2022 Abacus Medal: Mark Braverman

by Allyn Jackson

In the past decade, Mark Braverman has emerged as a major leader in theoretical computer science. He has an uncommon versatility and fearlessness that has allowed him not only to tackle specific outstanding problems, but also to work on deep theoretical questions. As a researcher, he exhibited exceptional maturity at a young age, producing results that brought new insights and stimulated new research.

Of the many subfields of computer science, the one known as theoretical computer science is the closest to mathematics. It draws on, as well as develops, abstract mathematical notions in order to address questions inspired by concrete problems in computation, communication, information transmission, and related areas. A major goal of theoretical computer science is to establish precise, mathematically rigorous results about how quickly and efficiently problems can be solved. Emblematic of this goal is the famous P versus NP problem, a major unsolved question in both theoretical computer science and mathematics.

At the age of 38, Braverman already has a publication list of more than 100 papers written with a total of more than 85 collaborators. Because his oeuvre is extensive and diverse, we focus here on three areas to which he has contributed results that exemplify the depth and power of his work.

Computing Julia Sets

Even in his earliest work Braverman took on fundamental questions. One of them centered on investigations of what becomes possible—and impossible—when one changes the theoretical basis for computing.

Modern computers are based on the model for computing formulated by Alan Turing in the 1930s and are essentially discrete systems: Any task a computer carries out boils down to manipulating 0s and 1s. By contrast systems in nature—the swinging of a pendulum, the development of a weather pattern, the fractal geometry of a coastline—are continuous rather than discrete. Mathematicians and computer scientists have therefore investigated alternative computational models that are continuous rather than discrete. Braverman has worked with one model that represents computations not as 0s and 1s but rather as real numbers (the set of real numbers contains all numbers, including ones like π that have infinite, nonrepeating decimal expansions).

The bedrock of mathematics shifted when Turing's model for computing revealed the concept of uncomputability: There exist numbers that can be described in a perfectly clear and precise way but that cannot be computed explicitly. What kinds of uncomputability phenomena arise with continuous computational models? This is the question that Braverman explored starting already with his master's thesis and in subsequent work, much of it with Michael Yampolsky.

Braverman focused on a continuous computing model that, intuitively, is based on the idea that a set is computable if it can be drawn pixel-by-pixel on a computer screen. Among the mathematical objects drawn in this way are Julia sets, which are fractals originally discovered by Gaston Julia in the early 20th century and popularized by Benoit Mandelbrot in the 1980s. The beauty and intricacy of Julia sets have made them the subject not only of art exhibitions but also of intensive study within the theory of dynamical systems, the branch of mathematics treating systems that evolve over time.

Although Julia sets can exhibit highly complicated behavior, each is characterized by a single parameter. Braverman and Yampolsky identified values of this parameter such that the associated Julia set is uncomputable in the continuous computing model they used. These parameter values are few and far between; you are not likely to hit upon one when entering a parameter in one of the many web programs that draw Julia sets. This points to a kind of instability of uncomputable structures and might offer hints about why they are rarely encountered in real-world problems.

In 2009 Braverman and Yampolsky published a book *Computability of Julia Sets*, which provides an overview of this area. Braverman has made further contributions to computability of other phenomena in dynamical systems. For example, in a 2007 paper with Ilia Binder and Yampolsky, Braverman investigated the computability of the Riemann mapping, a fundamental mathematical notion from the area of complex analysis. And in 2015 he published a paper with Jonathan Schneider and Cristóbal Rojas that presents a refinement of a pillar of computing theory, the Church-Turing Thesis.

Starting around 2010, Braverman's attention turned to information complexity, which we will discuss next. Here too the theme of discrete versus continuous arises in his work.

Information Complexity

In 1948, Claude Shannon published a paper that provided a comprehensive theory governing the transmission of information. He showed that, even when information is represented as discrete bits—that is, as strings of 0s and 1s—it can be modeled as a continuous quantity using probability theory. One can then define the notion of the *entropy* of a transmitted message, which intuitively speaking is the amount of information it contains. For example, suppose Alice sends Bob a message giving a year's worth of data about two daily events: Whether the sun rose that day, and whether it rained that day. Although she has 365 bits of sun-rising data, that information could be compressed into one bit; its entropy is very small. By contrast, the 365 bits of rain data could not be compressed so much; its entropy is higher. Shannon's theory shows that the entropy establishes a natural limit on how much a message can be compressed without losing information.

Now suppose that instead of the communication being one-way, it's two-way: Alice and Bob each have some information and send bits back and forth. Their goal might be, for example, to understand how their knowledge differs. It could happen that what Alice knows differs by only one bit from what Bob knows, but establishing that fact requires sending many bits back and forth. In such a case, the communication cost, which is the number of bits exchanged, is large, but the information cost, which is the amount of *information* exchanged, is low. Does information theory shed light on how to make such an exchange more efficient?

Starting around 1980, the area of communication complexity grew up around such questions. A good deal of progress was made addressing specific problems, and this had a big impact in applications to tasks like streaming algorithms and data structures. However, the theoretical underpinning remained somewhat undeveloped, partly because the necessary mathematical machinery was lacking. When Braverman came on the scene starting around 2010, he revived and expanded the field through a series of striking results that supplied new and more precise theoretical foundations.

Basic to this area is the direct sum question, which asks the following. Suppose C is the cost of Alice and Bob interacting to carry out a certain task once. If they carry out k independent repetitions of the task, is the final cost always equal to k times C? In the case of information cost, Braverman proved that the answer is yes. For communication cost the answer is generally no and depends on the nature of the task.

But in 2010, Braverman, together with Boaz Barak, Xi Chen, and Anup Rao, showed that the cost of k repetitions is at least $\sqrt{k}$ times the cost of doing the task once. The following year, Braverman and Rao proved an "amortization" result, establishing that in the limit as k gets very large, the average communication cost of one repetition approaches the information cost. This result has had a wide impact by providing a natural line of attack for proving results about efficiency for specific communication tasks. Braverman has also had important results in regimes where the communication can be corrupted by transmission errors or sabotage.

In the realm of computation, the dominant theoretical problem is P versus NP. Work on this problem has to a large extent remained in the discrete realm and has not benefited much from continuous tools from analysis, which is one of the most sophisticated and highly developed areas of mathematics. The work of Braverman and his collaborators on efficiency in communication might provide a glimmer of hope that continuous tools could one day have a larger impact on P versus NP than they have so far.

Mechanism Design

As algorithmic economics has provided the foundation for much of online commerce, mechanism design has grown into one of its most active subfields. Here too Braverman has made several significant contributions.

Right after his doctorate, Braverman held a research position at the Microsoft Research New England laboratory, where he worked with the lab's health care group. There he investigated machine-learning tools for studying factors leading to patient rehospitalization. This experience led him to realize that such questions are often more economic and game-theoretic than they are computational and sparked his interest in algorithmic economics.

An algorithm takes an input, carries out a step by step procedure, and produces an output. The algorithm carries out the same procedure regardless of what the input is; one might say that the input doesn't care what the algorithm is. But in many economic procedures, for example in auctions, the inputs are provided by agents who do care what the algorithm is and are seeking, by their inputs, to influence the output. This is the setting for mechanism design, which aims to construct protocols that take inputs elicited from agents having a stake in the output and that also drives the agents towards inputs that result in desirable output.

A well known example of a mechanism is the Vickrey auction. Bidders submit secret bids, and the person submitting the highest bid is allowed to buy the item, but at a price equal to the *second-highest* bid. This system drives bidders to be honest about what the item is actually worth to them: Underbidding cannot reduce the purchase price and could cause them to lose the opportunity to buy the item. In the more general Vickrey–Clarke–Groves (VCG) mechanism, multiple items are distributed among bidders, and each bidder must pay for the "harm" that buying an item causes to the others, thereby achieving a socially optimal solution. VCG produces excellent theoretical results, but its practical implementation is marred by instabilities and other problems.

In today's world of cheap computing and interconnectedness, algorithms are increasingly manipulated by strategic agents. A major goal is therefore to find ways to convert algorithms to mechanisms, and this is the focus of Braverman's latest research. In particular, he has been looking at how to incorporate the VCG mechanism into algorithms that are based on many implementations of local optimization. Such an algorithm contains many sub-algorithms, each of which uses local optimization on just one small chunk of the problem and takes incremental steps towards optimal solutions within that chunk. Those locally optimal solutions are then combined by the main algorithm to solve the problem. Braverman's idea is to bring the VCG mechanism into the algorithm at the level of the local optimization, where the larger problems of VCG can be effectively controlled. Because local optimization is used in many systems, including in machine learning, Braverman's approach has potential for wide impact in applications. It has already borne fruit in the realm of theory; in 2021 Braverman used it to strengthen an economics result from more than 40 years ago.

Problem-Solving Prowess and Theoretical Insight

This brief account of Braverman's work shows how he is able to make progress on difficult questions that require sustained focus and development over time. But he has also worked in a different mode, solving isolated and highly abstract open questions that called on his problem-solving prowess. An example of this is his 2010 proof of the Linial-Nisan conjecture. Too technical to describe here, this conjecture arose in the area of pseudorandomness and had stumped researchers since it was first proposed in 1990. Braverman's strikingly original solution stunned experts and was especially surprising because the problem lay so far from the areas in which he had been working.

Mark Braverman's combination of potent problem-solving ability and deep theoretical insight has produced results of exceptional impact. His work embodies the spirit of theoretical computer science, with its emphasis on marrying the power of abstract mathematics to the real-world struggle for speed and efficiency. His influence on the field, already large for such a young researcher, will no doubt continue to grow.

Biographical sketch of Mark Braverman:
`https://mbraverm.princeton.edu/about/brief-bio/`