TCS on the Web

BY

Stefan Neumann

TU Wien
Erzherzog-Johann-Platz 1
1040 Vienna, Austria
stefan.neumann@tuwien.ac.at
https://neumannstefan.com

Tim Roughgarden is a full professor at Columbia University. He works on algorithmic game theory and microeconomics, especially as applied to networks, auctions, and blockchains. He has made significant contributions to the design and analysis of algorithms, and his work has won many awards, including the 2009 Grace Murray Hopper Award, the 2012 Gödel Prize, and the 2016 Kalai Prize.

Tim is one of the most important theoretical computer scientists on YouTube[1], where he regularly uploads recordings of his lectures. He has more than 25,000 followers, and his videos have been watched more than 1 million times in total. In this column, he answers our questions about the origins of his YouTube channel, the teaching philosophy behind his videos, and how to communicate theoretical ideas to a broader audience.

[1] https://www.youtube.com/@timroughgardenlectures1861

Bringing Theoretical Computer Science to YouTube

A Conversation with Tim Roughgarden

Tim, thank you for taking the time for this interview. You created your YouTube channel back in 2013. Do you recall what initially motivated you to start uploading content?

First, let me say that it's an honor to do this interview. I've been reading the EATCS Bulletin — especially the excellent survey papers that have been published in it — since my days as a PhD student.

As to your question, the story begins in 2011, when there was tremendous appetite at Stanford (where I was at the time) for experimentation in online education. The companies Coursera and Udacity were both formed around this time by Stanford colleagues, and I pounced on the opportunity to develop my long-running undergraduate algorithms course into one of the first "MOOCs" (massive online open courses) on Coursera. The response was staggering, with hundreds of thousands of learners signing up for it. In addition to the sheer numbers, I was struck by how many of the learners were hungry for high-quality technical content but in no position to attend the exclusive schools where that content is traditionally taught (due to geography, family obligations, financial resources, etc.). To this day, I regularly receive emails from alumni of my online courses thanking me for the personal and professional advances that those courses unlocked for them. Since these experiences, I've been on a mission to make as many educational resources as widely available as possible.

You used to upload videos of board lectures of 60–90 minutes length. More recently, your videos are shorter—of about 30 minutes length—and now you write your content on a tablet (but still by hand). What has triggered that change and why do you not use slides?

To my taste, the gold standard for a video is a simulation of a one-on-one session with an expert tutor. The shorter videos in which I'm writing by hand are my best attempt at (a noninteractive version of) such a simulation. They are time-consuming to produce and have never contributed to my official teaching duties at Stanford or Columbia, so I can put them out only as time allows. The longer

videos at the white/blackboard are all of in-person courses that I delivered at Stanford to fulfill my teaching duties there. To all the instructors out there, especially if you're teaching a topic that doesn't already have adequate coverage on YouTube, I strongly encourage recording your lectures and making them public (with the permission of your employer). It increases your audience by something like 10–100x while increasing your workload by a far smaller amount (especially if you can enlist a teaching assistant or PhD student to assist with the video recording and production).

For theoretical computer science courses (and mathematical courses in general), as both a teacher and a student, I've always found white/blackboard lectures far more effective and engaging than slide-based ones. There's something about the way you can talk through math as you're writing it that seems quite difficult to simulate with a slide deck, no matter how much time you spend on your animations.

What do you think makes your lectures so popular?

That's really a question for others! I'm just happy that they seem to have been helpful to so many different people.

Beyond your research and YouTube channel, your lecture notes[1] *are also highly influential in our community. After you and Greg Valiant taught "Modern Algorithmic Toolbox" at Stanford, I noticed many instructors — including myself — started teaching consistent hashing. How did you succeed in popularizing your notes, and what do you think distinguishes them from others?*

The demand for high-quality and free lecture notes is high, and the supply is low (presumably because there are no short-term incentives for creating them). So if you've got a lecture or a course that you're proud of and that complements the free resources that are available already, by all means write it up and get it out there!

Sometimes my impression is that in TCS we have a lot of great results that would deserve more visibility in the general CS community. Do you have some suggestions to other TCS researchers on how we can improve our outreach?

I actually think that the TCS community has been quite strong on outreach for the past decade-plus (for example, through the programming at the Simons Institute) and is generally very well respected across computer science. For TCS readers looking to increase the audience of their own research, my general advice would be to: (i) identify the audience that you'd like to speak to (mathematicians? computer systems researchers? economists?); (ii) understand the typical

[1] `https://timroughgarden.org/notes.html`

background of that audience and what it tends to care about (mathematical depth? practical impact? policy implications?); and (iii) communicate your work to that audience accordingly. For example, if you'd like your work to attain visibility in two separate research communities (e.g., "traditional" TCS as well as some other area), it is sometimes worth writing two versions of the same paper (e.g., a formal TCS publication as well as an informal writeup or blog post that is optimized for the second audience). Also, give as many talks and seminars as you can!

Before we conclude the interview, could you please let us know what your current research is about?

Most of my research over the past 25 years resides at the interface of computer science and economics (a.k.a. "algorithmic game theory (AGT)" or "economics and computation (econ-CS)"). Many problems central to modern computer science, ranging from resource allocation in large networks to the design of blockchain protocols, fundamentally involve interactions between multiple self-interested parties. Economics and game theory offer a host of useful models and definitions to reason about such problems. The flow of ideas also travels in the other direction, with approximation and complexity notions playing an increasingly important role in the latest developments in economic theory.

Working in algorithmic game theory has led me on a journey across a number of application domains, such as network routing and design, online advertising, spectrum auctions, and data-driven mechanisms. For the last several years, I've been working full-time on developing foundations for blockchain protocols and the applications built on top of them — I've never seen a technology in which important incentive issues are so deeply baked into the technology itself! The three main thrusts of my research in the area thus far (done with a number of fantastic collaborators) have been: (i) a general theory of permissionless consensus protocols; (ii) a new branch of mechanism design that is tailored to the unique constraints of blockchain protocols; and (iii) mathematical foundations for decentralized exchanges.

Finally, is there anything else you want our readers to know?

I'm always happy to receive feedback on any of my videos/books/lecture notes/papers/etc., as well as suggestions about which topics would be most useful to cover in the future!

Thank you for this nice interview, Tim!