Mice and cages experimental design

Nicolas Wicker[1]

[1] *Université Lille 1 - UFR de Mathématiques Cité Scientifique 59655 Villeneuve d'Ascq FRANCE et nicolas.wicker@math.univ-lille1.fr*

Résumé. Un plan d'expérience est étudié où des souris doivent être placées dans des cages en respectant certaines contraintes. Trois méthodes différentes sont présentées pour résoudre deux problèmes différents. Dans le premier, les souris sont placées dans des cages avec la contrainte que les voisins doivent être évités. Dans le second, une nouvelle contrainte oblige les souris à changer de côté à chaque étape de telle sorte qu'une moitié des souris ne rencontre que l'autre moitié des souris. Une méthode est présentée pour le premier problème et deux pour le second, dont l'une exploite les corps finis d'une manière simple et semblable à ce qui se fait pour les carrés latins mutuellement orthogonaux.

Mots-clés. plans d'expériences, théorème du mariage, coloration d'arêtes.

Abstract. An experimental design is studied where mice should be placed into cages following a couple of constraints. Three methods are presented for two different problems. In the first one, mice are placed in cages with the constraint of changing cage at each step avoiding neighbours. In the second setting, an additional constraint compels mice to change side at each step so that one half of the mice meets only the other half. A method is presented for the first setting and two for the second one, one of the latter taking advantage of finite fields in a simple and very similar way to what happens when dealing with mutually orthogonal latin squares.

Keywords. experimental design, marriage theorem, edge coloring.

1 Introduction

In neuropsychology, studying depression is an important challenging trend. Of course, to study this kind of behaviour, animal models are usually considered with a lot of precaution as mice are not humans. Thus experimental design is particularly important and some propositions have arised lately Golden *et al.* (2011). Here, we propose two methods to produce experimental designs that can be useful for biomedical studies in the case where mice have to be placed in cages with particular constraints. In the first section, we present a method to place mice in cages with a given set of constraints implying for instance that each mouse must meet each other mouse, must change cage at each step. In the second section, the constraints are stronger, there are two parts in each cage and mice must swap side at each step. Then, the problem does not allow anymore all mice to meet one another. In both cases, trying to generalize the results leads to open problems.

2 No systematic swap setting

Given $2n$ mice and n cages we want to make mice meet in the following way. Each mouse encounters each other mouse once and only once. Cages are disposed in a linear way and contain each two parts; a left and a right side. There is a neighbourhood constraint standing as follows, if at time $t-1$ or at time $t+1$ two mice share the same cage, they cannot be neighbours at time t. Besides, we consider a refractory period r equal to 1, meaning that a mouse can return in a cage only after a refractory period equal to 1. The case $r > 1$ will not be considered so that it is an open problem to solve the experimental design with this additionnal constraint.

We will need in the following P. Hall's marriage theorem (van Lint, 2001) that is recalled here:

Theorem 1. *Given a graph $G(V, E)$ and two subsets X and Y of vertices V, a necessary and sufficient condition for there to be a complete matching from X to Y in G is that $|\Gamma(A)| \geq |A|$ for every $A \subset X$ where $\Gamma(A)$ stands for the neighbours of A.*

The meaning of vertices changes according to the case at hand, but whenever it is used, the idea it to establish if there is a perfect matching between two sets of points X and Y among which some pairs (x, y), with $x \in X$ and $y \in Y$, can match and others cannot. When a match is possible, then an edge exist between the two points of the pair.

The solution is given by the following procedure. First, we need a succession of perfect matchings between symbols so that symbols meet once and only once. This is possible as $2n$ is even, indeed in that case what is needed is an edge coloring of the complete graph K_{2n} (see figure 1) or put it differently a symmetric latin square. In a second step, for each time point the pairs of the perfect matching are assigned to the columns according to the given constraints. This can be done thanks to lemma 1 if $n > 2$. As last step, symbols can be permuted inside columns to satisfy the neighbourhood constraint thanks to lemma 2

Lemma 1. *It is always possible to associate a symbol pair to a column.*

Proof. A way to prove it is to follow the proof of theorem 17.1 in van Lint (2001) stating that a $t \times n$ latin rectangle can always be completed into a $n \times n$ latin square. First, we define B_j as the set of pairs that can be chosen for column j. Each pair can occur in $n-2$ sets B_j as each pair contains two symbols, both of which have visited a different column at time t. Besides, at time t column j has contained 2 different symbols so that B_j contains $n-2$ pairs as each of these symbols has been associated to a new symbol. Then, if we take l sets $B_{i1}, \ldots, B_{in}$, they contain at least $l(n-2)$ pairs and, as each pair belong to only $n-2$ columns, then there must be at least l different pairs in the l sets. Consequently, Hall's theorem 1 can be applied, so that each B_i is connected to a different symbol pair, meaning that there is a perfect matching between the pairs and the columns verifying the problem constraints. $\square$

Of course, lemma 1 can lead to neighbourhood constraint violations. That's why we present lemma 2 to show a way to correct these violations.

Lemma 2. *It is always possible to adjust a line t so that matchings in row $t+1$ satisfy the neighbourhood constraint if the edge coloring procedure described in figure 1 has been used and if $n > 2$.*

Proof. At this point, some notations are needed to distinguish the different column types at a given step. They will be denoted C_0, C_1, C_2 and C_3. Columns C_0 have no conflict at all. C_1 indicates that there is one conflict with the left or not exclusively with the right column, this is the case for column CD in row t as B and C are neighbours in row $t+1$. Type C_2 is for columns having a double conflict with the same column involving all sides. In the table below, in row t, GH and IJ are of type C_2 as G must avoid J, as G and J are in the same cage in row $t+1$, and H must avoid I also due to row $t+1$. In the last case, C_3, there is a double conflict of another kind, then a symbol is in conflict with both symbols of the neighbour column. This is the case for B which must avoid as well C and D.

$t-1$	C	E	A	I	H	J	B	D	G	F
t	A	B	C	D	E	F	G	H	I	J
$t+1$	H	I	A	E	G	J	B	C	D	F

With these notations, row t can be described by $C_3C_1C_1C_2C_2$. Then, conflicts are suppressed through the following two steps. First, all columns of type C_3 are swapped, which means that inside each column of type C_3 the left hand side and the right hand side are exchanged. In the second step, going from one side (let's say the left side), to the other side, each column having a conflict with the preceding one is swapped. With the given example, this leads to a new row t given by :

t	**B**	**A**	C	D	E	F	**H**	**G**	I	J

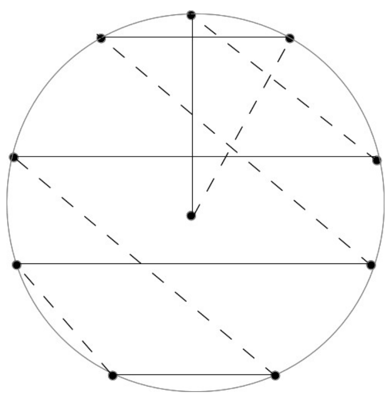
That way no conflict exists anymore. This is true as existing conflicts are suppressed and no new is created. To show the latter, two cases need to be considered. Firstly, when a column of type C_3 is swapped, it is changed into a C_0 column. Secondly, in the second step remaining conflicts are resolved.

Let us consider first a column of type C_3. Without loss of generality let us suppose that such a column contains symbols S_1S_2 followed by symbols S_3S_4 with a double conflict between S_2 and S_3 and between S_2 and S_4. Then, when S_1 and S_2 are swapped, S_2 cannot be in conflict with its preceding symbol as in that case S_2 would have three conflicts which is impossible. Besides, S_1 cannot be in conflict with S_3. Indeed, if there would be a conflict between them, then S_3S_4 would have been of type C_3 and for this reason also swapped. So, the last thing to be checked is that no new conflict arises between S_4 and S_1 if both their columns are swapped. Such a conflict would involve that in row $t-1$, we would have had either pairs S_1S_4, S_2S_3 or pairs S_1S_3, S_2S_4. By the edge coloring procedure, it is clear that four symbols cannot meet one another in closed group at two consecutive steps if $2n$ is larger than 4.

At last, now consider columns of type C_1 and C_2. Each time such a column is swapped, the conflict with the preceding column is suppressed as by definition such a column has a single conflict with the preceding column. Besides, it cannot create a double conflict of type C_3 with the following column as this would mean that the left symbol would have three conflicts which is impossible. Consequently, the two-steps procedure solves all conflicts.

□

Now, let us give an example for $n = 5$ on figure 1.



9	1	6	5	7	4	8	3	0	2
8	5	9	2	1	3	0	4	6	7
2	4	8	7	0	6	1	5	9	3
7	1	3	5	4	9	2	6	8	0
9	5	1	0	2	8	3	7	4	6
3	0	9	6	5	7	4	8	1	2
6	8	3	2	4	1	5	0	9	7
2	5	7	0	8	9	6	1	3	4
9	0	5	4	6	3	7	2	8	1

Figure 1: Illustration of edge-coloring and example of a solution for $n = 5$. Each column represents a different cage and each line a different experimental day.

3 Systematic swap setting

In the present setting, we keep the previous constraints and in addition mice must change side at each time point. So that, mice can only meet half of the total population which is divided into two equal parts. There are n mice of type L, n mice of type R and n cages. They are so-called because L mice are initially on the left side and R mice on the right side. At each time point, mice of type L encounter mice of type R and finally all mice of type L have encountered all mice of type R.

We consider first the simplest case where n is a prime number. This eases much the problem as calculations can be conducted in $\mathbb{F}_n$.

Lemma 3. *If n is prime, columns are chosen for symbols $x_1, \dots, x_n$ of type L and symbols $y_1, \dots, y_n$ of type R according to respectively formula $i + (t-1)k$ and $i + (t-1)l$ with $|k - l| > 1$ then prescribed constraints are respected.*

Proof. If n is a prime number, column choice can be obtained by the formula: $i+(t-1)k$ for row t. This means that initially, at time 1, i is the column of symbol x_i. Similarly, $i+(t-1)k$ indicates the position of symbol y_i of type R. We notice that $i+(t-1)k = j+(t-1)l$ has a unique solution, given by $t = 1+(i-j)(l-k)^{-1}$ which is possible as we are working in $\mathbb{F}_n$. Thus, each L mouse meets each R mouse once and only once. Besides, it is also obvious that each column is visited only once as: $i+(t-1)k = c$ has a unique solution $t = (c-i)k^{-1}+1$. Finally, the last constraint to be verified is that two neighbours at time t cannot be in the same column at time $t-1$ or at time $t+1$. This can be decomposed into two conditions, firstly $x+k \neq x+1+l$ where x and $x+1$ are two neighbour columns and secondly $x+k \neq x+l+1$ which is the same equation but means this time that if at a given time two symbols share the same column x, at next time point they cannot be in neighbour columns. Both conditions comes down finally to $|k-l| > 1$ as $k = l$ is obviously not possible. $\square$

In the general case, we proceed in a similar fashion to the method followed in the first setting. That is, first symbols are matched and then pairs of symbols are assigned to columns.

Lemma 4. *In the general case where n is not prime, a sufficient condition for the problem to be solved is that $n > 11$.*

Proof. At time $t=1$, no difficulty is encountered. At subsequent time points $t+1$, letting aside the constraint on neighbours, a symbol of type L can meet any symbol of type R except symbols already met, that makes $n-t$ possibilities. Using Hall's marriage theorem, as in the proof of lemma 1, this can be satisfied. Afterwards, the pair can be affected to $n-2$ columns. This is again made possible by Hall's theorem.

If now the neighbourhood constraint is considered, a difficulty arises. Indeed, a symbol can happen to be the neighbour at time t of a symbol it meets either at time $t-1$ or at time $t+1$. Fortunately, this problem can be tackled. First, let us call conflictual two symbols that are neighbours at step t and which should meet at time $t-1$ or $t+1$. Conflicts are then resolved iteratively. Let us consider the following situation occuring at time t:

P_1	P	P_2		Q_1	Q	Q_2
	$A \quad B$	$C \quad D$				

Here, we suppose without loss of generality that the conflict is between B and C and look for a pair Q that could be exchanged with P so as to reduce the number of conflicts. A permutation is only possible if the four following conditions are met, P_1 and P_2 should not be in conflict with Q and Q_1 and Q_2 not in conflict with P. P can be in conflict with three other column besides P_2 and two columns are not possible because of the refractory period, so $n-6$ columns are left as $P \neq Q$. Besides, Q cannot be in conflict with P_1 and P_2 so 3 columns can still be discarded and on top of that 2 owing to the refractory period. Finally, only $n-11$ columns are left. Therefore, if $n > 11$ it is possible iteratively to suppress all conflicts. $\square$

11	13	10	14	9	15	8	16	7	17	6	18	5	19	4	20	3	21	2	22	1	23	0	12
20	5	13	0	23	2	15	10	22	3	17	8	14	11	19	6	16	9	12	1	18	7	21	4
7	19	2	12	4	22	8	18	11	15	5	21	9	17	3	23	6	20	0	14	10	16	1	13
14	1	17	10	12	3	16	11	22	5	18	9	20	7	13	2	19	8	23	4	21	6	15	0
6	22	8	20	0	16	9	19	7	21	3	13	10	18	5	23	11	17	1	15	4	12	2	14
15	2	17	0	22	7	18	11	14	3	21	8	12	5	20	9	23	6	13	4	19	10	16	1
1	17	4	14	6	12	9	21	11	19	7	23	10	20	5	13	8	22	2	16	0	18	3	15
23	8	21	10	16	3	20	11	22	9	14	5	12	7	18	1	13	6	15	4	17	2	19	0
4	16	1	19	9	23	7	13	0	20	8	12	10	22	5	15	11	21	2	18	6	14	3	17
20	1	16	5	18	3	15	6	22	11	14	7	12	9	19	2	23	10	17	4	13	8	21	0
8	14	10	12	6	16	11	23	4	18	1	21	7	15	5	17	9	13	2	20	0	22	3	19
13	10	14	9	12	11	19	4	15	8	17	6	21	2	16	7	23	0	18	5	20	3	22	1

Figure 2: Example of a solution for $n = 12$ in the second setting. Again, each column represents a different cage and each line a different experimental day.

This method has been applied for $n = 12$ on figure 2.

4 Conclusion

Even if the proposed solutions are already useful for biologists, some extensions are needed. Indeed in both settings, only a refractory period r of length 1 has been considered, this should be generalized to prevent mice from going too often in the same cage and we presume that both problems can be solved systematically for higher values of r. Another extension of this work, would be to prevent mice from seeing each other twice consecutively.

Acknowledgements

The author is deeply grateful to Mathilde Henry who presented to him the problem of mice and cages experimental design.

Bibliographie

[1]Golden, S.A., Covington, H.E., Berton, O. and Russo, S.J. (2011), A standardized protocol for repeated social defeat stress in mice. *Nature Protocols*, 6(8), 1183–1191.
[2]Soifer, A. (2008), *The Mathematical Coloring Book*, Springer-Verlag.
[3]van Lint, J.H. and R.M. Wilson, R.M. (2001), *A course in combinatorics*, Cambridge University Press.