Some Social Aspects of Computers and Numbers

Neville Holmes, *University of Tasmania*

The use of computers is having a variety of effects on day-to-day activities, including playing cards.

Some three years ago, I shifted my home to near the center of a regional city after spending 20 years living in the countryside. This change encouraged me to resume playing duplicate bridge (tinyurl.com/2cmydj4) after a lapse of half a century.

The first 18 months of bridge once or twice a week were pleasant and sociable. Technically it was also quite familiar, at least in the card play itself. The bidding systems had changed a bit, but not uncomfortably. Then the local club began using a computer-driven card-dealing machine. For the administrators this was a boon. For the players it was very unsettling.

CARD DEALING

A computer runs a program to generate deals one at a time. For each hand, it runs a device that takes cards from a pack one at a time, scans their faces to identify them, and puts each in the appropriate pocket of the board that is to carry the computed deal. (See tinyurl.com/276lu25 for a detailed explanation of duplicate boards and their use, though computer controlled dealing isn't yet described there.)

To the more experienced players, however, and even me, the suit distributions seem abnormal. Each hand holds 13 cards, each card belonging to one of four suits. The expectation is that suit holdings of two to five cards are normal, of one (called a singleton) or six cards unusual, and of none (a void) or seven remarkable. This expectation isn't being met.

For example, in the 36 deals for a recent afternoon of bridge, there were 9 voids, 43 singletons, 22 six-card suits, 6 seven-card suits, and 1 eight-card suit.

When patterns like this recurred over the course of a few months, I sent an e-mail to those who had provided the software used for dealing our games. Their reply told me that the program uses BigDeal, "written some years ago by Dutch mathematicians and statisticians. It is considered the definitive deal generator and has the features to guarantee that every possible deal (2 to the 96 possible) can be generated, and further guarantees that a set of deals can't be duplicated. It also guarantees that the deals are truly random."

This startled me, especially the last sentence. I tried to find more about BigDeal through Google but was unable to navigate through the abundant "big deal" offers for this or that merchandise. So I decided to run a simple test on my own computer. The following suit distributions resulted from simulating three sessions of 3,600 deals.

0	1	6	7	8	9
755	4583	2466	504	62	3
773	4564	2387	491	65	6
764	4552	2434	498	61	6

The following shows the corresponding distributions for a few recent 36-deal sessions played at my club:

0	1	6	7	8	9
9	43	22	6	1	0
9	41	23	12	0	0
7	47	22	5	1	0
6	48	17	5	0	0

These figures strongly suggest that BigDeal imitates random dealing very well, at least in respect to the suit distributions. Why then does it seem abnormal? An explanation by Laurie Kelso appeared in the March 2010 issue of the *ABF Newsletter* (tinyurl.com/276sgom).

Continued on page 98

Published by the IEEE Computer Society

0018-9162/10/$26.00 © 2010 IEEE

Continued from page 100

"I sometimes hear comments about the differences in 'feel' of hand-shuffled versus computer-shuffled boards," stated Kelso.

"Theoretically, a deck of cards needs to be subjected to seven "riffle" shuffles to fully destroy any residual effects from the previous deal [for an explanation of riffling see tinyurl.com/2y6nug].

"Since most manual shuffling is at best cursory, it is really of little surprise that the results are sometimes flatter than the truly random hands produced via computer generation."

With the use of computers to deal the cards, some of the people I play with see the game becoming more of a gamble. They have become apprehensive about adverse distributions, and they see advantages in more preemptive bidding.

A pseudorandom number sequence is entirely predictable, but a good one has much the same statistical properties as a truly random number sequence.

RANDOMNESS

Players discussing the unusual nature of the computer-dealt hands typically blamed the computer. My early efforts to divert the blame to a programmer somewhere with a wry sense of humor were usually accepted. However, any attempt I made to point out the problem of having a machine designed to be completely predictable generate the numbers needed to deal cards randomly met mostly with disbelief.

Many players seemed to feel that computers could do anything numeric perfectly. Some who saw the relevance of predictability suggested that the computer could pick up random data from the system clock. The problem with this is that the program would have to pick up the random numbers needed from within a loop. Such a loop would be regular, so the numbers wouldn't be random.

A basic solution to this difficulty uses a noisy device and picks numbers from the noise. An early device of this kind, ERNIE (Electronic Random Number Indicator Equipment; tinyurl.com/272suu6), was unveiled in 1957 for a British government lottery and used a bank of neon tubes for noise. The more recent ERNIE 4 instead picks up thermal noise from transistors.

Hardware random-number generation is particularly significant for cryptography, with a variety of devices and methods in use (tinyurl.com/d2s4f). However, this approach is far from straightforward, and most applications requiring at least the appearance of randomness in their data use a variety of pseudorandom software (tinyurl.com/zgb4b).

A pseudorandom number sequence is entirely predictable, but a good one has much the same statistical properties as a truly random number sequence. Such sequences are ideal for stochastic simulations because the same sequences can be used repeatedly as models are developed and tested (tinyurl.com/2fjlyql).

The most popular and simplest method, linear congruential generation, uses a simple recurrence formula: $x_{n+1} = (ax_n + c) \bmod m$ (tinyurl.com/2qx2xb). That is, each number x_{n+1} in the sequence is calculated from x_n by multiplying it by a, adding c, then dividing by m to get the remainder for x_{n+1}. Any sequence starts with choosing the value of x_0, called the seed. Carefully choosing the values of a, c, and m, the pseudorandom sequence uses all the values between 0 and m-1 before starting the cycle again.

For example, suppose $a = 34$, $c = 53$, $m = 99$ and $x_0 = 0$. Then the full cycle is 0 53 73 60 14 34 21 74 94 81 35 55 42 95 16 3 56 76 63 17 37 24 77 97 84 38 58 45 98 19 6 59 79 66 20 40 27 80 1 87 41 61 48 2 22 9 62 82 69 23 43 30 83 4 90 44 64 51 5 25 12 65 85 72 26 46 33 86 7 93 47 67 54 8 28 15 68 88 75 29 49 36 89 10 96 50 70 57 11 31 18 71 91 78 32 52 39 92 13 0. Choosing a different seed will simply start the cycle at a different place.

This example shows that the arithmetic needed for the method is simple. It also shows that the method doesn't necessarily work well, as there are only five different gaps between consecutive numbers. Generators in practical use work much better, partly because the fixed-point arithmetic can deal simply with 32- or 64-bit integers, but the values of a, c, and m must be chosen carefully.

Better pseudorandom sequences can be computed with more complex arithmetic, such as in the various Mersenne Twister approaches (tinyurl.com/38g3gc), but linear congruential methods seem to be much more widely used (see the table at tinyurl.com/2qx2xb).

Taking the card sequence into account, there are 52! different bridge deals, a number that needs 226 bits or 68 decimal digits to represent it exactly, so it presents an arithmetic challenge. Since the sequence in which the cards in any hand is dealt is irrelevant, the number of distinct deals is much less: $52!/13!^4$ with 96 bits and 29 digits. Using a 32- or even a 64-bit generator can only deal a minute fraction of the possible deals, and that fraction always in the same sequence. Anybody writing software for dealing cards for any game should be aware of this.

GAMBLING

Randomness is behind many forms of gambling and has become a significant source of revenue for governments and gambling venue owners. It has also become a major spare-time occupation for many people. The encouragement of gambling has inevitably led to the problem of gambling addiction (tinyurl.com/2cbwzyt).

Much of the gambling machinery is now based on digital technology. Whereas once gamblers operated slot machines by pulling a handle, now they simply push a button that presumably operates a pseudorandom number generator to determine the result.

There are disturbing possibilities here. Many old-time stories about gambling describe "suckering," whereby skilled players would gang up on a newcomer and manage the cards so that he would win to start with, go on enthusiastically in the expectation that the winning would continue, and end up being fleeced of all his money. A similar possibility exists with slot machines run by software.

Winning excites and encourages continuance. Research has shown that you don't have to win to get this kind of excitement; near misses work just as well (tinyurl.com/25c7vnp). It would be simple to design gambling software for slot machines, for example, that would ramp up the near misses to promote addiction. Indeed, it could be cynically justified as increasing the entertainment value. I hope this doesn't happen, but when I read that children of parents with a gambling addiction are 10 times more likely to become problem gamblers (tinyurl.com/23ofzdu) and that some experts believe that arcade games are put into gambling venues to groom children for gambling (tinyurl.com/2dvgmky), I can't help worrying.

The worry is even greater with the rapid growth of online gambling, where regulation is much more difficult. The leading article of a special report on gambling in *The Economist* (10 July 2010) declared that "The internet is radically changing the business of gambling. Now policy must catch up." The lead article emphasized that "The move online threatens some traditional forms of gambling … but appears to benefit others, such as slot machines and lotteries."

Online gambling is socially isolating. Slot machines have a similar effect on many people, as any visitor to a casino can see. The best entertainment occurs through social interaction as, for example, when playing games like bridge or Scrabble. It's a pity that governments don't give the same support to such social games as they do to gambling.

When almost finished with writing this essay, I stumbled on what seemed to be an early (8 Sept. 2000) description of the BigDeal program by one of its designers, Hans van Staveren (tinyurl.com/287knch). Detailed and fascinating, it confirms that the program works using pseudorandom numbers, not truly random ones. However it does get the operator "to start typing until instructed to stop," and uses timing and key choices to carefully construct a 160-bit random seed.

Significantly, early discussion showed that using computer software as the basis for games where money or prestige is involved can provide openings for unscrupulous people to cheat. Computing professionals involved with such software should take note of the approach adopted by van Staveren and his colleagues.

***Neville Holmes** is an honorary research associate at the University of Tasmania's School of Computing and Information Systems. Contact him at neville.holmes@utas.edu.au.*