RS #161: Tom Griffiths and Brian Christian on, “Algorithms to live by”

Julia: Welcome to Rationally Speaking, the podcast where we explore the borderlands between reason and nonsense.

I'm your host, Julia Galef, and with me today we have two guests, Tom Griffiths, and Brian Christian. Tom is a returning guest to Rationally Speaking. He joined us a few months ago to talk about whether the brain is secretly rational, and when he's not appearing on Rationally Speaking, you can find him being a professor of Psychology and Cognitive Science at UC Berkeley, where he also directs the Computational Cognitive Science Lab.

Brian is an author who writes about Computer Science, Cognitive Science, and other related topics for publications like the Atlantic and Wired and The New Yorker. He's also the author of the best-selling book, *The Most Human Human*, which is pretty cool and you should check out as well.

Most recently, Tom and Brian have jointly published a book, called *Algorithms to Live By*. It's about some of the most crucial, integral algorithms that are used in Computer Science, and how those algorithms actually apply or can be applied to improve our human decision-making in our everyday life. Decisions like "How should I plan my day?" or "Should I quit my job?" That kind of thing.

That's what we're going to talk about today. Brian, welcome to the show, and Tom, welcome back.

Brian: Thanks so much.

Tom: Thank you.

Julia: This topic is something I've been thinking about for years. In fact I just gave an interview a few months ago to *Vice* Magazine in which I talked about a lot of similar material actually, although I hadn't read your book at that point.

I was talking about how algorithms like Bayes' Rule or understanding the trade-off between exploring and exploiting can help guide our decision-making and improve it. I thought I was being so nuanced and intelligent, and then the article came out and the headline was something like "Julia Galef wants humans to be more like robots!"

So, I've found this to be a difficult topic to communicate about publicly. I'm wondering if you guys have had any reaction like that to your book.

Brian: Yeah this is one of the things that we try to head off in the introduction of the book, we have a section where we step out of frame. Once we've teed up the idea that there's something to be learned at a daily human level from thinking about the problems we face in Computer Science terms, we then immediately back off and say "In case you are skeptical at this point and think that we're trying to advocate that we all turn into these robotic, Vulcan-like beings, no, in fact that is not what we're getting at here."

I think one of the most interesting themes of the book is that in fact, you can make a pretty powerful case grounded in Computer Science, that advocates for things that look at lot more like human intuition and not over-thinking things and being a little bit messy on occasion and trusting our instincts. There's a powerful opportunity to reaffirm some of the things that we either take for granted, or sometimes get a bad rap about the human side of reasoning. The two are not as divergent as I think people think.

Julia: The idea is also, I presume, that familiarizing oneself with these algorithms -- or holding them up against your intuitive decision-making and noticing differences -- that process doesn't just take you right back to the default intuitive strategies you were using, right? There's some difference?

Tom: I think one way of characterizing that is that a lot of these algorithms are things that work really well in very precise situations. If you're in a situation which exactly satisfies the assumptions that go into the algorithm, then that is exactly the right thing, and that's the thing that you should be able to do. There are cases where people are put in those situations, and then the thing they do isn't the right thing.

An example is one of the places that we really start in the book is the 37 percent rule. This is a strategy that you can use in any situation where you basically face a sequence of options -- for example, if you're trying to find an apartment. You might be going to open houses, and you have to make a decision on the spot if you're in the Bay Area about whether you are going to give a check to the landlord and make an offer on that place. Basically if you leave the open house without having done that, you're going to lose it.

You have a sequence of options. You have the chance to make an offer. If you don't make an offer, you lose it. Really what you have to be doing is to be building up a sense of ... how good are your options here? What are the options? At the same time as dealing with the cost of losing some of those options as you're gathering that information.

The right thing to do in that situation is you take 37 percent of the pool of options, or 37 percent of the time you are going to spend looking. If you're, say, looking for an apartment for 30 days, that's 11 days, you spend that time just gathering information. You leave your checkbook at home. You're just getting calibrated. Then after that, you make an offer on the first place you see that's better than any place you've seen so far.

If you're in that precise situation, that's exactly the right thing to do.

Julia: Before you talk about how to adapt the algorithm to match your real life situations, can you give some intuition for why 37 percent? That number seems weirdly ... I'm imagine that number must seem really specific to people who haven't read the explanation. Or random.

Tom: It's even more weirdly specific than that.

It's actually, it's 1 over e. Where e is Euler's constant, the thing that shows up when you're doing things like computing compound interest. That just basically falls out of the math, but a way to get an intuition for it is that ... what you're doing is trying to find the right trade-off between having information that can inform your action, and having enough room to take meaningful action. A way to get insight into this is, you can think about a case where, say you are only going to go to 3 open houses. You are trying to find the best place on the basis of that. One thing you could do is just say "I'll just make an offer on the first place. I'll just choose one at random essentially, and make an offer on that place." The probability that you get the very best place is 1 out of 3.

It turns out you can do better than 1 out of 3. If you go and look at the first place and don't make an offer, and then make an offer on the second place if it's better than the first place, otherwise make an offer on the third place. That is actually the thing that maximizes your chance of getting the best place, and it increases that probability to 50 percent.

Julia: Right.

Tom: If you keep on doing the math where you say "Now let's work out what happens with 4 places, and 5 places, and 6 places, and 7 places, and 8 places, so on ... where that threshold between looking and leaping lies. As the numbers go up to infinity, that threshold converges to 1 over e.

Julia: Cool. You were going to explain, "How does this hold up in real world situations? What are the options that might not be true in the real world?"

Tom: One way in which we ... what happens if you put human beings in this situation and you ask them to make a decision is, we characteristically tend to switch to early. We don't spend enough time getting calibrated. We see a good place and we're like "Okay, that's it, I'm going for it."

That's something which has been a little puzzling for people who are trying to figure out what people are doing here. One way of understanding it is that, when you look at the assumptions that go into this model, it assumes that there's not really any cost to spending time looking, right?

Julia: Right, right.

Tom: Whereas actually, when human beings face this kind of problem, there is a cost. You want to get where you're going to live figured out sooner rather than later. Or if you're in a psychology experiment you want to get out of that experiment sooner rather than later.

When you factor in a very small cost, then in fact making that switch around 31 percent, which is what people do, I think that corresponds to a cost of about 1 percent of the value of getting the very best place ultimately, and people are willing to make that trade-off.

That's something where what people are doing deviates from the rational model. If you're actually in exactly the situation that that model corresponds to, then you should do something different.

Knowing that you should do something different, if you're in exactly that situation, that's what we equip you with in terms of telling you about these different kinds of algorithms. In the book we talk about all of the variance on this. If you don't have 100 percent chance that you won't be able to go back to a place after you've passed it over, but there was some other probability, or if there's some chance your offers get rejected. There's a whole constellation of different variations on this for which we can actually give optimal solutions.

Julia: Yeah, what I've found in my experience is that even just being aware of what the parameters are, even if you don't know the value of those parameters -- Parameters like "What is the amount time I'm willing to spend? Or the number of options that I think there are? What do I expect the average or ... potentially highest quality option might be roughly? What is the rough switching cost?"

… Even before I come up with a value for those variables, just being aware of them I think makes my intuition somewhat better at solving the problem optimally.

Brian: Yeah, I think in the case of optimal stopping, the biggest genres of optimal stopping problem are what's called no information games, and full information games.

In a no information game, which is the example Tom gave with the apartments where you hit the classic 37 percent rule. That's based on the assumption that you encounter your options in a random order. But more to the point, when you encounter a particular option, you cannot say how much better than other options it is. You can only say in relative terms what is the relative rank of this. You could say, this is the second best thing that I've seen so far, but you can't say if it's closer to the first or closer to the third. This is called a no information game.

In contrast with that, the other main sort of branch of these problems is called full information games in which you have some sense of the distribution over which these options are being drawn. If you were hiring a typist for example, and you knew for certain that they were a 95th percentile typist, on some typing score, well then you find yourself in a full information game.

It turns out that the stopping rules for full information games are quite different. In fact, you don't need an initial looking period before being ready to leap. You do need to know how many candidates are in the pool in order to make an educated guess about whether there's a better candidate still remaining. The optimal strategy in this case just has a fundamentally different structure. This goes back to your point about developing an intuition for what the solution landscape looks like, and asking yourself as you enter into a problem, "Do I feel like I have full information, or do I feel like I have no information?"

Julia: Right, right.

Brian: The choice of whether you need to set this initial looking window is going to depend on that.

Julia: Great. Let's take another example of an algorithm from your book. Do either of you have a particular favorite out of your chapters?

Tom: Another good example, I think, one that we talk about that has implications both domestically and scientifically is algorithms that are for solving the problem of caching. Basically, in Computer Science-

Julia: That's spelled "caching," right?

Tom: That's right, yeah. In Computer Science these algorithms are used for managing the memory of computers.

You can think about memory as a constrained resource. There's only a certain amount of very fast memory in your computer, and then a larger amount of slower memory, just because fast memory is expensive. What the computer has to do is to figure out what it's going to keep in that fast memory in order to increase the probability that the things that you're actually looking for are going to be there. So that your computer can be as fast as possible, it wants to make it so that most of the time when it's looking for a piece of information that information is stored in the most accessible form of memory.

That's a problem that human beings face as well, we call it organizing our closet. You can ask the same kind of question about ... if you've got things that you could be putting in different places in your house, in your closet, in your basement, or maybe in a storage container or something like that. You have to make a decision about what are the things that you want to keep in the most accessible storage locations. That's a context where taking a look at the kinds of algorithms that are used in Computer Science and why it is that they work is something which is relevant.

Julia: Excellent. I'm tempted to go through, instead of talking about algorithms, and then discussing what real life situations they might apply to, I'm tempted to give you a couple of real life situations that I've dealt with, or that clients of mine have dealt with, and see if any of the algorithms that you've written about feel like they might be useful.

Does that sound good?

Brian: Sure, I'm game for that.

Julia: Cool, great. One common situation ... a lot of my clients are young. They're just out of college, or they're early 20's, late 20's, and one of the biggest most re-occurring things they're struggling with is setting up their career, basically. Figuring out what they want to specialize in, and more immediately, what job they should be searching for now, which is a somewhat separate question from what they want their career to look like in 20 years, et cetera.

I've been trying over time to figure out, are there good rough algorithms they could use for figuring out what kind of job to take, or are there ways to decide what features of a job to prioritize over other features? A job that pays well now, versus a job that will build up their skills, versus a job that will give them a lot of information about the field, et cetera.

I'm wondering if any of your algorithms feel like they have useful things to say to someone who's uncertain about what short-term job decisions to make to maximize their longer-term career.

Brian: To me, this feels very much like an explore/exploit situation.

In Computer Science, the explore/exploit trade-off is the idea of "How do you balance between spending your time and energy getting information, and using your time and energy, leveraging the information that you have to get some good outcome or some payout?"

I think that's relevant in a career context, in a life trajectory context, because you have to spend a certain amount of time trying stuff out and learning what you enjoy, what you're good at, and so forth. The key concept that emerges when you look at these explore/exploit problems, is that everything really depends on how much time you have and where you perceive yourself to be along some relevant interval of time. If you're at the beginning of a process, you should be much more highly exploratory. If you're at the end of a process, you should spend much more of your time and energy exploiting the knowledge that you've gained so far.

In concrete human terms, it's like if you first move to a city, you should spend the first month or more just relentlessly trying new things. The first restaurant you go to when you move to Berkeley is literally guaranteed to be the greatest restaurant you have ever been to in Berkeley. The second place you try has a 50/50 chance of being greatest place you've ever been to in Berkeley. The chance of making a great new discovery is greatest at the beginning. Moreover, the value of making a discovery is the highest when you have the most time left to enjoy it.

Julia: Right.

Brian: Finding an amazing new restaurant on your last night in town is kind of tragic because you think to yourself, "I wish I'd known about this years ago." For both of those reasons, we should be on this trajectory from exploration to exploitation.

Given that the group of people you're talking about are coming out of college, they're looking for their first jobs and so forth. I think it makes sense to approach it from the perspective that they have a long time ahead of them. It's worth spending time trying things out, even if they have a low probability of being good, because if they are good, they've got their whole lives and their whole careers to reap the fruits of that.

Julia: Yeah, there's something, just to go on a very brief tangent, there's something counter-intuitive about the idea of trying a bunch of things because some of them

will work, and then you can exploit those for a long time afterwards.

It sounds intuitive when I say it, I guess, but it somehow doesn't match our intuition when we're actually faced with situations. I think that if a friend of mine was like "Hey, Julia, do you want to try trapeze or something," my first intuition might be "That probably is something I'm not going to like, therefore I won't do it." The implicit algorithm that my brain seems to be using in these cases is: if it seems likely to work then I'll do it, if it seems likely to not work, then I won't do it.

I think my brain is failing to take into account the fact that only a small subset of things I try have to work in order for the entire set of things I tried to have been worth it in bulk.

Brian: Yeah, I think the key idea in explore/exploit problems, or in the multi-armed bandit problem, which is the canonical one, is that if something fails on you, it only fails once because you just don't do it again. If it's successful, then you can just keep going back to it again and again.

If you think about it from the perspective of taking up a new hobby or a new sport or something like that, the metric of what is most likely to be the most fun evening is probably not the right metric, because even if there's only a 1 in 20 chance that you enjoyed doing the trapeze, if you find that you do enjoy doing the trapeze, you can do it 50 times over the next year. That actually puts you well ahead in that situation.

Julia: Right, right.

Tom: The algorithm we talk about that I think engages with this is what's called upper confidence bound, which is a class of algorithms for solving explore/exploit problems.

What that says is the evaluation you should be making is not your expected value, not how likely you think this thing is to be good. Rather, you should be calculating the upper bound on that expected value. Kind of like thinking about a confidence interval around the expected value, and then you take the upper bound of that confidence interval.

It's kind of like "How good could this be, under my best guess at the best possible scenario?" ... and then comparing that across the different activities that you could do. I think it's a kind of optimistic attitude, one that favors things that you've got very little information about.

Julia: What do you think about the strategy ... I don't know if this really counts as an algorithm, but a strategy of choices that put you upwind? it's a concept, I forget where I read it, that's about doing things that preserve your option value in the future. Like taking jobs that will introduce you to many other jobs. Or taking jobs that, having had that first job, is going to be respected by the widest possible variety of other potential future employers across lots of different fields, that kind of thing.

Tom: Yeah, I think that sounds like a sensible strategy. It's not something we talk about in

terms of the algorithms we consider. I think what's good about it is that it makes it clear that you're not necessarily making a choice for life, or something like that, but rather making a choice which is a first choice.

I think that is something that engages with a different kind of consideration here which I think is really relevant to making these sorts of decisions, and it's something we talk about in the context of overfitting in the book. I think when we want to make a significant decision I think there's a tendency to try and collect all the information that we can, and try and really optimize that decision with respect to all of the criteria that we can identify.

One counter-intuitive thing, something that machine learning researchers and statisticians have discovered, is that that's something that can actually be counter-productive. The analogy is, if you're trying to make a prediction, say you're trying to predict the stock market into the future or something like that. Then the more complex you make your model, the less accurate it turns out being. There's some level of complexity that you need in order to capture the signal that's in the data, but once you exceed that level of complexity, you end up making your model worse.

It's called overfitting. Basically, you make your model so complex that it's not only able to fit the signal in the data, it's also able to fit the noise, and so it makes worse predictions because it's giving weight to lots of factors which in fact are useful only for predicting the variation that's in the data due to the noise. When you get new data, you've got the stuff you could measure from the past, but the stuff that really matters is what's going to happen in the future. The gap between those things is something that then means that a model which is made optimized for the things that you can measure, can end up doing a worse job in predicting the things that matter.

Julia: Would an example of that be "My relationship failed and my update from that experience is that I should avoid getting into relationships with people who have brown hair of this exact length and who I met on a Thursday?"

Tom: That's maybe a kind of overfitting. I think we're more prone to doing that in general. I think one way that this really manifests ... I feel like its manifested in my life, is failing to recognize that your utility function now is not going to be the same now as your utility function in 5 years or in 10 years. If you really optimize the decision for where you are right now, then that might be a decision which isn't actually the best thing for you down the line, right?

Julia: Right.

Tom: If you're finding the thing that's the perfect match for the current moment, then you're potentially losing some fit to how good that might be for you into the future. All your current idiosyncrasies are things that are contributing to that current decision. And you're putting so much effort into optimizing it, then as the idiosyncrasies fade, and maybe you develop other ones, it'll end up being a worse fit then maybe if you'd taken something that was perhaps a little more generic, and less perfectly optimized to your current situation.

Brian: Yeah, and Tom and I have talked about this from the perspective of buying a home, where you're in some sense trying to optimize for the happiness of your 5-years-from-now future self, who is somewhat unknowable.

I encountered the same thing recently when I bought a tuxedo. It's funny to buy something where you feel like you'll wear it 1-2 times a year for the next 7 years.

How do you optimize for something that is going to look good when you take it to a wedding in 5 years?

Just to use this banal example of men's fashion. Men's pants are much tighter than they were 10 years ago, currently. When I look at the jeans that I wore in the mid 2000s, they were like twice as much fabric, or something like that.

If I'm buying jeans, which is something where the use case of jeans is that you wear them almost every day for like 18 months and then they develop holes and you just throw them out or something. If I wanted to buy jeans I should buy tight jeans because that is the style of the mid 2010s.

But if I want to buy a tuxedo, then I should deliberately get something that is looser in the leg, because I'm just assuming that men's fashion is on this random walk. I don't want to nail the current trend right on the button, because I know that it's going to deviate from that later.

Julia: Right, right, right.

I have a selfish request, which is that I'm hoping that you guys can help me think about my current life problem… I have bunch of projects that I feel like I should be doing, like home improvement stuff or getting my finances in order, starting a budget, et cetera. All these things that I've been telling myself I should do for years. Going to the gym regularly, all these things.

Obviously I don't have the attention and energy to tackle all of them at once. In practice, I feel like I'm almost randomly picking a project to start at one time versus another, maybe whatever thing feels most tractable or enjoyable to me at a given time. I feel like there must be a better strategy for deciding what to prioritize when. Can you give me any advice?

Tom: We may be able to give you a disconcertingly wide array of pieces of advice.

Julia: Excellent.

Brian: We tackle these questions of time management in our chapter on scheduling theory. One of the upshots in scheduling is that there is an optimal strategy for every conceivable metric of how you want to measure what good time management means to you. If you want to make sure that no task goes too far beyond, it's deadline, then there's a strategy called Earliest Due Date, that you should follow. If you want to minimize the length of your to-do list at any given time, then there's another strategy that's called Shortest Processing Time.

Julia: So that would be like, to tackle the projects first that are just the shortest, that I can finish the fast, and then I'll a week from now have finished 3 projects and I'll feel so good.

Brian: Yeah, exactly, exactly. The assumption that that particular algorithm makes is that all tasks are of equal importance and the amount of time a specific task lingers on your list is not as important as just reducing the amount of things on the list. The optimal strategy in this case is just do the easiest things first.

Which sounds great, but those may not be the correct assumptions to make in your case. I think one of the critical things that comes out in our scheduling chapter is that before you can get the answer as to what you should be doing, you must first articulate exactly what problem you want to be solving.

Julia: Yeah.

Brian: That's not a step in most time management self-help books.

Julia: Yeah I think they often have an assumption about what the right optimization function is and then they just give advice premised on that assumption without having made that assumption explicit.

Tom: Yeah, I think that's pretty much right. I think there are nice heuristics that come out of this, though, in general. I think there are versions of these metrics that maybe approximate human metrics.

Rather than saying the Shortest Processing Time assumes that every job is weighted equally, but obviously that is not the case, some things are more important than others.

Then you get into what's called Weighted Shortest Processing Time. Basically the way that that works is you take the ratio of how important something is to how long it's going to take you, and then you prioritize jobs by that ratio. I think that gives you a reasonably good heuristic, which is something like "You should only take twice as long to do something if it's twice as important."

Julia: Right, right.

Tom: Right, and out of that comes a pretty general time management algorithm which is Weighted Shortest Expected Processing Time, with preemption.

This whole literature gets very hairy. There are all these modifications that you can make, but if you assume that you're in a situation where you can put down a job, you don't have to just sit on it all the time, that's the preemption part. You have weights and you care about minimizing the weighted shortest processing time overall.

Then the optimal strategy in a situation where you don't know exactly how long things are going to take and you don't know exactly what jobs you're going to have,

is you work on the job that has the highest ratio of importance to time remaining to complete it.

Then if a new job comes along, you immediately evaluate: "What's the ratio of importance to how long it's going to take to complete it, compared to my current job, where the time that I've already invested into it is taken into account, so it's the remaining time to complete it." Then you make a decision about what you're going to do on that basis.

I think that's a pretty good heuristic. It doesn't take into account the fact that human lives have cycles. It might be good to exercise a few times a week, or something like that, and it doesn't take into account the fact that the importance of things might be time-varying, or that your ability to execute jobs might be time-varying as well.

Those are extra complexities that can go into that.

Julia: Right.

Tom: I think using that heuristic calculation and adjusting it for where you are for that particular moment in your day is actually not a bad plan.

Julia: There was something that I highlighted, in I think it was this chapter, that really struck me. It was something you said about procrastination, which is a problem I've struggled with mightily over the years. You said basically that it might be the right strategy but for the wrong problem. Can you elaborate on that?

Brian: Yeah, this cuts back to this idea that even strategies like the straight version of the Shortest Processing Time -- it's a perfectly viable strategy. In fact, it's the optimal strategy for a specific flavor of a problem. It gives us a way of re-characterizing something like procrastination, which is to say, it is not a faulty strategy to the problem, it is the optimal strategy to the wrong problem.

There have been a bunch of really interesting studies in the psychology literature about intuitive human task completion, and one of my favorites involves a long hallway where there are two buckets of water, two large heavy buckets of water. One is in the middle of the hallway, and one is at the end, where the participant is standing.

The experimenter at the other side of the hallway says "Could you bring me one of those pails of water please?" What happens in more cases than not is the person immediately picks up the pail right next to where they're standing, and lugs it all the way down this hallway, walking by the other one that they could've picked up and walked only half the distance.

The authors of this study coined the term "pre-crastination" to refer to this process by which people do twice as much work as they actually needed to. I think it highlights the fact that ... my read on this is that subconsciously they're applying this Shortest Processing Time metric, and saying "There are two items on the to-do list now that I've been required to do. One is pick up a bucket, and the other is bring the

bucket to the guy." And it's like "Oh, I can accomplish half of my to-do list right now-
"

Julia: Right, right.

Brian: "By picking up the bucket that's right here." It's not that that's the wrong approach to the problem, it's the wrong characterization of the problem.

Julia: How would you characterize the problem better? What would be the right definition for what you're trying to do there?

Brian: I don't know in scheduling theoretic terms what the exact way to describe the-

Julia: Just something like minimizing total work?

Brian: Yeah, Tom, do you have an intuition about what the relevant formalization of the water bucket problem is? I'm just trying to think…

Tom: Yeah I think that's right, minimizing the total amount of time expended in completing tasks.

This itself actually is I think a really deep point, which is that the total amount of time required to complete all the tasks is called the makespan. In a single machine scheduling problem where you are doing all the work yourself, you can't delegate it or outsource it or anything, and you must do all of the tasks, then there is this funny anti-result that says "The total makespan of doing all the work yourself is the same regardless of the order that you do it in."

Now the water bucket problem violates that assumption because it just happens to be harder to walk when you're carrying a giant pail of water.

Julia: Right.

Tom: For most situations ... this in itself is this nice heuristic, which is that, if you have to do all the work yourself, and you must do all of it, then the total time it will take you to do all the work is the same regardless of the order that you do it in. This is a case where actually spending any time prioritizing at all could be counter-productive. You might as well just work in random order.

Julia: Right. There's a study that keeps coming to my mind, not just in this conversation but in general when I try to give people formalized advice for decision making. I can't remember the authors who did the study, but the gist was they had one group of people make some medium to large size decisions just using their gut, it might have been purchasing a car or purchasing a house, and then the other group of people were instructed to use a formal ... I don't know if it counts as an algorithm, I guess in a loose sense it does. It was a pro/con list. Maybe they were instructed to give weights to the different factors, like the speed of the car, the safety, the price, et cetera, and then make their decision using the result of that process, of that algorithm.

When the experimenters followed up with the two groups of people, the group that had gone with their gut was actually happier with their choice as compared to the group that used the algorithm. I think that the takeaway, the experimenters' story about this, was that there were some really important factors that the formal process group were neglecting, because it didn't seem to fit into their formal process. Like "How much do I enjoy this particular car?" Which might not have anything directly to do with its mileage or its price, et cetera. It just might be the look and feel of the car.

I think the results of that, and other experiments like it, should give us pause, and by us I mean people who are giving formal advice for making decisions. I don't think it's a fatal blow to the idea that we can give formal advice that can improve default human decision-making processes, I just think the takeaway's more nuanced that ... you want to make sure you're paying attention to subjective emotional factors that also matter in your overall satisfaction or preferences.

I'm wondering how you think about that risk, is there a downside risk to using these algorithms, and how do you try to account for it?

Tom: I don't think that's an argument against algorithms. I think it's an argument against bad algorithms.

Julia: Yeah actually it reminds me of people who point out bad science and then conclude from that "Therefore, we can't trust science or we shouldn't try doing science at all."

Tom: Right. For me that goes back to this point about overfitting. When I was talking about overfitting I said one of the critical things is you don't want to include too many variables in your model, because then you're going to be focusing too much on things that actually don't really matter in terms of the thing that you care about, which is in this case predicting your future satisfaction, right?

I think that decision procedure is exactly one that encourages people to overfit, to over think the problem, to include more factors than they should, and to put more weight on those factors just because they're on their list.

Whereas the fact that they might have had difficulty in coming up with those things ... it might have taken them awhile to do so… It's something that indicates maybe they're not that important in terms of making that decision.

When we talk about overfitting we really talk about it in the context of being a way of justifying using certain kinds of heuristic strategies. It says that there are going to be cases where the simpler thing is better, where having fewer reasons is better, where acting sooner rather thinking more is better. There are going to be those situations where that gap between what we can measure and what matters is larger.

If what you can measure is exactly the thing that really matters, then you should put all the effort into it. You should come up with as many factors as you can.

At the other extreme, if you were submitting a grant proposal to a committee, and

what that committee was going to do was take all the grant proposals they got, throw them up in the air, and then found the one which landed on top, you should invest no effort in what you put into it, because the decision process is completely random and what you can measure about your sense of the quality of the grant proposal is in fact completely dissociated from what matters in terms of it getting funded.

Most things are somewhere in between having all the information which is relevant, and it being a completely stochastic process. Most things, there's some element of stochasticity and then some element of predictability. As the predictability goes down, the amount of effort you should put into making that decision and the amount of effort you should put into coming up with reasons or these other kinds of things should be decreasing. I think there are lots of decisions where we have poor proxies for the thing that really matters, like our future satisfaction. By over thinking it, we're making worse decisions.

Julia: Right. There's one other important motivational point that I want to close on, which is that I often encounter people beating themselves up because they used some algorithm and it didn't turn out well for them. Maybe the algorithm is, “I should experiment with being more open with people.” The reason the person chooses that policy or that algorithm is that they think that overall in the long run, it's going to be better for them. They're going to get better at being open. They're going to calibrate their expectations about how much openness is okay.

But they try it and the second or third time they try it, the person really reacts badly to their openness, and so they're like "Dammit, I should never have done this. This was a huge mistake." They feel terrible.

I always want to tell people "Do you think this a good policy or not? Did you get unlucky, or was the policy actually a bad policy in expectation, and not just after the fact it turned out poorly?"

Brian: Absolutely.

Julia: Yeah, your book makes a really important point about paying attention to the quality of the algorithm and not the outcome.

Brian: Yeah, I absolutely agree. I think one of the other things is paying attention to how hard the problem is that you're trying to solve. I think one of the things that Computer Science does really well is give is a way of articulating how hard problems are. There are huge classes of problems that are just considered intractable. We should not expect to be able to reliably get the correct solution in an efficient and repeatable way.

Even some of the algorithms that we've discussed today, the 37 percent rule and the optimal stopping problem, if you read the fine print, the 37 percent rule only works 37 percent of the time. It just turns it out that optimal stopping in the no information case is a hard problem and even when you are following the optimal algorithm, you still fail 63 percent of the time.

Julia: Right.

Brian: So here's a case where not only does having the optimal strategy not guarantee that you'll succeed every time, it doesn't even guarantee that you'll succeed most of the time.

It just happens that that's best you can do. Similarly, in the explore/exploit case, the expected value of exploring is necessarily lower than the expected value of exploiting, and so when you try a restaurant and it's no good, that doesn't mean that you've done the wrong thing, that is just part of the problem. Because if it is good, then you get to multiply that over all the times that you return.

I think this is actually a really important theme of the book, which is that, as you say, we have a tendency to, if we don't get the result that we want, to immediately call into question the process that we used to arrive at that result.

I think having a grounding in the Computer Science of ... understanding what the landscape of these problems looks like, understanding what the solutions look like, gives us a way of resting easy even in those cases where we don't get what we want, to know that we followed a strategy that was a sensible strategy.

Julia: Right. A motivational hack that some of my friends use is to remind themselves of the multiverse, and keep in mind that actually that strategy did produce the best result, if you look across all the copies of me in all the different worlds, right? Maybe I'm in the world in which it didn't work out well, but overall, all the versions of me are better off because I used the strategy, definitely, not just "it could've happened that way."

Tom: As long as you can handle your quantum jealousy, right?

Julia: Right. "Motivational tricks for nerds, and their hazards," next episode.

Okay, cool, well I'm going to wrap up this section of the podcast and we'll link to the excellent *Algorithms to Live By* on our podcast website. I encourage everyone to read it.

But for now let's move on to the Rationally Speaking picks.

[interlude]

Julia: Welcome back. Every episode on Rationally Speaking, we invite our guest, or guests in this case, to give a Rationally Speaking pick of the episode. That's a book, or a website, or movie, or something that influenced their thinking in some way. Brian, let's start with you, what's your pick for the episode?

Brian: My pick is a small and very strange book called *Finite and Infinite Games* by James Carse. It came out in the 1980's and has just been reissued in reprint. It's a book that I was able to read in probably 3 hours or something like that, but I think about it

literally every day.

The basic premise of the book is you can draw this distinction of ... all human activity falls into one of two categories, either a finite game or an infinite game.

A finite game is something that you participate in to bring it to a close in some desired way. A boxing match is a great example, where each boxer throws every punch with the intent to bring the boxing match to a close.

In contrast, an infinite game is something that we participate in in order to prolong it or extend it. Conversation, or musical improv, or comedic improv, are all examples of things that ... we participate in them in order to prevent them from coming to an end.

Carse, having created this distinction, goes on to apply it to all sorts of things from law, to sex, to medicine, to ethics, and it's just a really, really riveting exploration of human motivation and it's a dichotomy that has stayed with me as I go through life.

Julia: Interesting. Okay, I'm not going to ask whether sex is a finite or infinite game for you. We'll just leave that as an exercise for the reader, or listener.

Tom, what's your pick for the episode?

Tom: My pick is also a book. It's a book called *Do The Right Thing* by Stuart Russell and Eric Wefald. This is a pretty technical book, but it really makes a very important point, and it's a point which is one of the implicit premises, I think, in our book.

The argument in this book is that really the way that we think about rationality should take into account the fact that we as agents have limited computational resources. Then it takes that premise and then it works from there and it comes up with a new way of looking at rationality, which they call bounded optimality. Basically, the idea is that being a rational agent isn't a matter of making the right decisions, it's a matter of using the right programs. That's very much consistent with the idea of our book which is that rationality is really a matter of following good algorithms.

Julia: Yeah, man, rationality really needs a full-time PR agent or something. I wonder if the job's open. I'll check.

Cool, well guys, thank you so much for coming on the show. I really enjoyed your book and will recommend it to others.

Brian: Thank you so much. Thanks for having us.

Tom: Yeah, great, thank you.

Julia: Well this concludes another episode of Rationally Speaking. Join us next time for more explorations on the borderlands between reason and nonsense.