# Gresham College Main logo

17 March 2014

**Might as well Toss a Coin:**

**How Random Numbers help us find Exact Solutions**

Professor Tony Mann

Throughout my life I have been fascinated by randomness. As a child I obsessively wrote numbers on slips of paper in order to simulate football scores by drawing lots, making great efforts to distribute the numbers to give the maximum realism in the results I obtained. In my first job, I wrote a random number generator for a minicomputer and spent weeks carrying out statistical tests to verify its randomness. (It passed every test I tried except one: sadly that one was essential to the objective of my work.) I have tried to persuade my Head of Department of the savings of time and my angst that could be achieved if I used random numbers to mark student work, but so far that idea hasn’t found favour.

Randomness came into my first experience of democracy. When I cast my first vote at the age of 18, the candidate I voted for tied with another, and the election was decided by the Returning Officer tossing a coin (which came down against my choice.) So with this lifelong interest in the mathematics of chance, I am delighted to have the opportunity to talk to you tonight about how randomness helps computers solve all sorts of mathematical problems.

Tossing a coin is generally not regarded as a good way to make important decisions. In football tournaments, drawn matches, which used to be decided on the toss of a coin, are now determined by penalty shoot-outs, which are perhaps just as random but which fit better into our love of story-telling. We don’t like to remember how random our lives are. We’d rather attribute a spectacular sporting success or failure to human strength or weakness, so that an unusually good sequence of results for a football team or the market-beating portfolio of a successful investment fund are seen as testimony to the skill of the coach or the fund manager, rather than simply random variation (although the latter is usually a very plausible explanation). A missed penalty is seen as the result of a loss of nerve by the striker, even though most penalty-takers might expect to miss one in five in normal circumstances. When things work out well in our lives, it is the result of our good decisions: when they don’t, it is due to bad luck!

But in some sports coin-tosses can be significant. Here are the results of some cricket matches between teams A and B:

|  |  |  |
| --- | --- | --- |
| Match number | Toss won by | Winner |
| 1 | B | B |
| 2 | B | B |
| 3 | A | Drawn: A on top |
| 4 | B | B |
| 5 | A | Drawn: A on top |
| 6 | A | A |
| 7 | A | A |
| 8 | A | A |
| 9 | A | A |
| 10 | B | A |

We see that in seven out of ten matches the team winning the toss won the match, and in the two drawn matches the team winning the toss would probably have won given more time. Only in one match, the last one, did the team which lost the toss win the match. So these data might suggest two evenly-matched teams. But these are the results of the two recent Ashes Test series, where you will remember that England totally outplayed Australia in last summer’s series (matches 1-5 in the table) only to be completely outclassed in the return series in Australia this winter (matches 6-10). So was there really such a dramatic turnaround in the performances of the two teams, or was the luck of the toss the decisive factor? We prefer to find explanations which don’t acknowledge the role of chance.

OK: let’s explore randomness. I’m going to see if I can determine someone’s random choices by reading their mind. I’d like a volunteer please (if you’re watching this online, or reading the transcript, you can do this too: I can even read your mind in these circumstances!)

I’d like my volunteer please to think of a random integer between 1 and 50, with two digits. Both digits should be odd, and not both the same. Please concentrate on that number. my next slide will tell you what I think it is. I’ve committed my prediction: please tell me what it is.

You say 37 – my prediction will now be revealed. I predicted 37. How was that? About 50 numbers to choose from and I got it right! (If I got it wrong, it’s because someone else in the room was thinking very hard of the number 37 at the same time and I read the wrong person’s mind. And if you are doing it online, and I got it wrong, then there must be someone else watching the lecture online at exactly this same time, who was thinking of 37!)

So about a 1 in 50 chance, because you had 50 numbers to choose from? Well, as I’m sure you’re aware, it wasn’t 1 in 50, because I applied some extra conditions, which reduced your choice. I required two digits, which rules out 1-9. I then said both digits should be odd, which removed all numbers in the 20s and 40s as well as even numbers in the teens and thirties. My requirement that the two digits should be different then removed 11 and 33. So the only numbers you could have chosen were 13, 15, 17, 19, 31, 35, 37 and 39. You had only eight numbers to choose from, so if you chose randomly from these, my chances were 1 in 8. And people tend to avoid extremes – 1s and 9s – and 5, which is too average. So a very high proportion of people will choose 37. And because I mentioned the range 1 to 50 first, people anchor on the idea that they have 50 choices, so they are more impressed when this trick works than they should be.

So perhaps a 1 in 8 chance coming off isn’t all that impressive. What about something that’s more than ten times as unlikely? I’d like my next volunteer to choose a random number between 1 and 100. Will I be able to predict your choice?

So what is your number? 88? You’ve learned lessons from the previous example. I’m going to reveal my prediction, and you will find that I have pulled off something very unlikely. My prediction is – “Your number is an integer”. Now there are infinitely many numbers between 1 and 100 which are not integers – all the fractions, all the irrational numbers like pi, and so on. So my prediction that you would choose an integer – of which there are only 100 amidst infinitely many non-integers – was very impressive.

But you might quibble that you misunderstood the options available to you and thought that I was asking for a whole number. So let’s do this once more, and I will be completely explicit about what you can choose. You can pick any random number you like – as small or large as you like, integer or non-integer, rational or irrational. If you want to, I’ll even allow you to choose a complex number. Again, I am making a prediction. Can I be right again?

So what is your number? Well, here’s my prediction:

“The number you chose is one which you can identify precisely in less time than has passed since the beginning of the universe.”

You’ve named your number in a few seconds, so, happily, I was right, otherwise the poor staff at Gresham College would have been kept waiting rather a long time before being able to lock up this evening. But why am I claiming I did well to make this prediction?

Well, there are infinitely many numbers you could have chosen – uncountably many, even. And given the finite number of words in the English language, and that you can only utter a few words per second, then only finitely many of these numbers can be expressed in any finite time. So almost all the numbers out there could not be described in the lifetime of the universe. The ones we can identify are very special indeed. If we were choosing a number truly at random, there is a probability of 0 that we would choose one which we can express. So once again, if you had followed my instructions and chosen randomly, the probability that my prediction would be correct was zero. I have (for the second time) made a correct prediction which was impossible in the sense that there was zero probability of my being right.

What I relied on in doing the impossible is that choosing a number really at random is hard for us! Indeed the whole concept of a random number needs to be very carefully thought through, even when we are assuming that all numbers are equally likely to be chosen. Here’s another example. What is the probability that a positive integer chosen at random is divisible by 7? Well, here is a list of all the positive integers:

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, ...}

and we can easily see that one integer out of every seven is a multiple of 7. So the probability that a positive integer chosen at random is a multiple of 7 is 1/7. That seems clear and straightforward.

OK, here is another list of all the positive integers:

{1, 7, 2, 14, 3, 21, 4, 28, 5, 35, 6, 42, 8, 49, 9, 56, 10, 63, 11, 70, 12, 77, 13, 85, 15, 91, ...}

Odd terms of this list – the first, third, fifth and so on - are those numbers not divisible by 7, and even terms – the second, fourth and sixth - are the multiples of 7. So this list contains every positive integer, exactly once. Clearly, now, every **second** positive integer is a multiple of 7 – that’s half of them - so my probability must be ½. The logic is clear, but we have two different answers: what has gone wrong? Again, this paradox (which caused a bitter row between the mathematicians R.A. Fisher and William Burnside about the foundations of probability theory in the 1920s) shows that we have to be very careful when we think about choosing things from infinite sets.

As a final illustration of the perils of random sampling, let me just mention the Doomsday argument, discovered (or invented?) by the astronomer Brandon Carter 20 years ago. In order of birth, I am number *n* out of all the human beings who have ever or will ever exist. If I regard my position in this list as random, because I could have been any one of all these people, then, from my position in this list, I can derive bounds for how many more people are likely to be born: if the total human population over all time were to be enormous, the probability that I would be so lucky as to be one of the first few billion of all human beings is very small. As a probability estimate, I can say on this basis that with 95% probability no more than 20 times *n* people will ever live, and the human race has, therefore, no more than about 9000 years left. You may (as I do) think this argument is dubious (or perhaps that it proves that the total human population will be infinite and therefore we have a long future ahead). But essentially the same argument was used by British statisticians to estimate the rate of production of German tanks during the Second World War based on observed serial numbers, and it gave the right answer!

Fine. So we’ve shown that the concept of a random number needs some careful handling. But if we do have randomness, then how can we use it? Take the toss of a coin – the epitome of apparent randomness. Can tossing a coin ever help us make a better decision?

Well, here’s one situation where it can. Consider the sad case of Buridan’s ass. This unfortunate donkey had access to two, equally appetising, and equally distant, large piles of hay. With no rational reason to choose to eat first from either pile rather than the other, the poor animal starved to death (providing a useful reference for future political cartoons, like this example from around 1900 regarding the siting of the Panama canal). Had he thought of tossing a coin to choose which pile of hay to start with, the ass would have survived and prospered.

Incidentally, although the name of the colourful 14th-century philosopher John Buridan is associated with this example, it does not appear in his surviving writings and it is not clear whether he had anything at all to do with it (perhaps it was invented by one of his opponents as a rebuttal of a point of Buridan’s philosophy).

I cannot resist mentioning a (highly unlikely) story of Buridan’s early life. He and one Pierre Roger were contending for the affections of the wife of a shoe-maker, and Buridan struck his rival on the head with a shoe. As a result of this blow, Roger developed a phenomenal memory, for which he became noted during his subsequent career as Pope Clement VI. I hope to tell you more about the fascinating mathematics of John Buridan, and the curious legends about his life, in a future lecture.

So tossing a coin might have saved Buridan’s donkey, and it might be a helpful way to choose between two equally attractive options in a situation of little importance – should I have vanilla or chocolate ice cream, for example. But is it really useful for us to toss a coin when we are faced with a decision that really matters? Well, I think it can be. Suppose I am undecided about a job offer, for example. I really can’t make up my mind whether to accept a new job or stay with my current employer. So I toss a coin: heads I stay, tails I move. As I see which way the coin falls, I am likely to experience a moment of either relief or disappointment. My true feeling, which up to now I haven’t been able to identify, reveals itself in the instant when I see what outcome the coin directs. So I ignore the result of the toss and make my decision based on that moment of clarity about what I really want.

This is, I think, how the *I Ching*, the Chinese *Book of Changes*, can work. If one wants guidance, one chooses randomly, essentially by tossing a coin six times, one of sixty-four possible hexagrams, each with a rather opaque and ambiguous text. How one instinctively interprets the text sheds light on one’s subconscious feelings and therefore may help one make the right choice.

But can coin-tossing do anything more than that? Can it solve mathematical problems? Well, let’s generalise slightly and use random numbers rather than tossing a coin. This isn’t really a big change because we can generate uniformly distributed random numbers between, say, zero and one by tossing a fair coin repeatedly to generate successively each of the binary digits of our random number. And computers can quickly and efficiently generate random numbers which are to all intents and purposes are uniformly distributed between 0 and 1.

Let’s take a simple mathematical problem – what is the value of the constant *π*, the ratio of a circle’s circumference to its diameter. This is a deterministic problem – we have calculated millions of digits of *π*, and Lu Chao has memorized the first 67,890 of them. The first few are 3.14159 26535. There are many nice formulae for *π*, such as the Gregory-Leibniz series

or Machin’s formula

or Ramanujan’s amazing formula

But what can we do if we don’t know any formula for calculating *π*? If all we know is that the area of a circle is *π* times the square of its radius, how can we find *π*?

Well, suppose I am a really bad darts player. I have a dartboard consisting of a circle of radius 1 inscribed in the centre of a square of side 2. If I throw darts at this square board, and my darts are so random that, when they land within the board, they are equally likely to land anywhere within it, then the proportion of those hitting the board which land within the circle will be the ratio of the area of the circle to the area of the square. The circle has area *π* times12, which is *π*, and the square has area 22 = 4, so the ratio will be *π*/4. So throwing random darts should give me an approximate value for *π*/4.

Now, I’m too good a darts player to do this. But computers can generate random numbers, and that’s just as good. For convenience, rather than use the whole dartboard, we’ll use the top right quadrant, with radius 1 and area *π*/4, embedded in a square of side 1 (and therefore area 1).

What we’re going to do is:

* Generate two random numbers between 0 and 1, which we will call *x* and *y*, representing a point (*x*, *y*) in the unit square.
* Test to see whether the point (*x*, *y*) lies within the quarter-circle of radius 1, which will be the case if and only if *x*2 + *y*2 < 1.
* Do this a large number of times and count the proportion of “darts” which land in the quarter-circle.
* This will give us an estimate for *π*/4.

Let’s try it. If we do it for ten or 100 darts, the results are not particularly impressive. But computers can throw darts much faster than I can, so we can do it for, say, one million tosses, and we will find we get a reasonable approximation. Indeed, once can calculate bounds on the likely error, so we know how good we can expect our answer to be.

Even if you aren’t Lu Chao, this is not the best way to find *π* when you need to have its value to hand: we need a lot of darts to guarantee accuracy as good as we could get by taking the numbers of letters in the successive words of the little poem

How I wish I

Could calculate pi.

or the longer but more useful (especially for mathematicians) everyday phrase, “May I have a large container of coffee?” (You have to remember to be impolite and not to insert the word “please” into the phrase!) If eight digits aren’t enough for you, Michael Keith’s book *Not A Wake* encodes the first ten thousand digits of *π* in this way: a computer would take a long time to throw enough darts for that!

But this does show the power of what we call the “Monte Carlo method” (named not from the place but from the activity for which it is famous) – the use of random numbers to get an approximate solution to a problem. Note how powerful it is. We didn’t need any sophisticated mathematics, any formula for *π*, any idea how to calculate *π*. Using only that the area of a circle of radius 1 is *π*, we have found a pretty reasonable approximation for *π*.

Actually this isn’t the most entertaining way to find an approximation to the value of *π* using the Monte Carlo method. In the eighteenth century the eminent French naturalist, the Comte de Buffon, proposed a method for estimating *π* by dropping needles on the floor! If we have a floor made of planks of uniform width, and we drop a needle randomly, so that both its position and its angle relative to the parallel lines of the planks are random, then the probability that the needle crosses a line between two planks depends on *π*. In fact a simple integration shows that the probability of intersection is 2*l*/*tπ* where *l* is the length of the needle and *t* is the distance between the planks. So if we drop *n* needles, and *m* of them cross lines between planks, I can estimate

*π* ≈ 2*ln* / *tm*

I’m going to demonstrate this now, and I’m hoping to get a good approximation. Unfortunately the risk assessment for my initial intention of having members of the audience throwing sharp needles randomly around the hall was not well received, so I am having to do this by computer simulation instead. (Sadly, in programming my simulation I have had to put in the value of *π* to specify the range for the random angle of the dropped needle, which makes the whole thing rather circular: but let’s gloss over that defect.) My programme is going to allow me to toss large numbers of simulated needles, like this:

[COMPUTER SIMULATION]

Dropping one needle at a time will keep us here too long so my programme allows the computer to execute many drops and report the result. If we do 100 drops with a needle length of 1 and a gap between floorboards of 2, we probably get a rather poor estimate: if we perform a million drops we do rather better.

But let’s see what happens, if we go back to individual drops. You may feel that a needle length of half the floorboard gap is rather too neat, so let’s use less round numbers. We also want rather large needles to be seen clearly in this hall, so I have found some virtual needles of length 710 cm, and the gap between the floorboards is 904 cm.

The first needle doesn’t cross a line, so my estimate for *π* after one drop is infinity, which is a bit out, when we remember that *π* begins 3.1415926535. The second needle does cross, and what do I have for *π*? I have 3.1415929, which is pretty good! I have gained an approximation which is accurate to seven significant figures, just by dropping two needles on the floor! How amazing is that?

Well, actually, it’s not really amazing, because although I haven’t exactly fixed the experiment – my random numbers really were random - I have given myself an excellent chance of getting such a good result. With two needles providing one intersection and one non-intersection, my ratio of successes is ½. This gives me an estimate for *π* of 4*l*/*t*, which for my chosen values comes out as 355/113, which is a well-known, very good approximation for *π*. Knowing the answer I was going for, I chose values which (if I had the right number of successes) would give me this ridiculously good approximation. Was I lucky in getting exactly one intersection from two needles? Yes, because it was only a 50/50 chance: but if I hadn’t, I could have continued to drop needles hoping for two intersections from 4 drops, or 3 from 6, or 4 from 8, and so on, and with a very high probability I will get my extremely good estimate from a small number of needles.

If I had preferred, I could have chosen values of the lengths which were not so convenient, but instead fixed the number of trials so that, if the number of intersections came out just right, I would get the same very accurate approximation. If I take *l*/*t* to be ½, and carry out my trials in batches of 355 needles, then sooner or later I may average exactly 113/355 intersections, at which point I will stop. In 1901 the Italian mathematician Mario Lazzarini used Buffon’s Needle to approximate *π* with results which seemed to be too good to be true: it is believed that this is the method he used.

What this shows is that if I know the answer in advance, and choose the right moment to stop, I can arrange for the Monte Carlo method to give apparently astonishingly good results. If I am using real needles, then of course I am likely to have a certain amount of difficulty in classifying borderline cases, which gives me even more opportunity to obtain the desired result.

But the true test of the Monte Carlo method arises when one doesn’t know the answer one is looking for. Simulation using random numbers is an invaluable tool. For example, a supermarket wants to find the best balance of staffing checkouts: too few staff and queues will build up: too many staff and, although the customers will get quick service, workers will be sitting idle and their wages could be saved. How many nurses should be on duty at Casualty on a Saturday evening, to deal with likely numbers of injured patients? These questions can be addressed mathematically, but simulation – modelling the situation using random numbers – is particularly effective because we are not seeking answers which are correct to ten decimal places, and because it is easy to simulate complex possibilities and to add new factors to the simulation which would make direct mathematical solution too difficult.

The power of large-scale simulation – computers can run the same simulation many times to see the spread of possible outcomes – makes this a very valuable tool for solving real-life problems.

Another example arises in the modelling of disease. A new epidemic strikes – how many people are going to be infected? Here we already have a good model based on parameters representing rates of infection, transmission and recovery, so the equations are standard, but we don’t know the parameters for our new disease. How easily is it transmitted from one person to another? How long are people infectious? What we can do here is simulate the disease for many different combinations of values of the parameters, running many simulations in each case. We then see which values of the parameters give results which match the outcomes we are currently observing. This allows us to gain very good, quick estimates of the likely spread of an epidemic, and often, to rule out the more extreme scenarios which are giving rise to panic. So in epidemic modelling random numbers are genuinely very, very useful.

Simulation is one mathematical use for random numbers. But there are others. Suppose we have an algorithm for solving a mathematical problem – a precise series of instructions which can be carried out to find a solution. As Buridan’s Ass shows us, we may need a way to resolve ties. What happens when two choices are equally promising? Some way to make a decision in this situation is needed.

Suppose we are trying to find the maximum value of a function of several variables. This is a common problem in many business situations. We want to find the combination of prices that will yield the most profit, or the operating conditions for a power plant which will safely generate the most power. The problem may be entirely deterministic – all the parameters are fixed, there is no chance involved at all in the situation which we are modelling – and yet random numbers can help us find the best solution.

We can think of this problem as travelling over a hilly landscape when we want to find the highest point (which is why this method is sometimes called “hill-climbing”). Unfortunately it is so misty that the only information available to us is where we are at the moment, and the slope of the ground at this point. An obvious way to proceed is to find the direction in which the slope is steepest, and take a step in that direction: then pause to see where that has taken us, and then do the same again. This is called the method of steepest ascent, and it works. But what do we do when there isn’t a single choice of direction because the ground slopes equally steeply in two or more directions? Well, we need to choose between them, by tossing a coin or choosing a random number!

This algorithm often works well, but it has a weakness: we want a global maximum but it tends to find a local one. That means that it will take us to the top of the nearest hill, but there may be a higher peak elsewhere, and having reached one hill-top, the method of steepest ascent won’t be able to take us any higher, because from where we are, every slope is downhill. So the basic algorithm will not always find the best solution. But if we combine this approach with randomness – we take random, reasonably large steps from our potential solution to see if we can find higher ground elsewhere – we have a much more robust method which may find a better outcome. So this is an example of randomness helping us solve a non-random business problem.

Of course, if you are a mathematical consultant advising serious business people, it is not always politic to reveal that your methods of solving their problems rely on chance. If I am CEO of a major company and I am paying someone large sums of money to solve a very important business problem, I don’t want to think that they are simply tossing a coin! So you might want to be careful how you present your analysis.

In other business situations randomness may appear more natural to the client. In the field of mathematics called Game Theory – the area for which mathematicians like John Nash have been awarded the Nobel Prize for Economics – the importance of making random choices is readily apparent. This is the mathematics of strategic thinking, which came to prominence in the middle of the last century, although it has recently been claimed in a recent book by Michael Sook-Young Chwe that game theory is systematically explored in the novels of Jane Austen!

If my company is competing with yours, then the decisions we each make will affect the other’s profitability. If I lower my prices I will do well if you maintain yours, since I will attract your customers, but if you respond by cutting your prices too, then we will end up in a disastrous price war which drives us both out of business. So I need to take account of your likely responses in taking my decisions. If you always take the same action in the same situation, then I can easily predict what you will do, and I can exploit that. But if your actions are, to some extent, random, then my task is more difficult.

We see this commonly in sport. A goalkeeper facing a penalty in a top-class football match will have studied the previous penalties taken by his opponent. If he always belts the ball, hitting it as hard as possible without worrying too much about direction, or alternatively if he always places it in the corner, the goalkeeper can plan for that, so his chance of saving it is much greater than if the striker varies his tactic. In this situation any pattern may be exploitable – so the more random your choice of penalty-taking strategy, the more successful it is likely to be! (Although random doesn’t have to mean that all choices are chosen equally often. If I think I am better at placing the ball in the corner, I might attempt that three times out of four, with the occasional belt to keep the goalkeeper guessing.)

When I prepared this talk I was going to mention Mikel Arteta’s retaken penalty in last week’s cup-tie, but there was an even better example of the dilemmas for penalty-taker and goalkeeper in yesterday’s match between Manchester United and Liverpool. Steven Gerrard put his first penalty to the bottom right. When he took his second penalty, did the goalkeeper expect him to do the same again, or to do the opposite? He put it in the same place again, and scored again. What about the third penalty in the match? This time Gerrard shot to the left (and hit the post with the goalkeeper beaten). He said after the match “I maybe got a bit cocky with the last penalty”, but perhaps he is just a good game theorist! The mathematics suggests that both striker and goalkeeper should make random decisions in these situations, which is why I expect that every penalty-taker in this summer’s World Cup will go into every match equipped with a supply of genuinely random numbers.

So I hope I have persuaded you that in some situations random numbers can help us solve mathematical problems. But here’s a variation: would we ever be interested in an algorithm which tells us the solution to our problems, but which is random in the sense that sometimes it will give us an incorrect answer?

Such algorithms are much studied by computer scientists. But what use is an algorithm which sometimes gives the wrong answer? Well, here’s a mathematical problem which actually has real importance in cryptography and security: given a large number *n*, how can I tell whether or not *n* is prime?

The obvious method – of testing for divisibility by each prime number up to the square root of *n* – works, but for large n it takes a very long time. In the jargon of computational complexity it takes **exponential** time, and what we would like is an algorithm which works in a time which is a **polynomial** function of *n*.

But prime numbers have some properties which we can use. The great mathematician Fermat showed that, if *p* is prime and *x* is any integer, then *xp – x* is a multiple of *p*. So here’s an idea. Generate some random numbers *x*, and test to see if, in each case, our large number *n* divides *xn – x.* If any one of my sample fails this test, then for sure *n* is not prime: if they all pass, then we have some reason to believe that *n* is prime.

Unfortunately, it turns out that there are some numbers *c* which are not prime but which will pass this test – whatever *x* we try, *xc – x* is a multiple of *c*. Examples are 561, 1105, 1729, 2465, and 2821. These are called Carmichael numbers, and there are infinitely many of them.

Nevertheless, tweaking this test gives us an algorithm which runs in polynomial time and which, if the input is a prime number, will always report that it is prime, and which, for a composite input, will tell us it is composite with a high probability.

At the time this method was devised by Miller and Rabin, in 1976, there was no guaranteed polynomial-time way of telling whether a given number is prime. That is no longer the case – in 2004 Agrawal, Kayal and Saxena found a non-random algorithm to answer that question in polynomial time. But randomised algorithms are still much faster in practice.

As is pointed out by Scott Aaronson, whose extraordinary book *Quantum Computing Since Democritus* was my source for this material, we can have randomised algorithms where the result may be incorrect but the probability of this unhappy outcome is much smaller than the chance that the computer will be destroyed by a meteorite in mid-calculation. Aaronson claims that such randomised algorithms are entirely sufficient for everyday practical purposes (such as administering radiation in hospitals or controlling nuclear missiles) and we should have reservations over their use only when it comes to really important applications like proving theorems in pure mathematics.

So we’ve seen randomness used to solve problems through simulation, through algorithms incorporating randomness, and through useful algorithms which give results that have a chance of being wrong. I’m going to finish by talking about what I think are the most remarkable random problem-solvers.

Perhaps the most powerful problem-solving method on earth is evolution. Random mutations operating over many generations under natural selection have built more complex, effective and resilient organisms than human engineers can dream of.

Evolutionary algorithms are mathematical methods which mimic evolution. You want to find the best solution to a problem? Start with some possible solutions, make a new generation populated by random changes to these original solutions, and choose the best of these as the parents of the next generation. Sometimes these algorithms can mimic biological evolution very closely, creating offspring by combining elements of two parents, or even combining analogues of chromosomes in explicit imitation of genetic recombination. The speed and power of modern computers enables many generations to be created in a short time, with each generation hopefully throwing up one or more solutions which improve on those previously found. Genetic and evolutionary algorithms are a fascinating current research tool and have applications in many problem areas. They have been found to be particularly suited to timetabling problems. Amongst successes of genetic algorithms are the design of systems of mirrors to reflect sunlight to a solar collector, generating an effective walking gait for two-legged robots, and designing the antenna for a spacecraft to give the optimal radiation pattern – the outcome of the last of these is shown in my picture.

What I love about these evolutionary algorithms is that we can use them when we have no idea how to solve a problem! We don’t need to have clever insights about possible solutions: we just use the power of natural selection over many generations to start with poor initial guesses and (if it goes well) we will end up with highly optimised results. It really is, for me, as close as we can get to magic!

Of course, the drawback is that we don’t gain any insights at all into the problem from our random, genetically-mutated outcome. Unlike what I might call a human approach to problem-solving, which develops through improved understanding of the nature of the problem leading to new ideas about how to exploit its features to find better solutions, an evolutionary algorithm may present an excellent solution but otherwise leaves us none the wiser. These algorithms are truly inscrutable.

So during the course of this evening I hope that I have shown you some of the ways in which coin-tossing, or its computer equivalent, can help computers solve mathematical problems. Randomness can address the weaknesses of some deterministic algorithms, such as the tendency of steepest-ascent methods to find local peaks rather than the sought-after highest mountain. Simulation can enable us to model complex situations and make excellent business decisions. Monte Carlo methods can give us good answers for little effort. Randomised algorithms probably give us the right answers, quickly. And evolutionary algorithms can find good solutions to very difficult problems where we don’t need to have any ideas of our own as to how to proceed.

I hope I have convinced you of the power of randomness in mathematical computing. Thank you for listening and, this being my last lecture in this series, I’d like to take this opportunity to thank the staff at Gresham College for all their help, and particularly to thank James and Alex who have coped admirably with all my technical demands.

© Professor Tony Mann