The Julius Silver, Roslyn S. Silver, and Enid Silver Winslow

DIALOGUES IN ARTS AND SCIENCE

CREATING AND SOLVING PUZZLES FOR A LIVING

Dennis Shasha

Silver Professor of Computer Science

NEW YORK UNIVERSITY

Creating and Solving Puzzles for a Living

In the early 2000s, when I was writing the puzzle column for *Scientific American*, the magazine asked me to write a brief description of my work. I suggested "Dennis Shasha creates and solves puzzles for a living." The magazine duly printed that after my byline. A few months later an indignant reader wrote: "Surely Professor Shasha does something more serious than create and solve puzzles." My thought was: not really.

The fact is that my philosophy of research is to break down every difficult problem I face into a puzzle that contains the essential features of the problem at hand. In this short paper, I'll present four examples from biology, physics, computer hardware, and drug anti-counterfeiting.

1. Plants

My plant biology colleague Gloria Coruzzi and I have had a long and fruitful collaboration together. (Gloria works on Arabidopsis, corn, and rice primarily, but also some fruits.) Early on in our collaboration, at a time when I thought that ALA might represent Alabama instead of Alanine, Gloria and I worked on the problem of trying to learn as much as possible from as few experiments as possible. To me, this sounded like a software testing problem.

While software testing might sound far from plant biology, safe-cracking might feel closer — at least it's tactile.

So consider a safe in which there are 15 toggle switches that can each take three different values, say high, middle, and low. Now, suppose that the safe will open if some pair of switches s1 and s2, and you don't know which two, have some combinations of settings. For example, perhaps switch 8 should be high and switch 13 should be medium. If that condition holds, then it doesn't matter which values the other switches have. How many switch combinations do you need to try to open the safe? At first the problem seems daunting because there are 3 to the power 15 (3^{15}) different switch settings which is about 14 million. However, that doesn't take advantage of all the structure of the problem, because any setting having switches s1 and s2 (whichever they are) set to their proper values will work. Using a variant of a technique called Combinatorial Design, I wrote a program to solve the problem with 21 tests.

How does this apply to plants? Well, if there are a number of different inputs that one can give to a plant each in various (but discrete) quantities, then one can use combinatorial design to explore the space of input dosages. Formally, what is guaranteed is that for any pair of inputs, every possible combination of dosages is tried by some experiment (thus the analogy to the safe). As for the safe, this approach vastly reduced the number of combinations we needed to try.

We then extended this scheme to find "minimal pairs," a concept I first

heard about in linguistics, in which say the dosage of one or a few "focus" nutrients could be varied and then a combinatorial design of all the other nutrients could be tried (e.g. low carbon and a combinatorial design of other nutrients vs. high carbon and the same combinatorial design of the other nutrients). This would suggest some contexts in which the focus nutrients were most effective. The entire approach became clear thanks to the puzzle.

2. Supernovas

My physics colleague Allen Mincer and I both live in Silver Towers. Though I had always observed his pleasant smile, I had never had a technical conversation with him. That all changed when both of our kids played on the same soccer team as 10 year olds. While the other parents were shouting helpful suggestions to their kids and the coaches, Allen and I discussed supernovas.

It turns out that supernovas can explode over a period that can range from milliseconds to days. The astronomers would like to aim their telescopes at the supernovas when they do explode, so the problem is to detect the longer-exploding ones quickly so the telescopes can change their aim. Fortunately, the supernovas leave a signature of a particle shower. The problem as Allen described it was therefore to find bursts of these particles. He told me that the density of bursts being much greater for shorter explosions than for longer ones. Allen further told me that these bursts were rare, occurring something like once a month.

My excellent graduate student Yunyue Zhu and I discussed this and observed that in order to detect bursts across many time intervals, we should use an exponential hierarchy of windows. For example the smallest window might be a millisecond over which we would count particles. Then two milliseconds, then four, then eight, then sixteen... So, for example, the length sixteen window would collect the count of all particles received over the 16 milliseconds covered by that window. Let's say that a burst lasted 27 seconds. We would like that to be captured in some length 32 window or at least in a length 64 window. The trouble was that a single hierarchy could not do that. For example, suppose a length 64 window finishes at millisecond 64. A 27 millisecond burst might go from millisecond 60 to 87. That would not be caught.

Fortunately, the image of vacuuming came to the rescue: one has to overlap the vacuum paths in order to clean the floor. So we extended the data structure so that there were two overlapping hierarchies. So if hierarchy A had 64 length windows from millisecond 1 to 64, 65 to 128, 129 to 192, and so on, then hierarchy B would have length 64 windows from millisecond 33 to 96, 97 to 160, etc. This ensured that any burst of size less than 32 would be captured by some 64 millisecond window of one of the hierarchies. This method could therefore find the infrequent bursts efficiently regardless of their size.

Parental disclosure: One day, Allen and I were in such deep discussion that I missed a goal my son Tyler had made. Tyler seemed not to mind when I confessed this to him after the game. His expectations of fatherly sports attention, apparently, were low.

3. Faulty Circuits

After I graduated from college in the late 1970s, I went to work for IBM which made big, expensive, and very popular mainframes at the time. My job was to design circuits. One of the problems we faced was that the computer hardware would fail sometimes, but only for a few microseconds. When the field engineer examined the machine, he or she would find no problem. So, my manager asked me to figure out ways to design circuits to check other circuits without replicating the other circuit entirely. To me, this was like a puzzle in which there were liars, but only occasional liars. As in life, it turns out that an occasional liar is much harder to unmask than a systematic one. You have to catch it in the act.

Consider the problem of checking a decoder circuit. Such a circuit takes some number k of inputs and sends signals out to 2^k output wires. Exactly one of those output wires will be a 1 and the others will be 0. So if k is 4, then the binary sequence 1001 would make output wire 9 a 1 and the other 15 output wires all 0s. So the problem is to check that exactly one output wire has a 1. Breaking this down, we want our checking circuit to confirm that there is at least one 1 and that there are no more than one 1s in the output wires. The at-least one part was easy: a big OR circuit. The no-more-than-one part was more challenging. The naive approach is to check every pair of output wires and to see if they are both 1. But that would require 120 AND circuits.

On my way back from vacation, I figured out a better way. (I had to pull over so that I wouldn't get distracted in the drive back to Poughkeepsie.) I reasoned as follows: each output wire corresponds to a particular binary combination of the k input bits. When k is 4, for example, there is one wire corresponding to 0000, a different one for 0001, all the way up to 1111. Now if two wires both have 1s, then those two wires must correspond to input bit combinations that differ by at least one bit. For example if the wire corresponding to 0101 is 1 and the wire corresponding to 1101 also has 1, the inputs differ in the first position. So, to detect that two output wires whose corresponding inputs differ only in the first position are both 1, do the following: consider all the output wires corresponding to a 0 in the first position and put them into an OR and do the same with all the output wires corresponding to a 1 in the first position and then use an AND circuit to see if the results of those ORs are both 1. The same trick can be used for the second, third, and fourth positions. This would require only 15 or so logical ORs and ANDs.

4. Drug Anti-Counterfeiting

Drug counterfeiting is a $100 billion dollar business worldwide and causes many deaths and suffering. Historically, the drug industry has fought back with fancier packaging, but my Chemistry colleague Mike Ward and I thought that there might be something we could do to the pills themselves. To me, it could be a probability game. Could we make it so that a would-be

counterfeiter would have to make consistently good guesses repeatedly to pull off a successful counterfeit? That would be like forcing the counterfeiter to be an incredibly lucky gambler.

In one simple version of the idea, suppose that the pills could include hidden food coloring in the coating that would make each pill either red or blue. Now if the pills in say a 20 pill bottle are numbered 1 through 20, then an end consumer could write down the bottle number, the pill number and the color and report that to the drug company via a website as he or she consumes the pills. If the drug company sees that some pill number and the color don't match, then the drug company knows a counterfeit has taken place. The counterfeiter has lost his bet.

There you have it. I like to convert difficult problems into puzzles that somehow touch on daily life. Virtually every piece of work I've done has evolved that way. If the problem feels like a puzzle, I want to solve it. It's my destiny. I don't care what the field.