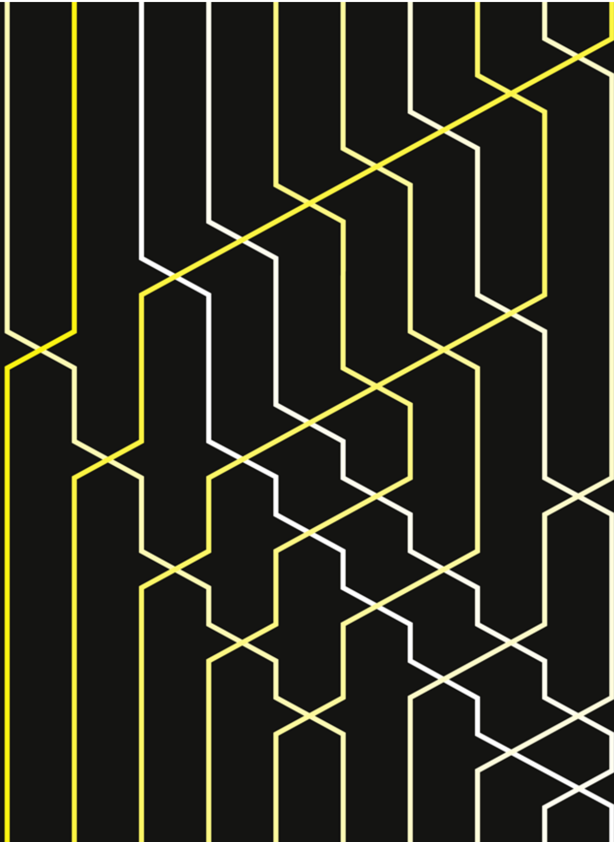


ALGORITHMS TO LIVE BY

THE COMPUTER SCIENCE OF HUMAN DECISIONS

BRIAN CHRISTIAN
& TOM GRIFFITHS

'Practical and highly enjoyable'
POPULAR SCIENCE

Brian Christian
Griffiths Griffiths

Algorithms to Live By: The Computer Science of Human Decisions

Аннотация

A fascinating exploration of how computer algorithms can be applied to our everyday lives.In this dazzlingly interdisciplinary work, acclaimed author Brian Christian and cognitive scientist Tom Griffiths show us how the simple, precise algorithms used by computers can also untangle very human questions. Modern life is constrained by limited space and time, limits that give rise to a particular set of problems. What should we do, or leave undone, in a day or a lifetime? How much messiness should we accept? The authors explain how to have better hunches and when to leave things to chance, how to deal with overwhelming choices and how best to connect with others.From finding a spouse to finding a parking spot, from organizing one's inbox to understanding the workings of human memory, Algorithms To Live By is full of practical takeaways to help you solve common decision-making problems and illuminate the workings of the human mind.

Algorithms to Live By

The Computer Science of Human Decisions

Brian Christian and

Copyright

William Collins
An imprint of HarperCollins*Publishers*
1 London Bridge Street
London SE1 9GF
www.WilliamCollinsBooks.com

This eBook first published in Great Britain by William Collins in 2016

First published in the United States by Henry Holt and Company, LLC in 2016

A catalogue record for this book is available from the British Library

Cover design by Jonathan Pelham

Source ISBN: 9780007547999
Ebook Edition © April 2016 ISBN: 9780007547982
Version: 2018-09-27

Dedication

For our families

Contents

Introduction

Algorithms to Live By

Imagine you're searching for an apartment in San Francisco—arguably the most harrowing American city in which to do so. The booming tech sector and tight zoning laws limiting new construction have conspired to make the city just as expensive as New York, and by many accounts more competitive. New listings go up and come down within minutes, open houses are mobbed, and often the keys end up in the hands of whoever can physically foist a deposit check on the landlord first.

Such a savage market leaves little room for the kind of fact-finding and deliberation that is theoretically supposed to characterize the doings of the rational consumer. Unlike, say, a mall patron or an online shopper, who can compare options before making a decision, the would-be San Franciscan has to decide instantly either way: you can take the apartment you are currently looking at, forsaking all others, or you can walk away, never to return.

Let's assume for a moment, for the sake of simplicity, that you care only about maximizing your chance of getting the very best apartment available. Your goal is reducing the twin, Scylla-and-Charybdis regrets of the "one that got away" and the "stone left unturned" to the absolute minimum. You run into a dilemma right off the bat: How are you to know that an apartment is indeed the best unless you have a baseline to judge it by? And how are you to establish that baseline unless you look at (and lose) a number of apartments? The more information you gather, the better you'll know the right opportunity when you see it—but the more likely you are to have already passed it by.

So what do you do? How do you make an informed decision when the very act of informing it jeopardizes the outcome? It's a cruel situation, bordering on paradox.

When presented with this kind of problem, most people will intuitively say something to the effect that it requires some sort of balance between looking and leaping—that you must look at enough apartments to establish a standard, then take whatever satisfies the standard you've established. This notion of balance is, in fact, precisely correct. What most people don't say with any certainty is what that balance is. Fortunately, there's an answer.

Thirty-seven percent.

If you want the best odds of getting the best apartment, spend 37% of your apartment hunt (eleven days, if you've given yourself a month for the search) noncommittally exploring options. Leave the checkbook at home; you're just calibrating. But after that point, be prepared to immediately commit—deposit and all—to the very first place you see that beats whatever you've already seen. This is not merely an intuitively satisfying compromise between looking and leaping. It is the provably optimal solution.

We know this because finding an apartment belongs to a class of mathematical problems known as "optimal stopping" problems. The 37% rule defines a simple series of steps—what computer scientists call an "algorithm"—for solving these problems. And as it turns out, apartment hunting is just one of the ways that optimal stopping rears its head in daily life. Committing to or forgoing a succession of options is a structure that appears in life again and again, in slightly different incarnations. How many times to circle the block before pulling into a parking space? How far to push your luck with a risky business venture before cashing out? How long to hold out for a better offer on that house or car?

The same challenge also appears in an even more fraught setting: dating. Optimal stopping is the science of serial monogamy.

Simple algorithms offer solutions not only to an apartment hunt but to all such situations in life where we confront the question of optimal stopping. People grapple with these issues every day—although surely poets have spilled more ink on the tribulations of courtship than of parking—and they do so with, in some cases, considerable anguish. But the anguish is unnecessary. Mathematically, at least, these are solved problems.

Every harried renter, driver, and suitor you see around you as you go through a typical week is essentially reinventing the wheel. They don't need a therapist; they need an algorithm. The therapist tells them to find the right, comfortable balance between impulsivity and overthinking.

The algorithm tells them the balance is thirty-seven percent.

* * *

There is a particular set of problems that all people face, problems that are a direct result of the fact that our lives are carried out in finite space and time. What should we do, and leave undone, in a day or in a decade? What degree of mess should we embrace—and how much order is excessive? What balance between new experiences and favored ones makes for the most fulfilling life?

These might seem like problems unique to humans; they're not. For more than half a century, computer scientists have been grappling with, and in many cases solving, the equivalents of these everyday dilemmas. How should a processor allocate its “attention” to perform all that the user asks of it, with the minimum overhead and in the least amount of time? When should it switch between different tasks, and how many tasks should it take on in the first place? What is the best way for it to use its limited memory resources? Should it collect more data, or take an action based on the data it already has? Seizing the day might be a challenge for humans, but computers all around us are seizing milliseconds with ease. And there’s much we can learn from how they do it.

Talking about algorithms for human lives might seem like an odd juxtaposition. For many people, the word “algorithm” evokes the arcane and inscrutable machinations of big data, big government, and big business: increasingly part of the infrastructure of the modern world, but hardly a source of practical wisdom or guidance for human affairs. But an algorithm is just a finite sequence of steps used to solve a problem, and algorithms are much broader—and older by far—than the computer. Long before algorithms were ever used by machines, they were used by people.

The word “algorithm” comes from the name of Persian mathematician al-Khwārizmī, author of a ninth-century book of techniques for doing mathematics by hand. (His book was called al-Jabr wa’l-Muqābala—and the “al-jabr” of the title in turn provides the source of our word “algebra.”) The earliest known mathematical algorithms, however, predate even al-Khwārizmī's work: a four-thousand-year-old Sumerian clay tablet found near Baghdad describes a scheme for long division.

But algorithms are not confined to mathematics alone. When you cook bread from a recipe, you're following an algorithm. When you knit a sweater from a pattern, you're following an algorithm. When you put a sharp edge on a piece of flint by executing a precise sequence of strikes with the end of an antler —a key step in making fine stone tools—you're following an algorithm. Algorithms have been a part of human technology ever since the Stone Age.

* * *

In this book, we explore the idea of human algorithm design—searching for better solutions to the challenges people encounter every day. Applying the lens of computer science to everyday life has consequences at many scales. Most immediately, it offers us practical, concrete suggestions for how to solve specific problems. Optimal stopping tells us when to look and when to leap. The explore/exploit tradeoff tells us how to find the balance between trying new things and enjoying our favorites. Sorting theory tells us how (and whether) to arrange our offices. Caching theory tells us how to fill our closets. Scheduling theory tells us how to fill our time.

At the next level, computer science gives us a vocabulary for understanding the deeper principles at play in each of these domains. As Carl Sagan put it, "Science is a way of thinking much more than it is a body of knowledge." Even in cases where life is too messy for us to expect a strict numerical analysis or a ready answer, using intuitions and concepts honed on the simpler forms of these problems offers us a way to understand the key issues and make progress.

Most broadly, looking through the lens of computer science can teach us about the nature of the human mind, the meaning of rationality, and the oldest question of all: how to live. Examining cognition as a means of solving the fundamentally computational problems posed by our environment can utterly change the way we think about human rationality.

The notion that studying the inner workings of computers might reveal how to think and decide, what to believe and how to behave, might strike many people as not only wildly reductive, but in fact misguided. Even if computer science did have things to say about how to think and how to act, would we want to listen? We look at the AIs and robots of science fiction, and it seems like theirs is not a life any of us would want to live.

In part, that's because when we think about computers, we think about coldly mechanical, deterministic systems: machines applying rigid deductive logic, making decisions by exhaustively enumerating the options, and grinding out the exact right answer no matter how long and hard they have to think. Indeed, the person who first imagined computers had something essentially like this in mind. Alan Turing defined the very notion of computation by an analogy to a human mathematician who carefully works through the steps of a lengthy calculation, yielding an unmistakably right answer.

So it might come as a surprise that this is not what modern computers are actually doing when they face a difficult problem. Straightforward arithmetic, of course, isn't particularly challenging for a modern computer. Rather, it's tasks like conversing with people, fixing a corrupted file, or winning a game of Go—problems where the rules aren't clear, some of the required information is missing, or finding exactly the right answer would require considering an astronomical number of possibilities—that now pose the biggest challenges in computer science. And the algorithms that researchers have developed to solve the hardest classes of problems have moved computers away from an extreme reliance on exhaustive calculation. Instead, tackling real-world tasks requires being comfortable with chance, trading off time with accuracy, and using approximations.

As computers become better tuned to real-world problems, they provide not only algorithms that people can borrow for their own lives, but a better standard against which to compare human cognition itself. Over the past decade or two, behavioral economics has told a very particular story about human beings: that we are irrational and error-prone, owing in large part to the buggy, idiosyncratic hardware of the brain. This self-deprecating story has become increasingly familiar, but certain questions remain vexing. Why are four-year-olds, for instance, still better than million-dollar supercomputers at a host of cognitive tasks, including vision, language, and causal reasoning?

The solutions to everyday problems that come from computer science tell a different story about the human mind. Life is full of problems that are, quite simply, hard. And the mistakes made by people often say more about the intrinsic difficulties of the problem than about the fallibility of human brains. Thinking algorithmically about the world, learning about the fundamental structures of the problems we face and about the properties of their solutions, can help us see how good we actually are, and better understand the errors that we make.

In fact, human beings turn out to consistently confront some of the hardest cases of the problems studied by computer scientists. Often, people need to make decisions while dealing with uncertainty, time constraints, partial information, and a rapidly changing world. In some of those cases, even cutting-edge computer science has not yet come up with efficient, always-right algorithms. For certain situations it appears that such algorithms might not exist at all.

Even where perfect algorithms haven't been found, however, the battle between generations of computer scientists and the most intractable real-world problems has yielded a series of insights. These hard-won precepts are at odds with our intuitions about rationality, and they don't sound anything like the narrow prescriptions of a mathematician trying to force the world into clean, formal lines. They say: Don't always consider all your options. Don't necessarily go for the outcome that seems best every time. Make a mess on occasion. Travel light. Let things wait. Trust your instincts and don't think too long. Relax. Toss a coin. Forgive, but don't forget. To thine own self be true.

Living by the wisdom of computer science doesn't sound so bad after all. And unlike most advice, it's backed up by proofs.

* * *

Just as designing algorithms for computers was originally a subject that fell into the cracks between disciplines—an odd hybrid of mathematics and engineering—so, too, designing algorithms for humans is a topic that doesn't have a natural disciplinary home. Today, algorithm design draws not only on computer science, math, and engineering but on kindred fields like statistics and operations research. And as we consider how algorithms designed for machines might relate to human minds, we also need to look to cognitive science, psychology, economics, and beyond.

We, your authors, are familiar with this interdisciplinary territory. Brian studied computer science and philosophy before going on to graduate work in English and a career at the intersection of the three. Tom studied psychology and statistics before becoming a professor at UC Berkeley, where he spends most of his time thinking about the relationship between human cognition and computation. But nobody can be an expert in all of the fields that are relevant to designing better algorithms for humans. So as part of our quest for algorithms to live by, we talked to the people who came up with some of the most famous algorithms of the last fifty years. And we asked them, some of the smartest people in the world, how their research influenced the way they approached their own lives—from finding their spouses to sorting their socks.

The next pages begin our journey through some of the biggest challenges faced by computers and human minds alike: how to manage finite space, finite time, limited attention, unknown unknowns, incomplete information, and an unforeseeable future; how to do so with grace and confidence; and how to do so in a community with others who are all simultaneously trying to do the same. We will learn about the fundamental mathematical structure of these challenges and about how computers are engineered—sometimes counter to what we imagine—to make the most of them. And we will learn about how the mind works, about its distinct but deeply related ways of tackling the same set of issues and coping with the same constraints. Ultimately, what we can gain is not only a set of concrete takeaways for the problems around us, not only a new way to see the elegant structures behind even the hairiest human dilemmas, not only a recognition of the travails of humans and computers as deeply conjoined, but something even more profound: a new vocabulary for the world around us, and a chance to learn something truly new about ourselves.

1 Optimal Stopping
When to Stop Looking

Though all Christians start a wedding invitation by solemnly declaring their marriage is due to special Divine arrangement, I, as a philosopher, would like to talk in greater detail about this …

—JOHANNES KEPLER

If you prefer Mr. Martin to every other person; if you think him the most agreeable man you have ever been in company with, why should you hesitate?

—JANE AUSTEN, EMMA

It's such a common phenomenon that college guidance counselors even have a slang term for it: the "turkey drop." High-school sweethearts come home for Thanksgiving of their freshman year of college and, four days later, return to campus single.

An angst-ridden Brian went to his own college guidance counselor his freshman year. His high-school girlfriend had gone to a different college several states away, and they struggled with the distance. They also struggled with a stranger and more philosophical question: how good a relationship did they have? They had no real benchmark of other relationships by which to judge it. Brian's counselor recognized theirs as a classic freshman-year dilemma, and was surprisingly nonchalant in her advice: "Gather data."

The nature of serial monogamy, writ large, is that its practitioners are confronted with a fundamental, unavoidable problem. When have you met enough people to know who your best match is? And what if acquiring the data costs you that very match? It seems the ultimate Catch-22 of the heart.

As we have seen, this Catch-22, this angsty freshman cri de coeur, is what mathematicians call an "optimal stopping" problem, and it may actually have an answer: 37%.

Of course, it all depends on the assumptions you're willing to make about love.

The Secretary Problem

In any optimal stopping problem, the crucial dilemma is not which option to pick, but how many options to even consider. These problems turn out to have implications not only for lovers and renters, but also for drivers, homeowners, burglars, and beyond.

The 37% Rule* derives from optimal stopping's most famous puzzle, which has come to be known as the "secretary problem." Its setup is much like the apartment hunter's dilemma that we considered earlier. Imagine you're interviewing a set of applicants for a position as a secretary, and your goal is to maximize the chance of hiring the single best applicant in the pool. While you have no idea how to assign scores to individual applicants, you can easily judge which one you prefer. (A mathematician might say you have access only to the ordinal numbers—the relative ranks of the applicants compared to each other—but not to the cardinal numbers, their ratings on some kind of general scale.) You interview the applicants in random order, one at a time. You can decide to offer the job to an applicant at any point and they are guaranteed to accept, terminating the search. But if you pass over an applicant, deciding not to hire them, they are gone forever.

The secretary problem is widely considered to have made its first appearance in print—sans explicit mention of secretaries—in the February 1960 issue of Scientific American, as one of several puzzles posed in Martin Gardner's beloved column on recreational mathematics. But the origins of the problem are surprisingly mysterious. Our own initial search yielded little but speculation, before turning into unexpectedly physical detective work: a road trip down to the archive of Gardner's papers at Stanford, to haul out boxes of his midcentury correspondence. Reading paper correspondence is a bit like eavesdropping on someone who's on the phone: you're only hearing one side of the exchange, and must infer the other. In our case, we only had the replies to what was apparently Gardner's own search for the problem's origins fiftysome years ago. The more we read, the more tangled and unclear the story became.

Harvard mathematician Frederick Mosteller recalled hearing about the problem in 1955 from his colleague Andrew Gleason, who had heard about it from somebody else. Leo Moser wrote from the University of Alberta to say that he read about the problem in "some notes" by R. E. Gaskell of Boeing, who himself credited a colleague. Roger Pinkham of Rutgers wrote that he first heard of the problem in 1955 from Duke University mathematician J. Shoenfield, "and I believe he said that he had heard the problem from someone at Michigan."

“Someone at Michigan” was almost certainly someone named Merrill Flood. Though he is largely unheard of outside mathematics, Flood’s influence on computer science is almost impossible to avoid. He’s credited with popularizing the traveling salesman problem (which we discuss in more detail in chapter 8), devising the prisoner’s dilemma (which we discuss in chapter 11), and even with possibly coining the term “software.” It’s Flood who made the first known discovery of the 37% Rule, in 1958, and he claims to have been considering the problem since 1949—but he himself points back to several other mathematicians.

Suffice it to say that wherever it came from, the secretary problem proved to be a near-perfect mathematical puzzle: simple to explain, devilish to solve, succinct in its answer, and intriguing in its implications. As a result, it moved like wildfire through the mathematical circles of the 1950s, spreading by word of mouth, and thanks to Gardner’s column in 1960 came to grip the imagination of the public at large. By the 1980s the problem and its variations had produced so much analysis that it had come to be discussed in papers as a subfield unto itself.

As for secretaries—it’s charming to watch each culture put its own anthropological spin on formal systems. We think of chess, for instance, as medieval European in its imagery, but in fact its origins are in eighth-century India; it was heavy-handedly “Europeanized” in the fifteenth century, as its shahs became kings, its viziers turned to queens, and its elephants became bishops. Likewise, optimal stopping problems have had a number of incarnations, each reflecting the predominating concerns of its time. In the nineteenth century such problems were typified by baroque lotteries and by women choosing male suitors; in the early twentieth century by holidaying motorists searching for hotels and by male suitors choosing women; and in the paper-pushing, male-dominated mid-twentieth century, by male bosses choosing female assistants. The first explicit mention of it by name as the "secretary problem" appears to be in a 1964 paper, and somewhere along the way the name stuck.

Whence 37%?

In your search for a secretary, there are two ways you can fail: stopping early and stopping late. When you stop too early, you leave the best applicant undiscovered. When you stop too late, you hold out for a better applicant who doesn't exist. The optimal strategy will clearly require finding the right balance between the two, walking the tightrope between looking too much and not enough.

If your aim is finding the very best applicant, settling for nothing less, it's clear that as you go through the interview process you shouldn't even consider hiring somebody who isn't the best you've seen so far. However, simply being the best yet isn't enough for an offer; the very first applicant, for example, will of course be the best yet by definition. More generally, it stands to reason that the rate at which we encounter "best yet" applicants will go down as we proceed in our interviews. For instance, the second applicant has a 50/50 chance of being the best we've yet seen, but the fifth applicant only has a 1-in-5 chance of being the best so far, the sixth has a 1-in-6 chance, and so on. As a result, best-yet applicants will become steadily more impressive as the search continues (by definition, again, they're better than all those who came before)—but they will also become more and more infrequent.

Okay, so we know that taking the first best-yet applicant we encounter (a.k.a. the first applicant, period) is rash. If there are a hundred applicants, it also seems hasty to make an offer to the next one who's best-yet, just because she was better than the first. So how do we proceed?

Intuitively, there are a few potential strategies. For instance, making an offer the third time an applicant trumps everyone seen so far—or maybe the fourth time. Or perhaps taking the next best-yet applicant to come along after a long "drought"—a long streak of poor ones.

But as it happens, neither of these relatively sensible strategies comes out on top. Instead, the optimal solution takes the form of what we'll call the Look-Then-Leap Rule: You set a predetermined amount of time for "looking"—that is, exploring your options, gathering data—in which you categorically don't choose anyone, no matter how impressive. After that point, you enter the "leap" phase, prepared to instantly commit to anyone who outshines the best applicant you saw in the look phase.

We can see how the Look-Then-Leap Rule emerges by considering how the secretary problem plays out in the smallest applicant pools. With just one applicant the problem is easy to solve—hire her! With two applicants, you have a 50/50 chance of success no matter what you do. You can hire the first applicant (who'll turn out to be the best half the time), or dismiss the first and by default hire the second (who is also best half the time).

Add a third applicant, and all of a sudden things get interesting. The odds if we hire at random are one-third, or 33%. With two applicants we could do no better than chance; with three, can we? It turns out we can, and it all comes down to what we do with the second interviewee. When we see the first applicant, we have no information—she'll always appear to be the best yet. When we see the third applicant, we have no agency—we have to make an offer to the final applicant, since we've dismissed the others. But when we see the second applicant, we have a little bit of both: we know whether she's better or worse than the first, and we have the freedom to either hire or dismiss her. What happens when we just hire her if she's better than the first applicant, and dismiss her if she's not? This turns out to be the best possible strategy when facing three applicants; using this approach it's possible, surprisingly, to do just as well in the three-applicant problem as with two, choosing the best applicant exactly half the time.*

Enumerating these scenarios for four applicants tells us that we should still begin to leap as soon as the second applicant; with five applicants in the pool, we shouldn't leap before the third.

As the applicant pool grows, the exact place to draw the line between looking and leaping settles to 37% of the pool, yielding the 37% Rule: look at the first 37% of the applicants,* choosing none, then be ready to leap for anyone better than all those you've seen so far.

Number of Applicants	Take the Best Applicant After	Chance of Getting the Best
3	1 (33.33%)	50%
4	1 (25%)	45.83%
5	2 (40%)	43.33%
6	2 (33.33%)	42.78%
7	2 (28.57%)	41.43%
8	3 (37.5%)	40.98%
9	3 (33.33%)	40.59%
10	3 (30%)	39.87%
20	7 (35%)	38.42%
30	11 (36.67%)	37.86%
40	15 (37.5%)	37.57%
50	18 (36%)	37.43%
100	37 (37%)	37.10%
1000	369 (36.9%)	36.81%

How to optimally choose a secretary.

As it turns out, following this optimal strategy ultimately gives us a 37% chance of hiring the best applicant; it's one of the problem's curious mathematical symmetries that the strategy itself and its chance of success work out to the very same number. The table above shows the optimal strategy for the secretary problem with different numbers of applicants, demonstrating how the chance of success—like the point to switch from looking to leaping—converges on 37% as the number of applicants increases.

A 63% failure rate, when following the best possible strategy, is a sobering fact. Even when we act optimally in the secretary problem, we will still fail most of the time—that is, we won't end up with the single best applicant in the pool. This is bad news for those of us who would frame romance as a search for "the one." But here's the silver lining. Intuition would suggest that our chances of picking the single best applicant should steadily decrease as the applicant pool grows. If we were hiring at random, for instance, then in a pool of a hundred applicants we'd have a 1% chance of success, and in a pool of a million applicants we'd have a 0.0001% chance. Yet remarkably, the math of the secretary problem doesn't change. If you're stopping optimally, your chance of finding the single best applicant in a pool of a hundred is 37%. And in a pool of a million, believe it or not, your chance is still 37%. Thus the bigger the applicant pool gets, the more valuable knowing the optimal algorithm becomes. It's true that you're unlikely to find the needle the majority of the time, but optimal stopping is your best defense against the haystack, no matter how large.

Lover's Leap

The passion between the sexes has appeared in every age to be so nearly the same that it may always be considered, in algebraic language, as a given quantity.

—THOMAS MALTHUS

I married the first man I ever kissed. When I tell this to my children they just about throw up.

—BARBARA BUSH

Before he became a professor of operations research at Carnegie Mellon, Michael Trick was a graduate student, looking for love. "It hit me that the problem has been studied: it is the Secretary Problem! I had a position to fill [and] a series of applicants, and my goal was to pick the best applicant for the position." So he ran the numbers. He didn't know how many women he could expect to meet in his lifetime, but there's a certain flexibility in the 37% Rule: it can be applied to either the number of applicants or the time over which one is searching. Assuming that his search would run from ages eighteen to forty, the 37% Rule gave age 26.1 years as the point at which to switch from looking to leaping. A number that, as it happened, was exactly Trick's age at the time. So when he found a woman who was a better match than all those he had dated so far, he knew exactly what to do. He leapt. "I didn't know if she was Perfect (the assumptions of the model don't allow me to determine that), but there was no doubt that she met the qualifications for this step of the algorithm. So I proposed," he writes.

"And she turned me down."

Mathematicians have been having trouble with love since at least the seventeenth century. The legendary astronomer Johannes Kepler is today perhaps best remembered for discovering that planetary orbits are elliptical and for being a crucial part of the "Copernican Revolution" that included Galileo and Newton and upended humanity's sense of its place in the heavens. But Kepler had terrestrial concerns, too. After the death of his first wife in 1611, Kepler embarked on a long and arduous quest to remarry, ultimately courting a total of eleven women. Of the first four, Kepler liked the fourth the best ("because of her tall build and athletic body") but did not cease his search. "It would have been settled," Kepler wrote, "had not both love and reason forced a fifth woman on me. This one won me over with love, humble loyalty, economy of household, diligence, and the love she gave the stepchildren."

"However," he wrote, "I continued."

Kepler's friends and relations went on making introductions for him, and he kept on looking, but halfheartedly. His thoughts remained with number five. After eleven courtships in total, he decided he would search no further. "While preparing to travel to Regensburg, I returned to the fifth woman, declared myself, and was accepted." Kepler and Susanna Reuttinger were wed and had six children together, along with the children from Kepler's first marriage. Biographies describe the rest of Kepler's domestic life as a particularly peaceful and joyous time.

Both Kepler and Trick—in opposite ways—experienced firsthand some of the ways that the secretary problem oversimplifies the search for love. In the classical secretary problem, applicants always accept the position, preventing the rejection experienced by Trick. And they cannot be "recalled" once passed over, contrary to the strategy followed by Kepler.

In the decades since the secretary problem was first introduced, a wide range of variants on the scenario have been studied, with strategies for optimal stopping worked out under a number of different conditions. The possibility of rejection, for instance, has a straightforward mathematical solution: propose early and often. If you have, say, a 50/50 chance of being rejected, then the same kind of mathematical analysis that yielded the 37% Rule says you should start making offers after just a quarter of your search. If turned down, keep making offers to every best-yet person you see until somebody accepts. With such a strategy, your chance of overall success—that is, proposing and being accepted by the best applicant in the pool—will also be 25%. Not such terrible odds, perhaps, for a scenario that combines the obstacle of rejection with the general difficulty of establishing one's standards in the first place.

Kepler, for his part, decried the "restlessness and doubtfulness" that pushed him to keep on searching. "Was there no other way for my uneasy heart to be content with its fate," he bemoaned in a letter to a confidante, "than by realizing the impossibility of the fulfillment of so many other desires?" Here, again, optimal stopping theory provides some measure of consolation. Rather than being signs of moral or psychological degeneracy, restlessness and doubtfulness actually turn out to be part of the best strategy for scenarios where second chances are possible. If you can recall previous applicants, the optimal algorithm puts a twist on the familiar Look-Then-Leap Rule: a longer noncommittal period, and a fallback plan.

For example, assume an immediate proposal is a sure thing but belated proposals are rejected half the time. Then the math says you should keep looking noncommittally until you've seen 61% of applicants, and then only leap if someone in the remaining 39% of the pool proves to be the best yet. If you're still single after considering all the possibilities—as Kepler was—then go back to the best one that got away. The symmetry between strategy and outcome holds in this case once again, with your chances of ending up with the best applicant under this second-chances-allowed scenario also being 61%.

For Kepler, the difference between reality and the classical secretary problem brought with it a happy ending. In fact, the twist on the classical problem worked out well for Trick, too. After the rejection, he completed his degree and took a job in Germany. There, he "walked into a bar, fell in love with a beautiful woman, moved in together three weeks later, [and] invited her to live in the United States 'for a while.'" She agreed—and six years later, they were wed.

Knowing a Good Thing When You See It: Full Information

The first set of variants we considered—rejection and recall—altered the classical secretary problem's assumptions that timely proposals are always accepted, and tardy proposals, never. For these variants, the best approach remained the same as in the original: look noncommittally for a time, then be ready to leap.

But there's an even more fundamental assumption of the secretary problem that we might call into question. Namely, in the secretary problem we know nothing about the applicants other than how they compare to one another. We don't have an objective or preexisting sense of what makes for a good or a bad applicant; moreover, when we compare two of them, we know which of the two is better, but not by how much. It's this fact that gives rise to the unavoidable "look" phase, in which we risk passing up a superb early applicant while we calibrate our expectations and standards. Mathematicians refer to this genre of optimal stopping problems as "no-information games."

This setup is arguably a far cry from most searches for an apartment, a partner, or even a secretary. Imagine instead that we had some kind of objective criterion—if every secretary, for instance, had taken a typing exam scored by percentile, in the fashion of the SAT or GRE or LSAT. That is, every applicant's score will tell us where they fall among all the typists who took the test: a 51st-percentile typist is just above average, a 75thpercentile typist is better than three test takers out of four, and so on.

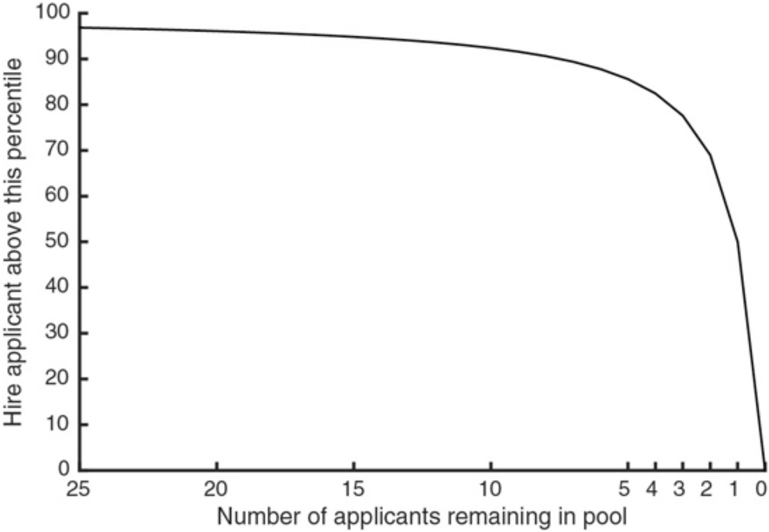
Suppose that our applicant pool is representative of the population at large and isn't skewed or self-selected in any way. Furthermore, suppose we decide that typing speed is the only thing that matters about our applicants. Then we have what mathematicians call "full information," and everything changes. "No buildup of experience is needed to set a standard," as the seminal 1966 paper on the problem put it, "and a profitable choice can sometimes be made immediately." In other words, if a 95th-percentile applicant happens to be the first one we evaluate, we know it instantly and can confidently hire her on the spot—that is, of course, assuming we don't think there's a 96th-percentile applicant in the pool.

And there's the rub. If our goal is, again, to get the single best person for the job, we still need to weigh the likelihood that there's a stronger applicant out there. However, the fact that we have full information gives us everything we need to calculate those odds directly. The chance that our next applicant is in the 96th percentile or higher will always be 1 in 20, for instance. Thus the decision of whether to stop comes down entirely to how many applicants we have left to see. Full information means that we don't need to look before we leap. We can instead use the Threshold Rule, where we immediately accept an applicant if she is above a certain percentile. We don't need to look at an initial group of candidates to set this threshold—but we do, however, need to be keenly aware of how much looking remains available.

The math shows that when there are a lot of applicants left in the pool, you should pass up even a very good applicant in the hopes of finding someone still better than that—but as your options dwindle, you should be prepared to hire anyone who's simply better than average. It's a familiar, if not exactly inspiring, message: in the face of slim pickings, lower your standards. It also makes clear the converse: with more fish in the sea, raise them. In both cases, crucially, the math tells you exactly by how much.

The easiest way to understand the numbers for this scenario is to start at the end and think backward. If you're down to the last applicant, of course, you are necessarily forced to choose her. But when looking at the next-to-last applicant, the question becomes: is she above the 50th percentile? If yes, then hire her; if not, it's worth rolling the dice on the last applicant instead, since her odds of being above the 50th percentile are 50/50 by definition. Likewise, you should choose the third-to-last applicant if she's above the 69th percentile, the fourth-to-last applicant if she's above the 78th, and so on, being more choosy the more applicants are left. No matter what, never hire someone who's below average unless you're totally out of options. (And since you're still interested only in finding the very best person in the applicant pool, never hire someone who isn't the best you've seen so far.)

The chance of ending up with the single best applicant in this full-information version of the secretary problem comes to 58%—still far from a guarantee, but considerably better than the 37% success rate offered by the 37% Rule in the no-information game. If you have all the facts, you can succeed more often than not, even as the applicant pool grows arbitrarily large.



Optimal stopping thresholds in the full-information secretary problem.

The full-information game thus offers an unexpected and somewhat bizarre takeaway. Gold digging is more likely to succeed than a quest for love. If you're evaluating your partners based on any kind of objective criterion—say, their income percentile—then you've got a lot more information at your disposal than if you're after a nebulous emotional response ("love") that might require both experience and comparison to calibrate.

Of course, there's no reason that net worth—or, for that matter, typing speed—needs to be the thing that you're measuring. Any yardstick that provides full information on where an applicant stands relative to the population at large will change the solution from the Look-Then-Leap Rule to the Threshold Rule and will dramatically boost your chances of finding the single best applicant in the group.

There are many more variants of the secretary problem that modify its other assumptions, perhaps bringing it more in line with the real-world challenges of finding love (or a secretary). But the lessons to be learned from optimal stopping aren't limited to dating or hiring. In fact, trying to make the best choice when options only present themselves one by one is also the basic structure of selling a house, parking a car, and quitting when you're ahead. And they're all, to some degree or other, solved problems.

When to Sell

If we alter two more aspects of the classical secretary problem, we find ourselves catapulted from the realm of dating to the realm of real estate. Earlier we talked about the process of renting an apartment as an optimal stopping problem, but owning a home has no shortage of optimal stopping either.

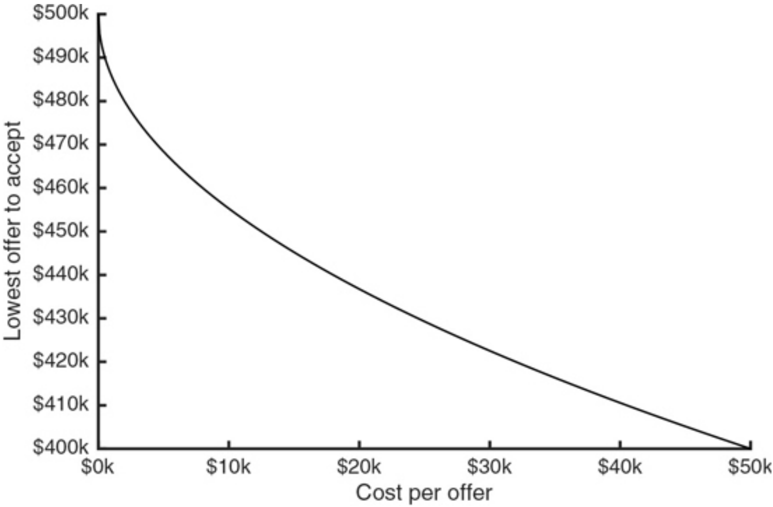
Imagine selling a house, for instance. After consulting with several real estate agents, you put your place on the market; a new coat of paint, some landscaping, and then it's just a matter of waiting for the offers to come in. As each offer arrives, you typically have to decide whether to accept it or turn it down. But turning down an offer comes at a cost—another week (or month) of mortgage payments while you wait for the next offer, which isn't guaranteed to be any better.

Selling a house is similar to the full-information game. We know the objective dollar value of the offers, telling us not only which ones are better than which, but also by how much. What's more, we have information about the broader state of the market, which enables us to at least roughly predict the range of offers to expect. (This gives us the same "percentile" information about each offer that we had with the typing exam above.) The difference here, however, is that our goal isn't actually to secure the single best offer—it's to make the most money through the process overall. Given that waiting has a cost measured in dollars, a good offer today beats a slightly better one several months from now.

Having this information, we don't need to look noncommittally to set a threshold. Instead, we can set one going in, ignore everything below it, and take the first option to exceed it. Granted, if we have a limited amount of savings that will run out if we don't sell by a certain time, or if we expect to get only a limited number of offers and no more interest thereafter, then we should lower our standards as such limits approach. (There's a reason why home buyers look for "motivated" sellers.) But if neither concern leads us to believe that our backs are against the wall, then we can simply focus on a cost-benefit analysis of the waiting game.

Here we'll analyze one of the simplest cases: where we know for certain the price range in which offers will come, and where all offers within that range are equally likely. If we don't have to worry about the offers (or our savings) running out, then we can think purely in terms of what we can expect to gain or lose by waiting for a better deal. If we decline the current offer, will the chance of a better one, multiplied by how much better we expect it to be, more than compensate for the cost of the wait? As it turns out, the math here is quite clean, giving us an explicit function for stopping price as a function of the cost of waiting for an offer.

This particular mathematical result doesn't care whether you're selling a mansion worth millions or a ramshackle shed. The only thing it cares about is the difference between the highest and lowest offers you're likely to receive. By plugging in some concrete figures, we can see how this algorithm offers us a considerable amount of explicit guidance. For instance, let's say the range of offers we're expecting runs from $400,000 to $500,000. First, if the cost of waiting is trivial, we're able to be almost infinitely choosy. If the cost of getting another offer is only a dollar, we'll maximize our earnings by waiting for someone willing to offer us $499,552.79 and not a dime less. If waiting costs $2,000 an offer, we should hold out for an even $480,000. In a slow market where waiting costs $10,000 an offer, we should take anything over $455,279. Finally, if waiting costs half or more of our expected range of offers—in this case, $50,000—then there's no advantage whatsoever to holding out; we'll do best by taking the very first offer that comes along and calling it done. Beggars can't be choosers.



Optimal stopping thresholds in the house-selling problem.

The critical thing to note in this problem is that our threshold depends only on the cost of search. Since the chances of the next offer being a good one—and the cost of finding out—never change, our stopping price has no reason to ever get lower as the search goes on, regardless of our luck. We set it once, before we even begin, and then we quite simply hold fast.

The University of Wisconsin–Madison's Laura Albert McLay, an optimization expert, recalls turning to her knowledge of optimal stopping problems when it came time to sell her own house. "The first offer we got was great," she explains, "but it had this huge cost because they wanted us to move out a month before we were ready. There was another competitive offer … [but] we just kind of held out until we got the right one." For many sellers, turning down a good offer or two can be a nerve-racking proposition, especially if the ones that immediately follow are no better. But McLay held her ground and stayed cool. "That would have been really, really hard," she admits, "if I didn't know the math was on my side."

This principle applies to any situation where you get a series of offers and pay a cost to seek or wait for the next. As a consequence, it's relevant to cases that go far beyond selling a house. For example, economists have used this algorithm to model how people look for jobs, where it handily explains the otherwise seemingly paradoxical fact of unemployed workers and unfilled vacancies existing at the same time.

In fact, these variations on the optimal stopping problem have another, even more surprising property. As we saw, the ability to "recall" a past opportunity was vital in Kepler's quest for love. But in house selling and job hunting, even if it's possible to reconsider an earlier offer, and even if that offer is guaranteed to still be on the table, you should nonetheless never do so. If it wasn't above your threshold then, it won't be above your threshold now. What you've paid to keep searching is a sunk cost. Don't compromise, don't second-guess. And don't look back.

When to Park

I find that the three major administrative problems on a campus are sex for the students, athletics for the alumni, and parking for the faculty.

—CLARK KERR, PRESIDENT OF UC BERKELEY, 1958–1967

Another domain where optimal stopping problems abound—and where looking back is also generally ill-advised—is the car. Motorists feature in some of the earliest literature on the secretary problem, and the framework of constant forward motion makes almost every car-trip decision into a stopping problem: the search for a restaurant; the search for a bathroom; and, most acutely for urban drivers, the search for a parking space. Who better to talk to about the ins and outs of parking than the man described by the Los Angeles Times as "the parking rock star," UCLA Distinguished Professor of Urban Planning Donald Shoup? We drove down from Northern California to visit him, reassuring Shoup that we'd be leaving plenty of time for unexpected traffic. "As for planning on 'unexpected traffic,' I think you should plan on expected traffic," he replied. Shoup is perhaps best known for his book The High Cost of Free

Parking, and he has done much to advance the discussion and understanding of what really happens when someone drives to their destination.

We should pity the poor driver. The ideal parking space, as Shoup models it, is one that optimizes a precise balance between the "sticker price" of the space, the time and inconvenience of walking, the time taken seeking the space (which varies wildly with destination, time of day, etc.), and the gas burned in doing so. The equation changes with the number of passengers in the car, who can split the monetary cost of a space but not the search time or the walk. At the same time, the driver needs to consider that the area with the most parking supply may also be the area with the most demand; parking has a game-theoretic component, as you try to outsmart the other drivers on the road while they in turn are trying to outsmart you.* That said, many of the challenges of parking boil down to a single number: the occupancy rate. This is the proportion of all parking spots that are currently occupied. If the occupancy rate is low, it's easy to find a good parking spot. If it's high, finding anywhere at all to park is a challenge.

Shoup argues that many of the headaches of parking are consequences of cities adopting policies that result in extremely high occupancy rates. If the cost of parking in a particular location is too low (or—horrors!—nothing at all), then there is a high incentive to park there, rather than to park a little farther away and walk. So everybody tries to park there, but most of them find the spaces are already full, and people end up wasting time and burning fossil fuel as they cruise for a spot.

Shoup's solution involves installing digital parking meters that are capable of adaptive prices that rise with demand. (This has now been implemented in downtown San Francisco.) The prices are set with a target occupancy rate in mind, and Shoup argues that this rate should be somewhere around 85%—a radical drop from the nearly 100%-packed curbs of most major cities. As he notes, when occupancy goes from 90% to 95%, it accommodates only 5% more cars but doubles the length of everyone's search.

The key impact that occupancy rate has on parking strategy becomes clear once we recognize that parking is an optimal stopping problem. As you drive along the street, every time you see the occasional empty spot you have to make a decision: should you take this spot, or go a little closer to your destination and try your luck?

Assume you're on an infinitely long road, with parking spots evenly spaced, and your goal is to minimize the distance you end up walking to your destination. Then the solution is the Look-Then-Leap Rule. The optimally stopping driver should pass up all vacant spots occurring more than a certain distance from the destination and then take the first space that appears thereafter. And the distance at which to switch from looking to leaping depends on the proportion of spots that are likely to be filled—the occupancy rate. The table on the next page gives the distances for some representative proportions.

With this occupancy rate (%)	Wait until this many spaces away, then take the next free spot
0	0
50	1
75	3
80	4
85	5
90	7
95	14
96	17
97	23
98	35
99	69
99.9	693

How to optimally find parking.

If this infinite street has a big-city occupancy rate of 99%, with just 1% of spots vacant, then you should take the first spot you see starting at almost 70 spots—more than a quarter mile—from your destination. But if Shoup has his way and occupancy rates drop to just 85%, you don't need to start seriously looking until you're half a block away.

Most of us don't drive on perfectly straight, infinitely long roads. So as with other optimal stopping problems, researchers have considered a variety of tweaks to this basic scenario. For instance, they have studied the optimal parking strategy for cases where the driver can make U-turns, where fewer parking spaces are available the closer one gets to the destination, and where the driver is in competition against rival drivers also heading to the same destination. But whatever the exact parameters of the problem, more vacant spots are always going to make life easier. It's something of a policy reminder to municipal governments: parking is not as simple as having a resource (spots) and maximizing its utilization (occupancy). Parking is also a process—an optimal stopping problem—and it's one that consumes attention, time, and fuel, and generates both pollution and congestion. The right policy addresses the whole problem. And, counterintuitively, empty spots on highly desirable blocks can be the sign that things are working correctly.

We asked Shoup if his research allows him to optimize his own commute, through the Los Angeles traffic to his office at UCLA. Does arguably the world's top expert on parking have some kind of secret weapon?

He does: "I ride my bike."

When to Quit

In 1997, Forbes magazine identified Boris Berezovsky as the richest man in Russia, with a fortune of roughly $3 billion. Just ten years earlier he had been living on a mathematician's salary from the USSR Academy of Sciences. He made his billions by drawing on industrial relationships he'd formed through his research to found a company that facilitated interaction between foreign carmakers and the Soviet car manufacturer AvtoVAZ. Berezovky's company then became a large-scale dealer for the cars that AvtoVAZ produced, using a payment installment scheme to take advantage of hyperinflation in the ruble. Using the funds from this partnership he bought partial ownership of AvtoVAZ itself, then the ORT Television network, and finally the Sibneft oil company. Becoming one of a new class of oligarchs, he participated in politics, supporting Boris Yeltsin's re-election in 1996 and the choice of Vladimir Putin as his successor in 1999.

But that's when Berezovsky's luck turned. Shortly after Putin's election, Berezovsky publicly objected to proposed constitutional reforms that would expand the power of the president. His continued public criticism of Putin led to the deterioration of their relationship. In October 2000, when Putin was asked about Berezovsky's criticisms, he replied, "The state has a cudgel in its hands that you use to hit just once, but on the head. We haven't used this cudgel yet.… The day we get really angry, we won't hesitate." Berezovsky left Russia permanently the next month, taking up exile in England, where he continued to criticize Putin's regime.

How did Berezovsky decide it was time to leave Russia? Is there a way, perhaps, to think mathematically about the advice to "quit while you're ahead"? Berezovsky in particular might have considered this very question himself, since the topic he had worked on all those years ago as a mathematician was none other than optimal stopping; he authored the first (and, so far, the only) book entirely devoted to the secretary problem.

The problem of quitting while you're ahead has been analyzed under several different guises, but perhaps the most appropriate to Berezovsky's case—with apologies to Russian oligarchs—is known as the "burglar problem." In this problem, a burglar has the opportunity to carry out a sequence of robberies. Each robbery provides some reward, and there's a chance of getting away with it each time. But if the burglar is caught, he gets arrested and loses all his accumulated gains. What algorithm should he follow to maximize his expected take?

The fact that this problem has a solution is bad news for heist movie screenplays: when the team is trying to lure the old burglar out of retirement for one last job, the canny thief need only crunch the numbers. Moreover, the results are pretty intuitive: the number of robberies you should carry out is roughly equal to the chance you get away, divided by the chance you get caught. If you're a skilled burglar and have a 90% chance of pulling off each robbery (and a 10% chance of losing it all), then retire after 90/10 = 9 robberies. A ham-fisted amateur with a 50/50 chance of success? The first time you have nothing to lose, but don't push your luck more than once.

Despite his expertise in optimal stopping, Berezovsky's story ends sadly. He died in March 2013, found by a bodyguard in the locked bathroom of his house in Berkshire with a ligature around his neck. The official conclusion of a postmortem examination was that he had committed suicide, hanging himself after losing much of his wealth through a series of high-profile legal cases involving his enemies in Russia. Perhaps he should have stopped sooner—amassing just a few tens of millions of dollars, say, and not getting into politics. But, alas, that was not his style. One of his mathematician friends, Leonid Boguslavsky, told a story about Berezovsky from when they were both young researchers: on a water-skiing trip to a lake near Moscow, the boat they had planned to use broke down. Here's how David Hoffman tells it in his book The Oligarchs:

While their friends went to the beach and lit a bonfire, Boguslavsky and Berezovsky headed to the dock to try to repair the motor.… Three hours later, they had taken apart and reassembled the motor. It was still dead. They had missed most of the party, yet Berezovsky insisted they had to keep trying. "We tried this and that," Boguslavsky recalled. Berezovsky would not give up.

Surprisingly, not giving up—ever—also makes an appearance in the optimal stopping literature. It might not seem like it from the wide range of problems we have discussed, but there are sequential decision-making problems for which there is no optimal stopping rule. A simple example is the game of "triple or nothing." Imagine you have $1.00, and can play the following game as many times as you want: bet all your money, and have a 50% chance of receiving triple the amount and a 50% chance of losing your entire stake. How many times should you play? Despite its simplicity, there is no optimal stopping rule for this problem, since each time you play, your average gains are a little higher. Starting with \$1.00, you will get \$3.00 half the time and \$0.00 half the time, so on average you expect to end the first round with \$1.50 in your pocket. Then, if you were lucky in the first round, the two possibilities from the \$3.00 you've just won are \$9.00 and \$0.00—for an average return of \$4.50 from the second bet. The math shows that you should always keep playing. But if you follow this strategy, you will eventually lose everything. Some problems are better avoided than solved.

Always Be Stopping

I expect to pass through this world but once. Any good therefore that I can do, or any kindness that I can show to any fellow creature, let me do it now. Let me not defer or neglect it, for I shall not pass this way again.

—STEPHEN GRELLET

Spend the afternoon. You can't take it with you.

—ANNIE DILLARD

We've looked at specific cases of people confronting stopping problems in their lives, and it's clear that most of us encounter these kinds of problems, in one form or another, daily. Whether it involves secretaries, fiancé(e)s, or apartments, life is full of optimal stopping. So the irresistible question is whether—by evolution or education or intuition—we actually do follow the best strategies.

At first glance, the answer is no. About a dozen studies have produced the same result: people tend to stop early, leaving better applicants unseen. To get a better sense for these findings, we talked to UC Riverside's Amnon Rapoport, who has been running optimal stopping experiments in the laboratory for more than forty years.

The study that most closely follows the classical secretary problem was run in the 1990s by Rapoport and his collaborator Darryl Seale. In this study people went through numerous repetitions of the secretary problem, with either 40 or 80 applicants each time. The overall rate at which people found the best possible applicant was pretty good: about 31%, not far from the optimal 37%. Most people acted in a way that was consistent with the Look-Then-Leap Rule, but they leapt sooner than they should have more than four-fifths of the time.

Rapoport told us that he keeps this in mind when solving optimal stopping problems in his own life. In searching for an apartment, for instance, he fights his own urge to commit quickly. "Despite the fact that by nature I am very impatient and I want to take the first apartment, I try to control myself!"

But that impatience suggests another consideration that isn't taken into account in the classical secretary problem: the role of time. After all, the whole time you're searching for a secretary, you don't have a secretary. What's more, you're spending the day conducting interviews instead of getting your own work done.

This type of cost offers a potential explanation for why people stop early when solving a secretary problem in the lab. Seale and Rapoport showed that if the cost of seeing each applicant is imagined to be, for instance, 1% of the value of finding the best secretary, then the optimal strategy would perfectly align with where people actually switched from looking to leaping in their experiment.

The mystery is that in Seale and Rapoport's study, there wasn't a cost for search. So why might people in the laboratory be acting like there was one?

Because for people there's always a time cost. It doesn't come from the design of the experiment. It comes from people's lives.

The "endogenous" time costs of searching, which aren't usually captured by optimal stopping models, might thus provide an explanation for why human decision-making routinely diverges from the prescriptions of those models. As optimal stopping researcher Neil Bearden puts it, "After searching for a while, we humans just tend to get bored. It's not irrational to get bored, but it's hard to model that rigorously."

But this doesn't make optimal stopping problems less important; it actually makes them more important, because the flow of time turns all decision-making into optimal stopping.

"The theory of optimal stopping is concerned with the problem of choosing a time to take a given action," opens the definitive textbook on optimal stopping, and it's hard to think of a more concise description of the human condition. We decide the right time to buy stocks and the right time to sell them, sure; but also the right time to open the bottle of wine we've been keeping around for a special occasion, the right moment to interrupt someone, the right moment to kiss them.

Viewed this way, the secretary problem's most fundamental yet most unbelievable assumption—its strict seriality, its inexorable one-way march—is revealed to be the nature of time itself. As such, the explicit premise of the optimal stopping problem is the implicit premise of what it is to be alive. It's this that forces us to decide based on possibilities we've not yet seen, this that forces us to embrace high rates of failure even when acting optimally. No choice recurs. We may get similar choices again, but never that exact one. Hesitation—inaction—is just as irrevocable as action. What the motorist, locked on the one-way road, is to space, we are to the fourth dimension: we truly pass this way but once.

Intuitively, we think that rational decision-making means exhaustively enumerating our options, weighing each one carefully, and then selecting the best. But in practice, when the clock—or the ticker—is ticking, few aspects of decision-making (or of thinking more generally) are as important as one: when to stop.

*We use boldface to indicate the algorithms that appear throughout the book.

*With this strategy we have a 33% risk of dismissing the best applicant and a 16% risk of never meeting her. To elaborate, there are exactly six possible orderings of the three applicants:

1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2, and 3-2-1. The strategy of looking at the first applicant and then leaping for whoever surpasses her will succeed in three of the six cases (2-1-3, 2-3-1, 3-1-2) and will fail in the other three—twice by being overly choosy (1-2-3, 1-3-2) and once by not being choosy enough (3-2-1).

*Just a hair under 37%, actually. To be precise, the mathematically optimal proportion of applicants to look at is 1/e—the same mathematical constant e, equivalent to 2.71828…, that shows up in calculations of compound interest. But you don't need to worry about knowing e to twelve decimal places: anything between 35% and 40% provides a success rate extremely close to the maximum. For more of the mathematical details, see the notes at the end of the book.

*More on the computational perils of game theory in chapter 11.

Конец ознакомительного фрагмента.