An "eulerian" approach to a class of matching problems

A.M. BERSANI
Dipartimento di Metodi e Modelli Matematici,
Università "La Sapienza" di Roma,
Via A. Scarpa 16, 00161 Roma, Italy
e-mail: bersani@dmmm.uniroma1.it

Abstract. We study a card game called *He Loves Me, He Loves Me Not* ($(HLM)^2N$), which can be considered as a generalization of the classical game *Mousetrap*. We present some results, based, on one hand, on theoretical considerations and, on the other one, on Monte Carlo simulations. Furthermore, we introduce a new technique which allows us to obtain the best result at least for French card decks (52 cards with 4 suits). This technique allows us to answer some open questions also related to the game *Mousetrap*.

1. Introduction

In 1857 Cayley **[2]** proposed a game called *Mousetrap*, played with a deck containing only one suit; here we report the description given in **[8, p. 237]**:

"Suppose that the numbers $1, 2, ..., n$ are written on cards, one to a card. After shuffling (permuting) the cards, start counting the deck from the top card down. If the number on the card does not equal the count, transfer the card to the bottom of the deck and continue counting. If the two are equal then set the card aside and start counting again from "one". The game is won if all the cards have been set aside, but lost if the count reaches $n + 1$."

Cayley posed the fundamental question **[3]**: "Find all the different arrangements of the cards, and inquire how many of these there are in which all or any given smaller number of the cards will be thrown out; and (in the several cases) in what orders the cards are thrown out."

Relatively few authors (in chronological order: **[3]**, **[18]**, **[8]**, **[10]**, **[12]**, **[9]**, **[11]**) have studied the problem, arriving, only recently **[9, 12]**, at partial results.

In **[8, p. 238]**, **[9]** and **[10]** Guy and Nowakowski consider another version of the game, called *Modular Mousetrap*, where, instead of stopping the game when no matching happens counting up to n, we start our counting again from "one", arriving either to set aside every card or at a loop where no cards can be set aside anymore. Obviously, in this game, if n is prime, we have only two possibilities: either *derangement*, where no coincidences occur, or winning deck. Guy and Nowakowski study only a small number of cases ($n \leq 5$), while in the case $n = 6$ they consider only decks starting with an "ace".

AMS 2000 Subject Classification: 05A05, 05A10, 05A15, 60C05, 65C50
Key words: permutations, derangement, Mousetrap, Treize, Monte Carlo simulations, Computational Combinatorics
OEIS: A000166, A001563, A002467, A002468, A002469, A007709, A007710, A007711, A007712, A018931, A018932, A018933, A018934, A028305, A028306, A055459, A067950

Guy and Nowakowski's paper is mostly focused on the problem of reformed permutations, i.e., decks obtained as lists of cards in the order they were set aside in a winning deck. This problem will be taken into consideration in another paper [**1**].

The games are studied in the case of only one suit. Here we introduce for the first time the generalized version of *Mousetrap* to the case of several suits (*"multisuit" Mousetrap*: $n = m \cdot s$).

Mousetrap rules could be generalized at least in two different ways: when the player has counted up to m, without coming to a card which ought to be thrown out, he can

a) either stop the game (*Mousetrap-like rule*)

b) or eliminate the last m cards and continue his counting, restarting from "one", up to the exhaustion of the deck, when all the cards have been eliminated or stored.

We choose the second option, that recalls a different *solitaire*, which we consider in Section 2. It is not known in the mathematical literature, but, as told in [**13**], it has been studied for a relatively long time. It is commonly called *He Loves Me, He Loves Me Not* ($(HLM)^2N$) or *Montecarlo*:

"From a deck with s suits and m ranks, deal all the cards into a pile one at a time, counting "one", "two", "three" etc. When a card whose value is k proves to be of the rank you call, it is *hit*. The card is thrown out and stored in another pile, the score is increased by k, the preceding $k-1$ cards are put at the end of the deck, in the same order in which they were dealt and you start again to count "one", "two", "three" etc. If you have counted up to m without coming to a card has been thrown out, the last m cards are "burned", i.e., definitively discarded and you begin the count afresh, counting "one", "two", "three" etc. with the residual cards. When the number n_c of cards in the residual deck is less than m, the count can arrive, at most, at the value n_c. The game ends when you have stored and/or "burned" all the cards and there are no more cards in the deck. The score of the game is given by the sum of the face values of all the stored cards."

The aim of the game is to achieve the greatest possible score.

Up to now, this game has been studied only by means of Monte Carlo simulations, separately by Andrea Pompili [**13**] and by the author.

In this paper we introduce a new technique, which allows us to obtain the number of winning decks for many values of m and s, without any need of simulations, not only for $(HLM)^2N$, but also for *Mousetrap* and *Modular Mousetrap*, in their "multisuit" version, too. The technique has been implemented in a computer program. New results have been obtained in a very efficient way and many others could be reached, if the algorithm could be implemented in a parallel calculus framework.

The paper is divided into six Sections.

In Section 2 we recall the most important results related to the game *Mousetrap*; moreover we consider the introductory notions of $(HLM)^2N$ and state two conjectures: a stronger one (SC) and a weaker one (WC), concerning the possibility to find at least one winning deck.

In Section 3 we briefly describe the algorithm, based on Monte Carlo simulations, with which we obtained winning decks just for a small number of cases; up to now, it represented the unique method used to validate the two conjectures.

In Section 4 we show a completely new method, which is highly performing and which allows us not only to give a positive answer to (SC) at least up to a deck of French cards ($m = 13; s = 4$), but, for a large range of m and s, gives the exact number of winning decks, i.e., of decks giving the best reachable score. Thanks to the new method, an answer to the question of the number of winning decks at $(HLM)^2N$ is given, up to

$s = 2$, $m = 15$; $s = 3$, $m = 9$; $s = 4$, $m = 7$.

Moreover, by means of the new technique, we give, in a very easy way, a positive answer to a question originally posed by Cayley [**2**].

In Section 5, adapting the new technique to the games *Mousetrap* and *Modular Mousetrap* and to their "multisuit" versions, we extend the results attained in [**9**] up to $m = 15$, $s = 1$ and to *"multisuit" Mousetrap*. Thanks to the new method introduced in Section 4, we give an answer to the question of the number of winning decks, up to $s = 1$, $m = 15$; $s = 2$, $m = 8$; $s = 3$, $m = 6$; $s = 4$, $m = 5$ for *Mousetrap*; $s = 1$, $m = 13$; $s = 2$, $m = 7$; $s = 3$, $m = 5$; $s = 4$, $m = 4$ for *Modular Mousetrap*.

In Section 6 we give a short review of open problems and perspectives.

2. Introductory notions and preliminary results on *Mousetrap* and $(HLM)^2N$

There are few results on *Mousetrap*, obtained, in particular, by Steen [**18**], already in 1878 and, much more recently, by Guy and Nowakowski [**9**] and Munfrom [**12**].

Cayley [**2**] proposed to investigate, at *Mousetrap*, whatever the number n of cards is, which permutations throw out the cards in the same order of their numbers. He obtained the corresponding permutations for $n \leq 8$:

$$1 \;\; ; \;\; 1\,2 \;\; ; \;\; 1\,3\,2 \;\; ; \;\; 1\,4\,2\,3 \;\; ; \;\; 1\,3\,2\,5\,4 \;\; ; \;\; 1\,4\,2\,5\,6\,3 \;\; ; \;\; 1\,5\,2\,7\,4\,3\,6 \;\; ; \;\; 1\,6\,2\,4\,5\,3\,7\,8 \,.$$

Guy and Nowakowski observed that not all the permutations are reformed permutations. On the other hand, the identity permutation $1\ 2\ \cdots\ n$ is always a reformed permutation. Since it is not possible, in general, to arrange the cards so that all the cards may be thrown out in a predetermined order, Cayley [**3**] posed the following questions:

1) for each n find the winning permutations of $1\ 2\ \cdots\ n$;

2) for each n find the number of permutations that eliminate precisely i cards for each i , $1 \leq i \leq n$.

He studied the game *Mousetrap* in the case $n = 4$, analyzing the $4! = 24$ different decks. Curiously, he made mistakes in six cases.

Steen [**18**], already in 1878 and, much more recently, Guy and Nowakowski [**9**] and Mundfrom [**12**], obtained deeper results. Steen calculated, for any n, the number $a_{n,i}$ of permutations that have i , $1 \leq i \leq n$, as the first card set aside and the numbers $b_{n,i}$ and $c_{n,i}$ of permutations that have "one" (respectively "two") as the first hit and i as the second. He obtained the following recurrence relations:

$$a_{n,1} = (n-1)! \quad , \quad a_{n,i} = a_{n,i-1} - a_{n-1,i-1} \quad , \quad b_{n,i} = a_{n-1,i-1} \quad , \quad \forall i = 2,..,n \tag{2.1}$$

$$c_{n,i} = c_{n,1} - (i-1)c_{n-1,1} + \sum_{k=2}^{i-2} (-1)^k \cdot \frac{i(i-1-k)}{2}\ c_{n-k,1} \qquad \forall n > i+1 \tag{2.2}$$

Denoting with $a_{n,0}$ the number of *derangements*; $a_n = \sum_{k=1}^{n} a_{n,k}$ the total number of permutations which give hits; $b_{n,0}$ the number of permutations giving "one" as the only hit; $b_n = \sum_{k=2}^{n} b_{n,k}$ the total number of permutations giving a second hit, "one" being the first; $c_{n,0}$ the number of permutations giving "two" as the only hit; $c_n = \sum_{k=1}^{n} c_{n,k}$ $(k \neq 2)$ the total number of permutations giving a second hit, "two" being the first; putting $a_{0,0} = 1$, Steen showed that, for $0 \leq i \leq n$

$$a_{n,0} = a_{n+1,n+1} \quad , \quad a_{n,0} = n a_{n-1,0} + (-1)^n \quad , \quad a_{n,i+1} = \sum_{k=0}^{i} (-1)^k \binom{i}{k} (n-1-k)! \tag{2.3}$$

$$b_{n,i} = a_{n-1,i-1} = a_{n-2,i-2} - a_{n-3,i-2} \quad , \quad b_{n,0} = a_{n,1} - b_n = a_{n,1} - a_{n-1} = a_{n-1,0} \tag{2.4}$$

$$a_n = n\ a_{n-1} + (-1)^{n-1} \quad , \quad b_n = a_{n-1} \tag{2.5}$$

$$c_{n,i} = \left[\sum_{k=1}^{i-3}(-1)^{k+i-1}\ \frac{k(k+3)}{2}\ (n-i+k-1)!\right] - (i-1)(n-3)! + (n-2)!\ . \tag{2.6}$$

Guy and Nowakowski [**9**] and Mundfrom [**12**] showed separately that Steen's formula (2.6) is not valid for $i = n-1$ and $i = n$ and gave the exact relations. We quote the expressions, together with the equation for $c_n = \sum_{k=1}^{n} c_{n,k}$, $k \neq 2$, as shown by Guy and Nowakowski [**9**], thanks to their compactness:

$$c_{n,n-1} = \sum_{k=0}^{n-3}(-1)^k \binom{n-3}{k}\ (n-k-2)! \tag{2.7}$$

$$c_{n,n} = (n-2)! + \left[\sum_{k=0}^{n-5}(-1)^{k+1}\left(\binom{n-4}{k} + \binom{n-3}{k+1}\right)\ (n-k-3)!\right] + 2(-1)^{n-3} \tag{2.8}$$

$$c_n = (n-2)(n-2)! - \left[\left[\frac{1}{e}((n-1)! - (n-2)! - 2(n-3)!\right]\right]\ , \tag{2.9}$$

where $[[x]]$ is the nearest integer to x.

Steen [**18**], Guy and Nowakowski [**9**] and Mundfrom [**12**] elaborated some tables related to formulas (2.1) - (2.9). The sequences there reproduced are quoted by Sloane [**15**], [**16**], [**17**] in the following way:

$\{a_n\}_{n\in\mathbb{N}}$ ([**18**]):	[**15**] N1423, [**16**] M3507, [**17**] A002467;
$\{a_{n,0}\}_{n\in\mathbb{N}}$ ([**18**]):	[**15**] N0766, [**16**] M1937, [**17**] A000166;
$\{a_{n,2}\}_{n\geq 2}$ ([**18**]):	[**15**] N1436, [**16**] M3545, [**17**] A001563;
$\{c_n\}_{n\geq 2}$ ([**9**], [**12**], [**18**]):	[**15**] N1186, [**16**] M2945, [**17**] A002468;
$\{c_{n,0}\}_{n\geq 2}$ ([**12**], [**18**]):	[**15**] N1635, [**16**] M3962, [**17**] A002469;
$\{c_{n,3}\}_{n\geq 3}$ ([**12**], [**18**]):	[**17**] A018931;
$\{c_{n,4}\}_{n\geq 4}$ ([**12**], [**18**]):	[**17**] A018932;
$\{c_{n,5}\}_{n\geq 5}$ ([**12**], [**18**]):	[**17**] A018933;
$\{c_{2,1}\} \cup \{c_{n,n}\}_{n\geq 3}$ ([**12**], [**18**]):	[**17**] A018934.

Let us observe that, owing to his mistakes in the formula for $c_{n,i}$, Steen reported erred sequences for $\{c_n\}_{n\geq 2}$, $\{c_{n,0}\}_{n\geq 2}$ and $\{c_{2,1}\} \cup \{c_{n,n}\}_{n\geq 3}$. The correct sequences, obtained by Mundfrom, are quoted as [**17**] A002468, A002469 and A018934. Guy and Nowakowski [**9**] extended the correct form of the sequence [**17**] A002468 up to the value $n = 20$.

Sequences [**17**] A000166 of *derangements* $\{a_{n,0}\}$ and A002467 of permutations with at least one fixed point arrive at $n = 21$, but can be easily improved by means of the following classical result, based on the *inclusion-exclusion principle* [**6, Ch. 4, pp. 88–103**], [**7, pp. 136–137**], [**14, Ch. 3, pp. 50–65**]:

Proposition 2.1. *The probability of derangement for the games Mousetrap (M) and Modular Mousetrap (MM) is*

$$P_{M,m}(0) = P_{MM,m}(0) = \sum_{k=0}^{m}\ \frac{(-1)^k}{k!}\ . \tag{2.10}$$

and

$$\lim_{m\to\infty} P_{M,m}(0) = \lim_{m\to\infty} P_{MM,m}(0) = Po_1(0)\ =\ e^{-1}\ ,$$

where $Po_1(k)$ is the poissonian distribution with characteristic parameter 1: $Po_1(k) = \dfrac{e^{-1}}{k!}$.

As a partial answer to question **2)** by Cayley, Guy and Nowakowski [**9**] produced a table, giving the numbers of permutations eliminating just i cards ($1 \leq i \leq 9$); the diagonal represents the numbers of winning permutations, i.e., permutations setting aside all the n cards and represents a partial answer to question **1)** by Cayley. Guy and Nowakowski computed the terms up to $n = 9$. Since the table does not derive from any closed formula, it was probably obtained by means of direct computations, considering that, for $n = 9$, it is possible to check, by means of a computer, all the permutations, whose number is equal to $9! = 362880$.

This table is quoted as [**17**] A028305, but only up to $n = 7$.

We can derive other sequences from this table: the first column is the sequence [**17**] A000166 of *derangements*. The second column is the sequence [**17**] A007710 ([**16**] M1695) of permutations eliminating just one card. The top diagonal is the sequence [**17**] A007709 ([**16**] M1608) of winning (or reformable) decks, i.e., of decks eliminating all the cards. The sums of the terms of each row, except the terms on the top diagonal, give the first nine terms of the sequence [**17**] A007711 ([**16**] M3546) of unreformed decks, i.e., of decks which do not eliminate all the cards.

Furthermore, Guy and Nowakowski proved the formula for the probability that only the card with value k is set aside from a deck of $n > 2$ cards and showed the related complete table of values, for $1 \leq k \leq n$, $1 \leq n \leq 10$, adding another table, for $11 \leq n \leq 17$, but $1 \leq k \leq 5$.

Sequence [**17**] A028306 quotes the table, but only up to $n = 8$.

Knowing general formulas giving the numbers of permutations that have i as the k-th hit, given the previous $(k-1)$ hits, would be very useful to arrive at a closed formula for the probability distribution of the game. But, as remarked by Steen, already the computations to obtain $c_{n,i}$ are very difficult and it is hard to expect more advanced results in this direction.

In Section 5 we present new results, based not on closed formulas but on Computational Combinatorics tools, which extend the results attained in [**9**] up to $m = 15$, $s = 1$ and to *"multisuit" Mousetrap*.

Finally, Guy and Nowakowski [**9**] yielded some results for the game *Modular Mousetrap*. Their paper is mostly focused on the problem of reformed permutations, i.e., decks obtained as lists of cards in the order they were set aside in a winning deck. We examine this problem in another paper [**1**].

The game *He Loves Me, He Loves Me Not* ($(HLM)^2N$), described in the Introduction, can be played with arbitrary values of m and s.

Since after every matching we start counting again from "one", the game recalls *Mousetrap*. On the other hand, the game differs from *Mousetrap* for the following reasons:

a) we record the sum of the values of the cards, not their number; obviously, in a deck of $m \cdot s$ cards, we can, at most, obtain

$$s \cdot \sum_{k+1}^{m} k \;=\; \frac{s}{2} m(m+1) \quad \text{points ;}$$

b) we "burn", i.e., we eliminate m cards, if no coincidences occur counting from 1 to m, but we do not stop the game and we continue our counting starting again from "one".

We can either stop the game when, remaining in the deck a number $n_c < m$ of cards, we don't obtain any matching counting up to n_c, or, following *Mousetrap* rules, continue our counting up to m; in this second case, if no matching happens counting up to m, the game stops; otherwise we can restart our counting, after having stored the last matching card. In the first case, we play $(HLM)^2N$; in the second we play the *"multisuit" Mousetrap*.

According to the author's opinion, *Mousetrap* and $(HLM)^2N$ are very intriguing, because there is no *a priori* information on any potential winning deck.

Moreover, the rule followed by *Mousetrap* allows the player to store all the $m \cdot s$ cards (in fact, at *Mousetrap*, if we remain with only one card in the deck, we know that we will store it, because we will count up to m visiting always the same card, whose values is, obviously, less or equal to m). Instead, thanks to the following Proposition we know that in $(HLM)^2N$ we can store at most $ms-1$ cards. In other words, when we consider *Mousetrap* with more than one suit, this game is easier than $(HLM)^2N$ and every deck winning at $(HLM)^2N$ wins at *Mousetrap*.

Proposition 2.2. *In $(HLM)^2N$, for every s, m we can at most store $ms-1$ cards and the score cannot exceed*

$$C_{max} := \frac{s}{2}[m(m+1)] - 2 \ . \tag{2.11}$$

Proof The proof is based on contradiction.

Let us suppose that we can store all the $n = m \cdot s$ cards. Since the storage mechanism implies that, once a card is stored, the number of residual cards in the deck is lowered by one, the last stored card lowers the residual deck from one card to no cards. Consequently, the only card storable as the last one is an "ace". Proceeding backward in the storage mechanism, when we store the last but one card, the deck passes from two cards to one. One of these two cards, as already observed, is an "ace". The second one, that must be stored, can be only an "ace" or a "two". But if we want to store the "two", the other card, which precedes it, cannot be an "ace" (otherwise, counting the two cards, we should have first stored the "ace"!). Thus the last two cards must be two "aces". Continuing our process backward and reasoning in the same way as before, since we want to store all the last three cards, the last but two cards must be an "ace", a "two" or a "three". But if the last but two cards is a "two" or a "three", if we want to store it we should not have an "ace" as the first of the three cards, in contradiction with the fact that the other two cards are two "aces". Consequently, the last three cards must be three "aces". The backward reasoning can be iterated, arriving at the conclusion that, for every k, the last k cards must be "aces". But the number of "aces" is equal to s, so, when $k > s$, we arrive at a contradiction. Formula (2.11) immediately follows from the first thesis. □

The crucial question is if it is always possible to find a deck from which we can store all the decks but a "two" and, consequently, we can obtain C_{max}.

We can state the following two conjectures:

Strong Conjecture (SC) For $s = 2 \ ; \ m \geq 6$ and $s \geq 3 \ ; \ m \geq 2$, there exists at least one deck from which we store $sm - 1 = n - 1$ cards, obtaining the best score, i.e.,

$$C_{max} \ = \ \frac{s}{2}m(m+1) - 2 \ .$$

Weak Conjecture (WC) For every $s \geq 2 \ ; \ m \geq 2$, there exists at least one deck from which we store $sm - 1 = n - 1$ cards.

Remark 2.3. For $s = 1$ it is impossible to obtain C_{max}. In fact, let us observe that, for $s = 1$, the only way to store the card with value m consists in putting it in the m-th place, without having any other coincidences in the previous $(m-1)$ places. Let us indicate with $X_1 \ X_2 \ X_3 \ ... \ X_{m-2} \ X_{m-1}$ an arbitrary *derangement* of the first $(m-1)$ cards; thus the m cards have the following sequence in the deck:

$$X_1 \ X_2 \ X_3 \ ... \ X_{m-2} \ X_{m-1} \ m \ .$$

But in the turn following the matching of the card m, the residual deck is formed by $(m-1)$ cards, placed in a *derangement*; consequently we cannot have any other coincidences.

Remark 2.4. For $s = 2$ there exist cases for which it is not possible to obtain the best score given by (2.11). The case $s = 2$, $m = 3$ (90 different decks) can be verified directly, "by hand". The best reachable score, in this case, is 9, instead of 10. In the other cases, with an increasing value of m, we need numerical simulations: for $s = 2$, $m = 4$ (2530 different decks) and for $s = 2$, $m = 5$ (113400 different decks), we obtain, respectively, 17 points, instead of 18 and 27 points, instead of 28. In Section 4 we prove this fact. For $s = 2$, $6 \leq m \leq 13$ we obtained the best score, given by (2.11).

3. Monte Carlo simulations

In order to obtain at least experimental answers to (SC) and (WC) for several values of m and $s \leq 4$, we built up a computer software, based on Monte Carlo simulations (which allow us to approximate the probability distribution by means of the frequency distribution of a sufficiently high number of experiments), according to the following, simple steps:

a) deck "shuffling", by means of random permutations of an initial deck;

b) playing the game: in a vector $\mathbf{C}$, with $\frac{s}{2}[m(m+1)]$ components, the first sm components are filled with the shuffled deck. A cursor passes through all the ordered components. When the first matching happens at a card, whose value is k_1, the preceding $(k_1 - 1)$ cards are put in the same order just after the last nonzero component of $\mathbf{C}$, filling the vector components from the $(ms+1)$-th position to the $(ms+k_1-1)$-th one. The cursor restarts from the $(k_1 + 1)$-th position, counting from "one". The card k_1 is stored and the actual score is increased by k_1 points. Subsequently, at the r-th matching, corresponding to the card k_r, we shift the preceding $(k_r - 1)$ cards, in the same order, just after the last nonzero component of the vector and so on.

Calling n_c the minimum value between the number of residual cards in the deck and m, when no coincidences happen after n_c cards, they are eliminated and if $n_c \leq m$ the game stops because there are no more cards to be "visited".

c) data storage: at the end of every game, if the score exceeds a determined threshold (for example, the previous best score), we store in a data file the score, the number of stored cards and the winning deck. If we are interested in the statistics, all the information for every deck is stored in frequency distribution vectors, letting the program compute the averages of scores, of the number of stored cards and of the values of stored cards. If we are interested only on the best score, when in a deck m consecutive cards are eliminated, due to no coincidences, the deck is discarded, because no longer able to improve the actual best score and the game restarts with a new deck.

The method is very efficient, considering the speed of execution and, in particular, the disk usage for the data storage; in fact, after the game, it is always possible to obtain back the deck we have examined, considering the first $m \cdot s$ components of the card vector $\mathbf{C}$.

The software has been written in FORTRAN code and implemented in a PC, equipped with a Pentium IV. On the other hand, Andrea Pompili, in **[13]**, used a Borland C language.

We can count in three different ways (they are the three ways of counting I know, from direct experience and from literature on *solitaires*, but many other ways could be chosen!):

a) ace (1), m, m-1, m-2, ..., 4, 3 , 2 ;

b) m, m-1, m-2, ..., 4, 3 , 2, ace (1) ;
c) ace (1), 2, 3, ..., m-1, m .

In this paper we choose the option **c)**.

The number of different decks, in $(HLM)^2N$ as in all the "multisuit" games we consider in this paper, is given by

$$N_{m\cdot s} = \frac{(m\cdot s)!}{(s!)^m} \ . \tag{3.1}$$

The presence of the denominator is related to what Doyle, Grinstead e Laurie Snell [**5**] define *rank-derangements*: when $s > 1$, a deck obtained from another one only exchanging the position between cards of the same rank is, playing $(HLM)^2N$ or *Mousetrap*, identical to the original.

Table 1 shows that the possibility to validate the conjectures becomes very hard when m and s increase too much. In order to give an idea of the computational complexity of the problem, let us observe that a French card deck has $\frac{52!}{(4!)^{13}} \sim 9.2\cdot 10^{49}$ permutations (without considering the *rank-derangements*). Supposing that each one of the over 6 billion Earth inhabitants could examine every day 20 billion decks, **each one different from the others and from the decks examined by the other players**, with a computer (this is the actual capacity of my FORTRAN program, implemented in a PC equipped with a Pentium IV), we should need more than $2\cdot 10^{27}$ years to test all the different decks!

The threshold for the number of decks to be checked beyond which the numerical simulations seem to become inadequate is around 10^{20}. Nevertheless, it is noteworthy the case $m = 10\ ,\ s = 2$. In fact, even after more than 600 billion simulations, no evidence of a winning deck appeared, though the number of different decks is "only" almost $2.58\cdot 10^{15}$. In this case, the Monte Carlo method has only given a positive answer to (WC), obtaining, at most, 106 points, instead of $C_{max} = 108$, as predicted in (2.11). This situation could be *a priori* related either to an effective negative answer to (SC) for $m = 10, s = 2$ or to the high number of different decks, in front of a too low number of winning decks. Actually, we answer the question in a surprisingly easy way in the next Section: there are only 656 winning decks and, consequently, the probability of finding one of them is $P_{10\cdot 2}(108) \sim 2.76\cdot 10^{-13}$. Then it seems that we should have needed $\mathcal{O}(10^{12} \div 10^{13})$ simulations to expect to find a winning deck.

4. The "eulerian" approach.

Here we introduce a new technique, which gives much more satisfactory answers than the numerical simulations, in a very efficient way, giving not only a positive answer to (SC) at least up to $m = 13\ ,\ s = 4$, i.e., for the classical deck of French cards (though it can be used to explore much larger decks), but also the exact number of winning decks, and consequently, the exact probability of winning, for a large number of cases, as shown in Table 2.

Let us first explain the method.

Trying an analogy with Rational Mechanics, we can interpret the way to deal the cards one after the other as an *eulerian* description of the game, since the observer is fixed and all the elements of the system, i.e., the cards, pass in front of it, while in the computer schemes another description, consisting in arranging sequentially all the cards and visiting them by means of a cursor, can be interpreted as a *lagrangian* description, because in this case we follow the evolution in time of the whole system, starting from its initial configuration.

The lagrangian description is much more useful in the numerical simulations: in fact, as already observed, after having assigned the first $m \cdot s$ components of the vector $\mathbf{C}$ (which can have, at most, $\frac{s}{2}m(m+1)-1$ components), after every matching the card with value k_1 giving this matching is stored and the preceding $k_1 - 1$ cards are put just after the last nonzero component of $\mathbf{C}$, ready to be visited again by the cursor, which, in this way, never comes back, but continues forward, up to the end of the game.

In other words, the lagrangian description allows us to generate, from every deck, a string, called *lagrangian string*, whose length is, at most, $C_{max} = \frac{s}{2}m(m+1)-1$ (in this case, we played with a winning deck and the last component in the string is a "two"), whose first $m \cdot s$ components give the initial configuration of the system, i.e., the original deck. A *derangement* corresponds to a string with length $m \cdot s$, coinciding with the original deck.

In every lagrangian string the number of times a card whose value is k is repeated varies from deck to deck. Let us consider, just as an example, the only two winning decks, found by means of the Monte Carlo method, in the case $m = 7, s = 4$, in at least 60 billion simulations:

$$\begin{array}{ccccccc} 4 & 3 & 1 & 4 & 7 & 7 & 2 \\ 6 & 5 & 4 & 7 & 3 & 1 & 3 \\ 3 & 6 & 2 & 2 & 1 & 7 & 6 \\ 6 & 5 & 1 & 5 & 4 & 5 & 2 \end{array} \quad (\mathbf{a}) \qquad \begin{array}{ccccccc} 4 & 3 & 7 & 3 & 1 & 6 & 7 \\ 2 & 7 & 5 & 3 & 5 & 2 & 2 \\ 4 & 4 & 6 & 2 & 1 & 7 & 5 \\ 6 & 3 & 1 & 6 & 5 & 1 & 4 \end{array} \quad (\mathbf{b}) \tag{4.1}$$

They generate, respectively, the following lagrangian strings, S_1 and S_2:

4 3 1 **4** 7 7 2 6 **5** 4 7 **3** **1** 3 3 6 2 2 1 **7** 6 6 5 1 **5** 4 5 2
4 3 1 7 7 2 **6** 4 7 **3** 3 6 2 2 1 **6** 6 5 1 **4** 5 **2** 3 1 7 7 2 4
7 3 6 2 2 1 **6** 5 1 5 3 1 7 **7** 2 4 **3** 6 **2** 2 1 5 1 **5** 3 1 7 2
4 **6** 2 1 5 1 3 1 **7** 2 4 2 1 **5** **1** 3 1 2 **4** 2 1 **3** **1** 2 **2** **1** 2

4 3 7 3 1 **6** 7 **2** 7 5 **3** 5 **2** 2 4 4 6 2 1 **7** 5 6 **3** **1** 6 5 1 **4**
4 3 7 3 1 7 **7** 5 5 2 **4** 4 6 2 1 **5** 6 6 5 1 4 3 **7** 3 1 7 5 **5**
2 4 6 2 1 **6** 6 5 1 **4** 3 3 1 7 **5** 2 4 6 2 1 **6** 5 1 **3** 3 1 7 2
4 **6** 2 1 5 1 3 1 **7** 2 4 2 1 **5** **1** 3 1 2 **4** 2 1 **3** **1** 2 **2** **1** 2

where the cards yielding matchings are written in boldface characters. In these strings, the number of times that the different cards are "visited" are respectively:
in S_1: ace: 23 times; 2: 23 times; 3: 15 times; 4: 12 times; 5: 12 times; 6: 12 times; 7: 14 times;
in S_2: ace: 23 times; 2: 19 times; 3: 15 times; 4: 14 times; 5: 15 times; 6: 13 times; 7: 12 times.

We can also consider another string, with the same length of the lagrangian one, formed by the values pronounced by the player, instead of the values "visited". In this case, all the strings of maximal length contain s times the value m, $2s$ times the value $m-1$, $3s$ times the value $m-2$, $s(m-k+1)$ times the value k ($k \neq 2$), down to $s(m-2)$ times the value 3, $sm = n$ times the value 1 e $s(m-1)-1$ times the value 2.

By construction, these new strings, which we call *eulerian*, have exactly $(n-1)$ matchings with the corresponding lagrangian strings, when they are generated by a winning deck.

Since our main goal is to study the winning decks, for the sake of simplicity in our presentation let us focus only on strings generated by winning decks, if not differently indicated.

It is immediate to note that, since in the eulerian strings every value k is preceded by the first $k-1$ values, we can indicate only with $[k]$, or, better, with k, the substring $123...k$. Consequently, we can represent

every eulerian string by means of a reduced string, which is formed by the cards which have given a matching, put in the same order in which they were stored, and, at the end, an "ace". Thus, the length of the reduced strings generated by winning decks is $n = m \cdot s$. In this case, if we substitute the final "ace" with the residual "two" of the winning deck, we form a new deck, i.e., a permutation of the original deck. In other words, the so modified winning reduced strings correspond to the reformed decks (or permutations) introduced by Guy and Nowakowski **[8,** §**E37]**, **[9]**, **[10]** in the game *Mousetrap*. We will call them *reduced (eulerian) string*.

In the above mentioned example, to the winning decks for $m = 7, s = 4$, we can associate the following eulerian strings

$$1234123451231123456712345123412345612312345612341212345$$
$$7123456123456712312123451234561234567123451123412311211 \quad ,$$

which generates the reduced eulerian string

$$45317546364276732567514312 12$$

and

$$1234561212312123456712311234123456712341234512345671234 5$$
$$1234561234123451234561231234561234567123451123412311211 \quad ,$$

which generates the reduced eulerian string

$$623273147457564563675143121 2 \ .$$

In the winning reduced strings the value 2 is always the final component. Consequently, the number of all the potential winning reduced strings is given by

$$S_{m \cdot s} = \frac{(n-1)!}{(s!)^{m-1} \cdot (s-1)!} \ . \tag{4.2}$$

If we had a bijective correspondence among the winning decks and the potential winning strings, we should know the number of winning decks and thus the winning probability, dividing $S_{m \cdot s}$ by the number of all the possible decks:

$$\frac{(n-1)!}{(s!)^{m-1} \cdot (s-1)!} \cdot \frac{(s!)^m}{n!} = \frac{s}{n} = \frac{1}{m} \ . \tag{4.3}$$

Unfortunately, we cannot have bijection. For the sake of simplicity, let us consider the case $m = 2 \ , \ s = 3$. Among the $\dfrac{6!}{(3!)^2} = 20$ decks, only four of them win. Here we show the winning decks and the associated reduced strings (or reformed decks):

the string 111222 is generated by the deck 111222;
the string 112122 is generated by the deck 112212;
the string 121122 is generated by the deck 122112;
the string 211122 is generated by the deck 221112.

The potential winning reduced strings are, in this case, $\dfrac{5!}{(3!) \cdot (2!)} = 10$:

$$\{221112\} \ ; \ \{212112\} \ ; \ \{211212\} \ ; \ \{211122\} \ ; \ \{122112\} \ ;$$
$$\{121212\} \ ; \ \{121122\} \ ; \ \{112212\} \ ; \ \{112122\} \ ; \ \{111222\} \ .$$

Actually, only the fourth, the seventh, the ninth and the tenth are generated by winning decks. However, even if we cannot have bijection between winning decks and potential winning reduced strings, we can use formula (4.3) as a rough upper bound for $P(C_{max})$.

This estimate can be highly improved, by means of Čebishev and Markov inequalities. This is the subject of a paper in preparation.

When we associate to a winning deck a reduced string we have a very deep information related to the fact that the procedure of string generation is (maintaining a physical language) reversible: knowing the generated string, we can rebuild the original deck. Considering example (4.1 **a)**) ($m = 7$, $s = 4$), let us consider a vector with 28 components. Let us put in the fourth component the first element of the string, i.e., the first stored card, which is clearly a "four". Then we will put in the $(4 + 5 =)$ ninth component the second stored card, i.e., a "five" and so on. When the counting arrives at 28, or, in general, at $m \cdot s$, we restart our counting from the first component, taking into account only the zero components, inserting the first 27 stored cards. The last card, i.e., a "two", will be put in correspondence with the last zero component. In this way we have rebuilt the original winning deck from the winning reduced string.

The "eulerian" approach can thus provide a very efficient method for the study of the winning decks, highly more efficient than the Monte Carlo simulations.

The technique, implemented in a computer program, rebuilds strings of continuing increasing length (up to the winning strings of length n, or n-strings), storing in data files only those ones so that the sub-decks, rebuilt from them, win playing $(HLM)^2N$, i.e., store all the cards but the final "two". The program, starting from a k-string, read in a data file, builds all the $(k + 1)$-strings, obtained adding at the beginning of the actual k-string all the allowed values from 1 to m; rebuilds the corresponding sub-decks; plays with the sub-decks. If a sub-deck sets aside all the cards, except for a "two" and generates the original string, the program stores the corresponding winning $(k + 1)$-string.

In order to save disk usage, the strings are stored as "characters" in the FORTRAN data files. Any idea regarding further memory saving improvements would be welcome.

More precisely, the algorithm is the following: starting from the last "two", we proceed backward, building all the sub-strings, of increasing length that can guarantee the storing of all the cards, apart from the last "two". Obviously, the last stored card can be only an "ace" or a "two"; similarly, the last but one can be only an "ace" or a "two": the drawing of a "three" as the last but one stored card is excluded by Remark (2.3). Continuing our reasoning, the last but two stored card can be only an "ace", a "two" or a "three", the last but three an "ace", a "two", a "three" or a "four" and so on, up to the last but $(m - 1)$ stored card, which cannot assume a value greater than $(m - 1)$. From the last but m stored card on, every card value is admitted.

Practically, let us recall that in the winning reduced n-strings the last position must be occupied by a card whose value is "two" and that, in order to have winning strings (since the strings of length $k \leq m$ (or k-strings), cannot be occupied by a card whose value is greater or equal to k), the position just before the last "two" can be occupied only by an "ace" or a "two"; thus we have only two winning final strings of length two: 12 and 22, which are respectively generated by the sub-decks 12 and 22.

The final strings of length three can be four: 112 ; 212 ; 122 ; 222. Clearly, the choice of these strings is related to s. If, for example, $s = 2$, the fourth string must be excluded, because it contains three identical cards.

Each one of these strings is in a one-to-one correspondence with a sub-deck generating it. In fact

from the string 112 we build the sub-deck 112, which generates the string 112 ;
from the string 212 we build the sub-deck 221, which generates the string 212 ;
from the string 122 we build the sub-deck 122, which generates the string 122 ;
from the string 222 we build the sub-deck 222, which generates the string 222 .

When we pass to the final strings of length four we have 12 possibilities:

$$1112\ ;\ 2112\ ;\ 3112\ ;\ 1212\ ;\ 2212\ ;\ 3212\ ;\ 1122\ ;\ 2122\ ;\ 3122\ ;\ 1222\ ;\ 2222\ ;\ 3222\ .$$

While we can associate to eight of them the corresponding generating winning deck, according to the following list:

$$\begin{aligned}
&\text{the sub}-\text{deck } 1112 \text{ generates the string } 1112\ ;\\
&\text{the sub}-\text{deck } 2211 \text{ generates the string } 2112\ ;\\
&\text{the sub}-\text{deck } 1221 \text{ generates the string } 1212\ ;\\
&\text{the sub}-\text{deck } 2132 \text{ generates the string } 3212\ ;\\
&\text{the sub}-\text{deck } 1122 \text{ generates the string } 1122\ ;\\
&\text{the sub}-\text{deck } 2212 \text{ generates the string } 2122\ ;\\
&\text{the sub}-\text{deck } 1222 \text{ generates the string } 1222\ ;\\
&\text{the sub}-\text{deck } 2222 \text{ generates the string } 2222\ ;
\end{aligned} \tag{4.4}$$

we realize that the strings 3112 ; 2212 ; 3122 ; 3222 have no corresponding winning deck. In fact, considering, for example, the string 3122, the deck generating it must have in the third position the card "three"; in the fourth position the card "ace" and, having no other components after, the second "ace" must be put in first position. Consequently, the card "two" must be put in the only place remained, that is in the second position. So the generating deck should be 1231. But it is evident that this deck, instead of the considered string, generates the loosing string formed only by an "ace", without any other coincidences.

Moreover, to the string 2212 corresponds the deck 1222, which generates the string 1222, which is still a winning string, but different from the original one. This last consideration shows that there is no bijective correspondence between decks and reduced strings: if every deck generates only one string, the reverse is in general not guaranteed: the same deck can be rebuilt from different strings!

In order to avoid these situations, the algorithm we have implemented contains a test where we check if the original string coincides with the reformed string obtained from the deck given back by the original string. Otherwise the string must be discarded.

Continuing the procedure, we select winning strings of continuing increasing length with the fundamental restriction that they must be generated by a deck, following the rules of $(HLM)^2N$.

By virtue of this technique we have been able to show that (SC) is true at least up to the case of French cards ($m = 13\ ,\ s = 4$), finding, in less than one second, four winning decks. The first winning deck of French cards found by the computer is the following:

$$\begin{array}{cccccccccccccc}
7 & 9 & 5 & 9 & 7 & 3 & 8 & 6 & 6 & 2 & 5 & 12 & 11 \\
4 & 12 & 9 & 7 & 7 & 10 & 2 & 4 & 5 & 3 & 11 & 13 & 2 \\
4 & 4 & 11 & 13 & 3 & 6 & 10 & 10 & 10 & 3 & 5 & 12 & 2 \\
1 & 1 & 1 & 1 & 12 & 9 & 11 & 13 & 8 & 8 & 6 & 8 & 13
\end{array}\ ,$$

while the first deck of Italian cards ($m = 10\ ,\ s = 4$) is

$$\begin{array}{cccccccccc}
6 & 8 & 9 & 7 & 5 & 5 & 3 & 6 & 6 & 10 \\
2 & 7 & 4 & 7 & 4 & 10 & 2 & 8 & 5 & 3 \\
9 & 2 & 4 & 4 & 3 & 6 & 10 & 7 & 10 & 3 \\
5 & 2 & 1 & 1 & 1 & 1 & 9 & 9 & 8 & 8
\end{array}\ .$$

The search for **at least** one winning deck is, in general, very fast. But, as shown in Table 2, we have, in many cases, found also the exact number of winning decks. Let us remark the fact that for the case $m = 10$, $s = 2$ (in comparison with un unsuccessful research of winning decks with Monte Carlo methods, after more than $6 \cdot 10^{11}$ simulations) we gained all the 656 winning decks, by virtue of the "eulerian" technique in less than one second.

Let us apply this method to prove the following

Proposition 4.1. *For $s = 2$, $m = 3, 4, 5$ there are no winning decks. For $s = 2$, $m = 6$ there exists only one winning deck.*

Proof Let us first consider strings with an arbitrary m and $s = 2$. Following the above described procedure, we must build all the winning strings of length, respectively, $3 \times 2 = 6$; $4 \times 2 = 8$; $5 \times 2 = 10$; $6 \times 2 = 12$, where, as already remarked, the last position must be occupied by a "two". According to the list (4.4) and recalling that $s = 2$, the 4-strings we are interested on are 2112 ; 1212 ; 3212 ; 1122. These 4-strings generate only the following 5-strings: 32112 ; 42112 ; 31212 ; 41212 ; 13212 ; 33212 ; 43212 ; 31122 ; 41122. Among them, only 42112 ; 31212 ; 13212 are generated by decks (respectively 21142 ; 21312 ; 12132). Continuing backward, we arrive at nine strings of length 6:

342112 ; 442112 ; 542112 ; 331212 ; 431212 ; 531212 ; 313212 ; 413212 ; 513212 ;

among them, only one (431212) is generated by a deck: 312421, which contains a "four". Thus, there are no winning 6-strings (and, consequently, winning decks) for $s = 2$, $m = 3$. Let us now build the four final 7-strings: 3431212 ; 4431212 ; 5431212 ; 6431212. Among them, only two are generated by decks:

3431212 is generated by 2133124;
5431212 is generated by 2421531.

Continuing: among the 9 strings of length 8, only four are generated by decks:

63431212 is generated by 33124621;
35431212 is generated by 31324215;
45431212 is generated by 53142421;
65431212 is generated by 21531624.

All of them contain cards whose value is greater than 4. Consequently, there are no winning decks for $m = 4$, $s = 2$.

The nine 9-strings generated by decks are:

563431212 is generated by 462153312;
663431212 is generated by 246216331;
435431212 is generated by 215431324;
345431212 is generated by 213531424;
545431212 is generated by 242155314;
845431212 is generated by 314242185;
365431212 is generated by 243215316;
665431212 is generated by 316246215;
765431212 is generated by 531624721.

In order to conclude the proof, let us now consider only strings where the cards assume at most value "six". Among all the 51 10-strings, only 17 are formed with cards whose value is at most "six". The strings generated by decks are 21. Among them, only 7 are formed with cards whose value is at most "six":

4563431212 is generated by 3124462153;
5563431212 is generated by 3312546215;
4663431212 is generated by 3314246216;
6345431212 is generated by 3142462135;
4365431212 is generated by 3164243215;
5365431212 is generated by 5316524321;
4665431212 is generated by 2154316246.

All of them contain at least one "six". Consequently, there are no winning decks for $m = 5$, $s = 2$. Finally, iterating the procedure only for cards whose value is at most "six", we arrive at 13 12-strings. Among them, only one, 534665431212, is generated by a deck: 316254632154. Then, for $m = 6$, $s = 2$, there is only one winning deck. □

In Table 2 we report the number of winning decks for $s = 2, 3, 4$.

Finally, recalling that in Remark (2.3) we have already shown that, for $s = 1$, it is not possible to reach C_{max}, we can, however, determine the best reachable score. Table 3 shows the best results obtained by virtue of a modified version of the computer program explained in this Section. The results we have achieved following this method coincide with the best scores obtained with Monte Carlo simulations when the number m is sufficiently small. For larger m, the simulations need too much time to reach the best score, while the "eulerian" method arrives at the correct answer very quickly.

5. Applications to the game *Mousetrap*

As already remarked in the Introduction, there are few results related to the game *Mousetrap*. In particular, there are no (even approximated) formulas giving the probability of winning decks. The technique introduced in the previous Section, adequately adapted to this game, allows us to obtain not a closed formula, but a sequence of values, giving the number $N_{max,m\cdot s}$ of winning decks and, consequently, the probability $P_{max,m\cdot s}$ for different values of m and s.

The main change consists in allowing the last card to assume whatever value, as allowed by the rules of this game.

Up to now, according to [**4**], [**15**], [**16**], [**17**], the sequence of values of P_{max} was obtained only for $s = 1$ and up to $m = n = 13$. In [**17**] this sequence cannot be read in A007709, but can be easily obtained from A007711, from the sequence of non-winning decks (or unreformed decks), because their number is, obviously, equal to $n! - N_{max,n}$.

According to Kok Seng Chua [**4**], this sequence has been obtained playing with all the $n! = m!$ decks, by means of a computer program generating all the permutations of a set of n elements.

Our new technique allows us to obtain the same results very quickly (my PC yielded the exact number of winning 13-decks in less than 2 hours, in comparison with one week job used by K.S. Chua [**4**]) and to extend the sequence, for $s = 1$, up to $m = 15$.

The new sequence of reformed decks (starting from $n = 1$), quoted in [**17**] as A007709, is thus

1 ; **1** ; **2** ; **6** ; **15** ; **84** ; **330** ; **1,812** ; **9,978** ; **65,503** ; **449,719** ; **3,674,670** ; **28,886,593** ;
266,242,729 ; 2,527,701,273

while the sequence of unreformed decks (i.e., the total number of non winning decks), quoted as A007711, is now

$$\mathbf{0}\ ;\ \mathbf{1}\ ;\ \mathbf{4}\ ;\ \mathbf{18}\ ;\ \mathbf{105}\ ;\ \mathbf{636}\ ;\ \mathbf{4,710}\ ;\ \mathbf{38,508}\ ;\ \mathbf{352,902}\ ;\ \mathbf{3,563,297}\ ;\ \mathbf{39,467,081}\ ;$$
$$\mathbf{475,326,930}\ ;\mathbf{6,198,134,207}\ ;\ 86,912,048,471\ ;\ 1,305,146,666,727$$

(the values in boldface were already quoted in [**17**] or in [**4**]).

We adapted the "eulerian" technique to the game *Modular Mousetrap*, too. Though experimentally the number of winning decks grows with m much faster than at *Mousetrap*, nevertheless the new technique has proved to be very powerful, for *Modular Mousetrap* too, as shown in Table 5.

Let us remark that, for example, in [**9**] the authors study *Modular Mousetrap* only in the cases $s = 1\ ,\ m \leq 5$, while in the case $s = 1\ ,\ m = 6$ they study only the decks where the first card is an "ace".

Furthermore, we have obtained a huge amount of results in the *"multisuit" Mousetrap* ($s > 1$), arriving, just as a test of the efficiency of the new technique, at $s = 2\ ,\ m = 8\ ; s = 3\ ,\ m = 6\ ; s = 4\ ,\ m = 5$ for *Mousetrap* and at $s = 1\ ,\ m = 13\ ;\ s = 2\ ,\ m = 7\ ; s = 3\ ,\ m = 5\ ; s = 4\ ,\ m = 4$ for *Modular Mousetrap*.

These results, shown in Table 5, can be extended to the cases $s > 4$ and, by means of parallel calculus, to higher values of m.

Remark 5.1. Let us denote with $P_{M,m\cdot s}(k)$ and $P_{MM,m\cdot s}(k)$ the probability of storing k cards, respectively at *Mousetrap* and *Modular Mousetrap*. As already observed in [**9**], at *Modular Mousetrap*, when $s = 1$ and m is prime, every deck which is not a *derangement* is a winning deck, because the cards have no possibilities to end in a loop. Consequently, while, if m is not prime, there is no *a priori* rule showing what is the winning probability, Table 5 shows that, when $s = 1$ and m is prime, it is very easy to know the **exact** winning probability:

$$P_{MM,m}(m) = 1 - P_{MM,m}(0) \qquad\qquad \forall m \quad prime\ .$$

Thus, knowing the sequence [**17**] A002467 of permutations with at least one fixed point, we immediately obtain the sequence of numbers of winning decks, for n prime:

224,837,335,816,336 for $n = m = 17$; 76,894,368,849,186,894 for $n = m = 19$ and so on.

For these cases the "eulerian" technique should have proved to be computationally too costly, for a single PC. Let us remark that, when n is prime, all the k-strings, with $k \leq n$, cannot end in any loop, i.e., are winning strings and must be stored. Consequently, in our rebuilding procedure, we must examine all the $n \cdot (n-1) \cdot (n-2) \cdot ... \cdot (n-k+1) = \binom{n}{k} k!$ k-substrings and, in particular, all the $n!$ strings of length n. Thus, playing *Modular Mousetrap*, when n is prime, our method coincides with Chua's technique, consisting in the analysis of all the $n!$ permutations.

Since, by Proposition (2.1), the probability of *derangement* for the games *Mousetrap* (M) and *Modular Mousetrap* (MM) is

$$P_{M,m}(0) = P_{MM,m}(0) = \sum_{k=0}^{m} \frac{(-1)^k}{k!} \tag{5.1}$$

and

$$\lim_{m\to\infty} P_{M,m}(0) = \lim_{m\to\infty} P_{MM,m}(0) = e^{-1} \sim 0.367879441\ ,$$

it follows that, at *Modular Mousetrap*, $\lim_{m\to\infty} P_{MM,m}(m) = 1 - \frac{1}{e} \sim 0.632120559$, if we consider only the sequence of prime numbers m (see Table 5).

For the other values of m, the winning probability seems to oscillate and tend to zero very slowly, when $m \to \infty$.

It is important to remark that, since our technique starts just from a permutation and tries to rebuild the deck from which the permutation is reformed, the "eulerian" technique can be easily adapted to check if any particular permutation is a reformed deck. In particular, we can very easily give, for every n, the deck producing as reformed permutation the identity $1\ 2\ \cdots\ n$, giving a positive answer to the original question by Cayley **[2]** ("investigate, whatever the number n of cards is, which permutations throw out the cards in the same order of their numbers").

Here we report the sequence of the requested decks up to $n = 13$, but it is a matter of seconds to find the answer for every value of n.

$1\ [Ca]\quad ;\quad 1\ 2\ [Ca]\quad ;\quad 1\ 3\ 2\ [Ca]\quad ;\quad 1\ 4\ 2\ 3\ [Ca]\quad ;\quad 1\ 3\ 2\ 5\ 4\ [Ca]\quad ;\quad 1\ 4\ 2\ 5\ 6\ 3\ [Ca]\quad ;$
$1\ 5\ 2\ 7\ 4\ 3\ 6\ [Ca]\quad ;\quad 1\ 6\ 2\ 4\ 5\ 3\ 7\ 8\ [Ca]\quad ;\quad 1\ 4\ 2\ 8\ 6\ 3\ 7\ 9\ 5\quad ;\quad 1\ 8\ 2\ 9\ 7\ 3\ 10\ 5\ 6\ 4\quad ;$
$1\ 10\ 2\ 9\ 6\ 3\ 5\ 8\ 7\ 4\ 11\quad ;\quad 1\ 6\ 2\ 7\ 5\ 3\ 11\ 12\ 8\ 4\ 9\ 10\quad ;\quad 1\ 8\ 2\ 5\ 10\ 3\ 12\ 11\ 9\ 4\ 7\ 6\ 13\quad .$

We have inserted the symbol [Ca] to indicate the permutations originally obtained by Cayley in **[2]**.

In the web site

http://www.dmmm.uniroma1.it/∼bersani/mousetrap.html

it is possible to read the decks up to $n = 100$ and it is possible to build new ones, by means of a specific FORTRAN file.

Remark 5.2. The new technique becomes computationally expensive when either m or s grows too much and we cannot achieve the number of winning decks for all the cases considered in Tables 4 and 5.

However, we have estimated, by means of Monte Carlo simulations, the winning probability for all the missing cases (up to $m = 13$, $s = 4$).

It is worthy to note that, playing *Modular Mousetrap*, when m is prime, we can only store $k \cdot m$ cards $(k = 0, 1, ..., s)$. In this case, we experimentally observe that

$$\lim_{m\to\infty} P_{MM,m}(m \cdot s) \ \sim\ 0.52 \quad \text{if} \quad s = 2$$

$$\lim_{m\to\infty} P_{MM,m}(m \cdot s) \ \sim\ 0.48 \quad \text{if} \quad s = 3$$

$$\lim_{m\to\infty} P_{MM,m}(m \cdot s) \ \sim\ 0.46 \quad \text{if} \quad s = 4\ .$$

The reason for these asymptotic values is, up to now, not clear.

On the other hand, when m is not prime, there are decks which store a number of cards strictly lying between zero and $m \cdot s$. In these cases, from Table 5 we can note that, for s fixed, the higher the number of divisors of m is, the lower the winning probability is. This is related to the fact that the deck has more chances to end in a loop than decks with less divisors.

Finally, the new technique has proved to be extremely efficient when applied to the study of reformed decks (or reformed permutations): as already recalled, when a deck wins at *Mousetrap*, *Modular Mousetrap* or $(HLM)^2N$, it generates a new deck, which corresponds to a reduced eulerian string or, equivalently, to a reformed deck. We can play again with this new deck, in order to check if it will win again.

Guy and Nowakowski first proposed the study of reformed decks in **[9]** and obtained only partial results, for *Mousetrap* and *Modular Mousetrap*.

K. S. Chua **[4]** achieved a substantial improvement only for the game *Mousetrap*. His results are quoted in **[17]** in the sequences A007711, A007712, A055459, A067950.

We have further improved their results, extending the sequences A007711, A007712, A055459, A067950 up to $m = 15$ for *Mousetrap*, but, first of all, obtaining very long sequences of reformed decks in the game *Modular Mousetrap* (for example, the 13-deck

6 2 5 11 1 8 13 12 7 9 10 3 4

is 51 times reformable) and lots of nontrivial cycles, i.e., sequences of reformed decks where, starting from a deck, after a sequence of reformed permutations we arrive again at the original deck; for example, the 17-deck

1 17 10 14 7 16 12 3 9 8 4 6 13 2 15 5 11

yields a 39554-cycle!

The reason why we can obtain very long sequences and cycles in *Modular Mousetrap* must be found in the fact that, as already remarked, in this game, when n is prime, we have always either a winning deck or a *derangement* and that the probability to find a winning deck is very high (see Table 5). Consequently, in this case it is very easy to obtain a reformed one from a deck.

The complete list of the most important results on reformed decks, for *Mousetrap*, *Modular Mousetrap* and $(HLM)^2N$ will be published in another paper **[1]**.

6. Conclusions and further developments

The "eulerian" technique here introduced has proved to be very powerful for the study of the games *He Loves Me, He loves Me Not, Mousetrap* and *Modular Mousetrap*. Clearly, it can give only the number of winning decks, without any possibility of reaching a closed formula. But the complexity of the game studied is so high making it very difficult to expect finding general closed formulas. In fact, as already remarked, only partial results have been obtained in the previous literature.

The contraindication of this "eulerian" method (which consists in rebuilding the winning decks starting from strings, of increasing length, formed by the last stored cards in the decks) is related to disk usage problems: in order to build all the strings of length $(k + 1)$, the program needs to store all the strings of length k.

Even if we should not be interested in the storage of all the winning decks, but only in their number, it is however necessary to store all the winning $(n - 1)$-strings. In the game *Modular Mousetrap*, in the case $m = 13\ ,\ s = 1$, the storage of all the winning n-strings needed a 112.5 GB memory, while the storage of all the winning $(n - 1)$-strings needed a 176.4 GB memory.

Moreover, in the case of French cards ($m = 13\ , s = 4$), considering the growth rate of the number of winning cards at $(HLM)^2N$ when m grows, for $s = 4$, we should expect, in the most cautious estimate, at least 10^{24} winning decks. Actually, a number absolutely unreasonable, for an actual PC.

Certainly, the usage of parallel computers or (as actually done playing the games in the most advanced cases) the storage of all the k-strings in several data subfiles, which can be processed separately, can help the search of all the winning decks for increasing values of m and/or s.

Anyway, the importance of the technique consists first in having shown that (SC) is true at least for $m = 13\ ,\ s = 4$ (but the test can be performed for much larger decks). The growth rate of the number of winning decks allows us to suppose that (SC) is true for every value of m and s, though the winning probability decreases with m. However this technique cannot give a definitive positive answer to (SC) for every value of m and s.

The results obtained here can be obviously extended, in particular by means of more powerful computers or by means of parallel calculus to which the technique is absolutely adaptable. Less disk usage expensive computer methods or other improvements could help too.

Acknowledgments I am deeply grateful to Prof. Gianpaolo Scalia Tomba, for his enlightening quotations of the appropriate literature at the beginning of my studies on this subject; to Francesco Scigliano and Joel Hazan, who helped me in the Monte Carlo simulations and to all the people who have provided me with even simple suggestions for improving the computer programs through which I have implemented the "eulerian" technique.

I will highly appreciate every information on improvements of the results reported in this paper and/or of the computer programs to obtain them.

REFERENCES

[1] A.M. Bersani, *Reformed permutations in Mousetrap and its generalizations*, preprint Me.Mo.Mat. n. 15/2005. http://www.dmmm.uniroma1.it/∼bersani/reformednew.pdf

[2] A. Cayley, *A problem in permutations*, Quart. J. Pure Appl. Math., vol. 1, 1857, p. 79.

[3] A. Cayley, *On the game of Mousetrap*, Quart. J. Pure Appl. Math., vol. 15, 1878, pp. 8 – 10.

[4] K.S. Chua, *private communication.*

[5] P.G. Doyle, C.M. Grinstead, J. Laurie Snell *Frustration Solitaire*, UMAP Journal, vol. 16, n. 2, 1995, pp. 137 – 145.

[6] W. Feller, *An introduction to Probability Theory and its applications*, Wiley and Sons, New York, 1957.

[7] M. Fréchet, *Les probabilités associées a un système d'événements compatibles et dépendants - Seconde partie: cas particuliers et applications*, Hermann and C., Paris, 1943.

[8] R.K. Guy, *Mousetrap*, § E37 in *Unsolved Problems in Number Theory*, second edition, Springer-Verlag, New York, 1994, pp. 237 – 238.

[9] R.K. Guy, R. Nowakowski, *Mousetrap*, Combinatorics, Paul Erdõs is Eighty, vol. 1, D. Miklós, V.T. Sós, T. Szonyi Eds., János Bolyai Mathematical Society, Budapest, 1993, pp. 193 - 206.

[10] R.K. Guy, R. Nowakowski, *Unsolved Problems - Mousetrap*, Amer. Math. Monthly, vol. 101, n. 10, 1994, pp. 1007 – 1008.

[11] R. Guy, R. Nowakowski, *Monthly Unsolved Problems, 1969 – 1995*, Amer. Math. Monthly, vol. 102, 1995, pp. 921 – 926.

[12] D.J. Mundfrom, *A problem in permutations: the game of Mousetrap*, European J. Combin., vol. 15, 1994, pp. 555 –560.

[13] A. Pompili, *Il metodo MONTE CARLO per l'analisi di un solitario*, http://xoomer.virgilio.it/vdepetr/Art/Text22.htm.

[14] J. Riordan, *An introduction to Combinatorial Analysis*, Princeton Univ. Press, Princeton, 1980.

[15] N.J.A. Sloane, *A Handbook of integer sequences*, Academic Press, San Diego, 1973.

[16] N.J.A. Sloane, S. Plouffe, *The encyclopedia of integer sequences*, Academic Press, San Diego, 1995.

[17] N.J.A. Sloane, *The On-Line Encyclopaedia of Integer Sequences*, http://www.research.att.com/∼njas/sequences/ .

[18] A. Steen, *Some formulae respecting the Game of Mousetrap*, Quart. J. Pure Appl. Math., vol. 15, 1878, pp. 230 – 241.

	$s=2$	$s=3$	$s=4$
$m=2$	6 4/4 (3/3) $-$ SC immediate	20 7/7 (5/5) $-$ SC immediate	70 10/10 (7/7) $-$ SC immediate
$m=3$	90 9/10 (5/5) $-$ WC immediate	1680 16/16 (8/8) $-$ SC immediate	34650 22/22 (11/11) $-$ SC immediate
$m=4$	2520 17/18 (7/7) $-$ WC immediate	369600 28/28 (11/11) $-$ SC immediate	63063000 38/38 (15/15) $-$ SC immediate
$m=5$	113400 27/28 (9/9) $-$ WC immediate	168168000 43/43 (14/14) $-$ SC $355,932$	$\sim 3.06\cdot 10^{11}$ 58/58 (19/19) $-$ SC $14,461,409$
$m=6$	7484400 40/40 (11/11) $-$ SC $4,530,195$	$\sim 1.37\cdot 10^{11}$ 61/61 (17/17) $-$ SC $123,289,316$	$\sim 3.25\cdot 10^{15}$ 82/82 (23/23) $-$ SC $314,429,118$
$m=7$	681080400 54/54 (13/13) $-$ SC $62,241,794$	$\sim 1.83\cdot 10^{14}$ 82/82 (20/20) $-$ SC $7,332,146,168$	$\sim 6.65\cdot 10^{19}$ 110/110 (27/27) $-$ SC $63,227,020,954$
$m=8$	$\sim 8.17\cdot 10^{10}$ 70/70 (15/15) $-$ SC $4,152,727,936$	$\sim 3.69\cdot 10^{17}$ 106/106 (23/23) $-$ SC $\sim 147,000,000,000$	$\sim 2.39\cdot 10^{24}$ 139/142 (31/31) $-$ WC $\sim 264,386,000,000$
$m=9$	$\sim 1.25\cdot 10^{13}$ 88/88 (17/17) $-$ SC $\sim 90,000,000,000$	$\sim 1.08\cdot 10^{21}$ 131/133 (26/26) $-$ WC $\sim 255,000,000,000$	$\sim 1.41\cdot 10^{29}$ 172/178 (34/35) $> 207,000,000,000$
$m=10$	$\sim 2.38\cdot 10^{15}$ 106/108 (19/19) $-$ WC $> 600,000,000,000$	$\sim 4.39\cdot 10^{24}$ 154/163 (28/29) $> 81,000,000,000$	$\sim 1.29\cdot 10^{34}$ 205/218 (37/39) $> 217,000,000,000$
$m=11$	$\sim 5.49\cdot 10^{17}$ 128/130 (21/21) $-$ WC $92,800,000,000$	$\sim 2.39\cdot 10^{28}$ 184/196 (31/32) $36,700,000,000$	$\sim 1.75\cdot 10^{39}$ 224/262 (39/43) $2,000,000,000$
$m=12$	$\sim 1.51\cdot 10^{20}$ 139/154 (22/23) $12,000,000,000$	$\sim 1.71\cdot 10^{32}$ 204/232 (33/35) $2,000,000,000$	$\sim 3.40\cdot 10^{44}$ 273/310 (43/47) $1,000,000,000$
$m=13$	$\sim 4.92\cdot 10^{22}$ 158/180 (22/25) $2,000,000,000$	$\sim 1.56\cdot 10^{36}$ 235/271 (34/38) $5,000,000,000$	$\sim 9.20\cdot 10^{49}$ 305/362 (45/51) $4,000,000,000$

Table 1 - BEST SCORES IN $(HLM)^2N$ OBTAINED WITH MONTE CARLO METHODS

In each box of this table we report the number $N_{m\cdot s}$ of different decks; the ratio between the best score and C_{max}; the ratio between the best number of stored cards and the number predicted by (WC); the number of simulations performed before achieving the first winning deck or performed without obtaining any winning deck. The number of simulations is given by the sum of the trials done by F. Scigliano and by myself, while I have no information about the number of simulations done by A. Pompili. The symbols SC and WC indicate respectively if we proved the strong or the weak conjecture.

	$s=2$	$s=3$	$s=4$
$m=2$	$3/6$; $P_{max}=0.5$	$4/20$; $P_{max}=0.2$	$15/70$; $P_{max}\sim 0.21$
$m=3$	$0/90$; $P_{max}=0$	$4/1680$; $P_{max}\sim 0.0024$	$5/34650$; $P_{max}\sim 0.00014$
$m=4$	$0/2520$; $P_{max}=0$	$9/369,600$ $P_{max}\sim 0.000024$	$229/63,063,000$ $P_{max}\sim 0.0000036$
$m=5$	$0/113400$; $P_{max}=0$	$63/168,168,000$ $P_{max}\sim 0.000000375$	$10568/3.06\cdot 10^{11}$ $P_{max}\sim 0.000000035$
$m=6$	$1/7,484,400$ $P_{max}\sim 1.34\cdot 10^{-7}$	$1177/1.37\cdot 10^{11}$ $P_{max}\sim 0.000000009$	$1,212,483/3.25\cdot 10^{15}$ $P_{max}\sim 3.73\cdot 10^{-10}$
$m=7$	$7/681,080,400$ $P_{max}\sim 1.00\cdot 10^{-8}$	$36144/1.83\cdot 10^{14}$ $P_{max}\sim 1.98\cdot 10^{-10}$	$411,488,689/6.65\cdot 10^{19}$ $P_{max}\sim 6.19\cdot 10^{-12}$
$m=8$	$8/8.17\cdot 10^{10}$ $P_{max}\sim 9.79\cdot 10^{-11}$	$1,677,968/3.69\cdot 10^{17}$ $P_{max}\sim 4.54\cdot 10^{-12}$	
$m=9$	$105/1.25\cdot 10^{13}$ $P_{max}\sim 8.40\cdot 10^{-12}$	$127,255,522/1.08\cdot 10^{21}$ $P_{max}\sim 1.18\cdot 10^{-13}$	
$m=10$	$656/2.38\cdot 10^{15}$ $P_{max}\sim 2.76\cdot 10^{-13}$		
$m=11$	$6745/5.49\cdot 10^{17}$ $P_{max}\sim 1.23\cdot 10^{-14}$		
$m=12$	$76823/1.51\cdot 10^{20}$ $P_{max}\sim 5.07\cdot 10^{-16}$		
$m=13$	$986,994/4.92\cdot 10^{22}$ $P_{max}\sim 2.00\cdot 10^{-17}$		
$m=14$	$17,175,636/1.86\cdot 10^{25}$ $P_{max}\sim 9.23\cdot 10^{-19}$		
$m=15$	$320,152,788/8.09\cdot 10^{27}$ $P_{max}\sim 3.96\cdot 10^{-20}$		

Table 2 - WINNING DECKS AT *HE LOVES ME HE LOVES ME NOT*

In each box we report the ratio between the number of winning decks and the total number of decks and the winning probability $P_{max}=P(C_{max})$.

	$s=1$	winning deck(s)
$m=2$	1/1 (1/1)	1 , 2
$m=3$	3/4 (1/2 *and* 2/2)	three decks
$m=4$	6/8 (3/3)	2 , 1 , 3 , 4
$m=5$	9/13 (3/4)	2 , 5 , 1 , 4 , 3
$m=6$	14/19 (4/5)	6 , 1 , 4 , 3 , 5 , 2
$m=7$	18/26 (4/6)	3 , 7 , 1 , 5 , 2 , 6 , 4
$m=8$	25/34 (5/7)	8 , 1 , 5 , 2 , 6 , 4 , 7 , 3
$m=9$	31/43 (7/8)	4 , 1 , 2 , 6 , 9 , 7 , 3 , 8 , 5
$m=10$	39/53 (6/9)	10 , 1 , 6 , 2 , 7 , 3 , 8 , 5 , 9 , 4
$m=11$	47/64 (8/10 *and* 9/10)	six decks
$m=12$	56/76 (7/11 *and* 10/11)	three decks
$m=13$	67/89 (11/12)	two decks
$m=14$	79/103 (12/13)	two decks
$m=15$	93/118 (13/14)	two decks
$m=16$	108/134 (14/15)	two decks
.....		

Table 3

In this table we report the ratio between the best score at $(HLM)^2N$ with one suit and C_{max} and the ratio between the number of stored cards and the number of cards satisfying (WC). In some cases it is possible to obtain the same best score with a different number of cards. When there is only one winning deck, we report it in the third column.

	$s=1$	$s=2$	$s=3$	$s=4$
$m=2$	$1/2\ ;\ P=0.5$ $[\mathbf{G-N}]$	$3/6\ ;\ P=0.5$	$4/20\ ;\ P=0.2$	$15/70\ ;\ P\sim 0.21$
$m=3$	$2/6\ ;\ P\sim 0.33$ $[\mathbf{G-N}]$	$12/90\ ;\ P\sim 0.13$	$90/1680\ ;\ P\sim 0.054$	$675/34650\ ;\ P\sim 0.019$
$m=4$	$6/24\ ;\ P=0.25$ $[\mathbf{G-N}]$	$147/2520\ ;\ P\sim 0.058$	$5232/369,600$ $P\sim 0.014$	$210,069/63,063,000$ $P\sim 0.0033$
$m=5$	$15/120$ $P=0.125$ $[\mathbf{G-N}]$	$2322/113,400$ $P\sim 0.020$	$476,042/168,168,000$ $P\sim 0.0028$	$119,375,881/3.06\cdot 10^{11}$ $P\sim 0.00039$
$m=6$	$84/720$ $P\sim 0.12$ $[\mathbf{G-N}]$	$71629/7,484,400$ $P\sim 0.0096$	$111,660,352/1.37\cdot 10^{11}$ $P\sim 0.00081$	$P\sim 0.000070$ $[\mathbf{MC}]$
$m=7$	$330/5040$ $P\sim 0.065$ $[\mathbf{G-N}]$	$2,214,258/681,080,400$ $P\sim 0.0033$	$P\sim 0.00016$ $[\mathbf{MC}]$	$P\sim 0.0000081$ $[\mathbf{MC}]$
$m=8$	$1812/40320$ $P\sim 0.045$ $[\mathbf{G-N}]$	$118,228,868/8.17\cdot 10^{10}$ $P\sim 0.0014$	$P\sim 0.000046$ $[\mathbf{MC}]$	$P\sim 0.0000015$ $[\mathbf{MC}]$
$m=9$	$9978/362,880$ $P\sim 0.027$ $[\mathbf{G-N}]$	$P\sim 0.00053$ $[\mathbf{MC}]$	$P\sim 0.000010$ $[\mathbf{MC}]$	$P\sim 0.0000002$ $[\mathbf{MC}]$
$m=10$	$65503/3,628,800$ $P\sim 0.018$ $[\mathbf{C-S}]$	$P\sim 0.00022$ $[\mathbf{MC}]$	$P\sim 0.0000026$ $[\mathbf{MC}]$	$P\sim 0.00000003$ $[\mathbf{MC}]$
$m=11$	$449,719/39,916,800$ $P\sim 0.011$ $[\mathbf{C-S}]$	$P\sim 0.000083$ $[\mathbf{MC}]$	$P\sim 0.0000006$ $[\mathbf{MC}]$	$2\cdot 10^{-9}<P<6\cdot 10^{-9}$ $[\mathbf{MC}]$
$m=12$	$3,674,670/479,001,600$ $P\sim 0.0077$ $[\mathbf{C-S}]$	$P\sim 0.000036$ $[\mathbf{MC}]$	$P\sim 0.000000084$ $[\mathbf{MC}]$	still under investigation
$m=13$	$28,886,593/6,227,020,800$ $P\sim 0.0046$ $[\mathbf{C-S}]$	$P\sim 0.000013$ $[\mathbf{MC}]$	$3\cdot 10^{-8}<P<5\cdot 10^{-8}$ $[\mathbf{MC}]$	still under investigation
$m=14$	$266,242,729/8.72\cdot 10^{10}$ $P\sim 0.0031$			
$m=15$	$2,527,701,273/1.31\cdot 10^{12}$ $P\sim 0.0019$			

Table 4 - WINNING DECKS AT *MOUSETRAP*

In each box we report the ratio between the number of winning decks and $N_{m\cdot s}$ and the winning probability $P := P_{M,m\cdot s}(m\cdot s)$. We indicate with **[G-N]** and with **[C-S]** the results already quoted respectively in **[9]** and in **[4]**, **[17]**. We indicate with **[MC]** the estimates obtained by means of Monte Carlo simulations.

	$s=1$	$s=2$	$s=3$	$s=4$
$m=2$	$1/2$; $P=0.5$ **[G-N]**	$5/6$; $P\sim 0.83$	$19/20$; $P=0.95$	$69/70$; $P\sim 0.986$
$m=3$	$4/6$; $P\sim 0.67$ **[G-N]**	$60/90$; $P\sim 0.67$	$1081/1680$; $P\sim 0.64$	$22898/34650$ $P\sim 0.66$
$m=4$	$9/24$; $P=0.375$ **[G-N]**	$1182/2520$; $P\sim 0.47$	$173,053/369,600$ $P\sim 0.47$	$29,642,185/63,063,000$ $P\sim 0.47$
$m=5$	$76/120$; $P\sim 0.633$ **[G-N]**	$63063/113,400$ $P\sim 0.56$	$86,636,303/168,168,000$ $P\sim 0.52$	$P\sim 0.49$ **[MC]**
$m=6$	$190/720$; $P\sim 0.26$	$1,797,350/7,484,400$ $P\sim 0.24$	$P\sim 0.23$ **[MC]**	$P\sim 0.22$ **[MC]**
$m=7$	$3186/5040$; $P\sim 0.632143$	$364,572,156/681,080,400$ $P\sim 0.54$	$P\sim 0.49$ **[MC]**	$P\sim 0.46$ **[MC]**
$m=8$	$11351/40320$ $P\sim 0.28$	$P\sim 0.24$ **[MC]**	$P\sim 0.22$ **[MC]**	$P\sim 0.21$ **[MC]**
$m=9$	$132,684/362,880$ $P\sim 0.37$	$P\sim 0.31$ **[MC]**	$P\sim 0.28$ **[MC]**	$P\sim 0.27$ **[MC]**
$m=10$	$884,371/3,628,800$ $P\sim 0.24$	$P\sim 0.20$ **[MC]**	$P\sim 0.18$ **[MC]**	$P\sim 0.18$ **[MC]**
$m=11$	$25,232,230/39,916,800$ $P\sim 0.632120561$	$P\sim 0.53$ **[MC]**	$P\sim 0.48$ **[MC]**	$P\sim 0.45$ **[MC]**
$m=12$	$50,436,488/479,001,600$ $P\sim 0.11$	$P\sim 0.085$ **[MC]**	$P\sim 0.077$ **[MC]**	$P\sim 0.073$ **[MC]**
$m=13$	$3,936,227,868/6,227,020,800$ $P\sim 0.632120559$ [**A002467**]	$P\sim 0.53$ **[MC]**	$P\sim 0.48$ **[MC]**	$P\sim 0.45$ **[MC]**

Table 5 - WINNING DECKS AT *MODULAR MOUSETRAP*

In each box we report the ratio between the number of winning decks and $N_{m\cdot s}$ and the winning probability $P := P_{MM,m\cdot s}(m\cdot s)$. We indicate with **[G-N]** the results already quoted in [**9**].

The result corresponding to $m=13$, $s=1$ can be also obtained subtracting the total number of *derangements* to the total number of decks, $n!=m!$ (because m is prime). We indicate it with [**A002467**]. We indicate with **[MC]** the estimates obtained by means of Monte Carlo simulations.