CS-BIGS 7(1):26-32 http://www.csbigs.fr

Mice and cages experimental design

Nicolas Wicker
University of Lille 1, France

An experimental design is studied where mice should be placed into cages according to two different constraint settings. In the first one, mice are placed in contiguous cages with the constraint of changing cage at each day t, avoiding neighbours of days $t-1$ and $t+1$ and meeting in the end all other mice. 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. Thus experimental design is particularly important and some propositions have arised lately (Golden et al., 2011). Here, we propose three 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. The experimental disposition of cages is represented in Figure 1. Mice can see each other if they are adjacent or in the same cage, for example mouse 2 and mouse 3 can see each other. Henceforth, they will be called neighbours.

In section 2, 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 and avoid mice from neighbour cages. In section 3, 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.

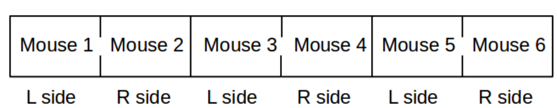
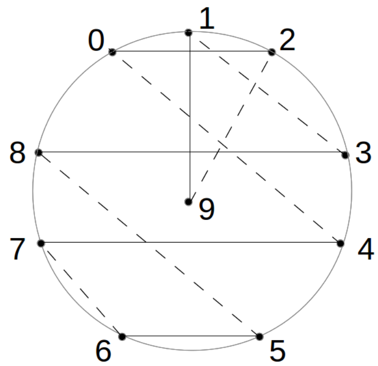


Figure 1: Three cages placement if the number of mice is $n = 6$ with representation of left and right sides. The small opening in the middle of each cage indicates the place where both mice of a cage can meet.

2. No systematic swap setting

Given $2n$ mice and n cages we want to make mice meet in the following way. Each mouse



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

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 day t two mice share the same cage, they cannot be neighbours at days $t-1$ and $t+1$. 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 and Wilson, 2001) that is recalled here:

Theorem 1. *Given a bipartite graph $G(V, E)$ with $V = X \cup Y$, 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 complete 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 exists between the two points of the pair.

The solution is given by the following procedure. First, we need a succession of complete matchings between mice so that mice 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 2) (Soifer, 2008) or put it differently a symmetric latin square. In a second step, for each day the pairs of the complete matching are assigned to the columns according to the given constraints. This can be done thanks to lemma 2 if $n > 2$ and is illustrated again with the same example from Figure 2 on Figure 3, where pairs of mice are connected to possible columns (letting aside at this step the neighbourhood constraints) for days 1 and 2. As last step, mice can be permuted inside columns to satisfy the neighbourhood constraint thanks to lemma 3

Lemma 2. *For each day $t \in 2n-1$, it is possible to associate mice pairs to columns $1, \ldots, n$ satisfying the refractory period constraint for $r = 1$ if $n > 2$.*

Proof. A way to prove it is to follow the proof of theorem 17.1 of Van Lint and Wilson (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 mice, both of which have visited a different column at day t. Besides, at day t column j has contained 2 different mice so that B_j contains $n-2$ pairs as each of these mice has been associated to a new mouse. Then, if we take l sets $B_{i1}, \ldots, B_{il}$,

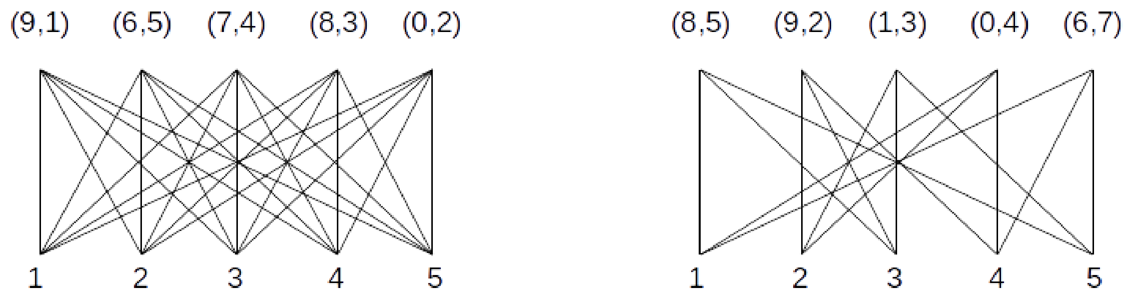


Figure 3: Bipartite graphs for days 1 and 2 using the example of Figure 2.

they contain $l(n-2)$ pairs and, as each pair belongs 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 mice pair, meaning that there is a complete matching between the pairs and the columns verifying the problem constraints. □

As, lemma 2 can lead to neighbourhood constraint violations, we present lemma 3 to show a way to correct these violations.

Lemma 3. *It is always possible to adjust a row t so that matchings in row $t+1$ satisfy the neighbourhood constraint if the edge coloring procedure described in Figure 2 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 and C_2. They are represented on Figure 4

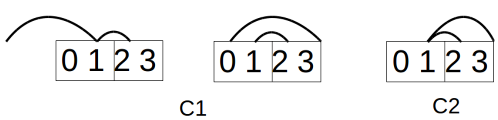


Figure 4: Conflict types C_1 and C_2 are represented with two examples for C_1.

Columns C_0 have no conflict at all. C_1 indicates that there is for each mouse in the column only one conflict with its neighbour column. For example in the table below, column $8-9$ in row t is of type C_1 as 8 is with 7 in row $t-1$. In the last case, C_2, there is a double conflict of another kind, a mouse is in conflict with both mice of the neighbour column. This is the case for 7 which must avoid as well 8 and 9.

$t-1$	9	4	7	8	2	0	1	3	6	5
t	0	1	2	3	4	5	6	7	8	9
$t+1$	7	9	0	4	6	8	1	2	3	5

With these notations, row t can be described by $C_2C_2C_1C_2C_1$. Then, conflicts are suppressed through the following two steps. First, all columns of type C_2 are swapped, which means that inside each column of type C_2 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	1	0	3	2	4	5	7	6	9	8

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_2 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_2. Without loss of generality let us suppose that such a column contains mice S_1S_2 followed by mice 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 mouse 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_2 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 mice 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. Each day 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_2 with the following column as this would mean that the left mouse would have three conflicts which is impossible. Consequently, the two-steps procedure solves all conflicts. □

Algorithm 1 Guide to the practitioner in the no systematic swapping setting.

Day 1 : define the set of mice pairs as $C_1 \leftarrow \{(2n-1,1),(0,2)\} \cup \bigcup_{i \in 2,\dots,n-1} (2n-i, i+1)$
for Day $t = 2$ to $2n-1$ **do**
  $C_t \leftarrow \emptyset$
  for all $(i,j) \in C_{t-1}$ **do**
    if $i = 2n-1$ **then**
      $C_t \leftarrow C_t \cup (i, j+1)$
    else
      $C_t \leftarrow C_t \cup ((i+1) \mod (2n-1), (j+1) \mod (2n-1))$
    end if
  end for
end for
for $t = 1$ to $2n-1$ **do**
  apply augmenting path algorithm to assign each pair in C_t to a column among columns 1 to n
end for
use lemma 3 to remove neighbourhood conflicts.

In algorithm 1 the practical steps to follow are given and to make the article self-contained, the augmenting path algorithm (Cormen et al., 1990) to recover a complete matching is recalled in algorithm 2 for the general case of a bipartite graph $G(V,E)$ with $V = X \cup Y$. The final result is provided for $n = 5$ on Figure 2.

Algorithm 2 Finding a complete matching.

initialize $M \leftarrow \emptyset$
repeat
  take a vertex $v \in X \setminus M$
  reach a vertex $w \in Y \setminus M$ taking alternatively edges in $E \setminus M$ and edges in M, this leads to a path P
  $M \leftarrow P \oplus M$ (symmetric difference)
until M is complete

3. Systematic swap setting

In the present setting, we keep the previous constraints and in addition mice must change side at each day. 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 of their cage and R mice on the right side. At each day, 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 the Galois field $\mathbb{F}_n$.

Lemma 4. *If n is prime, columns are chosen for mice $x_0,\dots,x_{n-1}$ of type L and mice $y_0,\dots,y_{n-1}$ of type R according to respectively formula $i+(t-1)k$ and $i+(t-1)l$ with $k \neq l$ and $k-l \neq \pm 1$ 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 day 1, i is the column of mouse x_i. Similarly, $i+(t-1)l$ indicates the position of mouse 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 mouse L meets each mouse R 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 day t cannot be in the same column at day $t-1$ or at day $t+1$. This means that if we have $i + (t-1)k = j + (t-1)l$ then $i + tk \neq j + tl \pm 1$ which implies that $k - l \neq \pm 1$. Besides, as obiously $k \neq l$ we can conclude. □

Algorithm 3 gives the steps to follow to design the systematic swap-setting experiment if n is prime.

Algorithm 3 Guide to the practitioner in the systematic swapping setting when n is prime.

Take k and l verifying $k \neq l$ and $k - l \neq \pm 1$
for Day $t = 1$ to n **do**
  mouse i of type L goes in cage $(i + (t-1)k) \mod n$ on the left side if $i = 1 \mod n$, on the right side otherwise
  mouse i of type R goes in cage $(i + (t-1)l) \mod n$ on the right side if $i = 1 \mod n$, on the left side otherwise
end for

An example of this experimental design is given for $n = 5$ in Table 1 taking $(x_0, x_1, \ldots, x_4) = (0, 1, \ldots, 4)$, $(y_0, y_1, \ldots, y_4) = (5, 6, \ldots, 9)$ and $(k, l) = (1, 3)$.

Table 1: Example of a solution for $n = 5$ in the second setting using the primality of 5. Again, each column represents a different cage and each row a different experimental day.

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

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

Lemma 5. *In the general case where n is not prime, a sufficient condition for the problem to be solved is that $n > 11$.*

Proof. At day $t = 1$, no difficulty is encountered. At subsequent days $t+1$, letting aside the constraint on neighbours, a mouse of type L can meet any mouse of type R except mice already met, that makes $n - t$ possibilities. Using Hall's marriage theorem, as in the proof of lemma 2, 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 mouse can happen to be the neighbour at day t of a mouse it meets either at day $t-1$ or at day $t+1$. Fortunately, this problem can be tackled. First, let us call conflictual two mice that are neighbours at step t and which should meet at day $t-1$ or $t+1$. Conflicts are then resolved iteratively. Let us consider the following situation occuring at day t:

P_1	P	P_2		Q_1	Q	Q_2
	0 1	2 3				

Here, we suppose without loss of generality that the conflict is between 1 and 2 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 columns 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. □

Table 2: Example of a solution for $n = 12$ in the second setting.

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

Algorithm 4 deals with the case where n is not prime. Basically, it is close to algorithm 1 making repeated use of algorithm 2.

Algorithm 4 Guide to the practitioner in the systematic swapping setting when n is not prime.

Day 1 : mice L are paired randomly with mice R
for Day $t = 2$ to n **do**
  create a graph where edges are between mice L and R which have not met each other yet
  realize the bipartite matching using algorithm 2
end for
for $t = 1$ to n **do**
  create a graph where edges are between mice pairs at day t and allowed columns
  realize the bipartite matching using algorithm 2
end for
use lemma 5 to remove neighbourhood conflicts.

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

4. Conclusion

Even if the proposed solutions are already useful for biologists, some extensions are needed. Indeed in both settings, the refractory period r after which a mouse can return to the same cage has only been considered to be equal to 1. 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. He would also like to thank the anonymous reviewer for his constructive comments. This work was supported in part by the Labex CEMPI (ANR-11-LABX–0007-001).

References

Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (1990). *Introduction to algorithms*. MIT Press.

Golden, S., Covington, H., Berton, O., and

Russo, S. (2011). A standardized protocol for repeated social defeat stress in mice. *Nature Protocols*, 6(8):1183–1191.

Soifer, A. (2008). *The Mathematical Coloring Book*. Springer-Verlag, New York.

Van Lint, J. and Wilson, R. (2001). *A course in combinatorics*. Cambridge University Press, Cambridge.

Correspondence: nicolas.wicker@math.univ-lille1.fr.