



# ۵۰ سال پیچیدگی محاسبه یک گزارش خبری

بن برویکر\*

چکیده. چه قدر دشوار است که ثابت کنیم حل مسئله‌ای سخت است؟ نظریه‌پردازان فرای پیچیدگی دهه‌هاست که به پرسیدن چنین سوال‌هایی مشغولند و سلسله‌ای از نتایج به یافتن پاسخ‌هایی انجامیده است. با این حال متخصصان پیچیدگی محاسبه هنوز در حال دست‌وینجه نرم کردن با دشوارترین مسئله‌ی خود هستند: خود نظریه‌ی پیچیدگی!

## ۱. خاستگاه

در اولین هفته‌ی ترم پاییز ۲۰۰۷، مارکو کارموسینو<sup>۱</sup> خود را به کلاس درس ریاضی‌ای کشاند که برای همه‌ی دانشجویان علوم کامپیوتر دانشگاه ماساچوست امهرست اجباری بود. کارموسینو یک دانشجوی سال دومی بود که می‌خواست برای آن که طراح بازی‌های کامپیوترا شود، ادامه‌ی تحصیل را رها کند. با این حال در آن کلاس بود که استاد با سوالی ساده مسیر زندگی او را تغییر داد: «از کجا می‌دانید که ریاضیات واقعاً کار می‌کند؟». کارموسینو که هم اکنون یک متخصص علوم کامپیوتر نظری در IBM است، به یاد می‌آورد: «آن سوال باعث شد که بنشینم و توجه کنم». او درس سینیاری اختیاری درباره‌ی آثار گودل برداشت. گودلی که استدلال‌های خود را جاعده‌اش برای اولین بار محدودیت‌های استدلال ریاضی را آشکار کرده بود و زیربنایی را برای تمام نتایج بعدی درباره‌ی محدودیت‌های اساسی محاسبه پی‌ریزی کرده بود. هضم آن همه مطلب برای کارموسینو دشوار بود. «صد درصدش را نفهمیدم، اما می‌دانستم که می‌خواهم بفهمم!».



شکل ۱. جایزه‌ی میلیون دلاری من کجاست؟! (منبع تصویر: [۱])

امروز حتی محققان کارکشته نیز وقتی با مسئله‌ی باز مرکزی علوم کامپیوتر نظری، یعنی  $P$  در برابر  $NP$  مواجه می‌شوند، نمی‌دانند که چطور می‌توانند به آن جوابی بدهنند. این پرسش درباره‌ی این است که آیا ممکن است تعدادی از مسائل محاسباتی که مدت‌هاست دشوار تلقی شده‌اند به سادگی (از راه میانبری مخفی که هنوز کشف نکردۀایم) قابل حل باشند، یا این که همان‌طور که اکثر محققان گمان می‌کنند، آن‌ها واقعاً دشوارند؟ در حقیقت این مسئله درباره‌ی چیزی جز ذات آن چه قابل دانستن است، نیست.

علی‌رغم دهه‌ها تلاش محققان پیچیدگی محاسبه<sup>۲</sup> — که به مطالعه‌ی چنین سؤالاتی درباره‌ی دشواری ذاتی مسائل مختلف پرداخته‌اند — پاسخ مسئله‌ی  $P$  در برابر  $NP$  هنوز ناشناخته است، و حتی مشخص نیست که برای یک اثبات احتمالی از کجا

\*این نوشته ترجمه‌ای از مقاله‌ی زیر است:

Ben Brubaker (2023) Complexity Theory's 50-Year Journey to the Limits of Knowledge. Quanta Magazine.

<sup>1</sup>Marco Carmosino

<sup>2</sup>computational complexity theory

باید شروع کرد. مایکل سیپسر<sup>۱</sup> می‌گوید: «هیچ نقشه‌ی راهی وجود ندارد..». او یک متخصص کهنه‌کار پیچیدگی محاسبه در MIT است که در تلاش برای حل این مسأله سال‌ها وقت صرف کرده است. او ادامه می‌دهد: «مثل این است که به برهوت بروی..».

این طور که به نظر می‌رسد اثبات این‌که حل برخی مسائل محاسباتی دشوار است، خود مسأله‌ای سخت است؛ اما چرا و چقدر؟ کارموسینو و دیگر محققان حوزه‌ی فرایچیدگی با چرخاندن لنز دوربین به سمت خود نظریه‌ی پیچیدگی، سؤالاتی این چنینی را مجدداً به صورت مسائل پیچیدگی فرمول‌بندی کردند تا تحقیقات این حوزه را پیش ببرند. راهول ایلانگو<sup>۲</sup>، دانشجوی تحصیلات تکمیلی در MIT، که به برخی از هیجان‌انگیزترین دست آورده‌ای اخیر در این زمینه دست یافته است، درباره‌ی این رویکرد می‌گوید: «ممکن است فکر کنید جالب است. شاید هم فکر کنید متخصصان پیچیدگی محاسبه عقلشان را از دست داده‌اند!».

با مطالعه‌ی این مسائل درون‌نگرانه، پژوهشگران یادگرفته‌اند که دشواری اثبات سختی مسائل، با سوال‌هایی بنیادی که در وهله‌ی اول ممکن است نامرتبط به نظر برسند، گره خورده است: تشخیص الگوهای پنهان در داده‌های به ظاهر تصادفی چه قدر مشکل است؟ و اگر مسائل واقعاً مشکل وجود دارند، تعدادشان چقدر زیاد است؟ اسکات آرانسون<sup>۳</sup>، متخصص پیچیدگی در دانشگاه تگزاس آستین<sup>۴</sup>، می‌گوید: «روشن شده که فرایچیدگی به حقیقت نزدیک است..».

این نوشته داستان مسیر طولانی و پریچ‌وخمی است که محققان را از مسأله‌ی  $P$  در برابر  $\NP$  به فرایچیدگی سوق داد. مسیری پر از پیچ‌های اشتباه و جاده‌های مسدود که دویاره و دویاره به نقطه‌ی اول خود برمی‌گشتند. با این حال برای متخصصان فرایچیدگی این سفر به سرزمین‌های ناشناخته خود پاداش خود است. والتنین کابانتس<sup>۵</sup>، متخصص پیچیدگی محاسبه در دانشگاه سایمون فریزر<sup>۶</sup> کانادا می‌گوید: «شروع کنید به پرسیدن سوالات به ظاهر ساده..»، و ادامه می‌دهد: «هیچ ایده‌ای ندارید که قرار است از کجا سر در بیاورید..».

۱.۱. ناشناخته‌های شناخته‌شده. مسأله‌ی  $P$  در برابر  $\NP$  نام بدون زرق و برقص را مدیون عادت نظریه‌پردازان پیچیدگی به دسته‌بندی کردن مسائل محاسباتی در گستره‌ای از کلاس‌های پیچیدگی است. یک مسأله‌ی محاسباتی، مسأله‌ای است که بتوان آن را با یک الگوریتم — یا به بیانی ساده با دنباله‌ی دقیقی از دستورالعمل‌ها — حل کرد. با این حال تمام الگوریتم‌ها به طور یکسانی مفید نیستند، و تنوع میان آن‌ها اشاره به تفاوت‌های بنیادی میان مسائل کلاس‌های مختلف دارد. چالش نظریه‌پردازان پیچیدگی این است که این اشارات را به قضیه‌هایی دقیق درباره‌ی روابط میان کلاس‌های پیچیدگی تبدیل کنند. این روابط — که دامنه‌ی آن‌ها فراتر از مزه‌های همه‌ی فناوری‌هاست — منعکس‌کننده‌ی حقایقی تغییرناپذیر درباره‌ی مفهوم محاسبه هستند. کابانتس می‌گوید: «این کار مانند کشف کردن قوانین کیهان است..».

$P$  و  $\NP$  دو نمونه از مشهورترین اعضای یک باع‌وحش در حال توسعه از صدھا کلاس پیچیدگی هستند. صرف نظر از جزئیات،  $P$  کلاس مسائلی است که به راحتی<sup>۷</sup> با یک الگوریتم حل می‌شوند، مانند مرتب کردن الفبایی یک لیست.  $\NP$  کلاس مسائلی است که درستی جواب‌هایش به راحتی قابل بررسی هستند، مانند حل سودوکو. از آنجا که تمام مسائلی که به راحتی حل می‌شوند به راحتی هم جواب‌هایشان قابل بررسی هستند، مسائل عضو  $P$  عضو  $\NP$  هم هستند. با این وجود، حل برخی مسائل  $\NP$  سخت به نظر می‌رسد؛ مثلاً در سودوکو بدون این که ابتدا بسیاری از حالات را امتحان کنید، نمی‌توانید جواب نهایی را بدست آورید<sup>۸</sup>. آیا ممکن است که این سختی ظاهری، فقط یک توهم باشد و یک ترفند ساده برای حل تمام مسائلی که درستی جواب‌شان به راحتی قابل بررسی است، وجود داشته باشد؟

اگر چنین باشد، آنگاه  $\NP = P$ : یعنی هر دو دسته مسأله با هم معادلنند. در این صورت باید الگوریتمی وجود داشته باشد که حل سودوکوهای عظیم، بهینه‌سازی مسیرهای حمل و نقل جهانی، شکستن پیشرفته‌ترین رمزها و ماشینی کردن اثبات قضایایی

<sup>1</sup>Michael Sipser

<sup>2</sup>Rahul Ilango

<sup>3</sup>Scott Aaronson

<sup>4</sup>University of Texas, Austin

<sup>5</sup>Valentine Kabanets

<sup>6</sup>Simon Fraser University

<sup>7</sup> در اینجا مقصود از حل پذیری راحت این است که الگوریتمی وجود دارد که در زمانی که بر حسب طول ورودی مسأله کران بالایی چند جمله‌ای دارد، مسأله را حل می‌کند [م..].

<sup>8</sup> به عبارت بهتر، به نظر می‌رسد برای هر الگوریتم که برای حل سودوکو وجود دارد، جدول‌های سودوکویی قابل تصورند که حل آن‌ها با الگوریتم مزبور آسان نیست [م..].

ریاضی<sup>۱</sup> را به سادگی انجام دهد. اگر  $\mathcal{N}\mathcal{P} \neq \mathcal{P}$ ، آنگاه بسیاری از مسائل محاسباتی که از نظر تئوری حل شدنی هستند، در عمل قابل حل نخواهند بود.

محققان مدت‌ها قبل از این‌که مسئله‌ی  $\mathcal{P}$  در برابر  $\mathcal{N}\mathcal{P}$  برای اولین‌بار بیان شود — در حقیقت مدت‌ها قبل‌تر از شروع علوم کامپیوتر مدرن — نگران محدودیت‌های استدلال‌های صوری ریاضی بودند. در ۱۹۲۱، دیوید هیلبرت<sup>۲</sup> ریاضیدان در تلاش برای پاسخ‌دادن به همان سوالی که نزدیک به یک قرن بعد توجه کارموسینو را جلب کرد، برای پی‌ریزی ریاضیاتی دقیق یک برنامه‌ی تحقیقاتی پیشنهاد کرد. او امید داشت که با شروع از تعداد کمی فرض ساده — به نام اصول موضوعه<sup>۳</sup> — یک نظریه‌ی ریاضی یک‌پارچه<sup>۴</sup> استخراج کند که دارای سه معیار کلیدی باشد. شرط اول هیلبرت سازگاری<sup>۵</sup> بود، شرطی ضروری برای این‌که ریاضیات عاری از تناقضات باشد: اگر دو گزاره‌ی متناقض را بتوان با شروع از اصول موضوعه‌ی یکسانی اثبات کرد، تمام نظریه‌ی غیرقابل نجات خواهد بود. با این حال یک قضیه‌ی می‌تواند عاری از تناقض، ولی هم‌چنان دست‌نیافتنی باشد. این انگیزه‌ای بود برای شرط دوم هیلبرت، تمامیت<sup>۶</sup>: التزام به این‌که تمام گزاره‌های ریاضی یا به طور قابل اثباتی درست باشند و یا نادرست. معیار سوم او، تصمیم‌پذیری<sup>۷</sup>، خواستار یک رویه‌ی بدون ابهام مکانیکی برای تعیین درست یا نادرست بودن هر گزاره‌ی ریاضی بود. هیلبرت در کنفرانسی در ۱۹۳۰ اعلام کرد: «شعار ما این است: ما باید بدانیم؛ ما خواهیم دانست.».

تها یک سال بعد گودل اولین ضریب را به رویای هیلبرت وارد کرد. او اثبات کرد که یک گزاره مانند «این گزاره اثبات‌پذیر نیست.» می‌تواند از هر مجموعه‌ی مناسی از اصول موضوعه استنتاج<sup>۸</sup> شود. اگر چنین گزاره‌ای به راستی اثبات‌پذیر نباشد، تمامیت نقض می‌شود و اگر اثبات‌پذیر باشد، نظریه ناسازگار خواهد بود، که نتیجه‌ی بدتری است. همچنین در همان مقاله، گودل ثابت کرد که هیچ نظریه‌ی ریاضی‌ای هرگز نمی‌تواند سازگاری خودش را ثابت کند.<sup>۹</sup>

محققان هنوز امید داشتند که ممکن است در آینده نظریه‌ای

ریاضی یافت شود که تصمیم‌پذیر باشد، هر چند چنین نظریه‌ای لزوماً ناتمام خواهد بود. ممکن است آن‌ها بتوانند روش‌هایی را توسعه دهنده که در حالی که از گزاره‌های آزاردهنده‌ای مانند گزاره‌های گودل دوری می‌کند، تمام گزاره‌های اثبات‌پذیر را شناسایی کند. مشکل آن‌جا بود که هیچ کس نمی‌دانست چگونه درباره‌ی این روش‌ها استدلال کند.



شکل ۲. در دهه‌ی ۱۹۲۰، هیلبرت (تصویر چپ) قصد داشت ریاضیات را بر مبانی محکم‌تری استوار کند. گودل (تصویر وسط) و تورینگ (تصویر راست) نشان دادند که رویای هیلبرت غیرممکن است.

بعدتر در ۱۹۳۶، یک دانشجوی تحصیلات تکمیلی ۲۳ ساله به نام آلن تورینگ<sup>۱۰</sup>، شرط تصمیم‌پذیری هیلبرت را به زبان در آن

زمان ناآشنای محاسبه بازنویسی کرد و ضریه‌ی مهلکی بر آن وارد کرد. تورینگ یک مدل ریاضی را فرمول‌بندی کرد که امروزه به نام ماشین تورینگ شناخته می‌شود، و می‌تواند تمام الگوریتم‌های ممکن را ارائه کند، و نشان داد اگر روش مکانیکی هیلبرت وجود داشته باشد، با این مدل قابل توصیف خواهد بود. او سپس از روش‌های خودارجاعی، مانند روش‌های گودل، استفاده کرد تا وجود گزاره‌های تصمیم‌نای‌پذیر را ثابت کند [۲]، یا معادلاً نشان داد مسائلی وجود دارند که هیچ الگوریتمی قادر به حلشان نیست. برنامه‌ی هیلبرت ویران شده بود: محدودیت‌های اساسی و همیشگی برای آنچه می‌توان اثبات کرد و آنچه می‌توان محاسبه کرد وجود خواهد داشت. با این حال هنگامی که کامپیوترا از اشیاء انتزاعی نظری به دستگاه‌های واقعی تبدیل شدند، محققان متوجه شدند که تمایز ساده‌ی تورینگ بین مسائل حل شدنی و حل نشدنی بسیاری از سوالات را بی‌پاسخ گذاشته است. تا دهه‌ی ۱۹۶۰، محققان علوم کامپیوترا الگوریتم‌های سریعی را برای حل کردن برخی مسائل توسعه داده بودند، و این در حالی بود که برای باقی مسائل الگوریتم‌های شناخته‌شده به طرز طاقت‌فرسایی کند بود. چه می‌شد اگر سوال فقط این نبود که آیا مسائل قابل

<sup>۱</sup>Automated theorem proving (ATP)

<sup>۲</sup>David Hilbert

<sup>۳</sup>Axioms

<sup>۴</sup>Unified mathematical theory

<sup>۵</sup>Consistency

<sup>۶</sup>Completeness

<sup>۷</sup>Decidability

<sup>۸</sup>Derivation

<sup>۹</sup>به بیان دقیق‌تر، گودل نشان داد که هیچ نظریه‌ی ریاضی‌ای که اقلًا شامل حساب باشد — با فرض سازگاری این نظریه — نمی‌تواند سازگاری خودش را ثابت کند [۳].

<sup>۱۰</sup>Alan Turing

حل هستند، بلکه این بود که حل آنها چقدر سخت است؟ کارموسینو می‌گوید: «یک نظریه‌ی غنی پدیدار شد و ما دیگر جواب‌ها را نمی‌دانیم.».

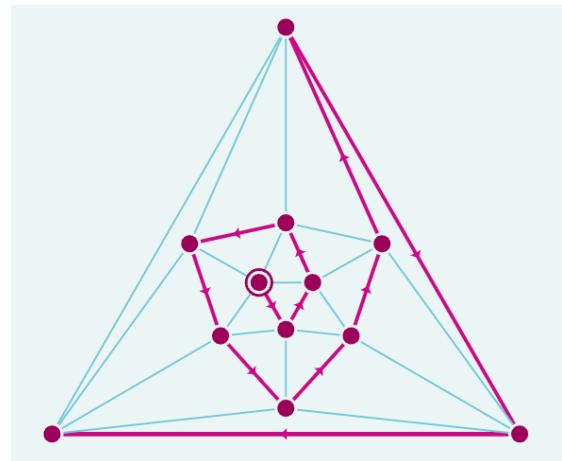
**۲۰.۱. مسیرهای واگرا.** برای این‌که نشان دهیم سوال کردن در مورد «سختی» تا چه اندازه می‌تواند بعرنج باشد، بگذارید دو مسئله در مورد گراف‌ها را که به طور نزدیکی با هم در ارتباطند در نظر بگیریم. گراف‌ها شبکه‌ای از نقاط یا رأس‌ها<sup>۱</sup> هستند که با خطوط یا یال‌هایی<sup>۲</sup> به هم وصل شده‌اند. محققان علوم کامپیوتر از گراف‌ها استفاده می‌کنند تا همه چیز — از محاسبات کوانتومی گرفته تا جریان ترافیک — را با آن مدل کنند.

فرض کنید گرافی به شما داده شده و از شما خواسته شده است که یک مسیر همیلتونی<sup>۳</sup> را در آن بیابید؛ مسیری که از تمام رأس‌ها دقیقاً یک‌بار می‌گذرد. این مسئله به وضوح از نظر تئوری حل شدنی است: تنها تعدادی متناهی مسیر ممکن وجود دارد؛ بنابراین می‌توانید هر یک از مسیرها را بررسی کنید. اگر تعداد کمی رأس وجود داشته باشد این روش ایده‌ی بدی نیست، اما وقتی اندازه‌ی گراف فقط کمی بزرگ‌تر می‌شود، تعداد حالت‌ها از کنترل خارج می‌شود و به سرعت این الگوریتم ساده را بی‌فایده می‌کند. راسل ایمپاگلیازو<sup>۴</sup> می‌گوید: «الگوریتم‌های مسیر همیلتونی

پیچیده‌تری وجود دارند که با این مشکل سرخشنانه‌تر مبارزه می‌کنند، اما زمانی که الگوریتم نیاز دارد که مسئله را حل کند، متناسب با اندازه‌ی گراف همیشه به طور نمایی رشد می‌کند. حتی قبل از این‌که گراف خیلی بزرگ شده باشد، بهترین الگوریتم‌هایی که محققان کشف کرده‌اند نیز نمی‌توانند مسیر را در زمان معقولی پیدا کنند. ایمپاگلیازو یک نظریه‌پرداز پیچیدگی در دانشگاه کالیفرنیا، سن دیگو<sup>۵</sup> است. او ادامه می‌دهد: «و از زمان معقول، منظورم قبل از پایان کیهان است.».

مسئله‌ی مسیر همیلتونی یک خاصیت جالب دیگر هم دارد. اگر کسی ادعا کند که یک مسیر همیلتونی در یک گراف خاص پیدا کرده است، شما می‌توانید به سرعت بررسی کنید که جواب او معتبر است یا نه، حتی اگر گراف خیلی بزرگ باشد. تنها کاری که لازم است انجام دهید این است که مسیر را دنبال کنید و هر رأس را یک‌یکی علامت بزنید، و نهایتاً بررسی کنید تا مطمئن شوید که هر رأس را دوبار علامت نزداید. اگر هیچ رأسی در پایان باقی نمانده باشد، آنگاه مسیر همیلتونی است. زمان لازم برای اجرای این الگوریتم بررسی جواب متناسب با اندازه‌ی گراف است، که آن را در دسته‌ی وسیع‌تری از الگوریتم‌های چندجمله‌ای قرار می‌دهد که زمان‌های اجرایشان متناسب با توابعی چندجمله‌ای از اندازه‌ی گراف افزایش پیدا می‌کند. رشد چندجمله‌ای در مقایسه با رشد نمایی رامتر است، در نتیجه الگوریتم‌های چندجمله‌ای حتی برای گراف‌های بزرگ نیز قابل اجرا خواهد بود. کارموسینو می‌گوید: «آن‌ها به طور چشمگیری کارآمدترند.».

مسئله‌ی مسیر همیلتونی عدم تقارن آشکاری دارد: شما می‌توانید یک جواب درست را با یک الگوریتم چندجمله‌ای سریع تایید کنید، اما برای پیداکردن جواب به یک الگوریتم کند نمایی نیاز دارید. این عدم تقارن ممکن است تعجب آور به نظر نرسد — تشخیص یک شاهکار هنری راحت‌تر از خلق‌کردنش است، یا بررسی یک اثبات ریاضی از اثبات‌کردن یک قضیه جدید راحت‌تر است. با این حال، مسئله‌ی دیگری که بسیار شبیه به مسیر همیلتونی است، کاملاً متفاوت رفتار می‌کند. مجدداً فرض کنید گرافی به شما داده شده است، اما این بار از شما خواسته شده که یک مسیر اویلری<sup>۶</sup> پیدا کنید — مسیری که از تمام یال‌ها دقیقاً یک بار می‌گذرد. مجدداً می‌توان دید که یک الگوریتم چندجمله‌ای برای بررسی جواب‌های ممکن وجود دارد، اما این بار



شکل ۳. مسیر همیلتونی، مسیری در گراف است که از هر رأس دقیقاً یک‌بار می‌گذرد.

<sup>1</sup>Node

<sup>2</sup>Edge

<sup>3</sup>Hamiltonian path

<sup>4</sup>Russell Impagliazzo

<sup>5</sup>University of California, San Diego

<sup>6</sup>Eulerian path

برای حل مسئله هم یک الگوریتم چندجمله‌ای وجود دارد. هیچ عدم تقارنی اینجا نیست. به نظر می‌آید در نظریه‌ی پیچیدگی بعضی مسیرها راحت‌تر از بقیه پیدا می‌شوند.

هر دو مسئله‌ی مسیر همیلتونی و مسیر اویلری در کلاس پیچیدگی  $\mathcal{NP}$  قرار دارند — کلاسی که شامل مسائلی است که تمام جواب‌هایشان با الگوریتمی چندجمله‌ای قابل بررسی هستند. مسئله‌ی مسیر اویلری در کلاس  $\mathcal{P}$  هم قرار می‌گیرد؛ زیرا یک الگوریتم چندجمله‌ای می‌تواند حلقه کند. با این حال آن‌طور که به نظر می‌رسد این برای مسئله‌ی مسیر همیلتونی صادق نیست. چرا این دو مسئله، که به طور اعجاب‌آوری شبیه هستند، به شدت متفاوت‌اند؟ این ذات مسئله‌ی  $\mathcal{P}$  در برابر  $\mathcal{NP}$  است.

۳.۱ به طور جهانی سخت. در وله‌ی اول، به نظر می‌رسید کلاس‌های پیچیدگی دسته‌بندی‌های مناسبی برای مرتب‌سازی مسائلی هستند که به یکدیگر شبیه‌اند اما مستقیماً به هم مرتبط نیستند. هیچ کس شک نکرد که پیداکردن مسیرهای همیلتونی ارتباطی با دیگر مسائل سخت محاسباتی داشته باشد. سپس در ۱۹۷۱، استفن کوک<sup>۱</sup>، ظرف یک سال نقل مکانش به دانشگاه تورنتو بعد از ردشدن درخواست استاد تمامی اش در ایالات متحده، نتیجه‌ی فوق العاده‌ای را منتشر کرد<sup>[۳]</sup>. او مسئله‌ی  $\mathcal{NP}$  خاصی را با یک ویژگی عجیب شناسایی کرده بود: اگر الگوریتمی چندجمله‌ای وجود داشته باشد که بتواند آن مسئله را حل کند، آنگاه می‌تواند هر مسئله دیگر  $\mathcal{NP}$  را نیز حل کند. به نظر می‌رسید مسئله‌ی «جهانی» کوک یکه ستونی است که مسائل ظاهرآ سخت را بالا نگه می‌دارد و آن‌ها را از مسائل راحت زیرشان جدا می‌کند. آن مسئله را حل کنید، و باقی  $\mathcal{NP}$  فرو می‌ریزد و پایین می‌آید. کوک گمان می‌کرد که هیچ الگوریتم سریعی برای مسئله‌ی جهانی اش وجود ندارد، و در نیمه‌های مقاله‌اش گفته بود: «من احساس می‌کنم اثبات این حدس ارزش تلاش قابل ملاحظه‌ای را دارد». به نظر می‌رسد که «تلاش قابل ملاحظه» دست کم گرفتن سختی مسئله بود. تقریباً در همان زمان، یک دانشجوی کارشناسی در اتحاد جماهیر شوروی به نام لئونید لوین<sup>۲</sup>، نتیجه‌ی مشابهی را ثابت کرد<sup>[۴]</sup> و افزون بر این، چندین مسئله‌ی جهانی متفاوت را نیز شناسایی کرد. هم‌چنین نظریه‌پرداز پیچیدگی امریکایی ریچارد کارپ<sup>۳</sup>، ثابت کرد<sup>[۵]</sup> که خاصیت جهانی بودن که توسط کوک (ولوین، اگرچه کوک و کارپ از کارهای لوین تا سال‌ها بعد اطلاعی نداشتند) شناسایی شده بود، به خودی خود جهانی است. تقریباً هر مسئله‌ی  $\mathcal{NP}$  بودن یک الگوریتم چندجمله‌ای — یعنی تقریباً تمام مسائلی که سخت به نظر می‌رسیدند — خاصیت یکسانی داشتند که به  $\mathcal{NP}$ -کامل بودن<sup>۴</sup> معروف شد. این یعنی تمام مسائل  $\mathcal{NP}$ -کامل — مسیر همیلتونی، سودوکو و هزاران چیز دیگر — به معنای دقیقی معادل هستند. ایلانگو می‌گوید: «شما تمام این مسئله‌های طبیعی مختلف را در اختیار دارید و حالا معلوم می‌شود که تمام‌شان به نحوی جادویی سوالی یکسان بودند، و هنوز نمی‌دانیم که آیا همان سوال حل می‌شود یا نه».

حل وفصل کردن سختی هر مسئله‌ی  $\mathcal{NP}$ -کامل برای حل سوال  $\mathcal{P}$  در برابر  $\mathcal{NP}$  کافی خواهد بود. اگر  $\mathcal{P} \neq \mathcal{NP}$  باشد، تمایز میان مسائل سخت و آسان توسط صدھا ستون که به یک اندازه قوی هستند نگه داشته می‌شود. اگر  $\mathcal{P} = \mathcal{NP}$  باشد، فروپاشی این عمارت لرzan در انتظار کوچک‌ترین تلنگر خواهد بود. به این ترتیب کوک، لوین و کارپ مسائل بسیاری را که به نظر می‌رسید نامرتبه باشند، یکی کردنند. حالا نظریه‌پردازان پیچیدگی تنها یک مسئله را باید حل می‌کردن:  $\mathcal{NP} = \mathcal{P}$  یا نه؟ پنجاه سال گذشته و این سوال هم‌چنان بی‌پاسخ مانده است. کابانتس استدلال‌های حول محدودیت‌های محاسبه را به بررسی یک قلمروی وسیع، بدون درک جامعی از عواقب این کار، تشییه می‌کند. موجودی با قدرت محاسباتی نامحدود می‌تواند از قله‌ی کوه به پایین نگاه کند و تمام چشم‌انداز را به یکباره ببیند، اما موجودات فانی ناچیز نمی‌توانند روی چینین مزیتی حساب کنند. او می‌گوید: «ما در پایین آن کوه می‌توانیم برای کمی بهتر دیدن بالا و پایین پیریم».

فرض کنید  $\mathcal{P} = \mathcal{NP}$  باشد. برای اثبات، محققان باید الگوریتم سریعی برای یک مسئله‌ی  $\mathcal{NP}$ -کامل پیدا کنند که ممکن است در گوشه‌ای از آن چشم‌انداز پنهان شده باشد. هیچ ضمانتی وجود ندارد که به این زودی‌ها پیدایش کنند: نظریه‌پردازان پیچیدگی گاه‌ویگاه بعد از دھه‌ها کار، الگوریتم‌های هوشمندانه‌ای برای مسائل ظاهرآ سخت (ونه  $\mathcal{NP}$ -کامل) کشف کرده‌اند. حال فرض کنید که  $\mathcal{NP} \neq \mathcal{P}$  باشد. اثبات آن حتی سخت‌تر به نظر می‌رسد. نظریه‌پردازان پیچیدگی باید نشان دهند که هیچ الگوریتم سریعی نمی‌تواند برای حل مسائل به ظاهر سخت وجود داشته باشد.

ندانستن این که از کجا باید شروع کیم بخشی از مشکل است؛ اما این‌گونه نیست که محققان هیچ تلاشی نکرده باشند. آن‌ها طی دھه‌ها از جهت‌های بسیاری به این مسئله حمله کرده‌اند و در هر مسیر به بن بست رسیده‌اند. کارموسینو می‌گوید:

<sup>1</sup>Stephen Cook

<sup>2</sup>Leonid Levin

<sup>3</sup>Richard Karp

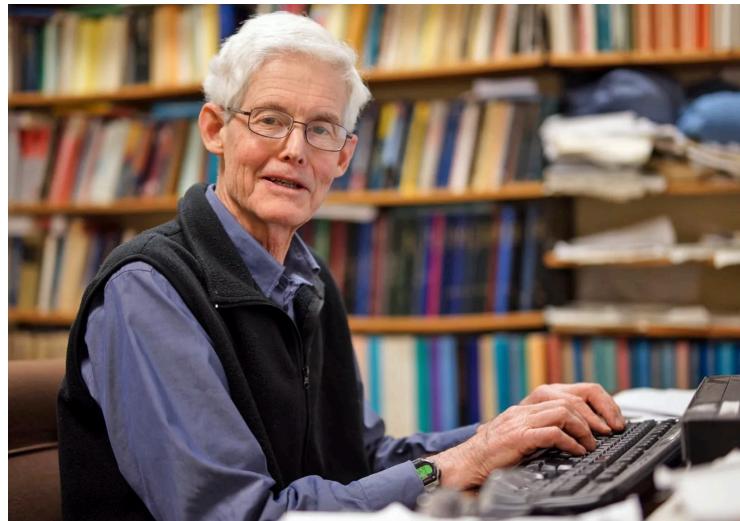
<sup>4</sup>NP-Completeness

«این یکی از آشکارترین حقایق در علوم کامپیوتر نظری است.» و ادامه می‌دهد: «وقتی با پدیده‌های مواجه می‌شوید که این قدر بادوام است، دلتنان می‌خواهد علتیش را بفهمید.».

## ۲. موضع

تا پایان آخرین سال تحصیل، حس کنجکاوی کارموسینو او را از گودل به یک کورس تحصیلات تکمیلی در نظریه‌ی پیچیدگی رسانده بود. او شگفت‌زده بود از این‌که متوجه شده بود زمانی که صرف انجام تکلیف‌هاییش می‌کرد از زمانی که برای انجام پروژه‌ای که از سر علاقه‌ی شخصی انجامش می‌داد — یک برنامه کامپیوتری که ساختار روابطی افسانه‌ها را یاد بگیرد و داستان‌های جدیدی تولید کند — بیشتر بوده است. کارموسینو به یاد می‌آورد: «با خودم فکر کردم: اووه! باید جدی بگیرم». طولی نکشید که او چنان جذب این موضوع شد که مشاورش با ملایمت پیشنهاد کرد در برنامه‌های پس از فارغ‌التحصیلی اش تجدیدنظر کند. کارموسینو می‌گوید: «او به من گفت که اگر بخواهم این

کار — علوم کامپیوتر نظری — را ادامه دهم، کافی است تحصیلات تکمیلی را شروع کنم». او بعد از گرفتن فوق‌لیسانس، در ۲۰۱۲ برای آن‌که دکتری اش را زیر نظر ایمپاگلیازو انجام دهد، به سن دیگو رفت.



شكل ۴. استفن کوک هم‌راستا با کارپ و لوین، مسئله‌ی  $P$  در برابر  $NP$  را در اوایل دهه‌ی ۱۹۷۰ صورت‌بندی کرد.



شكل ۵. تصویری از مارکو کارموسینو. جذابیت نتیجه‌ای که در دهه‌ی ۱۹۹۰ به دست آمده بود، بیست سال بعد الهام‌بخش او شد تا دستاورد مهمی را در فرایچیدگی حاصل کند.

در ابتدا هدف اصلی کارموسینو این بود که یک مقاله‌ی برجسته [۶] از دو دهه قبل را بهتر بفهمد. در آن مقاله الکساندر رازبروف<sup>۱</sup> و استیون رو دیچ<sup>۲</sup> نشان داده بودند که یک استراتژی «طبیعی» برای اثبات  $P \neq NP$  تقریباً به طور قطع شکست می‌خورد، چون این موفقیت با هزینه‌ی گرفای — زیوروکردن کامل رمزنگاری — به دست می‌آید، که محققان آن را بسیار دور از ذهن می‌دانند. محققان دست آورد رازبروف و رو دیچ را به عنوان یک مانع برای این رویکرد رایج برای اثبات  $P \neq NP$  تفسیر کردند.

این «مانع اثبات‌های طبیعی»<sup>۳</sup> تنها یکی از بسیاری موضع شناخته‌شده برای حل کردن مسائل باز در نظریه‌ی پیچیدگی است. هر یک از این

موضع مانند یک راهبند عمل می‌کند؛ هشدار می‌دهد که یک مسیر به ظاهر امیدوارکننده در واقع بنیست است. این موضع، با هم، نشان می‌دهند که هر اثباتی که مسئله‌ی  $P$  در برابر  $NP$  است، باید کاملاً با هر روشی که در گذشته استفاده شده است، متفاوت باشد؛ برای همین است که بیشتر محققان باور دارند جواب هنوز دور از دسترس است. با این حال موضع به ما می‌گویند

<sup>1</sup>Alexander Razborov

<sup>2</sup>Steven Rudich

<sup>3</sup>Natural proofs barrier

که کجا را نگاه نکنیم. ایلانگو می‌گوید: «نظریه‌ی پیچیدگی با مانع بسیار زیادی، هم نفرین شده است و هم مورد لطف قرار گرفته است.».

زمانی که کارموسینو با مانع اثبات‌های طبیعی مواجه شده بود، نزدیک به ۲۰ سال از عمر آن می‌گذشت. با این حال او احساس می‌کرد که مانع اثبات‌های طبیعی آموزه‌های بیشتری برای محققان دارند. آن احساس زمانی روشن شد که او و سه تن از همکارانش با بررسی مانع اثبات‌های طبیعی از منظر فرایپیچیدگی نتیجه‌ای اعجاب‌آور را اثبات کردند. اثبات آن‌ها یکی از محدود نتایج مهمی بود که باعث ایجاد علاقه‌ی جدیدی به فرایپیچیدگی و منجر به سیلی از پیشرفت‌ها در چند سال اخیر شده است. با این حال، برای دنبال‌کردن مسیری که از مانع اثبات‌های طبیعی به فرایپیچیدگی وجود دارد، باید به آن‌جا برگردید که محققان را در دهه‌ی ۱۹۷۰ رها کردیم — آن زمان که برای اولین بار با مسئله‌ی  $P \neq NP$  مواجه شدند. چه چیزی اثبات سختی مسائل را دشوار کرد؟

۱.۲. یک مسیر مداری. در آغاز محققان تلاش کردند به کمک اشکال دیگری از تکنیک‌هایی که تورینگ برای اثبات این‌که برخی مسائل توسط هیچ الگوریتمی حل شدنی نیستند، ثابت کنند  $P \neq NP$  — به این معنا که ثابت کنند برخی مسائل  $NP$  با هیچ الگوریتمی با زمان چندجمله‌ای حل نمی‌شوند. با این حال به سرعت اثباتی را پیدا کردند که نشان می‌داد آن روش‌ها کار نمی‌کنند [۵]، و این اولین مانع بزرگ برای حل سوال  $P \neq NP$  بود. آن‌ها در نتیجه شروع به جستجوی روش‌هایی دیگر کردند و طولی نکشید که روش دیگری در کار کلود شانون<sup>۱</sup>، که معاصر تورینگ بود، پیدا شد.

غريب به نظر می‌رسيد که شانون که در شهر کوچکی در شمال میشیگان بزرگ شده بود، آغازگر عصر اطلاعات باشد. با این حال، او ماهیت میان‌رشته‌ای حوزه‌ی نوظهور علوم کامپیوتر را نمایان کرد، و این در حالی بود که او در مهندسی برق و منطق ریاضی هم چیره‌دست بود. شانون در پایان نامه‌ی ارشدش [۶] نشان داد که چگونه مدارهای ساخته‌شده از کلیدهای الکترومکانیکی می‌توانند نمایان‌گر عبارات منطقی شامل متغیرهای بولی باشند. در این عبارات، متغیرهای بولی با «گیت‌های منطقی» OR و NOT به هم متصل شده‌اند. به عنوان مثال عبارت ساده‌ی  $A \text{ AND } B$  زمانی درست است که هم A و هم B درست باشند، و در غیر این صورت نادرست است. از طرفی دیگر  $A \text{ OR } B$  زمانی درست است که حداقل یکی از دو متغیر درست باشند. گیت NOT ساده‌تر است: مقدار یک متغیر را برعکس می‌کند. با تعدادی کافی از این بلوک‌های پایه‌ای می‌توان هر محاسباتی را انجام داد. ایلانگو می‌گوید: «وقتی در پایان روز به کامپیوتر خود نگاه می‌کنید، کامپیوتر شما دارد چه می‌کند؟ در واقع در حال اجرای یک مدار است.».

کار شانون روش جدیدی را به نظریه‌پردازان پیشنهاد داد تا در مورد دشواری مسائل محاسباتی فکر کنند؛ روشی که به «پیچیدگی مدار»<sup>۲</sup> موسوم است — گرچه مدارهای مورد بحث فقط مفاهیم انتزاعی ریاضی هستند. برای مدتی، محققان فکر می‌کردند که این رویکرد می‌تواند راهی برای حل  $P \neq NP$  باشد، اما در نهایت این مسیر در برابر مانع اثبات‌های طبیعی قرار گرفت.

چهارچوب پیچیدگی مداری مستلزم بازنگری مفاهیم بنیادی مدل محاسباتی تورینگ است. این‌جا محققان به جای مسائل محاسباتی و روش‌های حلشان، توابع بولی و مدارهایی که آن‌ها را محاسبه می‌کنند را مورد مطالعه قرار می‌دهند. یک تابع بولی متغیرهای بولی با مقادیر  $0$  و  $1$  را ورودی می‌گیرد و یکی از دو مقدار  $0$  یا  $1$  را خروجی می‌دهد. مشابه تعریف الگوریتم، یک



شکل ۶. تصویری از کلود شانون. او در پایان نامه‌ی کارشناسی ارشدش مدلی نظری برای محاسبه بر اساس مدارهای الکتریکی توسعه داده بود.

<sup>1</sup>Claude Shannon

<sup>2</sup>circuit complexity

مدار فرایندی را توصیف می‌کند که با توجه به هر ورودی، یک خروجی تعیین می‌شود. رایان ویلیامز<sup>۱</sup> می‌گوید: «به نظر من مردم شروع به کار بر روی پیچیدگی مداری کردند چون به این نتیجه رسیدند که ماشین‌های تورینگ خیلی پیچیده‌اند». ویلیامز یک نظریه‌پرداز پیچیدگی در MIT است. او ادامه می‌دهد: «ما می‌توانیم مدارها را گیت به گیت بررسی کنیم.».

همان‌گونه که برای حل هر مسئله‌ی محاسباتی، ممکن است الگوریتم‌های متفاوتی وجود داشته باشد — که بعضی نسبت به بقیه سریع‌ترند — مدارهای متفاوت زیادی نیز می‌توانند هر تابع بولی را محاسبه کنند — برخی با گیت‌های کم‌تری از بقیه. محققان پیچیدگی مداری یک تابع را با تعداد گیت‌های کوچک‌ترین مداری که آن را محاسبه می‌کند، تعریف می‌کنند. برای یک تابع با تعداد ثابتی متغیر ورودی، پیچیدگی مدار هم یک مقدار ثابت است — برای برخی توابع بیشتر از بقیه.

با این حال، در بسیاری از موارد می‌توان نسخه‌های پیچیده‌تری از یک تابع را با افزایش متغیرهای ورودی اش در نظر گرفت — همان‌گونه که می‌توان مسئله‌ی مسیر همیلتونی را با در نظر گرفتن گراف‌های بزرگ‌تر سخت کرد. این جاست که پژوهشگران همان سوالی را پرسیدند که هنگام مطالعه‌ی زمان اجرای یک الگوریتم پرسیده بودند: آیا با افزایش متغیرهای ورودی، حداقل تعداد گیت‌های لازم برای محاسبه‌ی یک تابع بولی به طور چندجمله‌ای رشد می‌کند یا نمایی؟ این دو دسته از توابع به ترتیب تابع «به راحتی محاسبه‌پذیر» و «به سختی محاسبه‌پذیر» نامیده می‌شوند.

یک تابع بولی به راحتی محاسبه‌پذیر مانند یک مسئله‌ی محاسباتی در کلاس  $\mathcal{P}$  است؛ مسئله‌ای که می‌توان آن را در زمان چندجمله‌ای حل کرد. با این حال توابع مشابه با مسائل  $\mathcal{NP}$ -سخت هم وجود دارند، به این معنا که بهترین روشی که محققان برای محاسبه‌ی نسخه‌های بزرگ‌تر تابع یافته‌اند، مستلزم افزایش نمایی تعداد گیت‌هاست؛ اگرچه درستی جوابشان به راحتی قابل بررسی است. اگر نظریه‌پردازان پیچیدگی می‌توانستند ثابت کنند که واقعاً روش بهتری برای محاسبه چنین توابعی وجود ندارد، این به معنای  $\mathcal{NP} \neq \mathcal{P}$  بود.

این استراتژی‌ای بود که بیشتر نظریه‌پردازان پیچیدگی در دهه‌ی ۱۹۸۰ دنبال کردند، و البته شانس با آن‌ها همراه بود. شانون در ۱۹۴۹ ثابت کرده بود [۱۰] که پیچیدگی مداری تقریباً هر جدول ارزش بولی (که تنها لیستی طولانی از ورودی‌های ممکن و خروجی‌های یک تابع بولی مشخص است) عملاً بیشترین حالت ممکن است. او از یک استدلال ساده‌ی خیره‌کننده استفاده کرد: راههای ممکن برای ترکیب تعداد کمی از گیت‌ها نسبت به راههای ترکیب تعداد زیادی از گیت‌ها بسیار کم‌تر است. آرانسون می‌گوید: «تعدادی کافی از مدارهای کوچک برای گشت‌وگذار وجود ندارد».

به این ترتیب نظریه‌پردازان پیچیدگی خود را در وضعیت عجیبی یافتند. اگر تقریباً تمام جدول‌های ارزش پیچیدگی مداری بالایی دارند، پس تقریباً هر تابع بولی به سختی محاسبه می‌شود. محققان تنها باید یک تابع را شناسایی می‌کردند که در کلاس  $\mathcal{NP}$  هم قرار بگیرد. مگر چقدر می‌تواند سخت باشد؟

۲.۲. برادران رمز. پیشرفت‌ها در آغاز سریع بود. در ۱۹۸۱، سیپسر و دو تن از همکارانش ثابت کردند [۱۲] که اگر از مدارهایی با قیود خاصی بر شیوه‌ی قرارگیری گیت‌های آن‌ها استفاده شود، یک تابع بولی خاص قطعاً محاسبه‌اش مشکل خواهد بود. سیپسر می‌گوید: «روبا یمان این بود که بتوانیم چیزهایی را در مورد این مدل‌های محدودشده اثبات کنیم و سپس بر اساس آنچه آموخته‌ایم با محدودیت‌های کمتر و کمتر کار کنیم».

در ۱۹۸۵، رازبروف قدم بزرگ بعدی را برداشت. او به تازگی تحصیلات تکمیلی اش را در موسکو شروع کرده بود و در حالی که داشت به مسئله‌ای در شاخه‌ی دیگری از ریاضیات می‌پرداخت، به طور اتفاقی به این تلاش پیوسته بود؛ جایی که فهمیده بود



شکل ۷. تصویری از اجزای اصلی کامپیوتر مارک ۱ هاروارد در سال ۱۹۴۴. کلیدهای الکترومکانیکی مشابه با همان‌هایی است که شانون در پایان نامه‌اش بررسی‌شان کرده بود.

<sup>۱</sup>Ryan Williams

حل مسئله‌ی  $P$  در برابر  $\mathcal{NP}$  یک پیشیاز برای کار او است. رازبروف می‌گوید: «من صرفاً خوششانس بودم که نمی‌دانستم این مسئله چقدر سخت است، و گرنه ممکن بود حتی شروعش هم نکنم».

رازبروف مدارهایی را که فقط شامل گیت‌های AND و OR می‌شدند تحلیل می‌کرد، و ثابت کرد [۱۲] که یک تابع خاص، هر طوری که گیت‌ها چیده شوند، به سختی با چنین مدارهایی محاسبه می‌شود — تابعی که  $\mathcal{NP}$ -کامل بودن آن ثابت شده بود. تنها کاری که محققان باید برای حل  $P$  در برابر  $\mathcal{NP}$  انجام می‌دادند این بود که تکنیک‌های رازبروف را به مدارهایی با گیت NOT گسترش دهند. رازبروف می‌گوید: «یک احساس جهانی وجود داشت که یک قدم دیگر، یک ضربه دیگر، و ما قرار است آن را بفهمیم»؛ اما این اتفاق نیفتاد. رازبروف خودش اثبات کرد که روش او، اگر گیت‌های NOT اضافه شوند، شکست خواهد خورد، و هیچ کس نتوانست راه دیگری برای پیش‌روی پیدا کند. با گذشت سال‌ها، او شروع به فکرکردن به این کرد که چه شد که آن مسیر از بین رفت.

در ایالات متحده، رودیج<sup>۱</sup> نیز داشت به همین سوال فکر می‌کرد. او و

ایمپاگلیازو همکلاسی‌های دانشگاه بودند که با هم به تحصیلات تكمیلی رفته بودند. دوستی آن‌ها به دلیل شیفتگی مشترکشان نسبت به اثبات‌های خودارجاعی گodel و تورینگ و پیامدهای آن‌ها برای پایه‌های ریاضیات و علوم کامپیوتر شکل گرفته بود. ایمپاگلیازو می‌گوید: «شوخی ما این بود که قرار بود دکمه‌ای بگیریم که رویش نوشته باشد خودارجاعی».<sup>۲</sup> به عنوان دانشجویان تحصیلات تكمیلی، رودیج و ایمپاگلیازو روی مبانی نظری پیچیدگی رمزگاری کار می‌کردند؛ موضوعی که شاید بهترین انگیزه‌ی عملی را برای فکرکردن روی اثبات  $P \neq \mathcal{NP}$  فراهم می‌کرد. رمزگاران پیام‌های سری را با پیچیدن آن‌ها لای «شبه‌تصادفی بودن»<sup>۳</sup> پنهان می‌کنند. پیامی که به این صورت رمزگذاری شده باشد برای هر شنودگری شبیه به رشته‌ی تصادفی درهم‌برهمی از اعداد خواهد بود، اما هم‌چنان می‌تواند توسط گیرنده رمزگشایی شود. با این حال چگونه می‌توان مطمئن شد که شکستن رمز برای یک شنودگر بالقوه بسیار مشکل خواهد بود؟

این جا جایی است که نظریه‌ی پیچیدگی وارد می‌شود. بیشتر روش‌های رمزگذاری که امروزه استفاده می‌شوند، مبتنی بر مسائل به ظاهر سخت  $\mathcal{NP}$  هستند. یک مهاجم برای رمزگشایی به یک الگوریتم — تاکنون کشف نشده — سریع برای حل کردن مسئله نیاز پیدا خواهد کرد. برای اثبات این‌که این روش‌ها واقعاً امن هستند، کاری که باید انجام دهید این است که نشان دهید  $P \neq \mathcal{NP}$ . آن‌طور که سیپسر می‌گوید، بدون یک اثبات تنها کاری که می‌توانید انجام دهید این است که «امیدوار باشید که آن کسی که دارید سعی می‌کنید چیزی را از او پنهان کنید، ریاضی دان بهتری از شما نباشد».



شکل ۹. الکساندر رازبروف (تصویر چپ) و استیون رودیج (تصویر راست) مانع اثبات‌های طبیعی را کشف کردند، که توضیح می‌داد چرا تلاش‌های پیشین برای اثبات  $P \neq \mathcal{NP}$  به نتیجه نرسیده است.

اگرچه رمزگاری به نوعه خود شکفتانگیز بود، اما به ظاهر عاری از استدلال‌های خودارجاعی‌ای بود که

در ابتدا رودیج و ایمپاگلیازو را به این حوزه کشانده بود. با این حال زمانی که رودیج در تلاش بود که بفهمد چرا پیچیدگی مداری به بن‌بست خورده است، متوجه شد که این دو موضوع چندان هم از هم دور نیستند. استراتژی‌ای که محققان برای اثبات  $P \neq \mathcal{NP}$  به کار گرفته بودند، یک ذات خودمناقض داشت که یادآور گزاره‌ی معروف «این گزاره اثبات‌پذیر نیست» گodel بود

<sup>1</sup> Steven Rudich

<sup>2</sup> pseudorandomness

و رمزنگاری می‌توانست به توضیح چراجی آن کمک کند. رازبروف در همان زمان در روسیه ارتباط مشابهی را کشف کرد. این‌ها بذرهای «مانع اثبات‌های طبیعی» بودند.

کشاکشی که در قلب مانع اثبات‌های طبیعی وجود دارد این است که مسئله‌ی تشخیص توابع با پیچیدگی بالا از توابع با پیچیدگی پایین مشابه مسئله‌ی تشخیص تصادفی‌های واقعی از شبه‌تصادفی‌ها در رمزگشایی پیام‌ها است. برای اثبات  $\mathcal{NP} \neq \mathcal{P}$  دوست داریم که نشان دهیم توابع با پیچیدگی بالا با قطعیت متفاوت از توابع با پیچیدگی پایین‌اند. با این حال از سوی دیگر برای قابل اعتماد بودن امنیت رمزنگاری، تمایل داریم که شبه‌تصادفی‌ها از تصادفی‌های واقعی غیرقابل تمایز باشند. شاید نمی‌توانیم هر دو را با هم داشته باشیم.

**۳.۲ یک لطیفه‌ی بی‌رحمانه.** در ۱۹۹۴ رازبروف و رو دیج متوجه شدند که به بینش‌های مشابهی رسیده‌اند و شروع به هم‌کاری کردند تا نتایجشان را باهم ترکیب کنند. آن‌ها در ابتدا مشاهده کردند که تمام تلاش‌های سابق برای اثبات  $\mathcal{NP} \neq \mathcal{P}$  با استفاده از پیچیدگی مداری، یک استراتژی کلی را اتخاذ کرده‌اند: ویژگی خاصی از یک تابع بولی  $\mathcal{NP}$ -کامل را شناسایی کنید، سپس اثبات کنید که هیچ تابع به راحتی محاسبه‌پذیری نمی‌تواند آن خاصیت را داشته باشد. این نشان خواهد داد که محاسبه‌ی تابع  $\mathcal{NP}$ -کامل انتخاب شده واقعاً سخت است و  $\mathcal{P} \neq \mathcal{NP}$  را ثابت می‌کند.

سیپسر، رازبروف و دیگران همین استراتژی را با موفقیت به کار گرفته بودند تا نتایج محدودتر خود را اثبات کنند و در تمام حالات، بیشتر توابع بولی دارای آن ویژگی خاص بودند که شناسایی شده بود. رازبروف و رو دیج برای اشاره به حالتی که آن ویژگی در اکثر توابع وجود دارد، اصطلاح «اثبات طبیعی» را ابداع کردند؛ صرفاً به این دلیل که هیچ جایگزین شناخته‌شده‌ای وجود نداشت. اگر اثبات‌های «غیرطبیعی» امکان‌پذیر باشند، باید بسیار ناشهودی باشند، و مستحق این نام خواهند بود.

پس از آن بود که رازبروف و رو دیج نتیجه‌ی اصلی‌شان را ثابت کردند: یک اثبات طبیعی برای  $\mathcal{P} \neq \mathcal{NP}$  مستلزم فهم گسترده‌ای خواهد بود از چگونگی تمایز توابعی که به راحتی محاسبه‌پذیرند از توابع به سختی محاسبه‌پذیر، و این درک می‌تواند انگیزه‌بخش الگوریتم سرعی برای شناسایی توابع به راحتی محاسبه‌پذیر باشد. اگر نظریه پردازان پیچیدگی در یک اثبات طبیعی  $\mathcal{P} \neq \mathcal{NP}$  موفق شده بودند، راهی تقریباً عاری از خطأ را نیز کشف می‌کردند که با نیمنگاهی به یک جدول ارزش دلخواه بتوان تعیین کرد که تابع متناظر آن پیچیدگی مداری زیاد یا کمی دارد — و البته این، نتیجه‌ای بسیار قوی‌تر و کلی تر است از آن چیزی که آن‌ها قصد داشتند ثابت کنند. کارموسینو می‌گوید: «تقریباً نمی‌توانید جلویش را بگیرید و بیشتر از آن چیزی که برایش چانه زدید به دست می‌آورید».

این مشابه آن است که تلاش کرده باشید یک گفته را صحبت‌سنگی کنید، ولی هر تلاشتان تبدیل به طرح ساخت یک دروغ‌سنجد همه‌کاره شده باشد — آن‌قدر خوب به نظر می‌رسد که باورکردنی نیست حقیقت داشته باشد. برای نظریه‌پردازان پیچیدگی، قدرت عجیب اثبات‌های طبیعی باعث شد که موفقیت کمتر محتمل به نظر برسد. اما اگر چنین اثباتی موفق می‌شد، به دلیل ارتباط میان پیچیدگی مداری و شبه‌تصادفی بودن، نتیجه‌ی غیرمنتظره‌اش خبر بدی برای رمزنگاری می‌بود.

برای درک این ارتباط، جدول ارزش یک تابع بولی با تعداد زیادی متغیر ورودی را تصور کنید. اگر آن تابع بولی پیچیدگی مداری زیادی داشته باشد، آن لیست طولانی ارزش‌ها اساساً از یک رشته‌ی واقعاً تصادفی از  $0$  و  $1$  قابل تمایز خواهد بود — رشته‌ای که مثلاً با پشت هم سکه‌انداختن به دست آمده باشد. با این حال اگر پیچیدگی مداری تابع کم باشد، آن رشته حتی اگر پیچیده به نظر برسد هم باید یک توصیف ساده و مختصر داشته باشد. این موضوع آن را بسیار شبیه به رشته‌های شبه‌تصادفی مورد استفاده در رمزنگاری می‌کند — رشته‌هایی که توصیف مختصرشان همان پیام مخفی مدفون در ظاهر تصادفی آن‌ها است. نتیجتاً دست آورده رازبروف و رو دیج نشان داد که هر اثبات طبیعی  $\mathcal{P} \neq \mathcal{NP}$  منتج به الگوریتم سرعی می‌شود که می‌تواند رشته‌های شبه‌تصادفی‌ای که دارای پیامی مخفی هستند را از تصادفی‌های واقعی تمایز دهد. به این ترتیب رمزنگاری امن غیرممکن خواهد شد، دقیقاً برعکس آن چیزی که محققان امیدوار بودند از اثبات  $\mathcal{P} \neq \mathcal{NP}$  به دست آورند.

از سوی دیگر اگر رمزنگاری امن ممکن باشد، آن‌گاه اثبات‌های طبیعی استراتژی ممکنی برای اثبات  $\mathcal{NP} \neq \mathcal{P}$  نخواهد بود — که پیش‌نیازی برای رمزنگاری است. این لب مطلب مانع اثبات‌های طبیعی بود. به نظر می‌رسید که نظریه‌پردازان پیچیدگی مخاطب یک لطیفه‌ی بی‌رحمانه بودند. کاباتنس می‌گوید: «اگر به سختی باور دارید، آنگاه باید بپذیرید که اثبات سختی سخت است».

۴.۲ پیش به سوی متأورس. ارتباط میان پیامدهای حدس  $\mathcal{NP} \neq P$  و سختی اثبات آن جالب، اما سر درآوردن از آن دشوار بود. یک دشواری این بود که مانع اثبات‌های طبیعی تنها جلوی یک رویکرد اثبات  $\mathcal{NP} \neq P$  را گرفت. دیگر این که، سختی اثبات  $\mathcal{NP} \neq P$  را نه به خود  $P \neq \mathcal{NP}$ ، بلکه به وجود رمزنگاری امن مرتبط کرد — به یک مسئله‌ی مشابه اما نه کاملاً معادل. برای فهم درست این ارتباط، محققان باید با فرایپیچیدگی<sup>۱</sup> خوبگیرند. ویلیامز می‌گوید: «این شهود وجود دارد که چون  $P \neq \mathcal{NP}$ ، پس اثبات  $\mathcal{NP} \neq P$  باید خیلی سخت باشد؛ اما برای این که به این شهود معنایی بدھید، باید به امر اثبات گزاره‌ای مانند  $\mathcal{NP} \neq P$  به عنوان یک مسئله‌ی محاسباتی نگاه کنید.».



شكل ۱۰. تصویری از والنتین کابانتس، که در دوران کارشناسی ارشدش مقاله‌ی تأثیرگذاری درباره‌ی یک مسئله‌ی اساسی در فرایپیچیدگی — که آن را مسئله‌ی حداقل اندازه‌ی مدار (MCSP) نامید — نوشته.

این کاری بود که کابانتس به عنوان یک دانشجوی تحصیلات تکمیلی انجام داد. او دو سال پس از سقوط جماهیر شوروی در اوکراین به دنیا آمد. در آشفتگی‌های پس از آن واقعه، او فرصت کمی داشت تا مباحث نظری‌ای که بیشتر به آن‌ها علاقمند بود را پی بگیرد. کابانتس به یاد می‌آورد: «من می‌خواستم کار آکادامیک‌تری انجام دهم، و به علاوه دوست داشتم دنیا را بگردم». او برای تحصیلات تکمیلی به کانادا رفت و آن‌جا بود که با مانع اثبات‌های طبیعی آشنا شد. کابانتس، مانند کارموسینو، مجدوب این نتیجه شده بود. او می‌گوید: «وجود چنین ارتباطی، خیلی عمیق به نظر می‌رسید».

او در ۲۰۰۰، اواخر تحصیلات تکمیلی اش، با صحبت‌هایی که با جین بی کای<sup>۲</sup> — یک نظریه‌پرداز پیچیدگی که در آن زمان برای فرصت مطالعاتی به تورنتو آمده بود — داشت، متوجه شد که موضوع مانع اثبات‌های طبیعی مدام در مکالماتشان مطرح می‌شود. آن‌ها تصمیم گرفتند که به عنوان یک بنیست بلکه به عنوان یک دعوت‌نامه نگاه کنند؛ فرضی برای بررسی دقیق این که چقدر سخت است که ثابت کنند مسائل سخت هستند. مقاله‌ای که آن‌ها در آن دیدگاه جدید را ارائه کردند [۱۴] به یکی از تأثیرگذارترین کارهای اولیه در حوزه‌ی نوظهور فرایپیچیدگی تبدیل شد.

مقاله‌ی کابانتس و کای روی یک مسئله‌ی محاسباتی مرکز می‌کند که در فرمول‌بندی مانع اثبات‌های طبیعی رازبروف و رودیچ به آن اشاره شده بود: با توجه به جدول ارزش یک تابع بولی، بررسی کنید که آیا پیچیدگی مداری آن زیاد است یا کم. آن‌ها آن مسئله را مسئله‌ی حداقل اندازه‌ی مدار<sup>۳</sup> یا MCSP نامیدند. MCSP یک مسئله‌ی بنیادی فرایپیچیدگی است: یک مسئله‌ی محاسباتی که موضوع مورد بحث آن نظریه‌ی گراف یا موضوع خارجی دیگری نیست، بلکه خود نظریه‌ی پیچیدگی است. در واقع، این مسئله مانند نسخه‌ای کمی از سوالی است که نظریه‌پردازان پیچیدگی را به درگیرشدن با  $\mathcal{NP}$  در مقابل  $\mathcal{NP}$  با استفاده از رویکرد پیچیدگی مدار در دهه‌ی ۱۹۸۰ سوق داد: کدام توابع بولی سخت محاسبه می‌شوند و کدام راحت؟ ایمپاگلیازو می‌گوید: «اگر یک الگوریتم MCSP بیابیم، مانند این خواهد بود که راهی برای ماشینی کردن کاری که در نظریه پیچیدگی انجام

می‌دهیم پیدا کرده باشیم؛ حداقل باید بینش فوق‌العاده‌ای در مورد این که چگونه کارمان را بهتر انجام دهیم، به ما بدهد.» نظریه‌پردازان پیچیدگی نگران این نیستند که این الگوریتم جادویی آن‌ها را از کار بیندازد. در واقع، آن‌ها اصلاً فکر نمی‌کنند که چنین چیزی وجود داشته باشد؛ چرا که رازبروف و رودیچ نشان دادند که هر الگوریتم چنینی برای تمیزدادن جدول ارزش‌های با پیچیدگی زیاد از پیچیدگی کم رمزنگاری را غیرممکن می‌کند. این یعنی MCSP احتمالاً یک مسئله‌ی محاسباتی سخت است، اما چقدر سخت؟ آیا مانند مسئله‌ی مسیر همیلتونی و تقریباً هر مسئله‌ی دیگری که محققان در دهه‌ی ۱۹۶۰ با آن درگیر بودند،  $\mathcal{NP}$ -کامل است؟ معمولاً پاسخ دادن به «چقدر سخت است؟» برای مسائل کلاس  $\mathcal{NP}$  آسان است؛ اما به نظر می‌رسید برای

<sup>1</sup>meta-complexity

<sup>2</sup>Jin-Yi Cai

<sup>3</sup>minimum circuit size problem

MCSP چیز دور از ذهنی باشد. کابانتس می‌گوید «ما تعداد کمی مسائل شناور داریم که با این که سخت به نظر می‌رسند، هنوز به جزیره‌ی  $\mathcal{NP}$ -کامل متصل نشده‌اند.».

کابانتس می‌دانست که او و کای اولين کسانی بودند که مسئله‌ای که آن‌ها MCSP نامیده بودند را بررسی کرده باشند. ریاضی‌دانان شوروی در ابتدای دهه‌ی ۱۹۵۰، در تلاشی اولیه برای درک دشواری ذاتی مسائل مختلف محاسباتی، مسئله‌ی بسیار مشابهی را مطالعه کرده بودند. لئونید لوین<sup>۱</sup> در دهه‌ی ۱۹۶۰ در جریان مطالعه‌ی چیزی که داشت به نظریه  $\mathcal{NP}$ -کامل بودن تبدیل می‌شد، با این مسئله گلاویز شده بود؛ اما توانسته بود  $\mathcal{NP}$ -کامل بودنش را ثابت کند، و مقاومت ماندگارش را بدون آن منتشر کرده بود. پس از آن برای ۳۰ سال آن مسئله توجه افراد کمی را به خود جلب کرد، تا این که کابانتس و کای به ارتباط آن با مانع اثبات‌های طبیعی اشاره کردند. کابانتس انتظار نداشت خودش این سوال را حل کند. او در عوض می‌خواست بررسی کند که چرا اثبات این که این مسئله‌ی ظاهراً سخت درباره‌ی محاسباتی واقعاً سخت بوده، انقدر مشکل بوده است. راهول ساتتانام<sup>۲</sup>، یک نظریه‌پرداز پیچیدگی در دانشگاه آکسفورد<sup>۳</sup> می‌گوید: «این به یک معنا فرافرای پیچیدگی است.».

اما آیا قرار بود این سختی تا آخر مسیر وجود داشته باشد، یا این که حداقل یک راه برای درک این که چرا محققان در اثبات  $\mathcal{NP}$ -کامل بودن MCSP موفق نشده بودند وجود داشت؟ کابانتس کشف کرد که بله، دلیلی وجود دارد: سختی درک‌دنن پیچیدگی مداری مانند یک مانع برای هر استراتژی شناخته‌شده برای اثبات  $\mathcal{NP}$ -کامل بودن MCSP عمل می‌کند — مسئله‌ای که خودش درباره‌ی سختی درک پیچیدگی مداری است. گریزگاهی از منطق خودمنافق و پیچ دریچ مانع اثبات‌های طبیعی نبود. ممکن است که  $\mathcal{NP}$ -کامل نباشد، اما این نیز دور از ذهن به نظر می‌رسد؛ چرا که برخی از انواع ساده‌تر مسئله پیشتر به عنوان مسائل  $\mathcal{NP}$ -کامل شناخته شده‌اند. ایمپاگلیازو می‌گوید: «مشکل این است که فقط جای خوبی برای فرادرادن آن نداریم، به گونه‌ای که مستقیماً آن را با تمام مسائل دیگری که مطالعه می‌کنیم مرتبط کند.».

کابانتس رفتار عجیب MCSP را روشن کرده بود، اما نمی‌دانست چگونه جلوتر برود. جریان تحقیقات فرای پیچیدگی خیلی کند شد. با این حال ۱۶ سال بعد، زمانی که محققان رابطه‌ی غیرمنتظره‌ای را با یک سوال بنیادی دیگر کشف کردند، دوباره شکوفا شد: حل مسائل چقدر سخت است، اگر فقط بخواهید در «بیشتر اوقات» پاسخ صحیح دریافت کنید؟

۵.۲. جنگ جهان‌ها. برای مسائل روزمره راهکارهایی که فقط بیشتر اوقات جواب می‌دهند هم کفایت می‌کند. مثلاً ما رفت و آمد هایمان را با الگوهای ترافیکی معمولی برنامه‌ریزی می‌کنیم نه با سناریوهای بدترین حالت. بیشتر نظریه‌پردازان پیچیدگی سخت‌تر راضی می‌شوند: آن‌ها تنها وقتی راضی می‌شوند مسئله‌ای را آسان اعلام کنند که الگوریتم سریعی پیدا کنند که جواب درست را برای هر ورودی ممکن به دست آورد. این مواجهه‌ی استاندارد مسائل را بر اساس آنچه محققان، پیچیدگی «بدترین حالت»<sup>۴</sup> می‌نامند، طبقه‌بندی می‌کند. با این حال یک نظریه پیچیدگی «حالات میانگین»<sup>۵</sup> هم وجود دارد که در آن مسائل ساده در نظر گرفته می‌شوند اگر الگوریتم سریعی که جواب درست را برای بیشتر ورودی‌ها به دست آورد، وجود داشته باشد.

این تمایز برای رمزنگاران اهمیت دارد. یک مسئله‌ی محاسباتی را تصور کنید که تقریباً برای هر ورودی، به غیر از چند حالت سرسرخت که بهترین الگوریتم در آن‌ها شکست می‌خورد، به راحتی حل می‌شود. پیچیدگی بدترین حالت آن مسئله را سخت در نظر می‌گیرد، اما برای رمزنگاری این مسئله بی‌فائده است: چه فایده‌ای دارد اگر رمزگشایی فقط برای برخی از پیام‌های شما دشوار باشد؟ این در واقع لوین بود که یک دهه بعد از کار پیشگامانه‌اش در  $\mathcal{NP}$ -کامل بودن، مطالعه‌ی دقیق پیچیدگی حالت میانگین را آغاز کرد. در آن فاصله، او با مقامات شوروی درگیر شده بود. لوین یک دردرساز بی‌پرده بود که گه‌گاه فعالیت‌های میهن‌دوستانه‌ی خود در گروه جوانان حزب کمونیست را لکه‌دار می‌کرد. لوین در سال ۱۹۷۲ به دلایل آشکارا سیاسی از مدرک دکترا محروم شد. ایمپاگلیازو می‌گوید: «برای این که به عنوان یک محقق جوان در جماهیر شوروی موفق شوید، نمی‌توانید خیلی صاحب‌نظر باشید، و تصور این که لئونید صاحب‌نظر نباشد سخت است.».

لوین در ۱۹۷۸، به ایالات متحده مهاجرت کرد و در نیمه‌ی دهه ۱۹۸۰، توجه خود را به پیچیدگی حالت میانگین معطوف کرد. او کار با دیگران را شروع کرد — از جمله ایمپاگلیازو که در آن زمان دانشجوی تحصیلات تکمیلی بود — تا نظریه را بیشتر

<sup>1</sup>Leonid Levin

<sup>2</sup>Rahul Santhanam

<sup>3</sup>Oxford

<sup>4</sup>worst-case

<sup>5</sup>average-case

توسعه دهد. با وجود این که آن‌ها پیشرفت کردند، ایمپاگلیازو متوجه شد اغلب محققان هرچند هر کدام حرف خودش را می‌زد، گمان می‌کردند که همگی دارند از یک موضوع سخن می‌گویند. او می‌خواست همه را همنظر کند، و البته این که مقاله‌های لوین به صورت مشهوری مختصر بود کمکی به آن نمی‌کرد — آن مقاله‌ای لوین [۱۵] که آغازگر حوزه‌ی پیچیدگی حالت میانگین بود، کمتر از دو صفحه بود. ایمپاگلیازو می‌گوید: «من می‌خواستم کارهای لئونید را به اصطلاحات فنی تر و در دسترس‌تر ترجمه کنم.». او تصمیم گرفت پیش از شیرجه‌زدن در ریاضیات ماجرا، با یک نمای کلی کوتاه و سرزنشه شروع کند. «آن بیشتر مقاله را گرفت، و البته این تنها بخشی است که همه به یاد می‌آورند.».

آن مقاله [۱۶] در ۱۹۹۵ منتشر شد و فوراً به یکی از بهترین‌های حوزه‌ی خودش تبدیل شد. ایمپاگلیازو نام‌های عجیب‌وغریبی را برای پنج جهانی که به وسیله‌ی درجه‌های متفاوت سختی‌های محاسباتی و قابلیت‌های رمزنگاری متفاوت تمایز شده بودند، ابداع کرده بود [۱۷]. ما در یکی از این جهان‌ها زندگی می‌کنیم ولی نمی‌دانیم کدام.



شکل ۱. لئونید لوین (تصویر راست) مطالعه‌ی پیچیدگی حالت میانگین را در اواسط دهه هشتاد آغاز کرد. راسل ایمپاگلیازو (تصویر چپ) بعدها این موضوع را در مقاله‌ای درباره‌ی پنج جهان محاسباتی که ممکن است در آن زندگی کنیم، درسترس‌تر کرد.

از زمانی که مقاله‌ای ایمپاگلیازو منتشر شد، محققان آرزوی حذف بخش‌هایی از متاورس مینیاتوری او را داشتند؛ با محدود کردن فضای احتمالات به وسیله‌ی ثابت کردن این که وجود برخی از جهان‌ها ممکن نخواهد بود. دو جهان هدف‌های وسوسه‌کننده‌ای بودند: آن‌هایی که رمزنگاری در آن‌ها ناممکن بود حتی اگر  $\mathcal{P} \neq \mathcal{NP}$  باشد. در یکی از آن جهان‌ها، به نام هیورستیکا<sup>۱</sup>، تمام مسائل  $\mathcal{NP}$ -کامل برای بیش‌تر ورودی‌ها به راحتی حل می‌شوند؛ اما الگوریتم‌های سریع گاهی اشتباه می‌کنند. بنابراین این مسائل با استانداردهای نظریه‌ی پیچیدگی

بدترین حالت، سخت در نظر گرفته می‌شوند. این جهانی است که رمزنگاری در آن ناممکن است چون تقریباً هر کدی به راحتی شکسته می‌شود. در جهان دیگر، به نام پسیلند<sup>۲</sup>، رمزنگاری به دلیل دیگر ناممکن است: تمام مسائل در حالت میانگین سخت است، اما رمزگذاری یک پیام آن را حتی برای گیرنده‌ی موردنظر هم ناخوانا می‌کند.

علوم می‌شود این دو جهان ارتباط تنگاتنگی با مسائل فرایمپاگلیازو دارند؛ به طور خاص، سرنوشت هیورستیکا به سوال طولانی‌مدت  $\mathcal{NP}$ -کامل بودن MCSP پیوند خورده است. آن سوالی که خیلی وقت پیش کابانتس مذوب آن شده بود و لوین را هاج و اوج کرده بود دیگر فقط یک کنجدکاوی نبود: سرنوشت یک جهان در خطر است.

برای رد کردن هیورستیکا، محققان باید تمایز میان پیچیدگی بدترین حالت و حالت میانگین را از بین ببرند، که این یعنی باید ثابت کنند که هر الگوریتم فرضی که یک مسئله‌ای  $\mathcal{NP}$ -کامل را به درستی برای بیش‌تر ورودی‌ها حل می‌کند، در واقع برای همه حالت‌ها جواب می‌دهد. این نوع ارتباط، که تحويل بدترین حالت به حالت میانگین نام دارد<sup>۳</sup>، برای مسائل خاصی وجود دارد، اما هیچ کدام‌شان  $\mathcal{NP}$ -کامل نیستند؛ بنابراین آن نتایج دلالت بر چیز کلی‌تری ندارند. نابودی هیورستیکا رمزنگاران را تا نیمه‌راه تحقق روایی رمزگذاری امن بر اساس فرض  $\mathcal{P} \neq \mathcal{NP}$  پیش می‌برد؛ اما نابود کردن یک دنیا کار کوچکی نیست. در ۲۰۰۳، دو نظریه پرداز پیچیدگی نشان دادند [۱۸] که رویکردهای موجود برای اثبات کردن تحويل بدترین حالت به حالت میانگین برای مسائل  $\mathcal{NP}$ -کامل شناخته شده پیامدهای عجیب و غریبی خواهد داشت که پیشنهاد می‌دهد احتمالاً چنین اثبات‌هایی ممکن نخواهند بود.

محققان باید یک رویکرد دیگر پیدا کنند و اکنون فکر می‌کنند MCSP همان مسئله‌ای است که نیاز دارند. با این حال این برای بیش از یک دهه روشن نشد. اولین توجه به این ارتباط از شیفتگی مداوم کارموسینو به مانع اثبات‌های طبیعی پدیدار شد.

<sup>1</sup>Heuristica

<sup>2</sup>Pessiland

<sup>3</sup>worst-case to average-case reduction

### ۳. فرصت‌ها

کارموسینو برای اولین بار به عنوان دانشجوی تحصیلات تکمیلی از طریق مقاله‌ی ۱۳ کابانتس و چهار محقق دیگر [۱۹] با تحقیقات فراییچیدگی رویرو شد، که رویکرد مانع اثبات‌های طبیعی را که کابانتس بیش از یک دهه قبل پیشگام آن شده بود، بیش‌تر توسعه داده بودند. این مقاله فقط اعتقاد او را که هنوز چیزهای زیادی برای بادگیری از مقاله‌ی کلاسیک رازبروف و رودیچ وجود دارد، راسخ‌تر کرد. کارموسینو می‌گوید: «من در آن زمان شیفتنه‌ی آن مقاله شده بودم، و هنوز هم هیچ چیز عوض نشده است.».

شیفتگی کارموسینو بالاخره در جریان بازدیدی از یک کارگاه یک ترمه در دانشگاه کالیفرنیا، برکلی —جایی که بیش‌تر وقت خود را به صحبت با ایمپاگلیازو، کابانتس و آتنونینا کولوکولوفا<sup>۱</sup>، یک نظریه پرداز پیچیدگی در دانشگاه مموریال نیوفاندلند<sup>۲</sup> که با کابانتس در مقاله‌ی ۱۳۰۱۶ اش همکاری کرده بود، صرف می‌کرد —ثمره داد. کارموسینو قبل از نفرشان کار کرده بود، و آن همکاری موفقیت‌آمیز به او این اعتمادبهنه نفس را داد که بارها و بارها سوالاتی درباره‌ی موضوعی که بیش از همه مجدوب آن شده بود را با آن‌ها مطرح کند. کابانتس به یاد می‌آورد: «او مردم را اذیت می‌کرد، البته به نحوی سازنده.» در آغاز کارموسینو ایده‌های جدیدی داشت برای اثبات‌کردن *NP*-کامل بودن نسخه‌ای از MCSP که در مقاله‌ی رازبروف و رودیچ درباره‌ی مانع اثبات‌های طبیعی آمده بود؛ اما آن ایده‌ها نتیجه نداد. در عوض، یک اظهارنظر بی‌مقدمه‌ی ایمپاگلیازو باعث شد که این چهار محقق متوجه شوند که مانع اثبات‌های طبیعی الگوریتم‌های قوی‌تری از آنچه که هر کسی تصور کرده بود به دست می‌دهد. ظاهراً یک نقشه‌ی مخفی روی راه‌بند حک شده بود.

این چهار محقق در مقاله‌ای در سال ۱۶۰۱ ثابت کردند [۲۰] که نوع خاصی از الگوریتم MCSP حالت میانگین را می‌توان برای ساختن یک الگوریتم بدترین حالت برای شناسایی الگوهای پنهان در رشته‌های به ظاهر تصادفی اعداد استفاده کرد —کاری که محققان علوم کامپیوتراز آن به عنوان یادگیری<sup>۳</sup> یاد می‌کنند. این نتیجه‌ی قابل توجهی است؛ چرا که یادگیری شهوداً کار سخت‌تری از رده‌بندی دودویی (با پیچیدگی زیاد یا کم) که توسط یک الگوریتم MCSP انجام شده باشد، به نظر می‌رسد، و در کمال تعجب این نتیجه پیچیدگی بدترین حالت یک کار را به پیچیدگی حالت میانگین یک کار دیگر مرتبط کرد. ایمپاگلیازو می‌گوید: «به هیچ وجه روش نبود که چنین ارتباطی اصلاً می‌تواند وجود داشته باشد.».

یک الگوریتم سریع برای MCSP مدارهای بولی در حالت

کلی کاملاً فرضی و نظری است: تا زمانی که نشان داده نشود MCSP یک مسئله‌ی محاسباتی راحت است، علی‌رغم تمام شواهد ضد آن، این الگوریتم نمی‌تواند وجود داشته باشد، و این یعنی الگوریتم یادگیری‌ای که در مقاله‌ی این چهار محقق به آن اشاره شده، به همان اندازه فرضی و نظری است.

با این حال برای برخی حالت‌های ساده‌تر MCSP و زمانی که مدار محدودیت‌های خاصی داشته باشد، برای تشخیص جداول ارزش با پیچیدگی زیاد از جداول با پیچیدگی کم سال‌هاست که الگوریتم‌های سریعی شناخته شده‌اند. مقاله‌ی کارموسینو، کابانتس، کولوکولوفا و ایمپاگلیازو نشان داد که این الگوریتم‌ها را می‌توان به الگوریتم‌های یادگیری‌ای تبدیل کرد که به طور مشابه محدود شده‌اند، اما هنوز قوی‌تر از هر الگوریتمی هستند که محققان سابقاً با این سطح از دقت نظری، درک کرده‌اند. ایلانگو می‌گوید: «این ویرگی خودارجاعی آن‌ها به نحوی شما را قادر می‌سازد که کارهایی را انجام دهید که ظاهراً نمی‌توانید با مسائل استاندارد بیش‌تری انجام دهید.».



شکل ۱۲. آتنونینا کولوکولوفا در ۱۶۰۱ همراه با کارموسینو، ایمپاگلیازو و کابانتس ارتباطی شگفت‌انگیز میان MCSP و یادگیری را ثابت کرد، که موجب جلب توجه‌ها به سمت فراییچیدگی شد.

<sup>1</sup> Antonina Kolokolova

<sup>2</sup> Memorial University of Newfoundland

<sup>3</sup> Learning

این دست آورد توجه نظریه پردازان پیچیدگی را که روی موضوعات دیگر کار می کردند، به خود جلب کرد. همچنین پیش نمایشی از ارتباطات بیشتر میان فرایپیچیدگی و پیچیدگی حالت میانگین را به نمایش گذاشت که در سال های آینده ظاهر خواهد شد. بیش از همه، این نتیجه گواهی بود بر این که محققان با پرسیدن سوالات ساده در مورد موانع که در ابتدا فقط مانع پیشرفت آن ها می شوند، تا چه حد می توانند پیش بروند. ایمپاگلیازو می گوید: «این نوع از دوگانی یک تم در جریان حداقل ۲۰ تا ۴۰ سال اخیر پیچیدگی است. موانع اغلب فرصت هستند».

۱.۳. اعتبارهای «جزئی». از زمانی که کارموسینو و همکارانش مقاله شان را منتشر کرده اند، پیشرفت ها شتاب گرفته است. کولوکولوفا می گوید: «اتفاقات جدیدی در حال رخدادن است، و محققان جوان خیلی خیلی باهوش زیادی وجود دارند».<sup>۱</sup> ایلانگو یکی از این محققان جوان است. او در سه سال اول تحصیلات تکمیلی اش، به مسئله باز و هولناک اثبات  $\mathcal{NP}$ -کامل بودن MCSP با استفاده از یک استراتژی دو وجهی حمله کرد: در حالی که  $\mathcal{NP}$ -کامل بودن حالت های مشابه MCSP [۲۱، ۲۲] را ثابت می کرد — همان گونه که محققان پیچیدگی مداری در دهه ۱۹۸۰ به  $\mathcal{P}$  در برابر  $\mathcal{NP}$  حمله کردند — در حال ثابت کردن  $\mathcal{NP}$ -کامل بودن حالت های پیچیده تر [۲۳] نیز بود، که شهوداً سخت تر به نظر می رسد و در نتیجه احتمالاً راحت تر می توان اثبات کرد که مشکل اند. ایلانگو علاقه ای خود به فرایپیچیدگی را مدیون اریک آلندر<sup>۲</sup> می دارد. او یک نظریه پرداز پیچیدگی در دانشگاه راتگرز<sup>۳</sup> است و یکی از معدود محققانی است که به کار روی فرایپیچیدگی در دهه ۲۰۰۰ و اوایل دهه ۲۰۱۰ ادامه دادند. ایلانگو می گوید: «اشتیاق او واگیردار بود».

یک محقق دیگر که از آلندر الهام گرفت، شوئیچی هیراها را<sup>۴</sup> بود، که اکنون استادی در موسسه ملی انفورماتیک توکیو<sup>۵</sup> است. او در ۲۰۱۸، در حالی که هنوز یک دانشجوی تحصیلات تکمیلی بود، سطح واقعی ارتباط میان فرایپیچیدگی و پیچیدگی حالت میانگین را که کارموسینو و همکارانش کشف کرده بودند، آشکار کرد. آن چهار محقق ارتباط میان پیچیدگی حالت میانگین یک مسئله (MCSP) و پیچیدگی بدترین حالت مسئله ای دیگر (یادگیری دودویی) را پیدا کرده بودند. هیراها را تکنیک های آن ها را بیشتر توسعه داد تا برای MCSP یک تحويل بدترین حالت به حالت میانگین به دست آورد [۲۴]. دست آورد او نتیجه می دهد که یک الگوریتم MCSP حالت میانگین فرضی — مانند الگوریتمی که کارموسینو و همکارانش در نظر گرفته بودند — در واقع به قدری قدرتمند خواهد بود که یک نسخه ای کمی متفاوت از MCSP را بدون هیچ اشتباهی حل کند.

نتیجه ای هیراها را هیجان انگیز است؛ زیرا بسیاری از محققان گمان می کنند که MCSP برخلاف تمام مسائل دیگری که برای آن ها تحويل بدترین حالت به حالت میانگین وجود دارد،  $\mathcal{NP}$ -کامل است. اگر آن ها بتوانند نتایج هیراها را گسترش دهند تا تمام الگوریتم های حالت میانگین را شامل شود و سپس ثابت کنند که  $\mathcal{NP}$ -کامل است، آنگاه ثابت می شود که ما در هیورستیک ارزندگی نمی کنیم. سانتانام می گوید: «این واقعاً نتیجه ای تکان دهنده ای خواهد بود». اثبات این که  $\mathcal{NP}$ -کامل است ممکن است کار دشواری به نظر برسد؛ چرا که این مسئله حدود ۵۰ سال است که مطرح شده است. با این حال پس از پیشرفت مهمی توسط هیراها در سال ۲۰۲۲ [۲۵]، اکنون محققان بسیار نزدیک تر از آن چیزی هستند که انتظار می رفت.

شكل ۱۳. راهول ایلانگو (تصویر چپ) و شوئیچی هیراها را (تصویر راست) اخیراً روش های جدید رمزگاری ای را توسعه داده اند که ثابت می کند نسخه هایی از  $\mathcal{NP}$ -کامل است.

هیراها  $\mathcal{NP}$ -کامل بودن را برای نوع دیگری از مسئله به نام MCSP جزئی<sup>۶</sup> اثبات کرد که در آن ورودی های خاصی را در جدول ارزش نادیده می گیرید. اثبات او مبتنی بر روش های توسعه یافته توسط ایلانگو بود تا نشان دهد که MCSP جزئی معادل



<sup>1</sup>Eric Allender

<sup>2</sup>Rutgers University

<sup>3</sup>Shuichi Hirahara

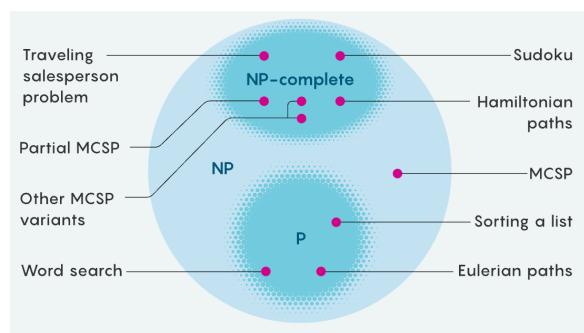
<sup>4</sup>National Institute of Informatics

<sup>5</sup>parital MCSP

مسئله‌ی به ظاهر نامربوطی است که شامل یک تکنیک رمزنگاری به نام تسهیم راز<sup>۱</sup> می‌شد. این روشی است برای تقسیم‌کردن یک پیام رمزنگاری شده میان تعداد زیادی از افراد، به طوری که پیام تنها زمانی رمزگشایی شود که تعداد خاصی از آن‌ها با یکدیگر همکاری کنند.

برای هر کاربرد واقع‌گرایانه‌ی رمزنگاری، پیش از هر چیز باید آن تعداد خاص را پیدا کنید؛ اما به کمک چند ترفند رمزنگاری می‌توانید یک سفاربی ناالمیدکننده بسازید که حتی در آن پیداکردن آن تعداد افرادی که باید با هم همکاری کنند هم دشوار باشد. هیراهارا راهی پیدا کرد تا ثابت کند این مسئله‌ی ساختگی رمزنگاری، NP-<sup>2</sup> کامل است و سپس نشان داد که این اثبات، NP-<sup>3</sup> کامل بودن MCSP جزئی را هم تضمین می‌کند. این دست آورد، حتی بیش از کارهای قبلی هیراهارا، به محققان انرژی داد و سایر محققان را متوجه این موضوع کرد؛ نظریه‌پرداز پیچیدگی لنس فورتو<sup>۴</sup> آن را دست آورد سال نامید. ویلیامز می‌گوید: «نتیجه‌ی شگفت‌انگیزی بود. همه فکر می‌کردند این مسائل جزئی تقریباً به همان سختی مسئله‌ی اصلی باشند.».

موانعی برای اثبات NP-<sup>5</sup> کامل بودن نسخه‌ی اصلی MCSP باقی مانده است؛ اما هیچ کدام از آن‌ها موانعی نیستند که نیاز به یک جعبه ابزار کاملاً جدید داشته باشند. ممکن است تنها مسئله، پیداکردن راهی درست برای ترکیب‌کردن تکیک‌های شناخته شده باشد. یک اثبات درنهایت وضعیت یکی از معدهود مسائلی را که از زمانی که نظریه‌ی پیچیدگی وجود داشته در برابر طبقه بندی مقاومت کرده است، حل می‌کند. لوین در ایمیلی نوشت: «این من را با نشان‌دادن این که احمق بودم که نتوانستم آن را ببینم، فروتن می‌کند.».



شکل ۱۴. ردیابی فعلی مسئله‌های مورد بحث در این نوشتہ.

**۲.۳. تکه‌های گم شده.** MCSP تنها مسئله‌ی فراپیچیدگی نیست که باعث پیشرفت بزرگی شده. در ۲۰۲۰، استاد رمزنگاری دانشگاه کرنل<sup>۶</sup>، رافائل پس<sup>۷</sup> و دانشجوی تحصیلات تکمیلی اش یانی لو<sup>۸</sup>، ارتباطی میان یک مسئله‌ی فراپیچیدگی دیگر را با یک پروتکل بنیادی رمزنگاری که مرز میان پسیلنند و هیورستیکا را مشخص می‌کند، و بدترین جهان‌های ایمپاگلیازو (جایی که مسائل NP-<sup>۹</sup> کامل در حالت میانگین مشکل‌اند ولی رمزنگاری هنوز غیرممکن است)، کشف کردند<sup>[۲۶]</sup>. این کشف، مسئله‌ی مورد مطالعه‌ی آن‌ها را به یک کاندیدای اصلی برای حمله به پسیلنند بدل می‌کند. به علاوه کارهای اخیر آن‌ها می‌تواند علیه هیورستیکا عمل کند<sup>[۲۷]</sup>. پس می‌گوید: «تکه‌های متفاوتی از پازل گم شده‌اند.»، و ادامه می‌دهد: «برای من امری جادویی است که این حوزه‌ها این قدر به هم مرتبط هستند.».

هیراهارا هشدار می‌دهد که هنوز چالش‌هایی در انتظار محققانی است که قصد دارند جهان‌هایی که ایمپاگلیازو ۳۰ سال پیش ساخته بود را از بین ببرند. او می‌گوید: «دوست دارم که بگویم در مقطعی هیورستیکا و پسیلنند کنار خواهد رفت، اما مطمئن نیستم چقدر به آن زمان نزدیک هستیم.».

بسیاری از محققان انتظار دارند که بزرگ‌ترین دشواری، پرکردن شکاف میان دو مدل متفاوت پیچیدگی حالت میانگین باشد. متخصصان رمزنگاری معمولاً الگوریتم‌های حالت میانگین را مطالعه می‌کنند که در هر دو جهت خطأ دارند — گهگاه رشته‌های تصادفی را به عنوان شبه‌تصادفی در نظر می‌گیرند و برعکس. با این حال، تحويل‌های بدترین حالت به حالت میانگین هیراهارا برای الگوریتم‌های حالت میانگینی جواب می‌دهد که فقط خطاهایی از نوع اول دارند. تمایزات ظرفی مانند این در نظریه‌ی پیچیدگی می‌تواند تفاوت‌های بزرگی ایجاد کند. اما علی‌رغم این مانع و موانع بسیار دیگر، آندر نمی‌تواند خوش‌بین نباشد. او می‌گوید: «من سعی می‌کنم نگذارم که خیلی هم معتقد باشم؛ چرا که سابقه‌ی کاملاً ثابت‌شده‌ای وجود دارد که هیچ چیز جواب نمی‌دهد. با این حال ما شاهد پیشرفت‌های بسیار هیجان‌انگیزی هستیم — راههایی برای مقابله با چیزهایی که مانند مانع بودند.».

<sup>1</sup> secret sharing

<sup>2</sup> Lance Fortnow

<sup>3</sup> Cornell Tech

<sup>4</sup> Rafael Pass

<sup>5</sup> Yanyi Liu

اگر یک درس وجود داشته باشد که محققان از سروکله زدن های خود با مسئله‌ی  $\mathcal{P}$  در برابر  $\mathcal{NP}$  یاد گرفته باشند، این است که نظریه‌ی پیچیدگی به خودی خود پیچیده است؛ اما این چالش دقیقاً همان چیزی است که جستجو را بسیار ارزشمند می‌کند. کارموسینو می‌گوید: «راستش خیلی خوب است که این قدر سخت است. عوضش هیچ وقت حوصله‌ام سر نمی‌رود!».

## مراجع

- [1] P Vs NP Problem In A Nutshell.. One of the unanswered questions in... | by Bilal Aamir | Medium. (n.d.). . Retrieved May 5, 2024, from <https://medium.com/@bilalaamir/p-vs-np-problem-in-a-nutshell-dbf08133bec5>
- [2] Turing, A.M. (1936) On Computable Numbers, with an Application to the Entscheidungsproblem. *The London Mathematical Society.*, Volume s2-42, Issue 1, 230-265.
- [3] Cook, Stephen A. (1971) The complexity of theorem-proving procedures. *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*.
- [4] Levin, L.A (1973) Universal Sequential Search Problems. *Problemy Peredachi Informatsii.*, Volume 9, Issue 3, 115-116.
- [5] Karp, R. M. (1972) Reducibility among Combinatorial Problems. *Complexity of Computer Computations.*, 85–103.
- [6] Razborov, A.A. and Rudich, S (1994) Natural proofs. *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*.
- [7] Baker, Theodore, John Gill, and Robert Solovay (1975) Relativizations of the  $P =? NP$  Question *SIAM Journal on Computing.*, Volume 4, Issue 4.
- [8] Tse, David (2020) How Claude Shannon Invented the Future. *Quanta Magazine*.
- [9] Shannon, Claude E. (1940) A symbolic analysis of relay and switching circuits. *Electrical Engineering.*, 713-723.
- [10] Shannon, Claude E. (1949) The synthesis of two-terminal switching circuits. *The Bell System Technical Journal.*, Volume 28, Issue 1, 59 - 98.
- [11] Hartnett, Kevin (2018) Why Mathematicians Can't Find the Hay in a Haystack. *Quanta Magazine*.
- [12] Furst, Merrick, James B. Saxe, and Michael Sipser. (1981) Parity, circuits, and the polynomial-time hierarchy. *Institute of electrical and electronics engineers*.
- [13] Razborov, Alexander. (1985) Lower bounds on the monotone complexity of some Boolean function. *Soviet Math. Dokl.*, Vol. 31., 354-357.
- [14] Kabanets, Valentine, and Jin-Yi Cai. (2000) Circuit minimization problem. *Proceedings of the thirty-second annual ACM symposium on Theory of computing*.
- [15] Levin, Leonid A. (1986) Average Case Complete Problems. *SIAM Journal on Computing.*, Volume 15, Issue 1, 285-286.
- [16] Impagliazzo, Russell. (1995) A personal view of average-case complexity. *Tenth Annual IEEE Conference*.
- [17] Klarreich, Erica (2022) Which Computational Universe Do We Live In? *Quanta Magazine*.
- [18] Bogdanov, Andrej, and Luca Trevisan. (2003) On worst-case to average-case reductions for NP problems. *44th Annual IEEE Symposium on Foundations of Computer Science*.
- [19] Chen, R., Kabanets, V., Kolokolova, A., Shaltiel, R. And Zuckerman, D. (2013) Mining Circuit Lower Bound Proofs for Meta-Algorithms.
- [20] Carmosino, M. L., Impagliazzo, R., Kabanets, V. And Kolokolova, A. (2016) Learning algorithms from natural proofs. *31st Conference on Computational Complexity (CCC 2016)*., Article 10, 1-24.
- [21] Ilango, Rahul. (2020) Constant Depth Formula and Partial Function Versions of MCSP are Hard. *IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*.
- [22] Ilango, Rahul. (2021) The Minimum Formula Size Problem is (ETH) Hard. *IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*.
- [23] Ilango, Rahul. (2020) Approaching MCSP from Above and Below: Hardness for a Conditional Variant and  $AC^0[p]$ . *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*.
- [24] Hirahara, Shuichi. (2018) Non-Black-Box Worst-Case to Average-Case Reductions within NP. *IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*.
- [25] Hirahara, Shuichi. (2022) NP-Hardness of Learning Programs and Partial MCSP. *IEEE 63rd annual symposium on foundations of computer science (FOCS)*.
- [26] Klarreich, Erica (2022) Researchers Identify 'Master Problem' Underlying All Cryptography. *Quanta Magazine*.
- [27] Liu, Yanyi And Pass, Rafael. (2022) On one-way functions from NP-complete problems. *Proceedings of the 37th Computational Complexity Conference.*, Article 36, 1-24.

مترجم: حامی عبادزاده سمنانی<sup>†</sup>

\* دانش آموخته‌ی دکتری فیزیک از دانشگاه بیل؛ عضو هیئت تحریریه‌ی مجله‌ی کواتا  
تارنما: <https://benbrubaker.com>

<sup>‡</sup>دانشجوی کارشناسی علوم کامپیوتر، دانشگاه صنعتی شریف