Reformed permutations in *Mousetrap* and its generalizations

Alberto M. Bersani [a]

[a] *Dipartimento di Metodi e Modelli Matematici, Università "La Sapienza" di Roma, Via A. Scarpa 16, 00161 Roma, Italy*

Abstract

We study a card game called *Mousetrap*, together with its generalization *He Loves Me, He Loves Me Not*. We first present some results for the latter game, based, on one hand, on theoretical considerations and, on the other one, on Monte Carlo simulations. Furthermore, we introduce a new combinatorial algorithm, which allows us to obtain the best result at least for French card decks (52 cards with 4 suits). We then apply the new algorithm to the study of *Mousetrap* and *Modular Mousetrap*, improving recent results. Finally, by means of our algorithm, we study the reformed permutations in *Mousetrap*, *Modular Mousetrap* and *He Loves Me, He Loves Me Not*, attaining new results which give some answers to several questions posed by Cayley and by Guy and Nowakowski in their papers.

Key words: Combinatorial Analysis, derangement, Mousetrap, reformed permutations, Monte Carlo simulations, Computational Combinatorics, Discrete Dynamical Systems, Deterministic solitaires, Combinatorial Game Theory
1991 MSC: 05A05, 05A10, 05A15, 60C05, 65C50

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

Email address: `e-mail: bersani@dmmm.uniroma1.it` (Alberto M. Bersani).
URL: `http://www.dmmm.uniroma1.it/ bersani/mousetrap.html` (Alberto M. Bersani).

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], [19], [8], [10], [12], [9], [11], [18]) have studied the problem, arriving, only recently [9], [12], [18], 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.

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 count up to m without any matching, the last m counted cards are "burned", i.e., definitively discarded and you begin the count afresh, counting "one", "two", "three" etc. with the successive 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 computing framework.

In their papers devoted to *Mousetrap*, Guy and Nowakowski proposed to study the so-called *reformed decks (or permutations)*: "consider a permutation for which every number is set aside. The list of numbers in the order that they were set aside is another permutation. Any permutation obtained in this way we call a *reformed* permutation. Characterize the reformed permutations."

The aim of this paper is to apply the new technique to the analysis of reformed decks, in the three games and to show new results which can answer some open questions proposed by Cayley and by Guy and Nowakowski.

The paper is divided into seven 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 $s = 2\ ,\ m = 16\ ;\ s = 3\ ,\ m = 10\ ;\ s = 4\ ,\ m = 7$.

In Section 5, adapting the new technique to the games *Mousetrap* and *Modular Mousetrap*, we extend the results attained in [9] up to $m = 16\ ,\ s = 1$ 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 $s = 1\ ,\ m = 16\ ;\ s = 2\ ,\ m = 9\ ;\ s = 3\ ,\ m = 6\ ;\ s = 4\ ,\ m = 5$ for *Mousetrap*; $s = 1\ ,\ m = 13\ ;\ s = 2\ ,\ m = 7\ ;\ s = 3\ ,\ m = 5\ ;\ s = 4\ ,\ m = 4$ for *Modular Mousetrap*.

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 6, applying the new technique to the study of reformed decks in the three games, we obtain many (even unexpected and curious) results. In particular, we produce the first 5-reformed deck, for *Mousetrap* ($m = 16, s = 1$). Moreover we discuss the existence of k-reformed decks, with $k > 5$, by means of probabilistic considerations.

In Section 7 we give a short review of open problems and perspectives.

The twenty tables, quoted in this paper, containing all the results here described, can be found, as online supplementary material, on the web page

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

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

There are few results on *Mousetrap*, obtained, in particular, by Steen [19], already in 1878 and, much more recently, by Guy and Nowakowski [9], Mundfrom [12] and Spivey [18].

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 $n \leq 8$:

$$1\quad ;\quad 1\ 2\quad ;\quad 1\ 3\ 2\quad ;\quad 1\ 4\ 2\ 3\quad ;\quad 1\ 3\ 2\ 5\ 4\quad ;$$

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 [19], already in 1878 and, much more recently, Guy and Nowakowski [9], Mundfrom [12] and Spivey [18], 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:

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

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

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 $0 \leq i \leq n$

$$a_{n,0} = a_{n+1,n+1}\ ,\ a_{n,0} = na_{n-1,0} + (-1)^n\ ,\ a_{n,i+1} = \sum_{k=0}^{i}(-1)^k \binom{i}{k}(n-1-k)! \quad (3)$$

$$b_{n,i} = a_{n-1,i-1} = a_{n-2,i-2} - a_{n-3,i-2} \quad , \quad b_{n,0} = a_{n,1} - b_n = a_{n,1} - a_{n-1} = a_{n-1,0} \quad (4)$$

$$a_n = n\ a_{n-1} + (-1)^{n-1} \quad , \quad b_n = a_{n-1} \quad (5)$$

$$c_{n,i} = \left[\sum_{k=1}^{i-3}(-1)^{k+i-1}\ \frac{k(k+3)}{2}\ (n-i+k-1)!\right] - (i-1)(n-3)! + (n-2)!\ . \quad (6)$$

Guy and Nowakowski [9] and Mundfrom [12] showed separately that Steen's formula (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:

$$c_{n,n-1} = \sum_{k=0}^{n-3}(-1)^k \binom{n-3}{k}\ (n-k-2)! \quad (7)$$

$$c_{n,n} = (n-2)! + \left[\sum_{k=0}^{n-5}(-1)^{k+1}\left(\binom{n-4}{k} + \binom{n-3}{k+1}\right)\ (n-k-3)!\right] + 2(-1)^{n-3} \quad (8)$$

$$c_n = (n-2)(n-2)! - \left[\left[\frac{1}{e}((n-1)! - (n-2)! - 2(n-3)!\right]\right]\ , \quad (9)$$

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

Spivey [18] approaches the game of *Mousetrap* using staircase rook polynomials ([14], Ch. 7, pp. 163–194) and determines the rook polynomial for the number of permutations in which card j is the only card removed and for the number of permutations in which card j followed by card k are the first two cards removed.

Setting $M_{n,j}$ as the number of decks in which card j is the only card removed, he shows that if $n \geq 4$

$$M_{n,2} = a_{n-1,0} - a_{n-2,0} - 2a_{n-3,0}\ .$$

Steen [19], Guy and Nowakowski [9] and Mundfrom [12] elaborated some tables related to formulas (1) - (9). The sequences there reproduced are quoted by Sloane [15], [16], [17] in the following way:

$\{a_n\}_{n\in I\!N}$ ([19]): [15] N1423, [16] M3507, [17] A002467;

$\{a_{n,0}\}_{n\in I\!N}$ ([19]): [15] N0766, [16] M1937, [17] A000166;

$\{a_{n,2}\}_{n\geq 2}$ ([19]): [15] N1436, [16] M3545, [17] A001563;

$\{c_n\}_{n\geq 2}$ ([9], [12], [19]): [15] N1186, [16] M2945, [17] A002468;

$\{c_{n,0}\}_{n\geq 2}$ ([12], [19]): [15] N1635, [16] M3962, [17] A002469;

$\{c_{n,3}\}_{n\geq 3}$ ([12], [19]): [17] A018931;

$\{c_{n,4}\}_{n\geq 4}$ ([12], [19]): [17] A018932;

$\{c_{n,5}\}_{n\geq 5}$ ([12], [19]): [17] A018933;

$\{c_{2,1}\} \cup \{c_{n,n}\}_{n\geq 3}$ ([12], [19]): [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):

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

$$P_{M,m}(0) = P_{MM,m}(0) = \sum_{k=0}^{m} \frac{(-1)^k}{k!} \; . \tag{10}$$

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, 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, 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 $m = 16$, $s = 1$ and to *"multisuit" Mousetrap*.

Finally, Guy and Nowakowski [9] yielded some results for the game *Modular Mousetrap*.

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 theorem, 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*.

Theorem 2.1 In $(HLM)^2N$, for every s, m we can store at most $ms-1$ cards and the score cannot exceed

$$C_{max} := \frac{s}{2}[m(m+1)] - 2 \ . \tag{11}$$

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 (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 cards but a "two" and, consequently, we can obtain C_{max}.

We can state the following two conjectures:

Strong Conjecture (SC) In $(HLM)^2N$, 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) In $(HLM)^2N$, for every $s \geq 2$; $m \geq 2$, there exists at least one deck from which we store $sm - 1 = n - 1$ cards.

Remark 2.1 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 2.2 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 2.3 For $s = 2$ there exist cases for which it is not possible to obtain the best score given by (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 $s = 2$, $6 \le m \le 13$ we obtained the best score, given by (11).

3 Monte Carlo simulations

In order to obtain at least experimental answers to (SC) and (WC) for several values of m and $s \le 4$, 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 $n_c \le m$ 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

$$N_{m \cdot s} = \frac{(m \cdot s)!}{(s!)^m} \ . \tag{12}$$

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), 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 (11). This situation could have been *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 backward 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.

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, playing the game we generate, from every deck, a 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.

We can also consider another string, formed by the cards which have given a matching, put in the same order in which they were stored. The length of the strings generated by winning decks is $n = m \cdot s$, with the residual "two" put in the last position. This is a new deck, i.e., a permutation of the original deck. In other words, the so modified winning strings correspond to the reformed decks (or permutations) introduced by Guy and Nowakowski ([9], §E37), [10],

[11].

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:

$$\begin{array}{ccccccc} 4 & 3 & 1 & 4 & 7 & 7 & 2 \\ 6 & 5 & 4 & 7 & 3 & 1 & 3 \\ 3 & 6 & 2 & 2 & 1 & 7 & 6 \\ 6 & 5 & 1 & 5 & 4 & 5 & 2 \end{array} \quad (\mathbf{a}) \qquad \begin{array}{ccccccc} 4 & 3 & 7 & 3 & 1 & 6 & 7 \\ 2 & 7 & 5 & 3 & 5 & 2 & 2 \\ 4 & 4 & 6 & 2 & 1 & 7 & 5 \\ 6 & 3 & 1 & 6 & 5 & 1 & 4 \end{array} \quad (\mathbf{b}) \tag{13}$$

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

$$4531754636427673256751431212$$

and

$$6232731474575645636751431212 \ .$$

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

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

$$S_{m \cdot s} = \frac{(n-1)!}{(s!)^{m-1} \cdot (s-1)!} \ . \tag{14}$$

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:

$$\frac{(n-1)!}{(s!)^{m-1} \cdot (s-1)!} \cdot \frac{(s!)^m}{n!} = \frac{s}{n} = \frac{1}{m} \ . \tag{15}$$

Unfortunately, we cannot have bijection. For the sake of simplicity, let us consider the case $m = 2$, $s = 3$. Among the $\frac{6!}{(3!)^2} = 20$ decks, only four of them win. Here we show the winning decks and the associated 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, $\frac{5!}{(3!)\cdot(2!)} = 10$:

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

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 (15) 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 string we have a very deep information related to the fact that the procedure of string generation is (using a physical language) reversible: knowing the generated string, we can rebuild the original deck. Considering example (13 **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 backward 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.

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 (2.1). 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:

$$
\begin{aligned}
&\text{the sub} - \text{deck } 1112 \text{ generates the string } 1112 \ ;\\
&\text{the sub} - \text{deck } 2211 \text{ generates the string } 2112 \ ;\\
&\text{the sub} - \text{deck } 1221 \text{ generates the string } 1212 \ ;\\
&\text{the sub} - \text{deck } 2132 \text{ generates the string } 3212 \ ;\\
&\text{the sub} - \text{deck } 1122 \text{ generates the string } 1122 \ ;\\
&\text{the sub} - \text{deck } 2212 \text{ generates the string } 2122 \ ;\\
&\text{the sub} - \text{deck } 1222 \text{ generates the string } 1222 \ ;\\
&\text{ the sub-deck } 2222 \text{ generates the string } 2222 \ ;
\end{aligned}
\tag{16}
$$

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 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 $(HLM)^2N$.

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.

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:

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

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

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

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 backward technique, in less than one second.

Let us apply this method to prove the following

Theorem 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 (16) 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 (2.1) 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 backward 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 algorithm 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 can be read in A007709 and can easily produce the sequence A007711 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 25 minutes, in comparison with one week job used by K.S. Chua [4]) and to extend the sequence, for $s = 1$, up to $m = 16$.

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

$$\mathbf{1}\ ;\ \mathbf{1}\ ;\ \mathbf{2}\ ;\ \mathbf{6}\ ;\ \mathbf{15}\ ;\ \mathbf{84}\ ;\ \mathbf{330}\ ;\ \mathbf{1,812}\ ;\ \mathbf{9,978}\ ;\ \mathbf{65,503}\ ;\ \mathbf{449,719}\ ;\ \mathbf{3,674,670}\ ;$$

$$28,886,593\ ;\ 266,242,729\ ;\ 2,527,701,273\ ;\ 25,749,021,720$$

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

$$\mathbf{0}\ ;\ \mathbf{1}\ ;\ \mathbf{4}\ ;\ \mathbf{18}\ ;\ \mathbf{105}\ ;\ \mathbf{636}\ ;\ \mathbf{4,710}\ ;\ \mathbf{38,508}\ ;\ \mathbf{352,902}\ ;\ \mathbf{3,563,297}\ ;$$

$$\mathbf{39,467,081}\ ;\ \mathbf{475,326,930}\ ;\mathbf{6,198,134,207}\ ;\ 86,912,048,471\ ;$$
$$1,305,146,666,727\ ;\ 20,897,040,866,280$$

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

We adapted the backward 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, substantially improving the results obtained in [9].

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 $s = 2$, $m = 9$; $s = 3$, $m = 6$; $s = 4$, $m = 5$ 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 Tables 4 and 5, can be extended to the cases $s > 4$ and, by means of parallel computing, 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:

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

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 backward 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 $n\cdot(n-1)\cdot(n-2)\cdot\ldots\cdot(n-k+1) = \binom{n}{k} k!$ 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 Lemma (2.1), the probability of *derangement* for the games *Mousetrap* (M) and *Modular Mousetrap* (MM) is

$$P_{M,m}(0) = P_{MM,m}(0) = \sum_{k=0}^{m} \frac{(-1)^k}{k!} \tag{17}$$

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 backward 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.

$$1\ [Ca] \quad ; \quad 1\ 2\ [Ca] \quad ; \quad 1\ 3\ 2\ [Ca] \quad ; \quad 1\ 4\ 2\ 3\ [Ca] \quad ; \quad 1\ 3\ 2\ 5\ 4\ [Ca] \quad ;$$

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

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 multisuit *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.

6 Searching for reformed decks

Thanks to its efficiency, the new technique has proved to be extremely useful 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 is called reformed deck. We can play again with this new deck in order to check if it will win again.

When we can repeat this operation k times, we will define *k-times reformable deck* the original deck and *k-reformed deck* the permutation obtained in the k-th reformation.

The reformation mechanism can produce peculiar situations like cycles.

A cycle is a sequence of reformed decks where one reformation coincides with one of the previous reformations (not necessarily the original deck). We classify the cycles more clearly in the second part of this Section, devoted to *Modular Mousetrap*.

Guy and Nowakowski [9] first proposed the study of reformed decks posing the following questions:

3) characterize the reformed permutations;

4) for a given n, what is the longest sequence of reformed permutations?

5) are there sequences of arbitrary length? are there any non trivial cycles, i.e., cycles other than

$$1 \;\rightarrow\; 1 \;\rightarrow\; 1\;... \qquad \textit{and} \qquad 1\;2 \;\rightarrow\; 1\;2 \;\rightarrow\; 1\;2\;...\;?$$

6) in *Modular Mousetrap* are there k-cycles for every k? what is the lowest value of n which yields a k-cycle?

Playing *Mousetrap*, they investigated the cases $s = 1$, $m \leq 9$. They achieved at most 3-reformed decks and did not find any non trivial cycle.

K. S. Chua [4] achieved a substantial improvement for the game *Mousetrap* finding for the first time a 4-reformed deck, for $s = 1$, $m = 11$. His results are quoted by Sloane [17] in the sequences A007711, A007712, A055459, A067950.

Here we further improve these results, extending the sequences A007711, A007712, A055459, A067950 up to $m = 16$ for *Mousetrap*, obtaining for the first time a 5-reformable deck:

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

for $m = 16$. With the new results, reported in Table 6, the first terms of the sequence of numbers of 4-times (but not 5-times) reformable permutations are

$$0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 2,\ 1,\ 1,\ 4,\ 14\ ,\ 57\ .$$

They are now classified on [17] as sequence A127966.

The discovery of the 5-reformed permutation represents an important step in the direction of a positive answer to question **5)**.

How many chances do we have to find 6-reformed decks?

The answer must be related to the probability $P_{M,m\cdot s}(m \cdot s)$, given by the ratio between the number of reformed decks and the number of all the permutations of the deck.

We can give a rough estimate of the number of at least k-reformed decks multiplying the number of at least $(k-1)$-times reformable permutations by $P_{M,m\cdot s}(m \cdot s)$. Obviously, the probability to obtain a reformed deck is, in general, not equal to the probability to obtain from all the reformed decks a 2-reformed one and, in general, from all the decks reformed at least k times, a $(k+1)$-reformed one. But experimentally all these probabilities are comparable. For example, indicating with $N_{\geq k,m\cdot s}$ the number of decks which are at least k-reformable, we have, in the case $s = 1\ ,\ m = 16$,

$$P_{M,16\cdot 1}(15) = \frac{N_{\geq 1,16\cdot 1}}{16!} \sim 0.00123\ ;\ \frac{N_{\geq 2,16\cdot 1}}{N_{\geq 1,16\cdot 1}} \sim 0.00124\ ;\ \frac{N_{\geq 3,16\cdot 1}}{N_{\geq 2,16\cdot 1}} \sim 0.00127\ ;$$

$$\frac{N_{\geq 4,16\cdot 1}}{N_{\geq 3,16\cdot 1}} \sim 0.00143\ ;\ \frac{N_{\geq 5,16\cdot 1}}{N_{\geq 4,16\cdot 1}} \sim 0.0172\ .$$

Multiplying the numbers $N_{\geq k,16\cdot 1}$ by $P_{M,16\cdot 1}(16) \sim 0.00123$, the estimates for the number of k-reformed decks (apart from the very peculiar case of $k = 5$) are very close to the real values quoted in Table 6.

The estimate 0.06 for the number of 5-reformed decks, obtained multiplying the number of 4-reformed decks by $P_{M,16\cdot 1}(16)$, was still too small to expect a 5-reformed deck. Nevertheless we fortunately and unexpectedly found the first (and up to now unique) 5-reformable deck

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

Clearly, for $m \leq 16$ we know the exact value of $P_{M,m\cdot s}(m\cdot s)$, together with the exact number of k-reformed decks, too. But when we have no information on the number of k-reformed decks, knowing even only an estimate of $P_{M,m\cdot s}(m\cdot s)$ could allow us at least roughly to predict for which value of m we can expect the first 6-reformed decks. To this aim, thanks to their high reliability, Monte Carlo simulations considerably help to know the estimate of $P_{M,m\cdot 1}(m)$ with sufficiently high accuracy. We have estimated, by means of Monte Carlo simulations, the winning probability for $17 \leq m \leq 35$. Analyzing the evolution of the values of $P_{M,m\cdot 1}(m)$, we can make a prediction on the order of number of k-reformed decks, for values of m that we have not yet studied with our technique.

Our prediction strongly depends on the competition between the growth rate of $N_{M,m\cdot s}$ and the decrease rate of $P_{M,m\cdot s}(m\cdot s)$.

The crucial observation is based on the fact that, up to now, for every $7 \leq m \leq 35$

$$0.6 \leq \frac{P_{M,(m+1)\cdot 1}(m+1)}{P_{M,m\cdot 1}(m)} \leq 0.7\ .$$

Starting from the experimental observation that varying k the ratios $\frac{N_{\geq(k+1),m\cdot 1}}{N_{\geq k,m\cdot 1}}$ are quite identical for the same m, we can obtain a rough estimate $N^e_{\geq k,m\cdot 1}$ of the number of decks k-reformed ($k \geq 6$) through the value

$$N^e_{\geq k,m\cdot 1} = m!\cdot\left(P_{M,m\cdot 1}(m)\right)^k\ , \tag{18}$$

the rough estimate of the number of 6-reformed decks can be computed multiplying the number $m!$ of decks by $[P_{M,m\cdot 1}(m)]^6$.

We obtain

$$P_{M,17\cdot 1}(m) \sim 0.00077\quad;\quad N^e_{\geq 6,17\cdot 1} \sim 0.000074\quad;$$

$$\begin{array}{ll}
P_{M,18\cdot 1}(m) \sim 0.00050\quad; & N^e_{\geq 6,18\cdot 1} \sim 0.00010\quad; \\
P_{M,19\cdot 1}(m) \sim 0.00031\quad; & N^e_{\geq 6,19\cdot 1} \sim 0.00011\quad; \\
P_{M,20\cdot 1}(m) \sim 0.00021\quad; & N^e_{\geq 6,20\cdot 1} \sim 0.00021\quad; \\
P_{M,21\cdot 1}(m) \sim 0.00013\quad; & N^e_{\geq 6,21\cdot 1} \sim 0.00025\quad; \\
P_{M,22\cdot 1}(m) \sim 0.000081\quad; & N^e_{\geq 6,22\cdot 1} \sim 0.00032\quad; \\
P_{M,23\cdot 1}(m) \sim 0.000052\quad; & N^e_{\geq 6,23\cdot 1} \sim 0.00051\quad;
\end{array}$$

$$
\begin{array}{ll}
P_{M,24\cdot 1}(m) \sim 0.000034 \quad ; & N^e_{\geq 6,24\cdot 1} \sim 0.00096 \quad ; \\
P_{M,25\cdot 1}(m) \sim 0.000021 \quad ; & N^e_{\geq 6,25\cdot 1} \sim 0.0013 \quad ; \\
P_{M,26\cdot 1}(m) \sim 0.000013 \quad ; & N^e_{\geq 6,26\cdot 1} \sim 0.0019 \quad ; \\
P_{M,27\cdot 1}(m) \sim 0.0000084 \quad ; & N^e_{\geq 6,27\cdot 1} \sim 0.0038 \quad ; \\
P_{M,28\cdot 1}(m) \sim 0.0000054 \quad ; & N^e_{\geq 6,28\cdot 1} \sim 0.0076 \quad ; \\
P_{M,29\cdot 1}(m) \sim 0.0000034 \quad ; & N^e_{\geq 6,29\cdot 1} \sim 0.0014 \quad ; \\
P_{M,30\cdot 1}(m) \sim 0.0000022 \quad ; & N^e_{\geq 6,30\cdot 1} \sim 0.030 \quad ; \\
P_{M,31\cdot 1}(m) \sim 0.0000014 \quad ; & N^e_{\geq 6,27\cdot 1} \sim 0.062 \quad ; \\
P_{M,32\cdot 1}(m) \sim 0.00000087 \quad ; & N^e_{\geq 6,32\cdot 1} \sim 0.11 \quad ; \\
P_{M,33\cdot 1}(m) \sim 0.00000055 \quad ; & N^e_{\geq 6,33\cdot 1} \sim 0.24 \quad ; \\
P_{M,34\cdot 1}(m) \sim 0.00000036 \quad ; & N^e_{\geq 6,34\cdot 1} \sim 0.64 \quad ; \\
P_{M,35\cdot 1}(m) \sim 0.00000023 \quad ; & N^e_{\geq 6,35\cdot 1} \sim 1.53 \ .
\end{array}
$$

(the study of greater values of m is still under investigation). Thus we can reasonably expect the first 6-reformed permutations for $32 \leq m \leq 35$.

In general we have the following

Theorem 6.1 If

$$
0.6 \leq \frac{P_{M,(m+1)\cdot 1}(m+1)}{P_{M,m\cdot 1}(m)} \qquad \forall m \geq 7 \tag{19}
$$

then there exists $\overline{m} = \overline{m}(k)$ such that

$$
N^e_{\geq k,m\cdot 1}(m) \geq 1 \qquad \forall m \geq \overline{m} \ .
$$

Proof Let us indicate with m_e the highest value for which it is known (by means of Monte Carlo simulations) the estimate of $P_{M,m_e\cdot 1}(m_e)$ (up to now $m_e = 35$). Thus, by virtue of (19),

$$
P_{M,m\cdot 1}(m) \geq (0.6)^{m-m_e} \cdot P_{M,m_e\cdot 1}(m_e) \qquad \forall m > m_e \ . \tag{20}
$$

From (18) it follows

$$
N^e_{\geq k,m\cdot 1} \geq m! \cdot (P_{M,m_e\cdot 1}(m_e))^k \cdot \left[(0.6)^{m-m_e}\right]^k \qquad \forall m > m_e \ . \tag{21}
$$

The right hand side of (21) is greater than 1 if and only if

$$
m! \cdot \left[(0.6)^k\right]^m \geq \frac{\left[(0.6)^{m_e k}\right]}{(P_{M,m_e\cdot 1}(m_e))^k} \ .
$$

Once fixed k and m_e, the right hand side of (21) is a constant. Since

$$a^n \cdot n! \longrightarrow_{n\to\infty} \infty \qquad \forall a > 0 \ , \tag{22}$$

we have the thesis.

Remark 6.1 Theorem (6.1) tells us that, under hypothesis (19), we can reasonably expect a positive answer to question **5)**. The importance of hypothesis (19) can be read in the limit (22) which cannot be used if

$$\frac{P_{M,(m+1)\cdot 1}(m+1)}{P_{M,m\cdot 1}(m)} \to 0 \quad for \quad m \to \infty \ .$$

Finally, lower bound 0.6 in (19) is given experimentally. Theorem (6.1) is still valid putting the more general hypothesis

$$\exists \alpha > 0 \quad s.t. \quad 0 < \alpha \leq \frac{P_{M,(m+1)\cdot 1}(m+1)}{P_{M,m\cdot 1}(m)} \qquad \forall m \geq 7$$

uniformly with respect to m.

The extension of our analysis to $s > 1$ confirms the above presented arguments proposed concerning the relationship between $P_{M,m\cdot s}(m\cdot s)$ and the appearance of k-reformed decks.

Also in this case, in order to look for 4-reformed permutations, we can compare the growth rate of $N_{M,m\cdot s}$ and the decrease rate of $P_{M,m\cdot s}(m \cdot s)$. In the most advanced cases we have examined (i.e., $m = 9$, $s = 2$; $m = 6$, $s = 3$; $m = 5$, $s = 2$) we found a large number of 3-reformed decks. Though $P_{M,m\cdot s}(m \cdot s)$ was rapidly decreasing, we expected 4-reformed decks already for $s = 2$, $9 \leq m \leq 11$; $s = 3$, $7 \leq m \leq 9$; $s = 4$, $7 \leq m \leq 8$. In fact very recently we studied the case $m = 9$, $s = 2$ and we found four 4-reformed decks:

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

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

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

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

Let us observe that, for $m = 3$, $s = 4$, we found the first (and up to now unique) non trivial 1-cycle: 1 1 1 1 2 2 3 2 2 3 3 3. Consequently, the second part of question **5)** receives a positive answer, but only in a “multisuit” framework, while a negative answer is highly probable for $s = 1$.

For what concerns reformed decks and cycles, *Modular Mousetrap* is much more intriguing. First we will need some terminology, in order to distinguish the different situations we will deal with.

We can interpret the reformation sequences as Discrete Dynamical Systems [1], [20], where every reformation A is a state and the deck preceding it is a pre-image of A. As shown by the deck 1 2 3 ... n, a deck A may have several different pre-images (their total number is the in-degree of A).

Decks without pre-images are known as *Garden of Eden states.*

Besides the k-reformed decks we must consider the cycles.

When a trajectory encounters a state that occurred previously, we have a cycle. The trajectory leading to the cycle is called *transient* or *pre-period.* The *period* of a k-cycle is the number k of states in it.

A 1-cycle can be seen as a fixed point of the dynamical system. The deck 1 2 3 ... n generates a 1-cycle, i.e., is a fixed point.

If the k-th reformation coincides with the h-th reformation ($1 \leq h < k$), we will divide the total k-trajectory into two parts:

i) a h-pre-period, where there is a sequence of h reformations;

ii) a $(k-h)$-cycle, starting from the h-th reformation and stopping at the k-th reformation, which coincides with the h-th one.

Guy and Nowakowski analyzed "by hand" all the reformed permutations for $s = 1$, $m \leq 5$. Clearly, this analysis cannot be easily performed for greater values of m. Indeed, for $m = 6$, they considered only decks where the first card is an "ace".

We have improved their results to many more cases and to $1 < s \leq 4$, as shown in Tables $10 - 13$.

Thanks to the high winning probability, in particular if n is prime, the game *Modular Mousetrap* has produced many interesting and intriguing results. In particular, we obtained very long sequences of reformed decks and cycles; the reason for it must be found in the fact that, as already remarked, in this game, when n is prime, we always have 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. Due to the highly increasing number of permutations when m grows, we were able to study all the decks in *Modular Mousetrap*, for $s = 1$, only up to $m = 13$.

The most complete and exhaustive investigation has been performed for $s = 1,\ m = 11$ and $s = 1,\ m = 13$.

Table 10 shows the huge increase of the number of cycles, with respect to smaller values of m. For $m = n = 11$, as shown in Table 14, we found the six 203-trajectories, starting respectively from

1 11 5 8 2 6 9 4 7 10 3 ; 1 11 5 8 2 6 9 10 7 4 3 ; 1 11 5 9 2 6 10 8 7 4 3 ;

1 11 5 8 2 6 4 9 7 10 3 ; 1 11 5 8 2 6 4 9 10 3 7 ; 1 11 5 9 2 6 8 10 7 4 3 ,

which, after 137 reformations, reach the state

1 2 3 4 7 5 6 8 9 10 11

which produces a 66-cycle.

For $m = 13$ the length of reformed decks grows: we have found eleven 51-reformable decks. One of them is

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

Longest cycles were discovered for $m = 13$, too: the deck

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

is characterized by a very long trajectory: after a 839-pre-period we obtain the deck

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

and the trajectory ends in a 1-cycle.

Curiously, for $m = 11$ the decks gave only $1,\ 2,\ 3,\ 4,\ 14,\ 15$ and 66-cycles. The number of decks entering in a 66-cycle is very high: $1,701,937$. For $m = 13$ we found only $1,\ 2,\ 3,\ 6,\ 7$ and 12-cycles.

Since we expect to achieve many more interesting results whenever n is prime, we have also examined the first 50 million reformed decks for $s = 1$ and $m = 17$ and the first 320 million reformed decks for $s = 1$ and $m = 19$.

We can understand the exhaustiveness of *Modular Mousetrap* considering that in our investigations, though we analyzed only few decks among all the $17! \sim 3.56 \cdot 10^{14}$ permutations and the $19! \sim 1.22 \cdot 10^{17}$ permutations, we obtained again a 51-reformed deck for $s = 1\ ,\ m = 17$, as for $s = 1\ ,\ m = 13$, and, mainly, a 39924-trajectory, ending in the trivial 1-cycle, from the deck

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

(clearly, we did not check the correctness of all the reformations, but we have sufficiently tested the computer program to believe it!).

Moreover, we found two 267-trajectories which, starting respectively from the decks

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

which, after 58 reformations, reach the state

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

which produces a 209-cycle.

Consequently, it is highly probable that the above mentioned scores could be improved, if we would study more cases for $s = 1$, m prime and $m > 13$.

Concerning the 1-cycles, for $s = 1$ there is no evidence of other cycles than the trivial one $(1\ 2\ \cdots\ m)$. When we pass to *"multisuit" Modular Mousetrap*, we not only have the trivial 1-cycle $1\ 2\ \cdots\ m\ 1\ 2\ \cdots\ m\ \cdots\ 1\ 2\ \cdots\ m$, but several other non trivial 1-cycles whose structure in general seems not to have any regularity: for example, the decks

1 1 2 2 ; 1 3 1 2 2 3 ; 1 4 3 1 2 2 3 4 ; 1 6 3 4 5 1 2 2 3 4 5 6 ; 1 3 4 5 1 2 2 3 4 5 ;

2 7 6 5 7 1 3 2 4 3 4 5 6 ; 2 6 6 5 7 1 1 3 2 4 3 4 5 7 ; 2 5 6 5 7 1 1 3 2 4 3 4 6 7 ; 2 7 6 5 1 1 3 5 2 4 3 4 6 7 ;

3 6 3 2 4 4 5 1 7 1 2 5 6 7 ; 1 3 4 5 6 7 1 2 2 3 4 5 6 7 ; 1 3 5 3 2 1 4 6 5 2 4 6 7 7 ;

1 1 1 2 2 2 ; 1 3 1 2 2 3 1 2 3 ; 1 1 1 2 2 3 3 2 3 etc.

are fixed points.

Up to now, we have achieved the greatest number of 1-cycles for $s = 3$, $m = 5$ and for $s = 4$, $m = 4$, where we found ten different 1-cycles. Since the two cases are the most advanced in our studies, we can suppose that we could obtain greater numbers of non trivial 1-cycles if we would continue our analysis for higher values of m.

The explosion of the number of k-cycles and k-reformed decks, already for $s = 1$, allows us to give a partial answer to questions **5)** and **6)** for *Modular Mousetrap*, as shown in Table 18. However, the results reported in this table seem to suggest a positive answer to the first part of question **6)**.

Due to the difficulty of reporting all the results for *Modular Mousetrap*, we have built the web page http://www.dmmm.uniroma1.it/∼bersani/mousetrap.html

where we show the numbers of trajectories, pre-periods, cycles and reformed decks for the different values of m and s we investigated. The page is still under construction and many documents are still written in Italian, but the meaning of the results is clear.

We extended the study of reformed decks to the game $(HLM)^2N$, too.

We can repeat the considerations related to the connection between the appearance of k-reformed decks and the probability to obtain the best score, which we indicate with $P_{max} := P(C_{max})$. Knowing the low probability to have winning decks at $(HLM)^2N$ (see Table 2), we cannot expect to easily attain k-reformed decks, with $k \geq 2$, at this game.

In fact, Table 19 shows that, excluding the trivial 1-cycles $1\ 1\ \cdots\ 1$ and $1\ 1\ \cdots\ 1\ 2\ 2\ \cdots\ 2$, there is no evidence of k-reformed decks ($k \geq 2$), apart from the unique, very special case $m = 2$, $s = 4$, where we attained the following four 2-times reformable decks:

$$2\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ ;\ 1\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ ;\ 1\ 1\ 2\ 2\ 2\ 2\ 1\ 1\ ;\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 1\ .$$

The very fast decrease of P_{max}, when m grows, seems not to allow us obtaining 2-reformed decks in other cases.

Thus, we have focused on the reformed decks satisfying (WC), instead of (SC). As shown in Table 20, we have found new 2-reformed decks only in the cases $s = 4$, $4 \leq m$, where the growth rate of the number of total reformed decks is sufficiently high to compensate the decrease of the probability $P_{max}(WC)$ and to produce decks satisfying (WC). Since $P_{max,6\cdot 4}(WC) \sim 4 \cdot 10^{-8}$, it is an open question if it is possible to obtain 3-reformed decks for higher values of m.

The case $s = 1$ has been studied only for $1 \leq m \leq 4$, because, as shown by the author (Table 3), (WC) is satisfied only for these values of m. For $m = 1$, the unique deck 1 is a 1-cycle. For $m = 2$, we have only the 1-cycle $1\ 2$. For $m = 3$ we have two 1-reformable decks: $1\ 3\ 2$ and $3\ 2\ 1$. For $m = 4$ we have only the 1-reformable deck $2\ 1\ 3\ 4$.

As already remarked, the existence of sequences of arbitrary length is still an open problem for *Mousetrap*. Thus, in some sense, it can be considered on the boundary between the classes of games producing reformed decks and of games without reformed decks.

7 Conclusions and further developments

The backward 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* and in particular for what concerns the reformed permutations. 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 backward 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, in our FORTRAN program it is however necessary to store all the winning $(n-2)$-strings.

In the game *Mousetrap*, for $m = 16$, $s = 1$, the storage of all the winning $(n-3 = 13)$-strings needed a 325 GB memory, while the storage of all the winning $(n-2 = 14)$-strings needed a 596 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. 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.

Moreover, up to now, the backward technique seems to be the unique one capable of giving more complete answers to questions **1) - 6)**. However, none of them has yet received a definitive answer. In particular, finding 5-reformed decks at *Mousetrap* brings to conjecture that, for increasing values of m, it is possible to find k-reformed decks for every value of k (question **4)**). As already observed, the answer strongly depends on the competition between the growth rate of the number of total reformed decks and the decrease rate of P_{max}, when m grows. It could be very useful to study the game for increasing values of m, by means not only of Monte Carlo simulations, but mainly of the backward technique implemented in a parallel computing framework in order to know the evolution, with m, not only of $P_{M,m\cdot 1}(m)$, but also of the different probabilities $P_{\geq k}$ to achieve decks which are at least k-reformable.

The improvement of the technique, mainly concerning the memory saving problems, could lead to more satisfactory results.

For example, it is highly probable that the scores attained at *Modular Mousetrap* by the deck

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

which is 51-times reformable, by the deck

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

which yielded a 39924-trajectory and by the 209-cycle

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

could be improved, if we would study more cases for $s = 1$, m prime and $m > 13$.

In order to encourage further suggestions to improve the memory saving and the algorithm we implemented, we inserted all the FORTRAN files used for our researches in our web page

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

together with all the results for *Modular Mousetrap*. The page is still under construction and the comments in the FORTRAN files are still written in Italian. However, until their translation into English is ready, I am at the disposal of everyone who would like to collaborate in this research in order to explain the passages in the FORTRAN files.

Some other problems can be explored in the games analyzed in this paper. For example, since *Modular Mousetrap* gives very long sequences of reformed decks, it could be interesting to determine the number of *Garden of Eden points*, or the in-degree of every cycle, in particular of the trivial one 1 2 3 ... n.

8 Acknowledgements

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 text of this paper and the computer programs through which I have implemented the backward technique.

I am also very grateful to Prof. N. J. A. Sloane and K. S. Chua, for their information on the most recent results on reformed decks.

The analysis of the most advanced cases was possible thanks to the usage of the servers of the Dept. "Me.Mo.Mat." Computing Center; the programs were compiled and executed on a Sun Fire X2100 M2 Cluster running Debian GNU/Linux Etch.

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] E. Berlekamp, J. Conway and R.K. Guy, *Winning Ways for your Mathematical Plays*, vol. 4, A K Peters Ltd. (Wellesley, Massachusetts, 2004).

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

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

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

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

[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, France 1943).

[8] R. K. Guy, Mousetrap, §*E*37 in *Unsolved Problems in Number Theory*, 3rd edition, Springer-Verlag (New York, 2004), 237 – 238.

[9] R. K. Guy and R. Nowakowski, Mousetrap, in D. Miklós, V.T. Sós and T. Szonyi, eds., *Combinatorics, Paul Erdős is Eighty*, vol. 1, János Bolyai Mathematical Society, Budapest, 1993, 193 – 206.

[10] R. K. Guy and R. Nowakowski, Unsolved Problems - Mousetrap, Amer. Math. Monthly **101**, 1007 – 1008 (1994).

[11] R. K. Guy and R. Nowakowski, Monthly Unsolved Problems, 1969 – 1995, Amer. Math. Monthly **102**, 921 – 926 (1995).

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

[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, New Jersey, 1980).

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

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

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

[18] M. Z. Spivey, Staircase Rook Polynomials and Cayley's Game of Mousetrap, accepted for publication on European J. Combin.

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

[20] A. Wuensche, Discrete Dynamical Networks and their Attractor Basins, Complexity International **6** (1998).