

The mathematical art of experiment design

Patrick Morris and Guillem Perarnau

October 2023

From Biology to Economics, experiments play a crucial role across the sciences. In many cases one has a lot of freedom in how to design an experiment. How do you design it well? How do you make sure it is fair and efficient? For answers to these questions, we can turn to Mathematics and in particular, a corner called *Combinatorics*, where such questions have been formalised and studied for centuries. Two key concepts arise - *randomness* and *combinatorial designs*. Despite huge progress in the mathematical understanding of these notions, one key challenge remains, namely how to marry these two ideas and *generate random designs*. This is a notoriously hard problem, but recent breakthrough results open up entirely new vistas and the future of the mathematical art of experimental design looks very bright.

Randomness: Making experiments fair. Suppose you are given the following task. Design an experiment that compares the effect of different fertilisers on the yield of a certain crop. Let us say you have n fertilisers (with n batches of each) and an n units by n units square field split into individual lots, each lot having size 1 by 1 and taking exactly one fertiliser. How do you arrange the fertilisers on the field in order to test and compare them? One way to do this would be to split your big square field into rows of lots and assign each row a fertiliser. See for example the distribution indicated in Figure 1 below. Whilst this is certainly a neat arrangement, it leads to an experiment that is not very *fair*. Indeed, it is susceptible to bias that will skew our results. Imagine, for example, that a pest infestation appears from the South. The last row, corresponding to a single fertiliser will then have very poor results, even though it could actually be the best one!

In order to avoid such unwanted biases, we want a distribution that is somehow far from neat, as any ordered rule for distribution will inevitably lead to a potential bias. How do we achieve such a distribution? A beautifully simple and yet extremely effective idea is to simply use a *random* distribution. That is, for each lot, we roll an n -sided die and assign the fertiliser corresponding to the outcome to that lot (of course, at some point you may run out of a given fertiliser, but then you can simply roll again until you get a choice where you have a free batch to use). See Figure 2 for an example of a random distribution. The great thing about this is that it is extremely fair. Indeed, it gives no (dis-)advantage to any particular fertiliser and will most likely be completely free of unwanted symmetry. Whilst it is impossible to know when the idea of using random objects first originated, it was pioneered by Paul Erdős in the second half of the 20th century, who realised the power of using randomness to get combinatorial objects with desired properties. Often such properties are difficult to obtain by deterministic means, but flourish

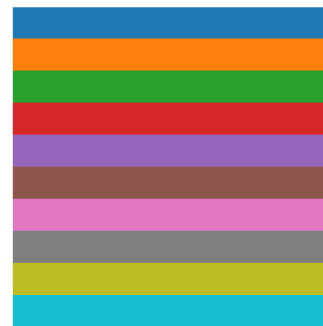


Figure 1: A row by row distribution of fertilisers

naturally with the use of randomness. Among the earliest results of the so-called “Probabilistic method”, we find the lower bound on diagonal Ramsey numbers and the existence of graphs with large girth and large chromatic number. The use of random objects to answer certain questions also spurred the study of random discrete structures in their own right, and the study of properties of random combinatorial objects is to this day a thriving and fascinating area at the intersection of Combinatorics, Probability Theory, Computer Science and Statistics.

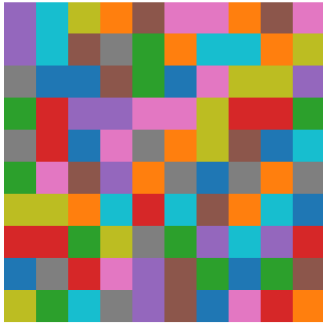


Figure 2: A random distribution of fertilisers

the random distribution as in Figure 2; there are several columns and rows that completely miss some fertilisers. There is a fix, but it comes at a price. Indeed, if we use a larger field, say our field is m units by m units, then placing our n fertilisers randomly, will most likely result in a distribution where each row and each column sees each fertiliser when (and only when) m is at least $n \log n$. This example is an instance of the famous *Coupon Collector Problem* from Discrete Probability. Actually in this case each row/column will also see each fertiliser around m/n times. Whilst this looks like a nice experiment distribution, it is very wasteful in resources as the extra $\log n$ factor means that we need lots more of each fertiliser, not to mention a larger field.

Combinatorial Designs: Making experiments efficient.

In many cases one can imagine that costs dictate that the design of the experiment should be optimally efficient. Thus, returning to our example, is it possible to use an n units by n units square and just use n copies of each fertiliser but still have each row and each column using each fertiliser once? Given that the random approach does not work for this, maybe it is time to revisit the ordered approach, as we did for Figure 1. Indeed, suddenly our initial arrangement doesn’t seem so bad anymore. At least every fertiliser is tested with each water level (in each column) and so we have comprehensive results for half of our task. In fact, it is not too hard to get an ordered arrangement that works, simply consider the diagonal arrangement depicted in Figure 3.

Mathematically speaking, what we are talking about is known as an n -Latin Square (*LS*); an $n \times n$ array with entries in $[n] = \{1, 2, \dots, n\}$ such that each number appears exactly once in each row and in each column. These objects were introduced by Leonhard Euler in the 18th century

Using randomness is highly effective in being robust to unwanted factors that could effect the outcome of our experiment, but what if there are critical factors that we actually *want* to test? Indeed, let us imagine a more complicated task for an experiment. Again you have different fertilisers which you want to place in a square of lots, but this time, you want to test the performance of each fertiliser in relation to other factors that correspond to the rows and columns of your square. For example, it could be that each row of your square is at a different height and each column receives a different amount of water. Our distribution should still be fair but, crucially, we should have data for each fertiliser and different height/water level, so that we can identify, for example, the optimal combination. Thus we see a problem with

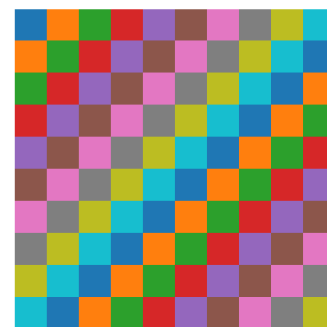


Figure 3: A *diagonal LS*

and are central in Design Theory. In fact, you have probably also seen a large number of LSs in your life time as they form popular games in the form of (completed) *Sudoku squares*, which are 9 by 9 LSs in which we additionally impose each number to appear exactly once in each of the 3×3 main subsquares. Euler realised that he could generate many different Latin squares and was interested in increasing the symmetrical requirements, looking for example, for the so called *Mutually Orthogonal Latin Squares (MOLS)* where two LSs overlap in a way that no pair is repeated in a square. See for example Figure 4. In the experimental design analogy, one can think that you are not only testing fertilisers but also pesticides in the same experiment and want to arrange them so that each fertiliser and each pesticide is tested at each height/water level and *additionally* every pair of fertiliser/pesticide is also tested together.

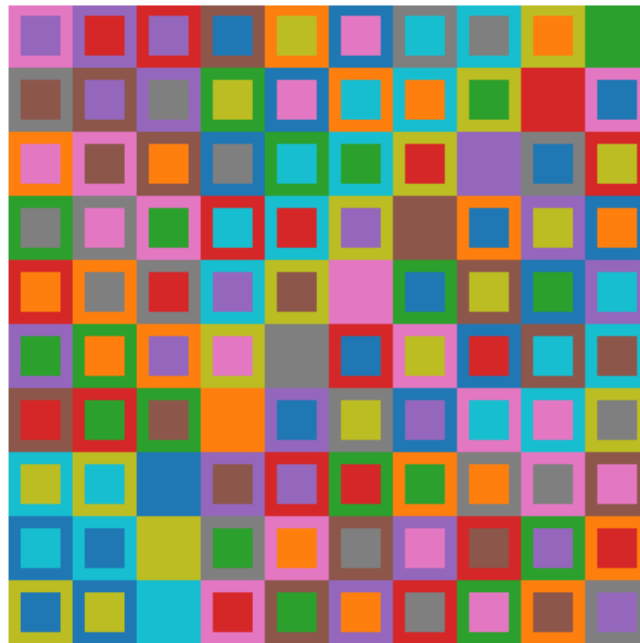
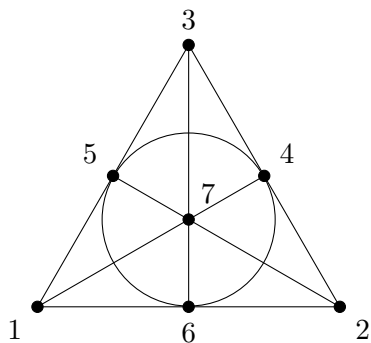


Figure 4: A pair of MOLSs

LSs and MOLSs are just one example of a wide family of mathematical objects known as *Combinatorial Designs*. Each is a collection of subsets of a finite set that have strong symmetric and balanced properties. Since Euler, these objects have been studied on a mathematical basis and a deep theory has been developed. They have also found many applications in different scientific disciplines. Indeed, their use in Experimental Design, as in the example above, was pioneered by Ronald Fisher in the 1920s-30s [3] as he was concerned with the agricultural applications of statistical methods. Another application is the construction of error-correcting and error-detecting codes, codes that are used to transmit information robustly and, preferably, efficiently. Indeed, to reduce the amount of additional bits transmitted, we would like to find optimal codes. An important family of optimal codes is *Perfect Codes*, rare objects that decompose the space of binary strings in a highly symmetrical way. A canonical example is constructed from the *Fano Plane*, an example of a combinatorial design known as a *Steiner Triple System (STS)* that can also be interpreted as a finite projective geometry. Take a collection $\{S_1, \dots, S_7\}$ of 3-subsets of $[7]$ such that each pair appears in exactly one triplet (see Figure 5). Now we construct a code as follows: for each $i \in [7]$ create a binary word v_i of length 7 by writing a 1 at entry j if and only if $i \in S_j$ (see Figure 5).



$S_1 = \{1, 3, 5\}$	$v_1 = 1010100$
$S_2 = \{2, 3, 4\}$	$v_2 = 0111000$
$S_3 = \{1, 2, 6\}$	$v_3 = 1100001$
$S_4 = \{1, 4, 7\}$	$v_4 = 1001001$
$S_5 = \{2, 5, 7\}$	$v_5 = 0100101$
$S_6 = \{3, 6, 7\}$	$v_6 = 0010011$
$S_7 = \{4, 5, 6\}$	$v_7 = 0001110$

Figure 5: Geometric realisation of Fano plane, collection of 3-subsets and their corresponding elements in the $(7,4)$ -Hamming code.

Add the zero word of length 7, 0000000, and also include all the complementary words (words obtained by replacing zeroes by ones, and vice versa). This collection of 16 strings is known as the $(7,4)$ -Hamming code. It transmits 4 bits of information by sending 7 bits, and it does so with some level of robustness: it can correct up to 1 error or detect up to 2. That is, if our goal is to correct errors and only 1 bit of the codeword is changed in the transmission (flipped from a 1 to a 0 or vice versa), the codewords are different enough that we can retrieve the original codeword. Similarly, if our goal is to detect errors and there are at most 2 bits changed, the word received will not match up with any codeword and so we will know that some error has occurred along the transmission.

Let us give one last example of a design, which is in fact also an STS and, like Sudoku squares, came from the world of recreational mathematics. This is known as *Kirkman's school girl problem*, a mathematical puzzle published in 1850 [7]: “Fifteen young ladies in a school walk out three abreast for seven days in succession: it is required to arrange them daily so that no two shall walk twice abreast”. This puzzle was the earliest explicit example of an STS (like the one in Figure 5), where one asks for a collection of subsets of some n -vertex set of size 3 such that every *pair* of vertices features exactly once. Thus Kirkman's problem seeks to find an n -vertex STS that, additionally, can be split into 7 partitions of $[n]$. Note that Kirkman's problem can also be interpreted as asking for an experimental design; if we are interested in a social experiment, we may want to ask each girl to rate the other girl's conversation and so it will be necessary for every pair of girls to walk together at least once. As with LSs, an STS represents the most efficient way of achieving this.

An interesting feature of Kirkman's problem is that it is not easy. Indeed, unlike the diagonal LS that we saw in Figure 3, there is no simple rule that can give the configuration necessary and it can take quite some time to come up with a solution, as a good recreational mathematics problem should! This is in fact very indicative of the development of Combinatorial Design Theory since its birth. We have seen some simple examples, but one can easily generalise definitions and require more from our designs, as Euler did already by asking for MOLS. A natural generalisation of STSs is the following: for integers $1 \leq t \leq k \leq n$ we can ask for a collection of k -subsets on n vertices such that every t -subset is covered exactly once, known as an (n, k, t) -Steiner system (*SS*).

Most of the research in Combinatorial Designs in the 20th century revolved around the question of whether the designs that we ask for actually exist? For some parameters, one can easily prove no such design exists via a combinatorial trick called *double counting*. For instance, if there exists an (n, k, t) -SS of size m , then the total number of t -subsets in $[n]$ is $\binom{n}{t}$, while the number of t -subsets covered by the collection of m k -subsets of the design is $\binom{k}{t}m$. Since m is integer, $\binom{k}{t}$ must divide $\binom{n}{t}$. These sort of conditions are known as *divisibility conditions* and are clearly necessary for the existence of designs. For other sets of parameters, more complicated proofs can show non-existence

despite the divisibility conditions being satisfied. On the other side, much effort has gone into constructing designs with various parameters and properties. Although this is sometimes easy, for most designs this very quickly gets hard, like with Kirkman’s problem. Solutions often involved using Geometry, as in Figure 5, and Algebra, as in Figure 3, where one can view the construction as placing, for each row i and column j , the fertiliser F_{k+1} where $k = i + j \pmod n$.

Random designs: The best of both worlds. We have seen how randomness leads to fair experiments whilst designs can be used to construct efficient experiments testing multiple different factors. Neither approach is perfect. Indeed we saw that randomness can lead to redundancy when we require multiple factors to be tested whilst it is clear that the unwanted symmetry of an arbitrary combinatorial design does not necessarily lead to a fair experiment. For instance, consider the example in Figure 3: if the central South-West to North-East diagonal gets the most light, then the fertilisers represented by (light-)blue will have a pretty sizeable advantage.

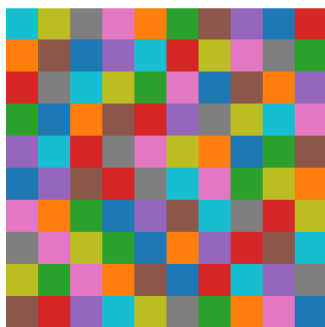


Figure 6: A random LS

In order to fix this, one idea is to construct the design in a random fashion. Yates already envisioned this in 1933. Discussing about experiment designing, he wrote “...it would seem theoretically preferable to choose a square at random from all the possible squares of given size”. For example, Figure 6 depicts a random LS of size 10. It certainly seems well-distributed and so robust against bias. In general though, Yates’ request poses a considerable challenge. One could hope to just have a list of all possible designs and then pick one at random but this quickly becomes intractable. There are already 7,580,721,483,160,132,811,489,280 distinct¹ LSs of size 10 and this number grows superexponentially with n . Therefore some new idea is needed to construct a random (or

close to random) design.

In the 90s there was a big breakthrough by Jacobson and Matthews, who gave a *Markov Chain Monte Carlo (MCMC)* algorithm for generating random LSs [5]. The MCMC method revolutionised the world of random algorithms in the last decades of the 20th century. The basic scheme of MCMC algorithms is the following: Given a class of objects one is interested in sampling from, define a local operation on them which allows us to navigate through the space of such objects. Then, we may follow a random trajectory (commonly referred to as a *random walk*) on our sample space by, at each step, performing a local operation chosen at random from all the possible ones. Under many circumstances, after just a relatively small number of steps (depending on the object size), the trajectory will lead to an object that is close to uniformly random. There is a large and rich theory developed for ways of proving that this is the case, know as *mixing times of Markov chains*. Due to their symmetry constraints, it is not easy to define a local operation on LSs. The pivotal contribution of Jacobson and Matthews was to find a “not so local” operation between them which yielded an MCMC algorithm for sampling an LS. Whilst one can prove that this algorithm will eventually lead to an almost uniform LS, and it seems to work well in practice, to this day we cannot provide mathematical evidence that the algorithm is efficient (that is, polynomial in n). For other designs (for example STSs), the situation is even worse, with a few similar algorithms proposed but very little proven about their uniformity or effectiveness.

Even if we cannot generate random designs, perhaps we can still say something about their

¹We think of two squares as the same if one can be obtained from the other by permuting rows and columns.

properties. Until recently, there has only been sporadic result on this. For example, Babai showed in the 80s that random STSs have a trivial automorphism group [1], providing evidence that random designs do indeed provide bias-free constructions for experimental designs.

Absorption: A new way to construct designs. In 2014, Peter Keevash announced a result that shook the mathematical community. He had found a new way to construct designs through the *absorption method* [6], provided that the divisibility conditions were satisfied. To think about Keevash’s method, consider the following random process which aims to build an n -LS. As when we considered the random distribution in Figure 2, we are going to use random dice rolls. In each step we pick a random row and a random column and fill the corresponding entry with a random symbol but this time, in contrast to the simple approach that we discussed at the beginning of the article, we make sure that we do not violate the rules of an LS. So when we use some symbol in a given row and column, we forbid that symbol from being used again in that row and column. This random process, known as the *semi-random method* or the *Rödl Nibble*, introduced by Vojtěch Rödl in 1985, has been used successfully to tackle similar problems. Rödl showed [11] that this process will get quite far and fill almost all of the cells but unfortunately, we will most likely get stuck. That is, before completing the LS, there will be no cells left where we can place symbols in a valid way, without getting two of the same symbol in the same row or column.

The idea of absorption is to fix this by first putting aside *absorbing structures* made up of some collections of rows, columns and symbols, that have lots of flexibility in how they can contribute to an LS. We then run the random process avoiding these absorbing structures, until we get stuck. At this point, almost all of the remaining entries of the LS have been filled and we reintroduce the absorbing structures using their flexibility to make some room and fill in the remaining empty cells (*absorb* them into the object) thus obtaining a full LS. Of course, this sketch is at a very high level and the real challenge is to define and find absorbing structures in an appropriate way so that they have the power to achieve this. Indeed, before Keevash’s result, both absorption and random processes were well-known techniques in the field of Combinatorics but it was not expected that this approach could handle structures so rigid as designs. Keevash managed to do so by using algebraic techniques to construct absorbing structures and shortly after, Glock, Kühn, Lo and Osthus [4] managed to bypass the need for algebra, adopting a new multi-round absorption process coined *iterative absorption*.

These sets of authors were the first who managed to incorporate randomness into the construction of designs, thus providing templates for experiments that, while not being uniformly random, enjoy many of the desired bias-free qualities that are present in them. Whilst this is a fantastic outcome, their main motivation came purely from showing the *existence* of designs. Indeed, as discussed previously, the construction of designs quickly gets complicated and the use of algebraic and geometric techniques is very limited. For example, Wilson [13] in the 1970s notoriously constructed $(n, k, 2)$ -SSs whenever their existence is not ruled out by the divisibility conditions (the case of STSs was proven already by Kirkman himself!). However in general, (n, k, t) -SSs were not constructed for all possible parameters and a longstanding conjecture of Steiner from 1853 stated they should always exist provided that the divisibility conditions are satisfied² and n is sufficiently large (see [12]). The method of construction of Keevash (and likewise the second group of authors) was flexible enough to tackle this and construct designs for all feasible parameters (with n large enough), which was a huge breakthrough. Indeed before Keevash, not even a single (n, k, t) -SS with $t \geq 6$ was known to exist.

²In fact, Steiner conjectured the existence of even more general designs, where one can choose how many times each t -set is covered. These are also given by the methods of Keevash.

These methods opened up completely new vistas for combinatorial designs. Not only could they construct a wide range of different designs but the methods actually construct *many* distinct designs with some fixed set of parameters, thus providing the best known lower bound for the *count* of designs. Recently, these methods have also been used to give designs with special properties, leveraging the control that we can use on the random process to mould our design in a certain desired way. Indeed, Kwan, Sah, Sawhney and Simkin built on the approach of Glöck et al. to construct so-called *high girth* STSs [9], establishing a famous conjecture of Erdős from 1973.

Towards understanding Random Designs. Unfortunately the absorption processes used to construct designs do not give designs that are close to uniform. Indeed, the absorbing structures used are very delicate and so skew the randomness given by the random part of the construction. Although many designs can be constructed this way, it is not even clear that more than a negligible proportion of the set of all designs are given by these methods. Thus Yates' problem of generating designs that are truly random (or close to random) remains elusive.

Nonetheless, somewhat surprisingly, Kwan realised that if certain properties hold with very high probability in the random process, then one can infer that uniformly random designs actually enjoy these properties also. His methods built on those of Keevash and involved providing lower and upper bounds on the number of ways in which a partial design can be completed in order to estimate how far the uniform distribution on designs skews the distribution on partial designs given by the random process. Using this ingenious analysis, he was able to unlock new properties of random designs. He showed that uniformly random STSs contain *perfect matchings* [8]: there is a collection of disjoint 3-sets covering the vertex set of the STS. Further work building on his methods has established more involved properties of STSs as well as LSs [2, 10].

Despite these recent developments that were unimaginable before the work of Keevash, it is clear that the study of random combinatorial designs is still very much in its infancy and many beautiful and interesting questions remain wide open, in particular the quest to actually generate random designs and thus have access to the perfect experiment constructions!

The first author of this article will start a project on this topic in April 2024, hosted by the second author of this article and funded by a Marie Curie postdoctoral fellowship awarded by the EU.

References

- [1] L. Babai. Almost all Steiner triple systems are asymmetric. *Annals of Discrete Mathematics*, 7:37–39, 1980.
- [2] A. Ferber and M. Kwan. Almost all Steiner triple systems are almost resolvable. *Forum of Mathematics, Sigma*, Vol. 8, p.e39, 2020.
- [3] R.A. Fisher *The design of experiments*. Edinburgh and London: Oliver & Boyd, pp.xi+252 pp, 1935.
- [4] S. Glock, D. Kühn, A. Lo and D. Osthus. The existence of designs via iterative absorption: hypergraph F -designs for arbitrary F . *Memoirs of the American Mathematical Society* 284, monograph 1406, 2023.
- [5] M. T. Jacobson and P. Matthews. Generating uniformly distributed random Latin squares *Journal of Combinatorial Designs*, 4(6):405–437, 1996 1996.
- [6] P. Keevash. The existence of designs *arXiv:1401.3665*, 2014.

- [7] T. P. Kirkman Note on an unanswered prize question. *Cambridge and Dublin Math. J.* 5:255–262, 1850.
- [8] M. Kwan. Almost all Steiner triple systems have perfect matchings *Proceedings of the London Mathematical Society*, 121(6):1468–1495, 2020.
- [9] M. Kwan, A. Sah, M. Sawhney and M. Simkin. High-girth Steiner triple systems *arXiv preprint arXiv:2201.04554*, 2022
- [10] M. Kwan, A. Sah, M. Sawhney and M. Simkin Substructures in Latin squares *Israel Journal of Mathematics*, 256:363–416, 2023.
- [11] V. Rödl On a packing and covering problem *European Journal of Combinatorics*, 6(1):69–78, 1985.
- [12] R. Wilson The early history of block designs *Rend. del Sem. Mat. di Messina*, 9:267–276, 2003.
- [13] R. M. Wilson An existence theory for pairwise balanced designs III. Proof of the existence conjectures *J. Combin. Theory Ser. A*, 18:71–79, 1975.