k=j),\ F_{n+1}(s)=0. \tag{2.8}$$

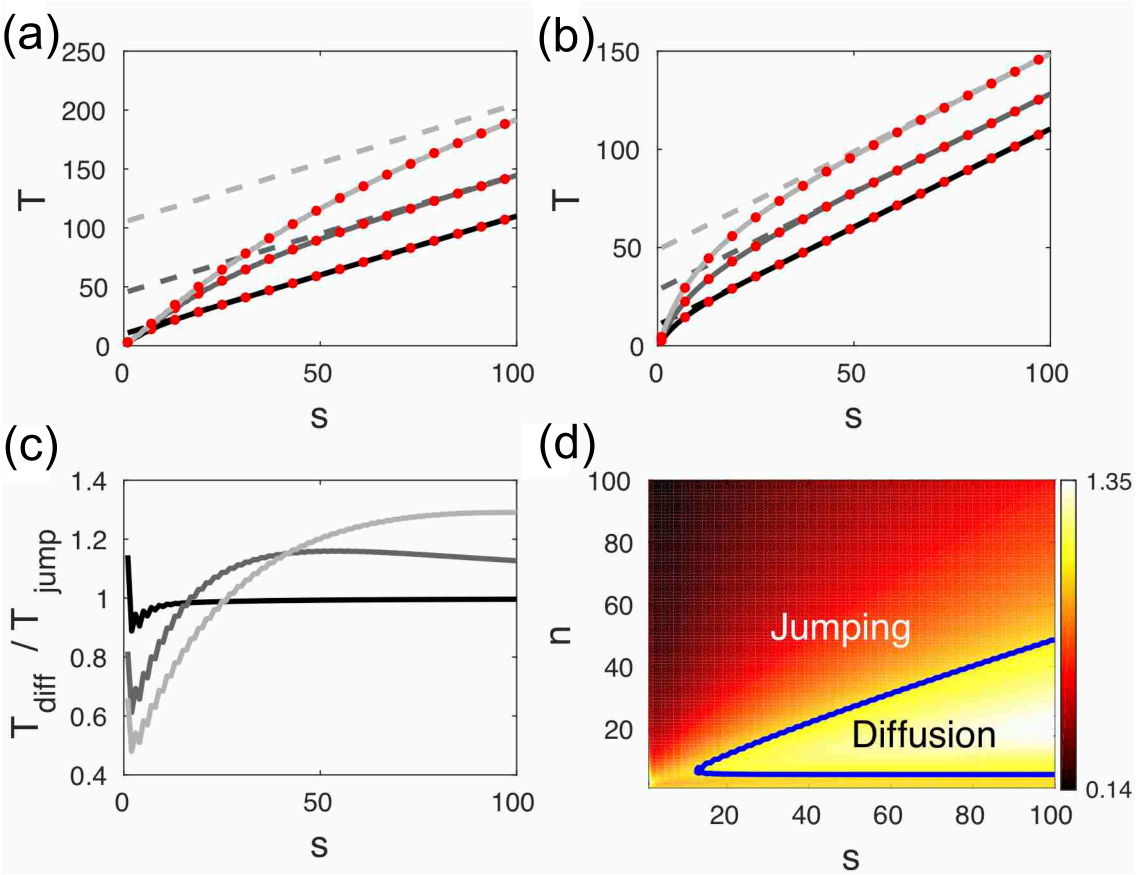

Figure 2.2: (a),(b) Forager lifetime computed from Eq. (2.9) in the case of pure diffusion (a) and pure jumping (b). Cover time approximations (dashed lines) computed from Eq. (2.1) agree in the limit of large s: $n = 15, 10, 5$ for black, dark grey, and grey. Red dots are means computed using 10^6 numerical simulations. (c),(d) Ratio of forager lifetime for pure diffusion to forager lifetime for pure jumping: $T_{\text{diff}}/T_{\text{jump}}$. The contour on (d) marks where the ratio is one, marking the boundary between where jumping vs. diffusion is the better strategy.

Note $F_1(s) \equiv 1$. Thus, $F_k(s)$ is the probability the forager survives long enough to consume the k^{th} piece of food having consumed the $k-1^{th}$ piece of food. We also wish to know the probability that the forager eats exactly k pieces of food before dying. This is given by

$$\mathbb{P}(k^* = k) = \Big(1 - F_{k+1}(s)\Big) \prod_{j=1}^{k} \Big(F_j(s)\Big).$$

The forager lifetime in each case can be computed first by conditioning on consuming exactly k pieces of food, $T_k | k^* = k$, which is simply the time it takes to eat the k pieces of food, plus s steps more until starvation. The expected lifetime T is then given by marginalizing over all possible values of k^*:

$$\begin{aligned} T &= \sum_{k=1}^{n} [T_k | k^* = k] \, \mathbb{P}(k^* = k) \\ &= s + \sum_{k=1}^{n} \Big(1 - F_{k+1}(s)\Big) \prod_{j=1}^{k} F_j(s) \sum_{i=1}^{k} \mathbb{E}(X_i | X_i \leq s). \end{aligned} \tag{2.9}$$

For the boundary cases of $p_j = 0$ and $p_j = 1$, we can derive an explicit formula for $\mathbb{P}(X_k = j)$ in Eq. (2.8), allowing us to explicitly calculate Eq. (2.9). In the limit of large s, we can approximate $F_j(s) = 1$ for all $j \leq n$ and reduce Eq. (2.9) to the sum of the expectations, $\mathbb{E}(X_i)$, which is the cover time upper bound given in Eq. (2.1).

2.4.1 Diffusion

We have a general Eq. (2.9) for $\mathbb{E}(T)$ that requires knowing $\mathbb{P}(X_k = j)$, the probability it takes j timesteps between consumption of the k^{th} and $(k-1)^{th}$ pieces of food. In what follows, we demonstrate how to explicitly compute this probability mass function in the case of a diffusive forager. In the next subsection, we study the case of a forager that purely jumps.

When $p_j = 0$, the forager moves by diffusion to carve out a food desert, a simply connected region without any food. As before, we label the sites of the desert so that site 1 is where the forager begins after consuming the $(k-1)^{th}$ piece of food, and the other desert boundary is site $k-1$. Following [8, 11], we can determine the probability mass function of X_k.

Let $u_{\ell,j}$ be the probability that it takes exactly j steps to first hit site 0 from site ℓ. We then have the following recursion relation:

$$u_{\ell,j+1} = \frac{1}{2}u_{\ell-1,j} + \frac{1}{2}u_{\ell+1,j} \tag{2.10}$$

$$u_{0,0} = 1,\ u_{j,0} = u_{0,j} = u_{k,j} = 0, \forall j > 0.$$

We then define the generating function

$$U_\ell(v) = \sum_{j=0}^{\infty} u_{\ell,j} v^j$$

and multiply Eq. (2.10) by v^{j+1}, so that by summing over j we obtain

$$U_\ell(v) = \frac{v}{2}U_{\ell-1}(v) + \frac{v}{2}U_{\ell+1}(v) \tag{2.11}$$

$$U_0(v) = 1, \qquad U_k(v) = 0.$$

The boundary conditions arise from the fact that the probability of hitting site 0 is $u_{0,0} = 1$ if starting there, but $u_{k,0} = 0$ if starting at the opposite food site. Considering solutions to Eq. (2.11) of the form $U_\ell(v) = \lambda^\ell(v)$, we obtain the characteristic equation

$$\lambda(v) = \frac{v}{2} + \frac{v}{2}\lambda^2(v).$$

This quadratic equation has two roots:

$$\lambda_\pm(v) = \frac{1 \pm \sqrt{1 - v^2}}{v}, \tag{2.12}$$

assuming $0 < v < 1$. Each root is a particular solution to Eq. (2.11). It follows that there are some functions $A(v), B(v)$ so the general solution has the form

$$U_\ell(v) = A(v)\lambda_+^\ell(v) + B(v)\lambda_-^\ell(v). \tag{2.13}$$

We can apply the boundary conditions $A(v) + B(v) = 1$ and $A(v)\lambda_+^k(v) + B(v)\lambda_-^k(v) = 0$ from Eq. (2.11) to determine $A(v)$ and $B(v)$. Finally, noting that by their definition, $\lambda_+(v)\lambda_-(v) = 1$, we have the explicit formula:

$$U_\ell(v) = \frac{\lambda_+^{k-\ell}(v) - \lambda_-^{k-\ell}(v)}{\lambda_+^k(v) - \lambda_-^k(v)}. \tag{2.14}$$

To determine $u_{\ell,j}$, we will decompose $U_\ell(v)$ with partial fractions. To start, we make the change of variables $v = \sec\phi$. Applying this to Eq. (2.12) and Eq. (2.14), we find

$$\lambda_\pm(v) = \cos\phi \pm i\sin\phi, \quad U_\ell(v) = \frac{\sin(k-\ell)\phi}{\sin k\phi}.$$

The denominator of $U_\ell(v)$ is zero for $\phi_m = \frac{m\pi}{k}$, $m = 0, ...k$, which correspond to $v_m = \sec\phi_m$. Furthermore, since the degree of the numerator exceeds the degree of the denominator by at most 1, $U_\ell(v)$ has a partial fraction decomposition with the form

$$\frac{\sin(k-\ell)\phi}{\sin k\phi} = Av + B + \frac{\rho_1}{v_1 - v} + ... + \frac{\rho_{k-1}}{v_{k-1} - v}. \tag{2.15}$$

To determine the value of ρ_m, we multiply both sides by $v_m - v$, then take $v \to v_m$ (and $\phi \to \phi_m$):

$$\rho_m = \frac{\sin\frac{\ell\pi m}{k}\sin\frac{\pi m}{k}}{k\cos^2\frac{\pi m}{k}}.$$

By decomposing each fraction of Eq. (2.15) into a geometric series, we find that the coefficient of v^j (which is $u_{\ell,j}$) is given by

$$\sum_{m=1}^{k-1}\frac{\rho_m}{v_m^{j+1}} = \frac{1}{k}\sum_{m=1}^{k-1}\cos^{j-1}\left(\frac{\pi m}{k}\right)\sin\left(\frac{\ell\pi m}{k}\right)\sin\left(\frac{\pi m}{k}\right).$$

We are interested in two possibilities, associated with the site the forager hits next, $h_k \in \{0, k\}$. Either, the forager can start at site 1 and hit site 0 ($h_k = 0$, corresponding to $u_{1,j}$), or the forager can start at site 1 and hit site k ($h_k = k$), which by symmetry is identical to the forager starting at site $k-1$ and hitting site 0 (corresponding to $u_{k-1,j}$). The probability it takes j steps from consuming the $(k-1)^{th}$ to the k^{th} food site is then the sum over the joint probabilities $\mathbb{P}(X_k = j, h_k)$. Thus, we have the following distribution for X_k:

$$\begin{aligned}\mathbb{P}(X_k = j) = &\sum_{h_k=0,k}\mathbb{P}(X_k = j, h_k) = u_{1,j} + u_{k-1,j} \\ =&\frac{1}{k}\sum_{m=1}^{k-1}\cos^{j-1}\left(\frac{\pi m}{k}\right)\sin\left(\frac{\pi m}{k}\right)\times\left((1-(-1)^m)\sin\left(\frac{\pi m}{k}\right)\right).\end{aligned} \tag{2.16}$$

We can compute the corresponding conditional expectations and cumulative distributions in the standard way, and then use Eq. (2.9) to compute the expected lifetime of the forager. For small values of s, the forager lifetime is initially super-linear in s, but that as s increases, the lifetime slowly converges to a linear function of s, as described by the cover time approximation [Fig. 2.2(a)]. Furthermore, the lifetime $T(n, s, p_j = 0)$ is generally insensitive to n for small values of s. This is because the forager will rarely ever consume all the food in its environment in these cases.

2.4.2 Jumping

For $p_j = 1$, the time X_k between consuming the $(k-1)^{th}$ and k^{th} food site is geometrically distributed with success probability $[n-(k-1)]/n$. Specifically, $\mathbb{P}(X_k = j)$ is the probability of $j-1$ visits to empty sites, each with probability $[k-1]/n$, followed by a visit to a food site, with probability $[n-(k-1)]/n$. Thus,

$$\mathbb{P}(X_k = j) = \left(\frac{k-1}{n}\right)^{j-1}\left(\frac{n-(k-1)}{n}\right).$$

We can compute the cumulative distributions and conditional expectations of a geometric random variable in the typical way, to yield the following formula for the forager lifetime from Eq. (2.9):

$$T(n, s, p_j = 1) = s + \sum_{k=1}^{n-1}\left(\frac{k+1}{n}\right)^s \prod_{j=1}^{k}\left(1-\left(\frac{j}{n}\right)^s\right) \times \sum_{i=1}^{k}\left[\frac{n}{n-i} + s + \frac{s}{(i/n)^s - 1}\right].$$

From Fig. 2.2(b) we again see that for small values of s the forager lifetime $T(n, s, p_j = 1)$ is initially super-linear in s and insensitive to n, but limits to the cover time approximation as s increases.

To compare the two strategies (pure diffusion vs. pure jumping), we compute the ratio $T_{\text{diff}}/T_{\text{jump}}$ of the forager lifetime for $p_j = 0$ to the forager lifetime for $p_j = 1$. For sufficiently large s, diffusion leads to longer lifetimes than jumping [Fig. 2.2(c)]. This is because the diffusive forager will cover the environment more slowly than the jumping forager, so they will not consume food as quickly. Note, the ratio drops as we change $s = 1$ to $s = 2$, since in the case of pure diffusion (and $s = 1$) the forager will live at least two timesteps, whereas the pure jumper may not. As soon as $s = 2$, this effect becomes negligible. Furthermore, this drop in the ratio becomes less severe for larger values of n, since the jumper will almost always live at least two timesteps. In Fig. 2.2(d), we display the ratio as a surface plot along both the s and n axis. Increasing s clearly expands the region (outlined) of n values, for which diffusion is a better strategy. Note that for very small values of n ($n \leq 5$) the cover time for diffusion is less than the cover time for jumping, leading to an advantage of jumping over diffusion at those parameter values. When s is large relative to n, the diffusive forager benefits from a larger cover time, so for these small values of n, it is consistently more beneficial to jump rather than diffuse. On the other hand, when s is small compared to n, it is better to jump since this will decrease the likelihood of starving before much of the environment's food has been consumed.

This concludes our analysis in the case of pure diffusion ($p_j = 0$) or pure jumping ($p_j = 1$). We now turn to two simpler instantiations of the jump-diffusion model of the starving forager: one that considers a very small environment ($n = 4$) and another that considers replacing diffusion with waiting. Both of these models exhibit the same nonmonotonicity of the lifetime T with respect to p_j, and admit some explicit analysis.

2.5 Tractable Models and Approximations of Jump-Diffusion

Given our insights from the extreme cases $p_j \in \{0, 1\}$, we now consider the full hybrid model, where the forager can both jump and diffuse. We have seen that when s is large is relative to n, it is more advantageous to diffuse rather than jump. To obtain explicit expressions of this result, we will consider two simplifications. First, for a sufficiently small system (small n and s), the forager lifetime can be explicitly determined either by combinatorial methods or by analyzing the probability transition matrix for the system. Secondly, we will consider a model that replaces diffusion with waiting. This altered model still yields qualitatively similar results to the jump-diffusion model, lending credence to our theory that diffusion acts as a way to prevent premature resource depletion. Both of these models demonstrate that it is most beneficial for the forager to use a mixture of jumping and diffusing (or waiting), specifically that $T(n, s, p_j)$ has an interior maximum on $p_j \in [0, 1]$. Furthermore, in the case of the jump-wait model, we will see that the larger s is relative to n, the smaller the optimal value of p_j becomes, consistent with our results for the jump-diffusion model.

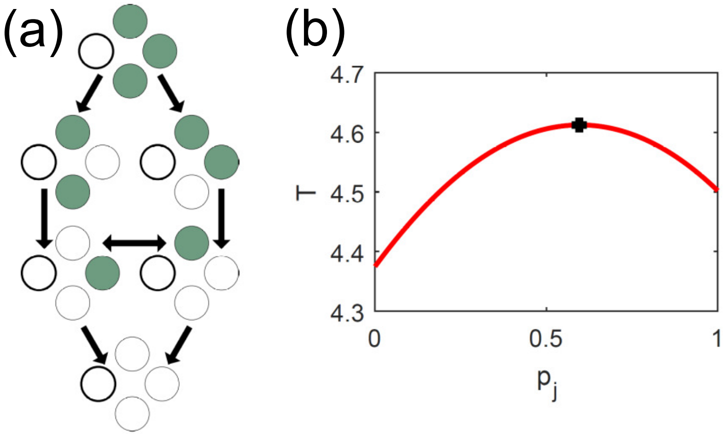

Figure 2.3: Jump-diffusion model of foraging for small environment size and survival time ($n = 4$, $s = 2$). (a) Enumeration of food/forager geometries for $n = 4$ and $s = 2$ system. The forager is always in the left (bold) site after a rotation of the system. The arrows denote admissible transitions between geometries. Note that since $s = 2$, every geometry can return to itself once. (b) Expected forager lifetime T computed explicitly as a function of p_j, Eq. (2.17), has an interior maximum (black dot).

2.5.1 Small System

For a system of small enough size, the lifetime of the forager can be analytically determined, either by enumerating all possible outcomes or by analyzing an associated discrete-time Markov chain. Here, we consider a lattice with $n = 4$ sites, and a starvation time of $s = 2$. The combination of food and forager states can be described as a thirteen state Markov chain [Fig. 2.3(a)]. State 1 corresponds to timestep 1, in which the starting site's food has been eaten. State 13 corresponds to the cemetery state, in which the forager has starved. Most of the remaining intermediate states are identical to at least one other state, when considering rotations, so there are only six elementary "live" states plus the cemetery state. Note that the forager can transition from most state geometries to death, by landing on a site without food more than $s = 2$ times in a row. The nonzero entries of the associated transition matrix Q corresponding to the probabilities to transition from state i to j are given in Appendix B.

To calculate the forager lifetime, we compute the mean absorption time into the thirteenth state (the cemetery state) as a passage time problem for Markov chains [18]. Let us denote by v be the vector of all zeroes save the first entry which is one. Let $\mathbf{1}$ be a vector of all ones. Finally, let $\tilde{Q}$ be the sub-matrix of the preceding probability transition matrix excluding the cemetery state. The expected forager lifetime is then given by

$$T(4, 2, p_j) = \mathbf{1}^T (I - \tilde{Q}^T)^{-1} v.$$

We can also determine the expected forager lifetime $T(4, 2, p_j)$ by enumerating outcomes directly (see Appendix A), yielding the following polynomial:

$$T(4, 2, p_j) = -\frac{3}{512} p_j^6 - \frac{3}{256} p_j^5 + \frac{15}{256} p_j^4 - \frac{11}{128} p_j^3 - \frac{39}{64} p_j^2 + \frac{25}{32} p_j + \frac{35}{8}, \tag{2.17}$$

which can be maximized numerically [Fig. 2.3(b)]. With either method of computation, the maximal forager lifetime is $T^{\max} \approx 4.612$ at $p_j \approx 0.598$, demonstrating it is optimal for the forager to both jump and diffuse in this simple case. Examining Fig. 2.3(a), we expect that the forager lifetime is lengthened by allowing the system to dwell in the intermediate states preceding the bottom cemetery state. This is the same intuition as in the large system: the optimal forager balances a reduction in their probability of starving before eating all the food with a reduction in the rate at which food is consumed.

2.5.2 A Jump-Wait Model

We now consider a modification of our jump-diffusion model which admits explicit analysis as a function of the jumping probability p_j. In this case, the forager remains at the same site (rather than diffusing) with probability $1 - p_j$. This mimics diffusion in food ‘deserts,’ which will generally arise in large domains when p_j is not too large. The forager can thus only consume food and reset their starvation clock by jumping. As we saw in our discussion of cover times, a diffusing forager covers the domain more slowly than a jumping forager. Waiting, just like diffusing, acts to slow the rate at which the forager consumes the food, providing qualitatively similar non-monotonic lifetimes in p_j [Fig. 2.4(a)].

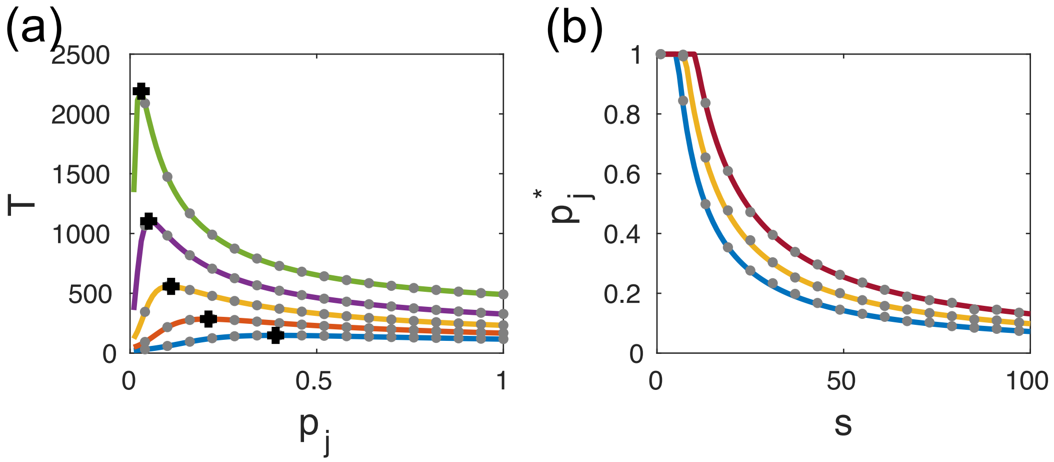

Figure 2.4: Jump-wait model of foraging. (a) Forager lifetime as a function of p_j. The maximal lifetime is marked in black: $n = 40$ and $s = 320, 160, 80, 40, 20$ top-to-bottom as in Fig. 2.1(a). Means computed using 10^6 numerical simulations per point are given by grey dots. (b) The value of p_j that maximizes the forager lifetime as a function of s: $n = 400, 80, 20$ top-to-bottom as in Fig. 2.1(b). Maxima computed using golden-section search [21] using 10^6 simulations per point are given by grey dots.

In this case, we can obtain an analytic expression for the forager lifetime, by noting that the inter-feeding times X_k are geometrically distributed with success probability $p_j \frac{n-(k-1)}{n}$. This can be derived by noting that the probability that the forager lands on a piece of food is the probability that the forager jumps at all, p_j, multiplied by the probability that the forager lands on a site with food, $\frac{n-(k-1)}{n}$. The probability that $X_k = j$ is thus the probability of $j-1$ visits to empty sites multiplied by the probability of a visit to a site with food:

$$\mathbb{P}(X_k = j) = \left(1 - p_j \frac{n-(k-1)}{n}\right)^{j-1} \left(p_j \frac{n-(k-1)}{n}\right).$$

The conditional expectation and cumulative distributions for a geometric random variable can be computed in the standard way, giving the forager lifetime from Eq. (2.9):

$$\begin{aligned}\mathbb{E}(T) =& s + \sum_{k=1}^{n-1} \left[\left(\frac{p_j(k-n+1)+n}{n}\right)^s \prod_{\ell=1}^{k} (1 - \left(\frac{p_j(\ell-n)+n}{n}\right)^s) \right. \\ & \left. \sum_{i=1}^{k} \left[s + \frac{n}{(n-i)p_j} + \frac{s}{(\frac{p_j(i-n)+n}{n})^s - 1} \right] \right]\end{aligned} \tag{2.18}$$

Taking a large s limit of this expression, we obtain:

$$\mathbb{E}(T) \approx s + \frac{n}{p_j} \sum_{k=1}^{n-1} \frac{1}{k}, \tag{2.19}$$

which is exactly the approximation Eq. (2.1) for the mean cover time plus the starvation time s. The cover time is equal to that from the case of pure jumping, Eq. (2.5), scaled by $\frac{1}{p_j}$. This demonstrates that for sufficiently large s, the smaller the value of p_j, the longer the expected forager lifetime.

By examining Fig. 2.4(a), we see that the jump-wait model shares important characteristics with the jump-diffusion model. The forager lifetime is non-monotonic in p_j and the optimal value of p_j decreases as s increases. Additionally, the optimal p_j decreases as a function of s, but increases as a function of n [Fig. 2.4(b)]. Similar to the optimal p_j curves for the jump-diffusion model, the curves have sections of relatively rapid change for intermediate values of s. Thus, our findings for the jump-wait model again suggest that a starving forager can maximize their lifetime by balancing a decrease in the rate of food consumption (by lowering p_j) with an increase in the probability of surviving until most of the food is consumed (by increasing p_j).

2.6 Jump-Penalty Models

Up to this point, we have considered a model in which jumps and diffusive movements both take a single timestep. However, foragers exploring distant food patches typically require more time for these excursions [12]. Thus, we consider the effects of introducing a cost function associated with the jumps. Specifically, we consider two different cost functions: (i) a constant penalty, where each jump takes c timesteps, and (ii) a distance-based penalty function where each jump takes $c \cdot d$ timesteps for a distance d. During the jump, the forager cannot consume any food, and in the simplest version of this extended model, it may starve mid-jump if the jump cost is large enough. Thus, we also consider a model in which the forager only makes jumps if it will not starve mid-jump. In certain cases, models in which jumps are penalized have longer survival times, due to the inclusion of large epochs in which the agent is traveling and not exhausting the finite food supply.

2.6.1 Constant Penalty

We begin by considering a constant penalty function. Any time the forager jumps, it takes $c \in \mathbb{N}$ timesteps to do so. This provides a benefit to the forager by limiting premature resource depletion, but every jump the forager makes brings it much closer to starvation. Despite these additional risks, it is more beneficial for the forager to jump than to diffuse for $c > 1$ [Fig. 2.5(a),(c)]. Thus, the added risk incurred by decreasing the number of jumps needed to starve the forager is outweighed by the slowing of food depletion. However, a larger c does not necessarily correspond to a greater lifetime. The forager lives longer with $c = \frac{s}{2}$ than for $c = s$ [Fig. 2.5(e)]. This is because for $c = s$, any time the forager makes a jump to an empty site, it is guaranteed to starve, whereas with $c = \frac{s}{2}$, if the forager lands on an empty site, it can jump once more, substantially reducing its odds of starving.

In the case of pure jumping ($p_j = 1$) we can calculate an explicit formula for the forager's lifetime as long as c evenly divides s, following along similar lines to Section 2.4. Let $s = c \cdot$

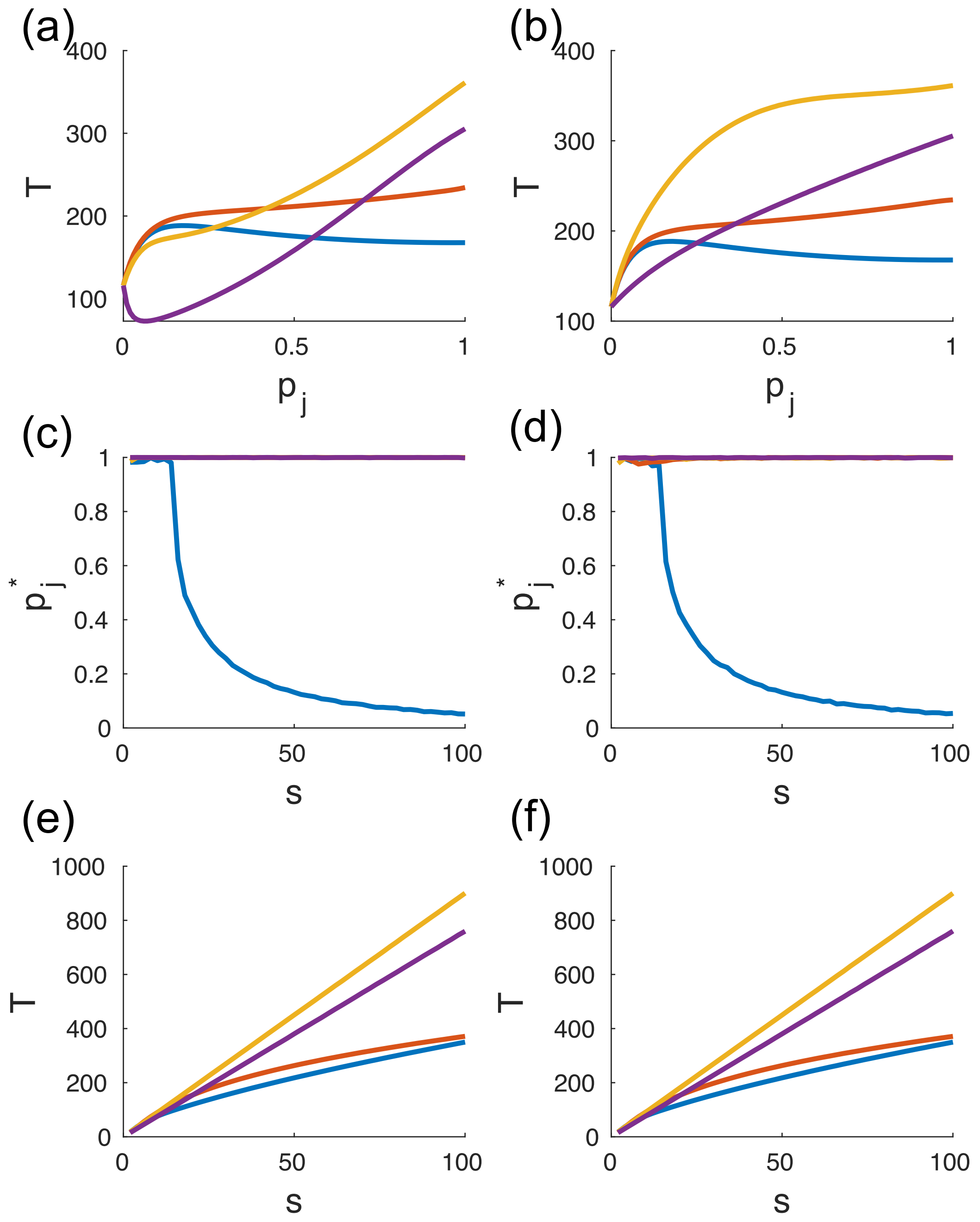

Figure 2.5: Jump-diffusion model of foraging with constant penalty c where the forager can [(a),(c),(e)] and cannot [(b),(d),(f)] starve during a jump. (a),(b) The expected lifetime of a forager as a function of p_j with $s = 40$. Means are computed with 10^6 numerical simulations per point. (c),(d) The optimal value of p_j as a function of s. Maxima are computed using golden-section search [21] with 10^7 simulations per point. (e),(f) The expected lifetime of a forager moving with the optimal p_j. Means are computed using 10^6 numerical simulations per point. $n = 40$ and blue, red, yellow, and purple correspond to $c(s) = 1, 2, s/2,$ and s.

q, for $c, q \in \mathbb{N}$. Let us define the geometric random variable X_k so that

$$\mathbb{P}(X_k = j) = \left(\frac{k-1}{n}\right)^{j-1} \left(\frac{n-(k-1)}{n}\right). \tag{2.20}$$

The equation for the lifetime is fairly similar to Eq. (2.9) with some modifications. Now, interfeeding times are given by $c \cdot X_k$ rather than X_k. Thus, the cumulative distribution of interfeeding times is computed by summing over Eq. (2.20):

$$\begin{aligned} F_k(s) &= \mathbb{P}(c \cdot X_k \leq s) = \sum_{j=1}^{q} \mathbb{P}(X_k = j), \\ F_{n+1}(s) &= 0, \end{aligned} \tag{2.21}$$

where $q = s/c \in \mathbb{N}$ as assumed. It follows that if $F_k(s)$ is the probability the forager survives long enough to consume the k^{th} piece of food after consuming the $(k-1)^{th}$ piece of food, then the probability the forager eats exactly k pieces of food is

$$\mathbb{P}(k^* = k) = \Big(1 - F_{k+1}(s)\Big) \prod_{j=1}^{k} \Big(F_j(s)\Big). \tag{2.22}$$

Finally, we calculate the conditional expectation, noting that $\mathbb{E}(c \cdot X_k | c \cdot X_k \leq s) = c \cdot \mathbb{E}(X_k | X_k \leq q)$. Replacing terms in Eq. (2.9) as such, and calculating the terms explicitly, using Eq. (2.20-2.22), we have the following result:

$$T(n, s, p_j = 1, c) = s + c \sum_{k=1}^{n-1} \left(\frac{k+1}{n}\right)^q \prod_{j=1}^{k} \left(1 - \left(\frac{j}{n}\right)^q\right) \times \sum_{i=1}^{k} \left[\frac{n}{n-i} + q + \frac{q}{(i/n)^q - 1}\right].$$

Indeed, the theoretical curves generated from $T(n, s, p_j = 1, c)$ match precisely with the results from numerical simulations in Fig. 2.5(e) when $p_j = 1$.

We can also consider an alteration to this model where the forager will not jump if it can die mid-jump. If the forager has $s_{current}$ steps left until starvation, then if $s_{current} < c$, the forager will strictly diffuse until it finds another piece of food or starves. If $s_{current} \geq c$, the forager will jump with probability p_j and diffuse with probability $1 - p_j$, exactly as before. This addition does not affect the optimal value of p_j [Fig. 2.5(b),(d),(f)]. For $c = 1$, the inability to die does not matter, since as long as the forager is alive, $s_{current} > 1 = c$. Furthermore, for $c > 1$, the addition of not being able to take a fatal jump does not make jumping any less favorable, so the optimal p_j continues to be 1.

2.6.2 Distance-Based Penalty

We now consider a penalty function for the jump that depends on the distance traveled d. If we enumerate our sites on the periodic lattice so that site i is adjacent to sites $i \pm 1$, for $i = 2, ...n$ and so site 1 and site n are adjacent, then the distance between sites x and y are $d(x, y) = \min(|x - y|, n - |x - y|)$. The number of timesteps for a jump from site x to site y is then given by $c \cdot d(x, y)$. Note that for c large enough, the average lifetime is a monotone decreasing function of the jump probability [Fig. 2.6(a)]. From Fig. 2.6(c), we see that for small values of s, it is better to solely diffuse, regardless of c. If s is small compared to the average jump penalty ($\approx c \cdot n/4$), the forager will likely starve mid-jump, possibly even on the first jump. However, for s sufficiently large compared to the average jump penalty, the benefits of slower food consumption become apparent. As in the case of constant jump penalties, the perils of dying mid-jump are outweighed by the slowed rate of food depletion. At this point the optimal strategy flips, and it is optimal to purely jump, which results in more rapid growth of the optimal survival time T as a function of s [Fig. 2.6(e)]. However, this never occurs for $c = \frac{s}{2}, s$, since the forager has a very small probability of surviving a jump, so it is always better to diffuse.

We can calculate the lifetime of the forager in the case where $p_j = 1$ and $c = s$ explicitly. When $c = s$, the forager can only survive jumps if it jumps to an adjacent site that still contains food. Since it takes zero time to jump to the site it is currently at, the forager essentially has $n - 1$ choices each time it needs to jump. For the first jump, it can survive if it jumps to either of the two adjacent sites, with probability $\frac{2}{n-1}$. If it jumps to any other site with probability $\frac{n-3}{n-1}$, it will starve mid-jump in s timesteps. For the forager to survive $s \cdot k$ steps for $1 < k < n$, it must survive the first jump with probability $\frac{2}{n-1}$ (by jumping to either of the two adjacent sites), it must survive $k - 2$ jumps with probability $\frac{1}{n-1}$ (by jumping to the one adjacent site with food), and then it must starve with probability $\frac{n-2}{n-1}$ (jumping to any site other than the adjacent site with food). The

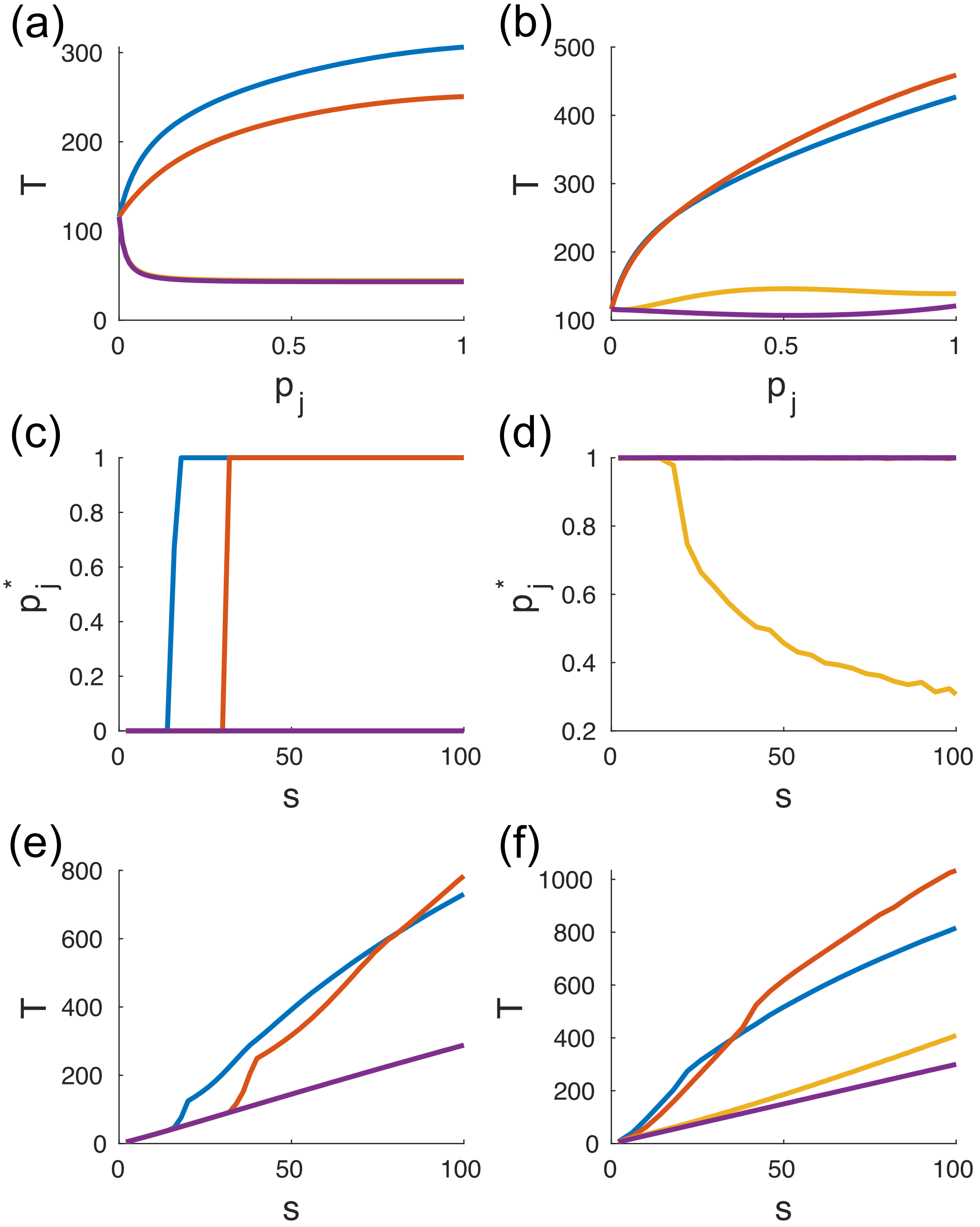

Figure 2.6: Jump-diffusion model of foraging with distance based penalty $c \cdot d$ where the forager can [(a),(c),(e)] and cannot [(b),(d),(f)] starve during a jump. (a),(b) The expected lifetime of a forager as a function of p_j with $s = 40$. Means are computed with 10^6 numerical simulations per point. (c),(d) The optimal value of p_j as a function of s. Maxima are computed using golden-section search [21] with 10^7 simulations per point. (e),(f) The expected lifetime of a forager moving with the optimal p_j. Means are computed using 10^6 numerical simulations per point. $n = 40$ and blue, red, yellow, and purple correspond to $c(s) = 1, 2, s/2,$ and s.

forager will live $s \cdot n$ steps if it survives the first jump with probability $\frac{2}{n-1}$, and it survives $n-1$ jumps each with probability $\frac{1}{n-1}$. The expected lifetime is

$$T(n, s, p_j = 1, c = s) = s\frac{n-3}{n-1} + 2s(n-2)\sum_{k=2}^{n-1} k\left(\frac{1}{n-1}\right)^k + \frac{2sn}{(n-1)^n}.$$

It is clear from Fig. 2.6(c) that it is always better to diffuse for such large values of the penalty scaling c.

We can also consider a modification to the model where the forager will only execute a jump if it does not starve mid-jump. In models discussed prior to this one, the forager jumps by selecting a uniformly randomly distributed site on the full domain to jump to [Fig. 1.1(a)]. In the modified model we consider now, a jumping forager restricts the space of possible next site locations to the set $V = \{y : c \cdot d(x, y) \leq s_{current}\}$, where $d(x, y)$ is the distance between the current x and next possible y site. This restriction prevents the forager from dying mid-jump. If $V = \{x\}$, then the forager will only diffuse until it consumes another piece of food, or it starves. The inability to die mid-jump substantially alters optimal strategy [Fig. 2.6(b),(d)]. For small values of s, it is now optimal to strictly jump for $c = 1, 2$, since the forager cannot make a jump that it will not survive. Thus, jumping outperforms diffusion for even smaller s. Even more altered is the optimal strategy for $c = s$. The forager is only permitted to move to neighboring sites, which will typically have a probability 1/2 of containing food, as a desert is carved out in the forager's wake. Now that the forager survives a jump with probability no less than $\frac{1}{2}$ instead of $\frac{1}{n-1}$, it is always optimal to jump. Finally, for $c = \frac{s}{2}$, the optimal values of p_j look qualitatively similar to the optimal values of p_j in the original model (Fig. 2.1B). Lifetimes are longer when c is relatively small as compared to s [Fig. 2.6(f)]. However, once s becomes large enough that the forager can comfortably reach the entire domain, the optimal strategy for $c = 2$ leads to longer lifetimes than $c = 1$.

We can again calculate the forager lifetime in the case of pure jumping with $c = s$. The forager is guaranteed to survive at least the first jump. Every subsequent jump is survived with probability $\frac{1}{2}$ if it jumps to the adjacent site with food. Otherwise, it will starve upon reaching the empty adjacent site. Thus, the proability that the forager lives $c \cdot k$ timesteps is $\frac{1}{2^{k-1}}$ for

$1 < k < n-1$. To survive $s \cdot n$ steps, the forager needs to survive the first jump with probability 1, and $n-2$ subsequent jumps each with probability $\frac{1}{2}$. The expected forager lifetime is

$$T(n, s, p_j = 1, c = s) = s \sum_{k=2}^{n-1} \frac{k}{2^{k-1}} + \frac{sn}{2^{n-2}} = s(3 - 2^{2-n}),$$

which matches the points generated from numerical simulations in Fig. 2.6(f).

2.7 Conclusions

We have extended the recently-developed starving forager model [5] to account for the possibility of long-range motion via jumping. The combination of these two modes of movement is related to Lévy-type motion often found in the dynamics of motile organisms' foraging strategies [17]. By analyzing cover times, we have shown that jumping consumes food more rapidly than diffusion. This provides an explanation for why a mixture of jumping and diffusion is optimal: excessive jumping leads to rapid food depletion, excessive diffusion leads to earlier starvation of the forager who gets stuck in food 'deserts.' In a sense, the forager optimally balances exploration (via jumping) and exploitation (via diffusion) when using a mixture of these modes of movement. This explanation is further validated by the qualitative similarities of the jump-diffusion and jump-wait models. In either model: Making s larger or n smaller lead to situations in which the forager maximizes their lifetime by diffusing more, while the converse corresponds to more jumping being optimal.

The non-monotonicity of the forager's lifetime in p_j breaks down when considering a model with time-penalized jumping. For constant time penalties, foragers obtain the longest lifetime by practicing pure jumping. The advantage formerly gained by diffusing, the slowing of food consumption, is now accomplished via the time penalties: No food is consumed while the forager is in the midst of a jump. When jumps are penalized as a function of their distance, pure diffusion becomes optimal, as most jumps will result in sudden death. However, if foragers enact a strategy of not jumping if it causes them to die, the best strategy is then for foragers to make long-distance jumps unless they will die mid-jump, in which case they diffuse. This more sophisticated strategy recovers an exploration/exploitation tradeoff that depends on the the forager's nearness to starvation.

Our model of a starving forager with a mixture of movement modes suggests several other possible extensions. Throughout this work, jumping has represented movement with equal probability to any lattice site. However, a forager executing a jump may more often select a site that is further away, to avoid revisiting empty sites. They may also be less likely to make extremely large jumps. This would suggest a model where the jump process is associated with a nonuniform distribution of jump distances. Our extension to a model that considers distance-penalized jumping has partially incorporated such a strategy by only allowing jumps below a certain distance, when the forager wishes to avoid starvation. However, we could also consider strategies whereby the forager only takes jumps above a certain size, to try and promote movement out of food deserts.

Our work has also only considered a periodic one-dimensional lattice environment. The behavior of the forager in higher dimensions is still open, and it would be interesting to see how the forager lifetime depends on domain size and geometry in higher dimensions (e.g., plane, torus, or sphere). Another relevant extension would be for the forager to retain some information about its previous actions. For example, p_j could increase, as the number of steps without food increases. This would provide a strategy in which the forager only executes long range movement if they are starving, which will probably limit the rate at which food is consumed and increase the overall lifetime T. Our model could also incorporate greed (or anti-greed) as a parameter [6]. As shown in previous work, the lifetime of foragers increases in one-dimensional environments if their diffusion is biased away from food. This finding mirrors our own conclusion, that foragers maximize their lifetime by balancing a reduction in the probability of early starvation with the conservation of resources.

Chapter 3

Collective Search

3.1 Introduction

This chapter discusses searching for a target in a patchy environment, or similarly, foraging for food in an environment where food is sparsely located. In Sec. 3.2, we formulate the search process as a Partially Observable Markov Decision process. This formulation allows us to numerically solve the problem, and compare the results to the rest of our analysis. We consider the case of a single searcher in Sec. 3.3 and demonstrate that a specific strategy yields good results. This work is extended to the case of collective search in Sec. 3.4, and we demonstrate that the optimal strategies for collective search are qualitatively different than those for a single searcher. Specifically, groups of foragers should spread themselves out among the patches to most efficiently find the target.

3.2 Partially Observable Markov Decision Processes

Partially Observable Markov Decision Processes (POMDPs) offer a flexible way of describing the movement of agents throughout a random environment [25]. Our collective search process can be precisely modelled as a POMDP as follows. We define the set of states $S = \{0, 1, 2, ..., z-1, \lambda\}$ where each state corresponds to a patch, so that the system is in state i if and only if the target is in patch i. State λ is a placeholder state, which is entered only when the target has been found. The set A of actions denotes all configurations of the y searchers among the z patches. For example, if there are two searchers and two patches, then the actions A are i) both foragers search the first patch ii) both foragers search the second patch and iii) one forager searches the first patch and the other searches the second patch. We define the extended state variable $\widetilde{S} = S \times A$ as follows. At time t, the system is in extended state $\widetilde{s_t} = (s, a)$, signifying that it is in state s_t and that the action at time $t-1$ was a_{t-1}. This offers us a way of storing the previous action in the state variable, which ensures that our process is Markov.

The goal for our search process is to minimize the amount of time to find the target. POMDPs are typically written to maximize a reward function, so we define our reward function $R(\widetilde{s}, a)$ to give us the reward of implementing action a while in extended state $\widetilde{s} = (s, b)$.

$$R(\widetilde{s}, a) = \begin{cases} 0 & s = \lambda \quad \text{Target has been found, nothing more to do} \\ -1 & s \neq \lambda \text{ and } a = b \text{ Takes one time step to search, no reorganization} \\ -\tau - 1 & s \neq \lambda \text{ and } a \neq b \text{ Takes } \tau \text{ time steps to reorganize, one time step to search,} \end{cases}$$

The reward is structured so that if the current action a differs from the prior action b, then the foragers must reorganize (which takes τ time steps).

We also need to describe how we transition among the states. The transition function $F(s, a, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$, the probability of transitioning from state s at time t to state s' at time $t+1$ after performing action a. For $\widetilde{s} = (s, b), \widetilde{s'} = (s', a)$, we have

$$F(\widetilde{s}, a, \widetilde{s'}) = \begin{cases} 1 & s = s' \text{ and } 0 \text{ of the foragers are in the same patch as the target} \\ 1 - (1-\alpha)^j & s \neq \lambda, s' = \lambda,\ j \text{ of the foragers are in the same patch as the target} \\ (1-\alpha)^j & s = s',\ j \text{ of the foragers are in the same patch as the target} \\ 1 & s = s' = \lambda, \text{ once the target has been found, the system stops evolving} \end{cases}$$

Essentially, the transitions are solely determined by how many of the foragers are in the same patch as the target. At this point, we can see that if we know which state $s \in S$ the system is in, then our search problem is very easy to solve. Namely, if the system is in state i, then we should have all of our foragers search patch i constantly until the target is found. Hence, to accurately model the search process, we need to introduce partial observability, so that we do not know what state the system is in until the target is found. We define the set $O = \{o_1, o_2\}$ to denote the two meaningful observations we can obtain after executing an action. Observation o_1 corresponds to searching for the target and not finding it, while observation o_2 corresponds to searching for the target and finding it. We define $Z(\widetilde{s}, a, o) = \mathbb{P}(o_t = o | a_t = a, \widetilde{s}_t = \widetilde{s})$, the probability of getting observation o after executing action a in state $\widetilde{s}$. We define it so that

$$Z(\widetilde{s}, a, o) = \begin{cases} 1 & o = o_1 \text{, and 0 of the foragers are in the same patch as the target} \\ 1 - (1-\alpha)^j & o = o_2 \text{, } j \text{ of the foragers are in the same patch as the target} \\ (1-\alpha)^j & o = o_1 \text{, } j \text{ of the foragers are in the same patch as the target} \end{cases}$$

The state transition function F and the observation function Z act similarly, where an observation of o_2 moves the system to the placeholder state λ, while the observation of o_1 does not change the state at all. Receiving observation o_2 simply indicates that the system has terminated, and provides no meaningful information beyond that. However, with o_1, we can define the notion of a belief state, motivated by the idea that if one patch is repeatedly searched without finding the target, then we are less likely to believe the target is in that patch. To formalize this, let there be m_1 foragers in patch 1, m_2 foragers in patch 2, continuing until there are m_z foragers in patch z where $m_1 + m_2 + ... + m_z = y$. Then, by applying Bayes' rule, we have that

$$\mathbb{P}(s = k | o_1) = \frac{\mathbb{P}(o_1 | s = k)\mathbb{P}(s = k)}{\mathbb{P}(o_1)} = \frac{(1-\alpha)^{m_k}\mathbb{P}(s = k)}{1 - \sum_{i=1}^{z}(1 - (1-\alpha)^{m_i})\mathbb{P}(s = i)}$$

We define the vector $b = (x_1, x_2, ..., x_z)$ so that x_k is our current belief that the target is in patch k. After performing our search with the given configuration of foragers, we have that

$$b := (\frac{(1-\alpha)^{m_1}x_1}{1 - \sum_{i=1}^{z}(1 - (1-\alpha)^{m_i})x_i}, \frac{(1-\alpha)^{m_2}x_2}{1 - \sum_{i=1}^{z}(1 - (1-\alpha)^{m_i})x_i}, ..., \frac{(1-\alpha)^{m_z}x_z}{1 - \sum_{i=1}^{z}(1 - (1-\alpha)^{m_i})x_i})$$

We begin with a uniform prior so that $b = (\frac{1}{z}, \frac{1}{z}, ..., \frac{1}{z})$, and then update b according to Bayes' rule as above. The history of the POMDP up to time t is the sequence $h_t = a_1o_1a_2o_2...a_to_t$ describing the actions and observations up to the present time. By design the belief vector b encapsulates all meaningful information from the history. The question is whether given a history (or belief state vector), we can determine the optimal action to take next so we find the target in the least amount of time. There exists a large literature on numerically solving POMDPs. Some methods include approximating Bellman's Equation or discretizing the belief space [25], but here we will use a variant of Monte-Carlo tree search from [26] called POMCP (Partially Observable Monte Carlo Planning). This type of approximate solution method uses Monte-Carlo simulation to peer a little into the future, and then selects the optimal action based on this look-ahead. While we will not delve in-depth into numerical solutions for POMDPs, applying the POMCP algorithm to our POMDP will give us numerical results, which we can use to gauge the success of the subsequent heuristics. Namely, we will say that a strategy has good performance if it can find the target in less time (or an approximately equal amount of time) than it takes the policy found by POMCP to find the target.

3.3 Single Searcher

Instead of trying to minimize the time to find the target over the set of all possible sequences of actions (or policies), we can instead fix a type of policy, and then optimize over its parameters instead. To do this, we will begin with the simplest variant of the problem, that of a single searcher. A thresholding scheme, where the single searcher visits the patches sequentially for fixed amounts of time, has good performance. We will be able to examine the effects of the time delay τ and the number of patches on the patch residence time.

The easiest version of our search problem to solve is that of a single searcher trying to find the target which can be in one of two patches. A reasonable strategy is to search in one of the patches for some amount of time, transition to the next patch, search in that patch, transition to the first patch, continuing until the target is found. For simplicity, we will say that the forager searches for n_0 time steps on its first patch visit, and n_1 time steps on every subsequent patch visit. If the agent starts in the target domain, the probability of finding the target in this first visit is

$$P_1(n_0) = \alpha \sum_{j=0}^{n_0-1} (1-\alpha)^j = 1 - (1-\alpha)^{n_0}, \tag{3.1}$$

and the mean time to find the target within that visit is

$$T_1(n_0) = \frac{\alpha}{P_1(n_0)} \sum_{j=0}^{n_0-1} (j+1)(1-\alpha)^j = \frac{1-(1+\alpha n_0)(1-\alpha)^{n_0}}{\alpha(1-(1-\alpha)^{n_0})}. \tag{3.2}$$

Otherwise, the agent will spend a total of n_0 timesteps searching and not finding the target. Subsequently, we can estimate the probability of finding the target and corresponding search time within each future visit as

$$P_1(n_1) = 1-(1-\alpha)^{n_1}, \quad T_1(n_1) \approx \frac{1-(1+\alpha n_1)(1-\alpha)^{n_1}}{\alpha(1-(1-\alpha)^{n_1})}. \tag{3.3}$$

Otherwise, the agent spends n_1 timesteps on each future visit. Note, the time spent in the nontarget patch will be n_0 on the first visit, if starting there, but otherwise approximately n_1. We can therefore approximate the total time to find the target, given the agent begins in the target domain half the time, and the non-target domain the other half of the time as

$$\begin{aligned}\overline{T}(\alpha, \mathbf{n}) = &\frac{1}{2}[P_1(n_0)T_1(n_0) + (1-P_1(n_0))P_1(n_1) \\ &\sum_{j=0}^{\infty}(1-P_1(n_1))^j(T_1(n_1) + n_0 + (2j+1)n_1 + 2(j+1)\tau) \\ &+ P_1(n_1)\sum_{j=0}^{\infty}(1-P_1(n_1))^j(T_1(n_1) + n_0 + 2jn_1 + (2j+1)\tau)]\end{aligned} \tag{3.4}$$

where $\mathbf{n} = (n_0, n_1)$.

This type of strategy also generalizes naturally to a single searcher and more than two patches. If there are z patches, then we define $\mathbf{n} = (n_0, n_1, n_2, ...n_{z-1})$ so that the forager spends n_0 time steps on its first patch visit, n_1 time steps on its next patch visit, continuing until it spends n_{z-1} time steps on the z^{th} patch visit. At this point, the forager will continue to spend n_{z-1} time steps on every subsequent patch visit. Since the forager has no prior information about the location of the target, we assume without loss of generality that it visits patch 0 first, patch 1 second, continuing until it visits patch $z-1$ last, at which point it begins at patch 0 once more. The lifetime of the forager is then given by

$$\overline{T}(\alpha, \mathbf{n}) = \frac{1}{z}\Bigg[\sum_{i=0}^{z-1} P_1(n_i)[i\tau + T_1(n_i) + \sum_{j=0}^{i-1} n_j]$$
$$+ \sum_{k=1}^{\infty}\sum_{i=0}^{z-1} P_1(n_i)(1 - P_1(n_i)^k[(kz+i)\tau + T_1(n_i) + \sum_{j=0}^{z-1} n_j + (i + (k-1)z)n_{z-1}]\Bigg]$$

where $P_1(n_i), T_1(n_i)$ are defined similarly to Eq. (3.3) as

$$P_1(n_i) = 1 - (1-\alpha)^{n_i}, \qquad T_1(n_i) \approx \frac{1 - (1 + \alpha n_i)(1-\alpha)^{n_i}}{\alpha(1 - (1-\alpha)^{n_i})}. \tag{3.5}$$

Eq. (3.5) gives the time to find the target as a function of $\mathbf{n}$ (assuming we fix α). Hence, to find an optimal searching strategy, we need to optimize $T(\alpha, \mathbf{n})$ over $\mathbf{n}$. This is seemingly difficult, since it involves optimizing a nonlinear function over integer valued variables (since $\mathbf{n}$ describes the number of time steps in each patch visit). However, we can get appreciably good results by implementing a naive approach as follows. We can simply ignore the integer constraints, optimize $T(\alpha, \mathbf{n})$ over $\mathbf{n}$ subject to each of the $n_i \geq 1$. After doing this, we round each value of the vector to the nearest integer. Though this does not optimize over the integers, and though we have no guarantees about the convexity of our objective function, we can see from Fig. 3.1 that these rough techniques are sufficient to get results which match or out-perform the results of the POMCP numerical simulation. As we would expect, a larger time constant increases the time to find the target [fig. 3.1(a)], as does an increase in the number of patches [fig. 3.1(c)]. An increase in τ corresponds to an increase in patch residence time [fig. 3.1(b)]. This also follows intuition, since a more costly transition increases the incentive to more thoroughly search the current location. Finally, we see that an increase in the number of patches corresponds to a decrease in the patch residency times, since the target is less likely to be in any given patch.

3.4 Multiple Searchers

We will now examine a more complicated problem by permitting multiple searchers. From the previous section, we have seen that a thresholding scheme is optimal in the case of a single searcher. A natural followup is to see if some type of a thresholding scheme can be extended to the

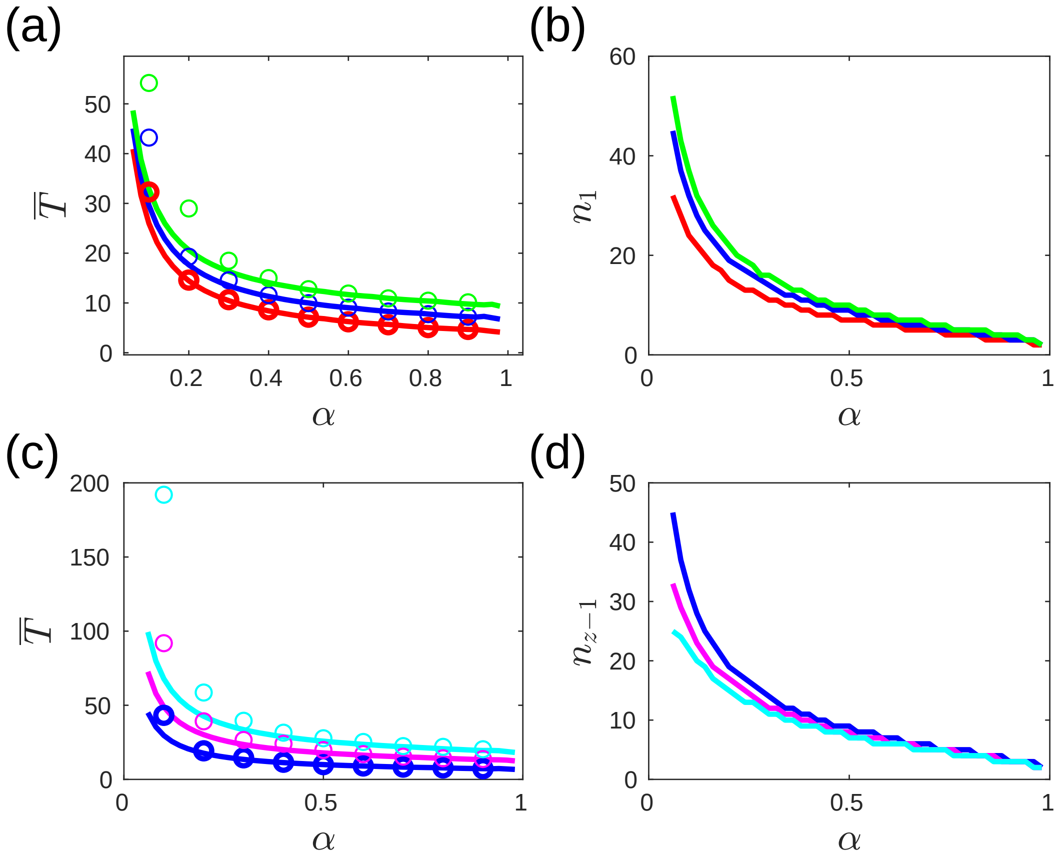

Figure 3.1: (a),(c) Time to find the target. Solid lines given by optimizing Eq. (3.4), dots given by applying POMCP [26] to the POMDP formulation. (b),(d) Final patch dwell time n_{z-1}, given from the arguments of optimizing Eq. (3.4). One-searcher, two-patch case with $\tau = 5, 10, 15$ corresponds to red, blue, and green. One-searcher with $\tau = 10$ and two, three, and four patches corresponds to blue, magenta, and cyan.

case of multiple searchers, and whether the performance is comparably good to numerical results. The simplest way of extending the previous work to multiple searchers is to consider a strategy where all searchers are always grouped together in the same patch. The group of foragers then transitions to each patch sequentially. The analysis of this is almost identical to the single searcher case. The time to find the target is given by

$$\begin{aligned}\overline{T}(\alpha, \mathbf{n}) = \frac{1}{z}\Bigg[&\sum_{i=0}^{z-1} P_y(n_i)[i\tau + T_y(n_i) + \sum_{j=0}^{i-1} n_j] \\ &+ \sum_{k=1}^{\infty}\sum_{i=0}^{z-1} P_y(n_i)(1 - P_y(n_i)^k[(kz+i)\tau + T_y(n_i) + \sum_{j=0}^{z-1} n_j + (i + (k-1)z)n_{z-1}]\Bigg]\end{aligned} \tag{3.6}$$

where

$$\begin{aligned} P_y(n_i) &= (1 - (1-\alpha)^y)\sum_{j=0}^{n_i - 1}(1-\alpha)^{yj} = 1 - (1-\alpha)^{yn_i} \\ T_y(n_i) &= \frac{(1-(1-\alpha)^y)}{P_y(n_i)}\sum_{j=0}^{n_i-1}(j+1)(1-\alpha)^{yj} = \frac{1 + (1-\alpha)^{n_i y}(-n_i - 1 + n_i(1-\alpha)^y)}{(1-(1-\alpha)^y)(1-(1-\alpha)^{n_i y})}\end{aligned} \tag{3.7}$$

While this method follows from the single searcher case, it is also worth considering whether there are any strategies that can exploit the existence of multiple searchers to more efficiently find the target. We will consider four cases, where the number of foragers is much larger, larger, equal, and smaller than the number of patches. We begin with the case of two searchers and two patches. A simple strategy is to assign each searcher to its own patch, and never have the searchers transition over to the other patch. The expected time to find the target is given by a standard geometric random variable, so

$$\overline{T}(\alpha) = \sum_{j=0}^{\infty} \alpha j (1-\alpha)^{j-1} = \frac{1}{\alpha} \tag{3.8}$$

We next consider the case of two searchers and three patches. Here we consider a strategy where the searchers alternate between two positions. In the first positions, one searcher is in the first patch, while another is in the second patch. In the second position, both searchers are in the third patch. This configuration allows the foragers to simultaneously search multiple patches without having to incur the cost of additional transitions. The time to find the target in this case is given by

$$\begin{aligned}\overline{T}(\alpha) = \frac{2}{3}P_1(n_0)T_1(n_0) + \frac{1}{3}\sum_{k=0}^{\infty} P_2(n_1)(1-P_2(n_1))^k(n_0 + k(n_1+n_2) + (2k+1)\tau + T_2(n_1)) \quad (3.9)\\ +\frac{2}{3}P_1(n_2)(1-P_1(n_0))\sum_{k=0}^{\infty}(1-P_1(n_2))^k(n_0 + (k+1)n_1 + kn_2 + 2(k+1)\tau + T_1(n_2))\end{aligned}$$

where P_1, P_2, T_1, T_2 are defined as in Eq. (3.7).

In the case of two searchers and four patches, a natural strategy is to partition the domain, so that the first forager explores the first and second patches, while the second forager explores the third and fourth patches. The time to find the target is then identical to the one searcher two patch case, given by Eq. (3.4).

Finally, we consider the case of three searchers and two patches. This differs from previous cases in that it is possible for all the patches to have at least one searcher. The choice is then how to assign the extra searchers. We consider the simplest strategy where two foragers reside in one patch, and the third forager resides in the other patch. Given that the target is in the first patch with probability $\frac{1}{2}$ and the second patch with probability $\frac{1}{2}$, the expected time to find the target is

$$\overline{T}(\alpha) = \frac{1}{2}[1-(1-\alpha)^2+\alpha]\sum_{j=0}^{\infty}(j+1)(1-\frac{1}{2}[1-(1-\alpha)^2+\alpha])^j = \frac{2}{3\alpha-\alpha^2} \quad (3.10)$$

Fig. 3.2 summarizes the results. We can see that in every case, our strategies which spread the foragers more evenly across the patches yield better results than grouping them all together. This result is intuitive, since by searching more patches simultaneously, the foragers can avoid the costly penalty required to reorganize themselves among the patches. The heuristics also perform as well, if not better, than the numerical results from POMCP. This follows from the fact that POMCP chooses policies by using Monte Carlo simulations to look into the future. Since only finite simulations are performed, the solver can sometimes choose sub-optimal strategies, and this trend is particularly exacerbated when α is small, and when the set of actions is large.

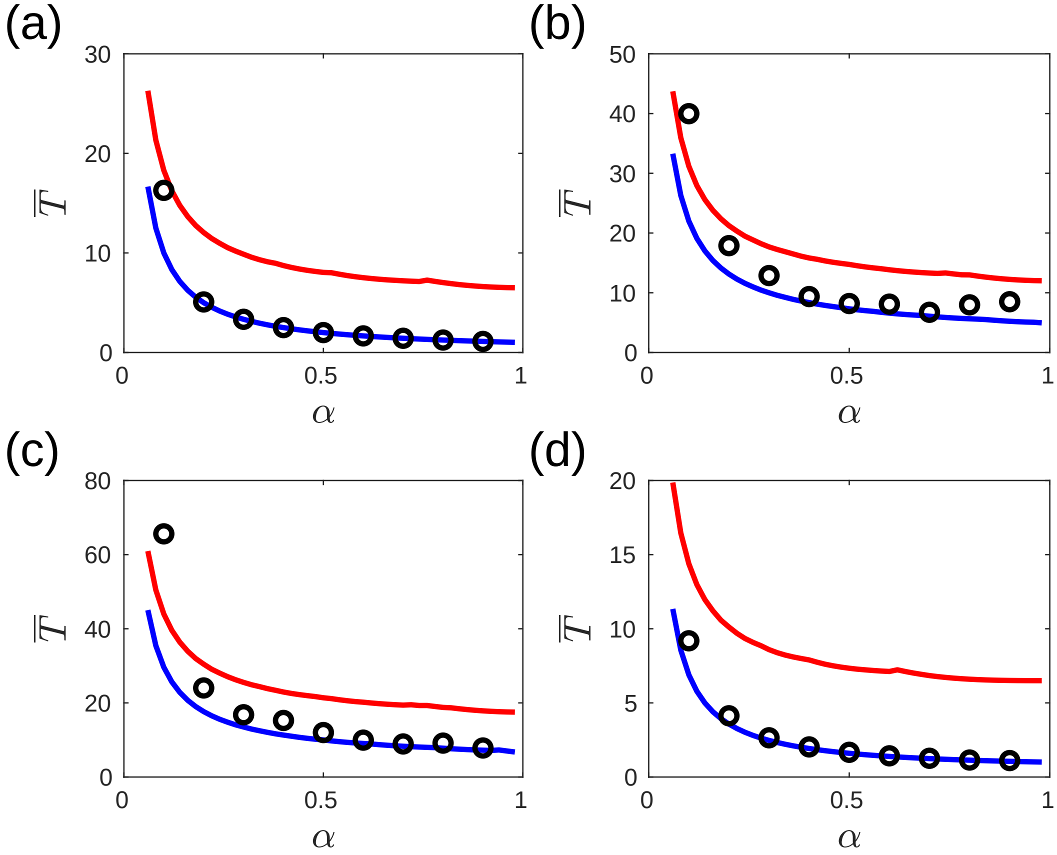

Figure 3.2: Time to find the target for different configurations of patches and searchers and $\tau = 10$. Blue denotes the thresholding strategy found by optimizing Eq. (3.6), while the dots denote the results of applying POMCP [26] to the POMDP formulation. (a) Two-searcher two-patch, red denotes Eq. (3.8). (b) Two-searcher three-patch, red denotes optimizing Eq. (3.9). (c) Two-searcher four-patch, red denotes optimizing Eq. (3.4). (d) Three-searcher two-patch, red denotes Eq. (3.10)

3.5 Conclusions

We have applied collective decision making to a search process in a patchy environment. By developing and testing a thresholding scheme for a single forager, we were able to demonstrate its effectiveness. We showed that as it became more expensive for the forager to reposition, it spent more time in each patch. As the number of patches increased, the time spent in each patch decreased, since the searcher needed to explore more patches to find the target. We also demonstrated that groups of searchers should search in a significantly different manner than a single searcher. Groups should spread themselves out, to search as many patches as possible without having to take costly transitions and reposition themselves. The strategies associated with this spreading significantly outperformed the extension of the thresholding rule to multiple searchers. Further study could include multiple targets, heterogeneous time delays or search probabilities, or delays in how the searchers can communicate information.

Chapter 4

Conclusions

We have explored two models of foraging in random environments. The first was a memoryless jump-diffusion model. Analysis showed that there was an optimal balance between the two movement paradigms, with enough jumping so that the forager was kept from starving, and enough diffusion so that the forager could conserve resources. The extension where jumping took more time than diffusion was also explored. This modification tended make jumping more optimal, since with the additional delay, jumping consumed resources far more slowly. However, if the forager was not allowed to die mid-jump than a hybrid strategy can become beneficial once more.

The other model we explored was searching for a target in a patchy environment, akin to foraging for sparsely located food. In the case of a single searcher, smaller penalties for transitioning between patches and a greater number of patches corresponded to lesser patch residency times. This is another example of the exploit-explore tradeoff we examined in the jump-diffusion model. By applying collective decision making to the process, we found that spreading the searchers more evenly across the patches reduced the time required to find the target.

Bibliography

[1] F. Bartumeus, J. Catalan, U. L. Fulco, M. L. Lyra, and G. M. Viswanathan. Optimizing the encounter rate in biological interactions: Lévy versus brownian strategies. Phys. Rev. Lett., 88:097901, Feb 2002.

[2] Willilam J. Bell. Searching Behaviour. Chapman and Hall Animal Behaviour Series. Chapman and Hall, 1990.

[3] O. Bénichou, U. Bhat, P.L. Krapivsky, and S. Redner. Optimally frugal foraging. Phys. Rev. E, 97(022110), 2018.

[4] O. Bénichou, C. Loverdo, M. Moreau, and R. Voituriez. Intermittent search strategies. Rev. Mod. Phys., 83:81–129, Mar 2011.

[5] Olivier Bénichou and S Redner. Depletion-controlled starvation of a diffusing forager. Physical review letters, 113(23):238101, 2014.

[6] U. Bhat, S. Redner, and O. Bénichou. Does greed help a forager survive? Phys. Rev. E, 95:062119, 2017.

[7] U Bhat, S Redner, and O Bénichou. Starvation dynamics of a greedy forager. Journal of Statistical Mechanics: Theory and Experiment, 2017(7):073213, 2017.

[8] G Blom and D Sandell. Cover times for random walks on graphs. Math. Scientist, 17:111–119, 1992.

[9] Eric L. Charnov. Optimal foraging, the marginal value theorem. Theoretical Population Biology, 9(2):129–136, 1976.

[10] M. Chupeau, O. Bénichou, and S. Redner. Universality classes of foraging with resource renewal. Phys. Rev. E, 93(032403), 2015.

[11] William Feller. An Introduction to Probability Theory and Its Applications, chapter 14. John Wiley & Sons, Inc., 1968.

[12] Chris H Fleming, Justin M Calabrese, Thomas Mueller, Kirk A Olson, Peter Leimgruber, and William F Fagan. From fine-scale foraging to home ranges: a semivariance approach to identifying movement modes across spatiotemporal scales. The American Naturalist, 183(5):E154–E167, 2014.

[13] Thomas T. Hills, Peter M. Todd, David Lazer, A. David Redish, and Iain D. Couzin. Exploration versus exploitation in space, mind, and society. Trends in Cognitive Sciences, 19(1):46–54, 2015.

[14] Nikhil Krishnan and Zachary P. Kilpatrick. Optimizing a jump-diffusion model of a starving forager. Phys. Rev. E, 98:052406, Nov 2018.

[15] JR Krummel, RH Gardner, G Sugihara, RV O'neill, and PR Coleman. Landscape patterns in a disturbed environment. Oikos, pages 321–324, 1987.

[16] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. Markov Chains and Mixing Times. American Mathematical Society, Providence, Rhode Island, 2008.

[17] Pascual López-López, José Benavent-Corai, Clara García-Ripollés, and Vicente Urios. Scavengers on the move: Behavioural changes in foraging search patterns during the annual cycle. PLOS ONE, 8(1):1–9, 01 2013.

[18] J.R. Norris. Markov Chains. Cambridge University Press, New York, New York, 1997.

[19] M.J Plank and A James. Optimal foraging: Lévy pattern or process? Journal of The Royal Society Interface, 5(26):1077–1086, 2008.

[20] Stephen C. Pratt, Eamonn B. Mallon, David J. Sumpter, and Nigel R. Franks. Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant leptothorax albipennis. Behavioral Ecology and Sociobiology, 52(2):117–127, Jul 2002.

[21] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. Numerical Recipes, chapter 10. Cambridge University Press, 2007.

[22] C.L. Rager, U. Bhat, O. Benichou, and S. Redner. The advantage of foraging myopically. arXiv, 2018.

[23] Gabriel Ramos-Fernández, José L. Mateos, Octavio Miramontes, Germinal Cocho, Hernán Larralde, and Bárbara Ayala-Orozco. Lévy walk patterns in the foraging movements of spider monkeys (ateles geoffroyi). Behavioral Ecology and Sociobiology, 55(3):223–230, Jan 2004.

[24] Thomas D. Seeley, Scott Camazine, and James Sneyd. Collective decision-making in honey bees: how colonies choose among nectar sources. Behavioral Ecology and Sociobiology, 28(4):277–290, Apr 1991.

[25] Guy Shani, Joelle Pineau, and Robert Kaplow. A survey of point-based pomdp solvers. Autonomous Agents and Multi-Agent Systems, 27(1):1–51, Jul 2013.

[26] David Silver and Joel Veness. Monte-carlo planning in large POMDPs. Advances in Neural Information Processing Systems 23 (NIPS), 2010.

[27] David W. Stephens. Decision ecology: Foraging and the ecology of animal decision making. Cognitive, Affective, & Behavioral Neuroscience, 8(4):475–484, 2008.

[28] David W. Stephens and John R. Krebs. Foraging Theory. Princeton University Press, Princeton, New Jersey, 1987.

[29] D Thompson and MA Fedak. How long should a dive last? a simple model of foraging decisions by breath-hold divers in a patchy environment. Animal Behaviour, 61(2):287–296, 2001.

[30] Yuuki Y. Watanabe, Motohiro Ito, and Akinori Takahashi. Testing optimal foraging theory in a penguin–krill system. Proceedings of the Royal Society of London B: Biological Sciences, 281(1779), 2014.

Appendix A

Calculating Small System Transitions

With a four site geometry, the forager has two types of movement. It can either move to the opposite site or remain at the current site both with probability $p_+ = \frac{p_j}{4}$ or it can move to a specific adjacent site with probability $p_- = \frac{2-p_j}{4}$. For brevity, we denote $\mathbb{P}(T(4,2,p_j) = i)$ as p_i, and enumerate all possible paths

$$\begin{aligned}
p_2 =& p_+^2, \qquad p_3 = \frac{p_-}{2} + 4p_+^3 \\
p_4 =& p_+p_3 + 2(p_-^2 + p_-p_+)(2p_-^2 + 3p_-p_+ + 4p_+^2) + 2(p_+p_-)(2p_-^2 + 6p_-p_+ + p_+^2) \\
p_5 =& p_+(2(p_-^2 + p_-p_+)(2p_-^2 + 3p_-p_+ + 4p_+^2) + 2(p_+p_-)(2p_-^2 + 6p_-p_+ + p_+^2)) + \\
& 2(p_-^3 + p_-^2p_+ + p_-p_+^2) + (1/2)p_-(2p_-^2 + 3p_-p_+ + 4p_+^2) + 4p_-p_+^2(2p_-^2 + 6p_-p_+ + p_+^2) \\
p_6 =& 2(p_-^2 + p_-p_+)(3p_-p_+) + 2p_-p_+(2p_-^2 + p_+^2) + p_+2(p_-^3 + p_-^2p_+ + p_-p_+^2) + p_-(p_-^2 + p_+p_-) + \\
& 4p_+^3p_- + p_+((1/2)p_-(2p_-^2 + 3p_-p_+ + 4p_+^2) + 4p_-p_+^2(2p_-^2 + 6p_-p_+ + p_+^2)) \\
p_7 =& (3/2)p_-^2p_+ + 4p_+^2p_-(2p_-^2 + p_+^2) + p_+(2(p_-^2 + p_-p_+)(3p_-p_+) + 2p_-p_+(2p_-^2 + p_+^2)) + \\
& p_+(p_-(p_-^2 + p_+p_-) + 4p_+^3p_-) \\
p_8 =& p_+((3/2)p_-^2p_+ + 4p_+^2p_-(2p_-^2 + p_+^2))
\end{aligned}$$

We have $\mathbb{E}(T(4,2,p_j)) = \sum_{i=2}^{8} ip_i$ which yields Eq. (2.17).

Appendix B

Transition matrix for small system

We define the states of the small system with $n = 4$ sites and survival time $s = 2$ according to the relative location of the forager and the arrangement of food sites remaining. This numbers thirteen distinct states, with a transition matrix Q for the update of the state vector $S_{t+1} = Q^T S_t$ where $Q_{1,2} = Q_{1,3} = Q_{2,3} = Q_{2,13} = Q_{9,10} = Q_{9,11} = Q_{10,11} = \frac{p_j}{4}$, $Q_{1,5} = Q_{2,5} = Q_{9,8} = \frac{2-p_j}{2}$, $Q_{3,4} = Q_{4,13} = Q_{7,8} = \frac{p_j}{2}$, $Q_{3,9} = Q_{4,9} = Q_{7,10} = Q_{7,11} = Q_{8,11} = \frac{2-p_j}{4}$, $Q_{5,6} = Q_{5,7} = Q_{6,7} = Q_{6,13} = \frac{1}{2}$, $Q_{8,13} = \frac{2+p_j}{4}$, $Q_{10,13} = \frac{4-p_j}{4}$, $Q_{11,12} = Q_{12,13} = Q_{13,13} = 1$. Note that the thirteenth state is the absorbing cemetery state.