

# How to Introduce the Basis of Algorithmics? Thanks to the Enumeration and Composition of All Riffle Shuffles from an N Card Deck Used in MathMagic

Pierrot Schott

Art et Recherche NUMérique (ARNUM), Ecole Supérieure d'Informatique,  
d'Electronique et d'Automatisme (ESIEA), Paris, France

Email: [schott@esiea.fr](mailto:schott@esiea.fr), [pierrot.schott@laposte.net](mailto:pierrot.schott@laposte.net), [magic.carte@laposte.net](mailto:magic.carte@laposte.net), [arnum@esiea-recherche.eu](mailto:arnum@esiea-recherche.eu)

Received April 17<sup>th</sup>, 2012; revised May 22<sup>nd</sup>, 2012; accepted May 30<sup>th</sup>, 2012

Why use magic for teaching combinatory, algorithms and finally informatics basis as tables, control structure, loops and recursive function? Magicians know that once the surprise has worn off, the audience will seek to understand how the trick works. The aim of every teacher is to interest their students, and a magic trick will lead them to ask “how?” and “why?” and “how can I create one myself?” In this article we consider a project I presented in 2009, the subject of which was “How many riffle shuffles does exist from an n card deck? Find the composition of each possible riffle shuffle”. The aim of the paper is not only to describe the project scope, the students’ theoretical studies, their approach to this problem and their computer realizations, but also to give ideas for a course or project using pedagogy. That is why only remarkable students’ realizations are shown. In order to complete the given project, the students must answer three steps: the first one is to answer to the following question: “how can I find all possible riffle shuffles with few cards?” (for example 3, 4 or 5 cards) the second one (to go further) is to answer to the following question “how can I generalize this solution through an algorithm?” the last one (to obtain the results!) is to program the algorithm with a recursive and a non-recursive solution). Each step of the Matlab™ solution code is associated with an informatics basis. Whatever the student’s professional ambitions, they will be able to see the impact that originality and creativity have when combined with an interest in one’s work. That’s why, two ameliorations of the “basic” algorithm are proposed and a study of the gain thanks to these ameliorations is done. The students know how to “perform” a magic trick for their family and friends thanks to the use of riffle shuffle in Gilbreath’s principles, a trick that they will be able to explain and so enjoy a certain amount of success with. Sharing a mathematical/informatics demonstration is not easy and the fact that they do so means that they will have worked on and understood and are capable of explaining this knowledge. Isn’t this the aim of all teaching?

*Keywords:* Higher Education; Engineer; Educational Method; Informatics; Algorithm

## Introduction

I am fascinated by magic (or rather, conjuring) and for many years I have used this way of teaching, both in my physics classes and as higher education teacher trainer.

In this paper I present all the notions, a brief history of magic and finally the researchers who have used this art to teach and/or to research.

The aim of the magician is to hide the principles he uses (using maths, physics, psychology, sleight of hand, etc.) by disguising the trick so that the audience has no way of discovering how it is done; thus allowing the magic to remain.

**The teacher can do exactly the opposite: unraveling a magic trick to highlight the principles used!**

## A Brief History of Magic

From the beginning of time people have feared what they don’t understand, and sought logical explanations for inexplicable phenomena. To begin with, they considered them to be the work of magic, then the work of the gods, then the work of

God himself. The church discouraged the spread of the conjurer’s art as it preferred not to have rational explanations for what was considered supernatural.

The first magic tricks were performed in the Middle Ages by clowns and con-men who would entertain passers-by by getting them to bet on the position of a ball hidden in one of three up-turned goblets, the bet being usually lost. This trick is known nowadays by the name “cups and balls” (Mayol, 2000). The first book considered to be about modern magic (or should we say, “conjuring”) was written and printed in the 16th Century, and was about magic with ropes (Ammar, 1998). It wasn’t until the end of the 19th Century that the word magic took on its present-day meaning when the famous magician Houdini, father of modern magic, made it the art it is today.

Since then, a good number of principles have been invented and improved on by magicians and gamblers (Erdnase, 1902), especially for bets with card tricks. Since the 1980s, the secrets that were once passed on from master to apprentice are now universally available through the use of video cassettes and modern communication technology, and magic has become a big business.

## Card Magic as the Vector of Research and Teaching

From 1886 until 1896, Poincaré occupied the chair of “probability Calculus” in the Paris University “La Sorbonne”. He wrote a work named “probability calculus” (Poincaré, 1912; Sheynin, 1991) which was printed for the first time in 1896. In the second edition, he brings very fundamental new reflections on the groups and the hypercomplex systems and on the ergodic theory. He is brought to these innovations by the study of the card shuffling and liquid mixing. The problems with the card shuffling and liquid diffusion studied by Poincaré are application cases of the ergodic theory, which is at the center of the probability leveling phenomenon: if the deck was shuffled for a long time, all the possible permutations have the same probability.

Both principles of Gilbreath (Gilbreath, 1958; Gilbreath, 1966; Gilbreath, 1989) are fascinating principles, allowing one to do extraordinary card tricks!

Some Mathematicians, such as M. Gardner (Magid, 2005; Gardner, 1958; Gardner, 2005), P. Diaconis (Diaconis, 1998; Diaconis, 2003; Assaf, 2009) or C. Mulcahy (Mulcahy, 2003; Mulcahy, 2004; Mulcahy, 2007), and some computer specialists, such as G. Huet (Huet, 1991)—one of the creators of the COQ language which allows automatic mathematical proofs to be made—studied this principle (and they are not the only ones).

We shall not present this principle here but it is necessary to know that it is based on a card shuffle commonly used on both sides of the Atlantic Ocean: the American riffle shuffle.

### Summary

The subject was given for the 4<sup>th</sup> semester students in a higher education school in France (named “Grandes écoles d’ingénieurs”) as a Multi Disciplinary Project (MDP) described in (Schott, 2010). The students had to solve and illustrate the following problem:

**“How many riffle shuffles does exist from an n card deck? Find the composition of each possible riffle shuffle.”**

A theoretical solution of the number of riffle shuffles is given by D. Aldous and P. Diaconis (Aldous, 1986). However, the composition was not given. “A first mathematical and algorithmically study is given in french by A. Lachal (Lachal, 2012). I introduce here firstly many algorithmic ameliorations and the use of data files and secondly a pedagogic point of view by giving, through the article construction, an possible approach for teaching an high level informatics language

By giving this project, I would like the students to:

- Find and understand the paper in which a solution part is written;
- Understand the whole problem and determine the way that the human being can solve it;
- Transliterate this “human solution” into a real algorithm;
- Write the algorithm using an informatics language; I choose Matlab™;
- Discover all of the basis of the sequential language (data format, table, loop, recursively, etc.);
- Discover a fascinating area and improve their creativity.

**The Aim of the Paper: To Not Only Show the Result of a High Education Project, But Also Give Ideas for a Course or Using Project Pedagogy**

The description of the higher education in France, and the

scopes and the educational objectives of the MDP are described in (Schott, 2010).

I do not want to describe the given subject, the composition of the student group, my motivations, my expectations and the realization of the students.

I prefer to write the “true” solution of the problem in order to give some ideas for an example in an algorithmic course or for a course using project pedagogy.

However, the order and the name of titles in this paper could be the course structure. In fact, I introduce step by step all of essentials programming basis in the following order:

- Modeling the problem,
- Solving the problem thanks a analytics solution,
- Finding algorithms,
- Programming the algorithms by:
  - Introducing the data types,
    - Introducing the data types,
    - Programming with the “for” and “while” loops,
    - Using the branch instruction as “if” ... “then” ... “else”,
    - Writing results in files,
    - Recursively programming,
    - Changing a recursive function by a simple ‘while’ loop.
- Results analysis versus computing time.

### Number and Composition of All of the Riffle Shuffles from a 2 Card Deck, 3 Card Deck and 4 Card Deck (From the Problem to an Algorithm: Step 1)

If we take an n card deck, how can we find the composition? It is humanly “impossible” if the number of cards is greater than 7 because the number of different riffle shuffles is:

$$M(n) = 2^n - n \quad (1)$$

This relation is proved in Appendix A.

In order to find a general algorithm, we have to understand the way of solving the problem with a deck of only a few cards.

### Mathematical Notations

We introduce three following notations:

- $\mathcal{M}_1(n)$ : the number of possible riffle shuffles,
- $\mathcal{M}(n)$ : the number of different possible riffle shuffles,
- $\mathcal{M}_i(p,q)$ : the number of possible riffle shuffles between a p card deck and a q card deck (the insertion of the first card of the p card deck is under the i<sup>th</sup> card of the q card deck) with  $p + q = n$ .

Each card will be represented by the quantity  $C_i^{(j)}$  where i is the initial position of the card in the deck and j is the position of this card after j shuffles. This will be written more simply by  $C_i$ .

### Definition of Riffle Shuffle

Let an n card deck be cut into two portions, called subdecks. Each subdeck is riffled with the hands, and the cards of each subdeck become entangled, as presented in **Figure 1**.

### Number and Composition of All Possible Riffle Shuffles of a 2 Card Deck

Let us take a 2 card deck in this order:  $C_1-C_2$ . There is only one way possible to cut this deck into two subdecks as pre-

sented in **Figure 2**.

There are two different possible riffle shuffles, presented in **Figure 3**.

In summary, there are only 2 riffle shuffles from a 2 card deck and Equation (1) is verified.

So the number of riffle shuffles  $\mathcal{M}_T(2)$  is :

$$\mathcal{M}(2) = \mathcal{M}_T(2) - 0 = 2 \text{ and } 2 = 2^2 - 2 = \mathcal{M}(2) \quad (2)$$

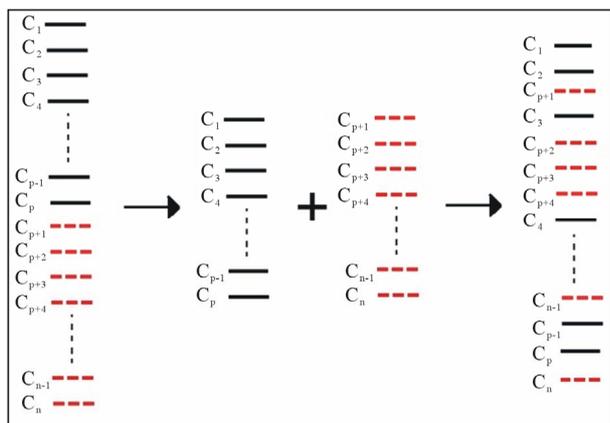
### Number and Composition of All Possible Riffle Shuffles of a 3 Card Deck

Let us take a 3 card deck in this order:  $C_1-C_2-C_3$ . There are only two ways possible to cut this deck into two subdecks as presented in **Figure 4**.

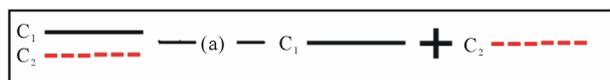
So, the number of all riffle shuffles  $\mathcal{M}_T(3)$  is worth:

$$\begin{aligned} \mathcal{M}_T(3) &= \mathcal{M}_0(2,1) + \mathcal{M}_0(1,2) \\ \mathcal{M}_T(3) &= \sum_{p=1}^2 \mathcal{M}_0(p,3-p) \end{aligned} \quad (3)$$

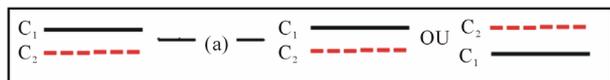
For each of the configurations presented in **Figure 4**, there



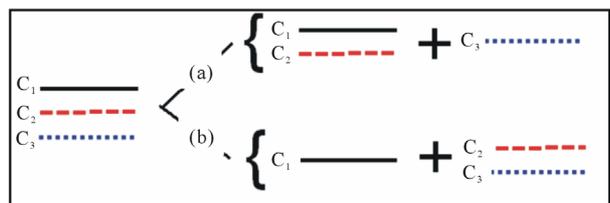
**Figure 1.** Example of a cut for an n card deck and then a riffle shuffle.



**Figure 2.** All possible cuts of a 2 card deck.



**Figure 3.** All possible riffle shuffles of a 2 card deck.



**Figure 4.** All possible cuts of a 3 card deck.

are several possible shuffles, which are presented in **Figure 5**.

In summary, there are 2 possible cuts and 6 riffle shuffles (of which 5 are different) from a 3 card deck and the Equation (1) is verified.

The number of riffle shuffles between a 1 card subdeck and a 2 card subdeck,  $\mathcal{M}_0(1,2)$ , is the same as the number of shuffles between a 2 card subdeck and a 1 card subdeck,  $\mathcal{M}_0(2,1)$ !

$$\mathcal{M}_0(1,2) = 3 = \mathcal{M}_0(2,1) \quad (4)$$

Thanks to Equations (3) and (4), the number of riffle shuffles- $\mathcal{M}_T(3)$ -comes easily:

$$\mathcal{M}_T(3) = \mathcal{M}_0(2,1) + \mathcal{M}_0(1,2) = 6 \quad (5)$$

The number of different riffle shuffles  $\mathcal{M}(3)$  is worth:

$$\mathcal{M}(3) = \mathcal{M}_T(3) - 1 = 5 \text{ and } 5 = 2^3 - 3 = \mathcal{M}(3) \quad (6)$$

### Number and Composition of all Possible Riffle Shuffle of a 4 Card Deck

Let us take a 4 card deck in this order:  $C_1-C_2-C_3-C_4$ . Only three ways are possible to cut this deck into two subdecks as presented in **Figure 6**.

So, the number of all riffle shuffles  $\mathcal{M}_T(4)$  is worth:

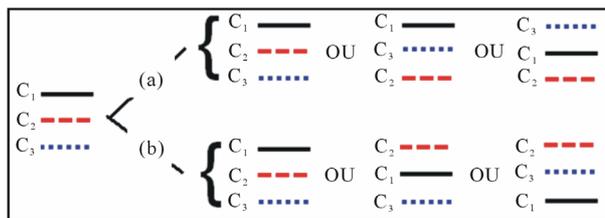
$$\begin{aligned} \mathcal{M}_T(4) &= \mathcal{M}_0(3,1) + \mathcal{M}_0(2,2) + \mathcal{M}_0(1,3) \\ \mathcal{M}_T(4) &= \sum_{p=1}^3 \mathcal{M}_0(p,4-p) \end{aligned} \quad (7)$$

For each of three configurations presented in **Figure 6**, there are several possible shuffles, which are presented in **Figures 7 and 9**.

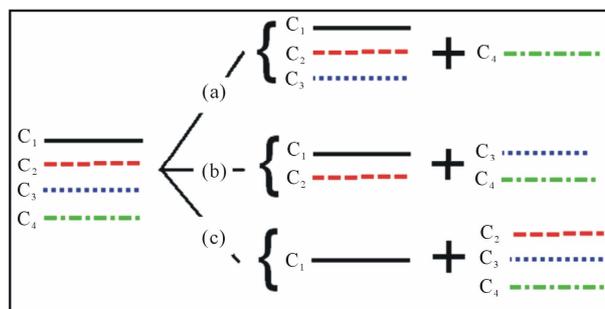
For the first cut—noted (a), and the third cut—noted (c), all of possible riffle shuffle are presented in **Figure 7**.

For each configuration (a) and (c), a one card deck is shuffled with a 3 card deck:

$$\mathcal{M}_0(3,1) = \mathcal{M}_0(1,3) = 4 \quad (8)$$



**Figure 5.** All possible riffle shuffles of a 3 card deck.



**Figure 6.** All possible cuts of a 4 card deck.

For the (b) configuration, the first card (here  $C_1$ ) is put away. Thus the new configuration is so: one subdeck with one card ( $C_2$ ) and the second unchanged subdeck ( $C_3-C_4$ ). This configuration is well known and the riffle shuffles are presented in **Figure 8**.

Three riffle shuffles are obtained. The  $C_1$  card is shuffled in these 3 shuffled subdecks. What is essential to note is that the  $C_1$  card must be inserted over the  $C_2$  card. All of the possible riffle shuffles so obtained are presented in **Figure 9**.

We thus have found the value of  $\mathcal{M}_0(2,2)$ . Mathematically, it becomes:

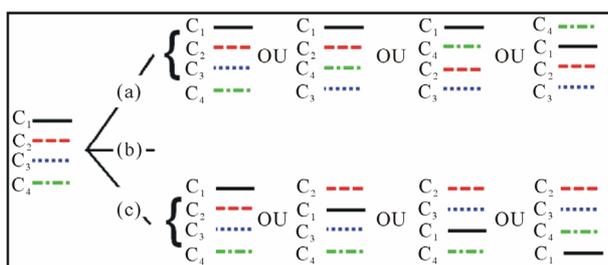
$$\begin{aligned} \mathcal{M}_0(2,2) &= \mathcal{M}_1(1,3) + \mathcal{M}_2(1,3) + \mathcal{M}_3(1,3) \\ \mathcal{M}_0(2,2) &= \sum_{k=1}^3 \mathcal{M}_k(1,3) \end{aligned} \tag{9}$$

Thanks to **Figure 9**, we can notice that:

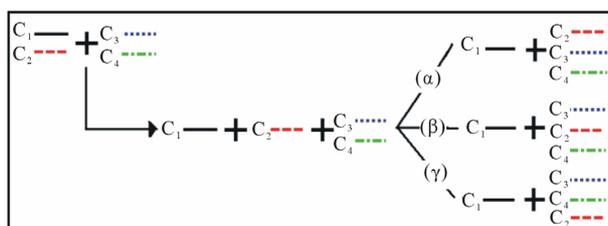
$$\mathcal{M}_1(1,3) = 3; \mathcal{M}_2(1,3) = 2; \mathcal{M}_3(1,3) = 1 \tag{10}$$

So, the value of  $\mathcal{M}_0(2,2)$  is worth:

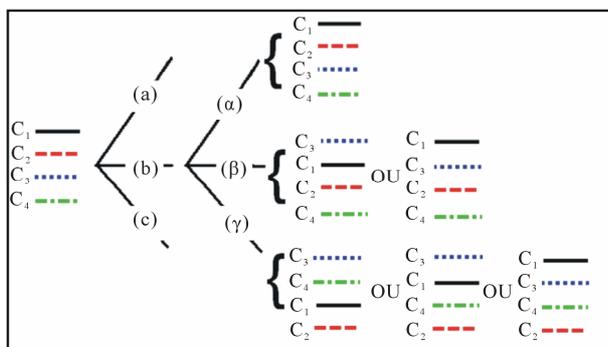
$$\mathcal{M}_0(2,2) = 3 + 2 + 1 = 6 \tag{11}$$



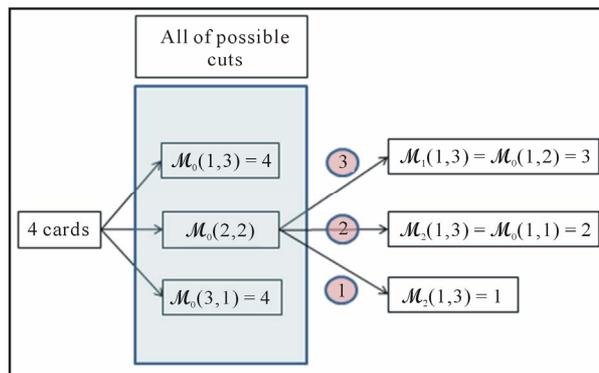
**Figure 7.** All possible riffle shuffles of a 4 card deck [cut (a) and (c)].



**Figure 8.** All possible riffle shuffles between  $(C_1-C_2)$  and  $(C_3-C_4)$ .



**Figure 9.** All possible riffle shuffles of a 4 card deck [configuration (b)].



**Figure 10.** All possible riffle shuffles of a 4 card deck.

Thanks to the Equations (7), (8) and (11),  $\mathcal{M}_T(4)$  can easily be calculated:

$$\mathcal{M}_T(4) = \mathcal{M}_0(1,3) + \mathcal{M}_0(2,2) + \mathcal{M}_0(3,1) = 14 \tag{12}$$

**Figure 10** summarises the whole process to calculate  $\mathcal{M}_T(4)$ . In summary, there are 3 possible cuts and 14 riffle shuffles (of which 12 are different) from a 4 card deck and the Equation (1) is verified.

$$\mathcal{M}(4) = \mathcal{M}_T(4) - 2 = 12 \text{ and } 12 = 2^4 - 4 = \mathcal{M}(4) \tag{13}$$

### Number and Composition of All of Riffle Shuffles of an N Card Deck (From the Problem to an Algorithm: Step 2)

Four points are presented in order to find the number and composition of all riffle shuffles of an n card deck:

- All possible cuts of an n card deck: relation between  $\mathcal{M}_T(n)$  and  $[\mathcal{M}_0(i,j)]_{i+j=n}$ ;
- Main program algorithm;
- All possible riffle shuffles between a p card subdeck and a q card subdeck  $\mathcal{M}_0(p,q)$ ;
- The obtained riffle shuffle from a 1 card subdeck and a q card deck:  $\mathcal{M}_0(1,q)$  and  $\mathcal{M}_1(1,q)$ .

For each point, the human approach and its transliteration into an algorithm are given. The informatics bases are given in the next section because I would like to describe first the algorithm, and then the programming solution!

### All Possible Cuts from an N Card Deck: Relation between $\mathcal{M}_T(n)$ and $[\mathcal{M}_0(i,j)]_{i+j=n}$

I present, firstly, all possible cuts of an n card deck, secondly the mathematics formulation with the  $[\mathcal{M}_0(i,j)]_{i+j=n}$ , and finally the main program algorithm.

### Decomposition “By Hand” and Mathematics Formulation

Let us take an n card deck in this order:

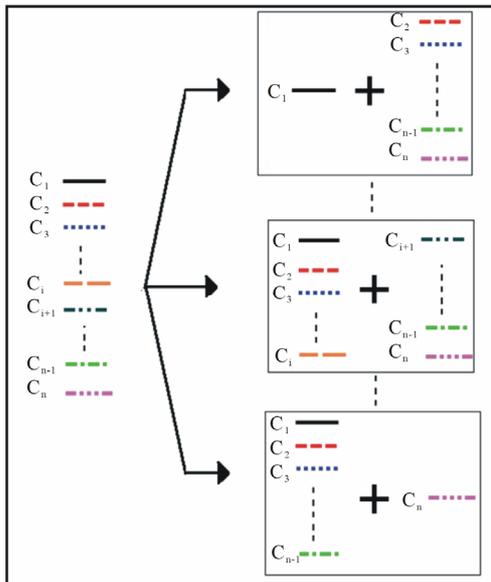
$C_1-C_2- \dots -C_i- \dots -C_N$ . Only  $(N - 1)$  ways are possible to cut this deck in two subdecks as presented in **Figure 11**.

Mathematically, in order to calculate all of the possible riffle shuffles, it is necessary:

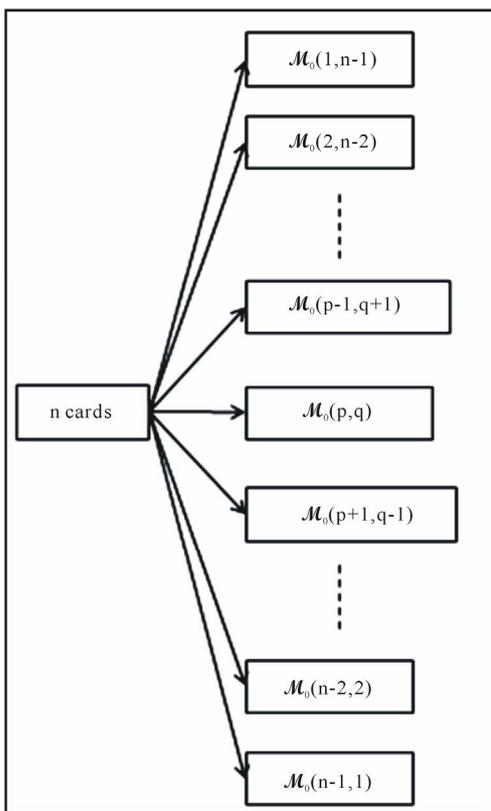
- Firstly, to find all possible cuts of the n card deck, so  $(n - 1)$  possibilities, as presented in **Figure 12**;
- Secondly, to calculate the number of possible riffle shuffles  $[\mathcal{M}_0(i,j)]_{i+j=n}$ .

Thus Equation (14) generalizes Equation (3) [resp. (7)] found where  $n = 3$  [resp.  $n = 4$ ]:

$$\mathcal{M}_r(n) = \sum_{p=1}^{n-1} \mathcal{M}_0(p, n-p) \tag{14}$$



**Figure 11.**  
All possible cuts of an  $n$  card deck.



**Figure 12.**  
All possible cuts of an  $n$  card deck.

Let us remark a coarse relation: the number of possible riffle shuffles between a  $p$  card deck and a  $q$  card deck is the same as the ones between a  $q$  card deck and a  $p$  card deck:

$$\mathcal{M}_0(p, q) = \mathcal{M}_0(q, p) \tag{15}$$

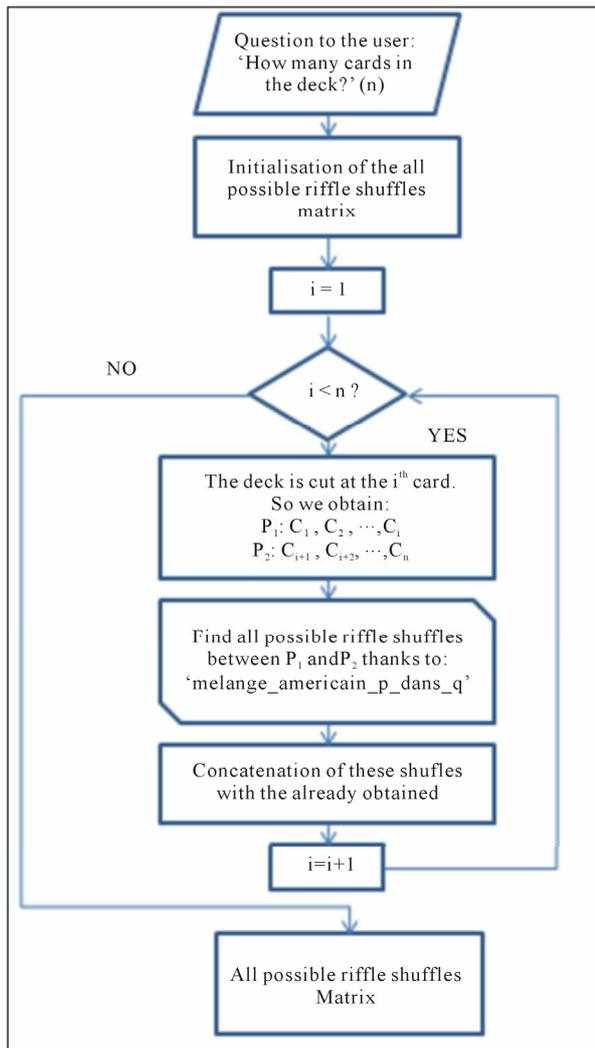
So the number of terms in the sum of Equation (14) can be divided by 2. Lastly, Equation (16) generalizes Equation (4) [resp. (8)] found where  $n = 3$  [resp.  $n = 4$ ] and Equation (15) too:

$$\mathcal{M}_i(p, q) = \mathcal{M}_i(q, p) \tag{16}$$

**Main Program Algorithm**

Thanks to the proposed decomposition by Equation (14), the main program algorithm comes quite naturally, as presented in **Figure 13**. Every rectangle in the “loop” requires of course an algorithm much more detailed in order to program all of the software.

The Matlab™ main program is called “tous\_melanges\_americains” (translated as “all\_riffle\_shuffles” in English). It is given in Appendix B.



**Figure 13.**  
Main program algorithm called “tous\_melanges\_americains”.

### All Possible Riffle Shuffles between a p Card Subdeck and a q Card Subdeck $\mathcal{M}_0(p,q)$

To determine the number and the composition of all possible riffle shuffles  $[\mathcal{M}_1(n)]$ ,  $(n - 1) * \mathcal{M}_0(p, n - p)$  must be determined.

To determine them, thanks to the study with a few cards, we have to follow this strategy, described in **Figure 15**.

1) If the p card subdeck contains:

- More than one card, the top card is put away and the new subdeck—which contains  $(p - 1)$  cards—must be shuffled with the q card subdeck, as presented in **Figure 14**;
- Only a single card (say  $C_p$ ), this card is shuffled in the q card subdeck, as presented in **Figure 18**.

2) All the calculated riffle shuffles must be now shuffled with the card  $C_{p-1}$ . All calculated riffle shuffles must be now be shuffled with the card  $C_{p-2}$  and so on, as presented in **Figure 19**.

### The Enumeration Representation of All Riffle Shuffles between a p Card Deck and a q Card Deck $\mathcal{M}_0(p,q)$ Thanks to a Tree

All riffle shuffles between a p card deck and a q card deck thanks to a tree are presented in **Figure 16**. The aim is to calculate the tree's number having the value  $(q + 1)$  at the step p (and not before), then having the value  $(q)$ , then  $(q - 1)$ , etc. associated to the riffle shuffle number  $\mathcal{M}_0(1,i)_{0 \leq i \leq q}$ .

The first iteration of the given algorithm in **Figure 16** in order to calculate  $\mathcal{M}_0(p,q)$ :

$$\mathcal{M}_0(p,q) = \sum_{k=1}^{q+1} \mathcal{M}_k(p-1,q+1) \quad (17)$$

By using Equation (20), the following equation is obtained:

$$\mathcal{M}_0(p,q) = \sum_{k=1}^{q+1} \mathcal{M}_0(p-1,q+1-k) \quad (18)$$

In order to program this decomposition, both choices are possible:

1) To calculate step by step (vertically), as presented in **Figure 17**, for which the Matlab™ code is given in Matlab program 6 in Appendix B;

2) To calculate step by step (horizontally—the branches are calculated one after another) as presented in **Figure 23** for which the Matlab™ code is given in Matlab program 5 in Appendix B.

Whatever the chosen algorithm, we need to obtain the number and the composition of all riffle shuffles between a 1 card deck and a q card deck.

### The Obtained Riffle Shuffle from a 1 Card Subdeck and a q Card Deck: $\mathcal{M}_0(1,q)$

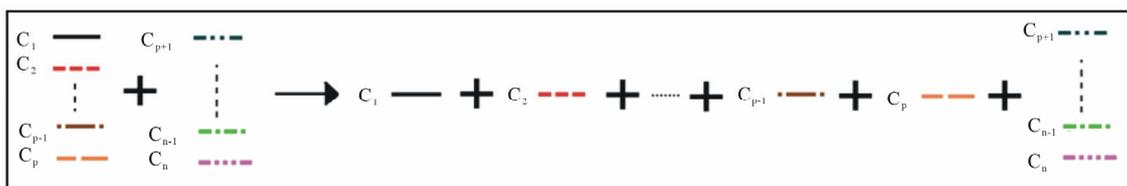
The operating mode used in order to calculate  $\mathcal{M}_0(1,q)$  is presented in **Figure 18**.

All riffle shuffles between a 1 card deck and a q card deck is  $(q + 1)$ . It generalizes Equation (4) (resp. (8)) found where  $n = 1 + q = 3$  (resp.  $n = 4$ ):

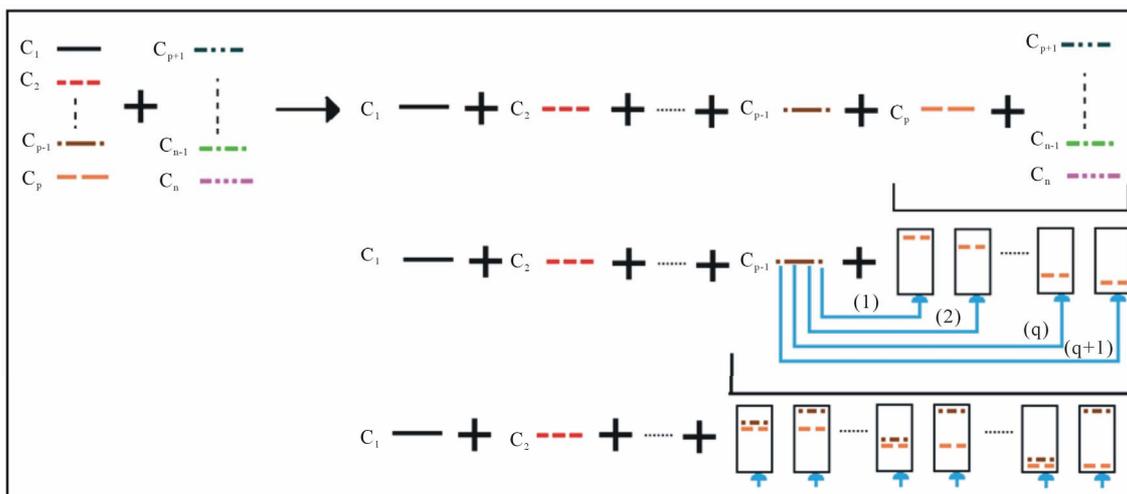
$$\mathcal{M}_0(1,q) = q + 1 \quad (19)$$

### The Obtained Riffle Shuffle from a 1 Card Subdeck and a q Card Deck over the $i^{\text{th}}$ Card Subdeck: $\mathcal{M}_i(1,q)$

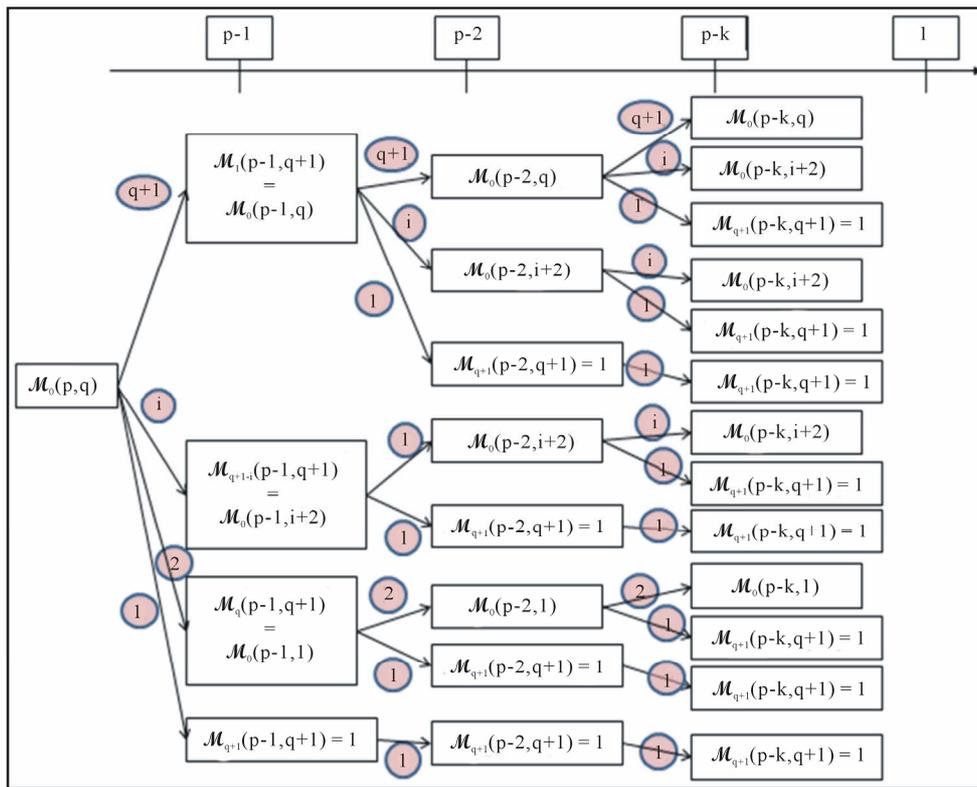
The operating mode used in order to calculate  $\mathcal{M}_i(1,q)$  is presented in **Figure 19**. Let us note that all cards must be in-



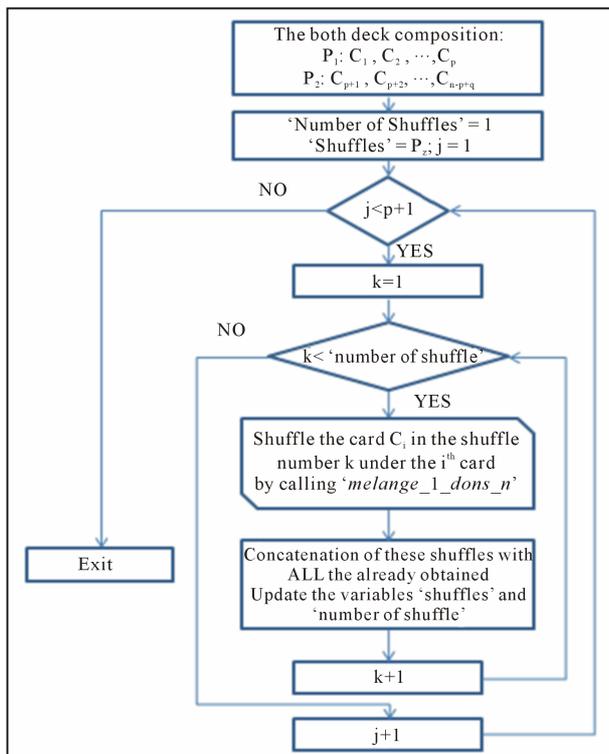
**Figure 14.** Decomposition of the p card subdeck in order to shuffle with a q card subdeck.



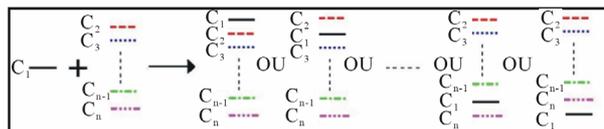
**Figure 15.** Algorithm to calculate and determine all of the compositions of possible riffle shuffles between a p card subdeck and a q card subdeck.



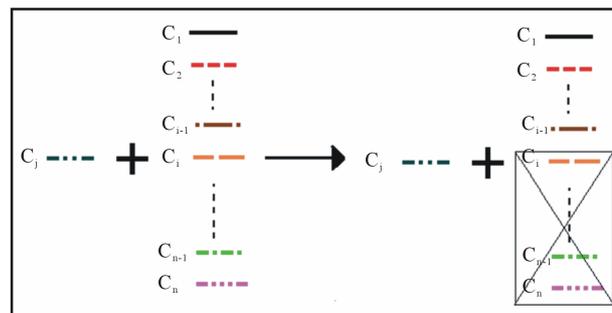
**Figure 16.** All riffle shuffles between a p card deck and a q card deck enumeration thanks to a tree.



**Figure 17.** Algorithm of the function which calculates all riffle shuffles between a p card deck with a q card deck  $\mathcal{M}_0(p, q)$ .



**Figure 18.** Modus operandi to find all riffle shuffles between a 1 card deck and a q card deck enumeration and decomposition.  $\mathcal{M}_q(p, q)$ .



**Figure 19.** Modus operandi to find all riffle shuffles between a 1 card deck and a q card deck enumeration and decomposition.  $\mathcal{M}_i(1, q) = \mathcal{M}_0(1, i-1)$ .

sorted over card  $C_i$ .

All riffle shuffles between a 1 card deck and a q card deck over the  $i^{th}$  position is worth (i). It generalizes Equation (10) found where  $n = 4$ :

$$\mathcal{M}_i(1, q) = \mathcal{M}_0(1, i-1) \tag{20}$$

Thanks to a similar approach, the number of riffle shuffles between a p card deck and a q card deck, when the first inserted card in the q card deck must be over the position 1 is:

$$\mathcal{M}_1(p,q) = p + 1 \tag{21}$$

We can define last equation where q = 0 ... which is physically impossible!

$$\mathcal{M}_0(p,0) = 1 \tag{22}$$

### From the Algorithm to a Matlab™ Program

The algorithms are found and it is necessary to program them. Thanks to each algorithm shown, one new notion is introduced:

- The tables: card deck representation;
- The “FOR” loop: to generate all of the possible cuts;
- The “IF THEN ELSE” control: to generate all of the possible riffle shuffles between a 1 card deck and a q card deck [ $\mathcal{M}_1(p,q)$ ] with the first insertion at the  $i^{\text{th}}$  card of the q card deck;
- The “WHILE” loop: to generate all of the possible riffle shuffles between a p card deck and a q card deck [ $\mathcal{M}_0(p,q)$ ].
- The recursive function (vs “WHILE” loop): to generate all of the possible riffle shuffles between a p card deck and a q card deck [ $\mathcal{M}_0(p,q)$ ].

### Card Deck Representation: Introduction to the Table

To represent the card deck, we shall allocate a value to each single card of the n card deck:

- Card value between 1 and 52;
- Card value between 101 and 113, 201 and 213, 301 and 313 and finally 401 and 413 (the hundred value represents the card family. The value between 1 and 13 represents the card value in the growing order from Ace to King)...

Thus, whatever the choice, a bijection between each single card of the deck and a number is done.

A new bicycle™ card deck is presented in **Figure 20**.

This new bicycle™ card deck in the informatics program is presented in **Figure 21** with the card representation between 1 and 52.

This new bicycle™ card deck in the informatics program is presented in **Table 1** with the card representation with hundred numbers.

The corresponding Matlab™ program is shown below.

```
n=52;
Pi(1:13)=101:113;
Pi(14:26)=Pi(1:13)+100;
Pi(27:40)=313:-1::301;
Pi(41:52)=Pi(27:40)+100;
```

Matlab™ program 1.  
Generation of the card deck.

### To Generate All of the Possible Cuts: The “FOR” Loop

The algorithm of all possible cuts generation is presented in **Figure 22**. The associated Matlab™ program, using the “for”

**Table 1.**  
Used Abbreviations for the cards values.

Card Value	Ace of Heart	2 of Heart	...	King of Heart	Ace of Club	...	King of Club	King of Diamond	...	Ace of diamond	King of Spade	...	Ace of Spade
Informatics representation	101	102	...	113	201	...	213	313	...	301	413	...	401



**Figure 20.**  
A new bicycle™ card deck.



**Figure 21.**  
A new bicycle™ card deck informatics representation with card numbered between 1 and 52.

loop, is given too.

```
n = 52;
Pi = 1:n;
for i = 1:n-1
    P1 = Pi(1:i)
    P2 = Pi(i+1:n);
end
```

Matlab program 2.  
Matlab™ program: Generation of all possible cuts of an n card deck: example of the “for” loop.

### To Generate all of the Possible Riffle Shuffles between an 1 Card Deck and a q Card Deck [ $\mathcal{M}_1(p,q)$ ] with the First Insertion at the $i^{\text{th}}$ Card of the q Card Deck: The “IF THEN ELSE” Control

The algorithm of all possible riffle shuffles (between a 1 card deck and q card deck) generation has been presented in **Figures 18 and 19**. The associated sanitized Matlab™ program is given using the “if then else” control sequence.

```
function [M,place_carte] = melange_1_dans_n(i,c1,P2)
n2 = length(P2);
if (i <= 0)
    fin = n2 + 1;
elseif (i > n2+1)
    fin = n2 + 1;
else
    fin = i;
end
for k = 1:fin
    ...
end
```

Matlab™ program 3.  
Bowdlerized Matlab™ program: generation of all possible riffle shuffles between a 1 card deck and a q card deck.

### To Generate All of the Possible Riffle Shuffles between a p Card Deck and a q Card Deck [ $\mathcal{M}_0(p,q)$ ]: The “WHILE” Loop

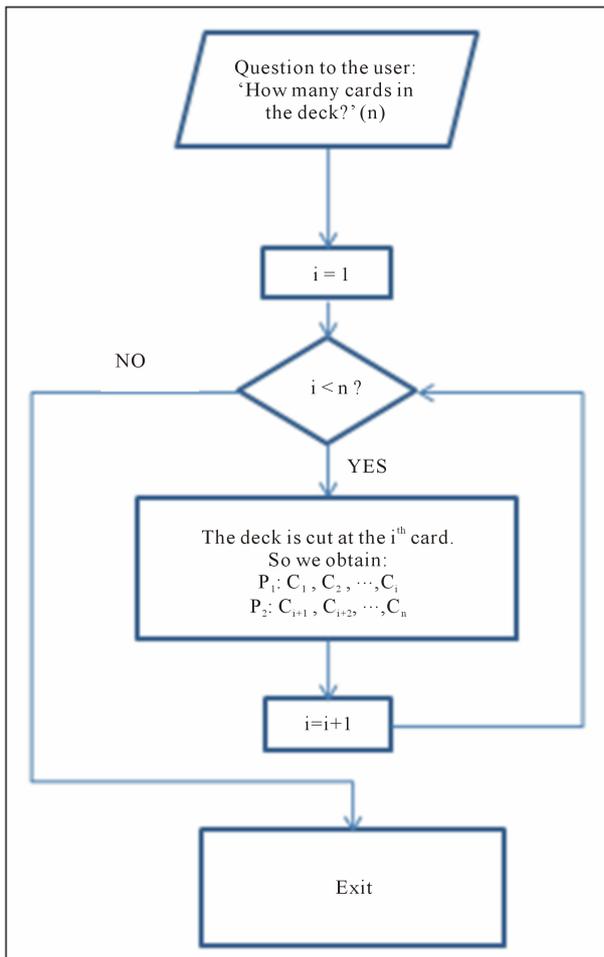
The algorithm of all possible riffle shuffles (between a p card

deck and q card deck) generation has been presented in **Figure 17**. The associated sanitized Matlab™ program is given using the “while” loop (and the “for” loop”).

```
function [M,place_carte] = melange_p_dans_n_non_recurisif(P1,P2)
n1 = length(P1);
n2 = length(P2);
nb_cartes_restantes = n1;
while (nb_cartes_restantes > 0)
...
    tab = size(Mi);
    nb_ligne_Mi = tab(1);
    for i=1:nb_ligne_Mi
        [Mii,place_carte_ii,nb_appel,nb_insertion_carte]
        =melange_1_dans_n(place_carte_i(i),carte_dessus,Mi(i,:));
        ...
    end
nb_cartes_restantes = nb_cartes_restantes - 1;
end
```

Matlab™ program 3.

Bowdlerized Matlab™ program: Generation of all possible riffle shuffle between a pa card deck and a q card deck (non recursive function): example of the “while” loop.



**Figure 22.** Algorithm: generation of all possible cuts of an n card deck: example of the “for” loop.

**To Generate All of the Possible Riffle Shuffles between a p Card Deck and a q Card Deck [ $\mathcal{M}_0(p,q)$ ]: The Recursive Function (vs “WHILE” Loop)**

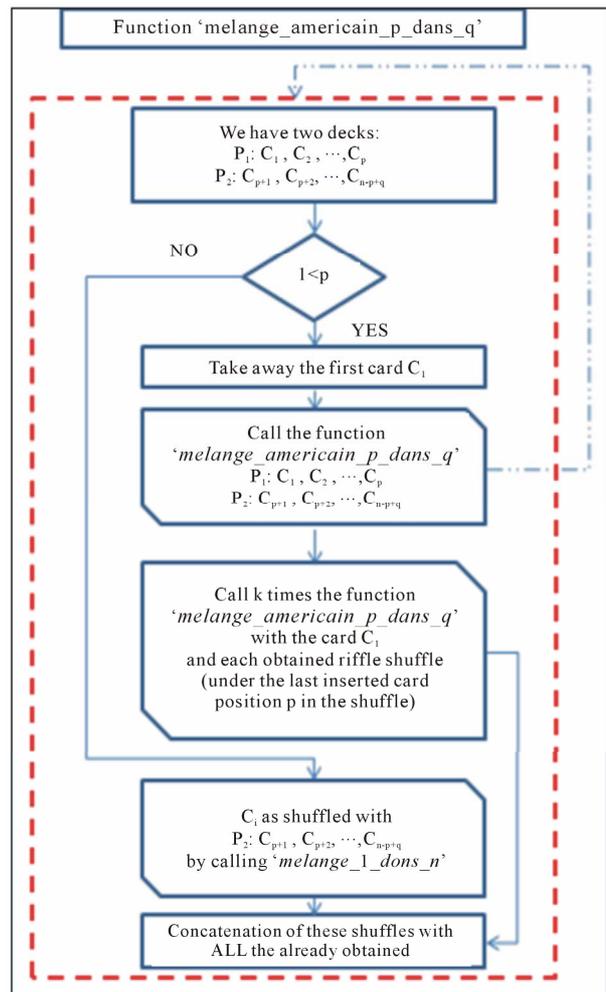
The algorithm of all possible riffle shuffles (between a p card deck and q card deck) generation has been presented in **Figure 17**. It can be programmed in two different ways; with a “while” loop or with a recursive function presented in **Figure 23**.

Indeed, at the end of the decomposition presented in **Figure 15**, only a single card is shuffled in a k card deck! Thus, the program is called as many times as possible until arriving to shuffle a 1 card deck with a q card deck, giving (q + 1) shuffles and quitting the program automatically. The second card is shuffled with each of (q + 1) shuffles and so on thanks to the recursivity!

The associated Matlab™ program is given in Appendix B in Matlab™ program 5.

**Computing Time Comparison between the Recursive Module and the WHILE Loop Module**

All the simulations were carried out with the help of a computer COMPAQ “Presario CQ62”. The main characteristics are



**Figure 23.** Algorithm of the function which calculates all riffle shuffles between a p card deck with a q card deck  $\mathcal{M}_0(p,q)$ : recursive version.

the following: the processor is a 2.2 GHz AMD with a random access memory of 2 Go.

Every computing time under a second was voluntarily interpreted as zero (for the application it's practically a real time answer). Over 20 cards, the computer has not enough memory. That's why **Figure 24** shows the computing time when the recursive module is used (full red line) and when the WHILE loop module is used (dotted blue line).

We notice that the difference between the 2 curves is not significant to be able to conclude that one method is faster than the other!

### Improvement of the Initial Algorithm

We present alternately 2 improvements of the initial recursive algorithm. The first one uses a not yet used parity property whereas the second one is only purely algorithmic by using files—which allows us to introduce the notion of file in an unusual way.

### Use of the Problem Symmetries

Let us use the break-down shown in **Figure 11**. It is easy to verify that shuffle a 1 card deck into a (q - 1) card deck is similar to shuffling a (q - 1) card deck into a 1 card deck, which can be mathematically written by:

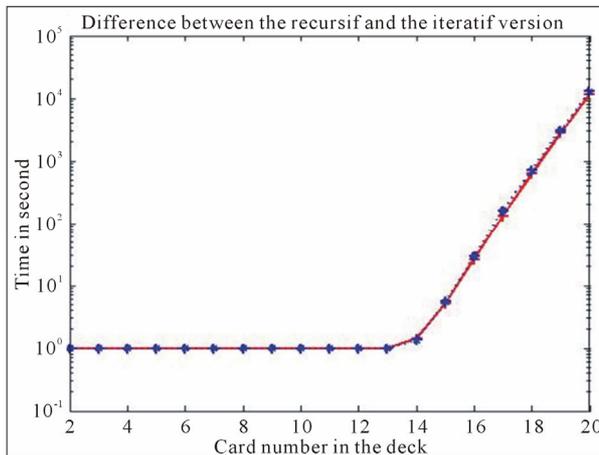
$$\mathcal{M}_0(1,q) = q + 1 = \mathcal{M}_0(q,1) \quad (23)$$

However if no improvement is added to the current algo-

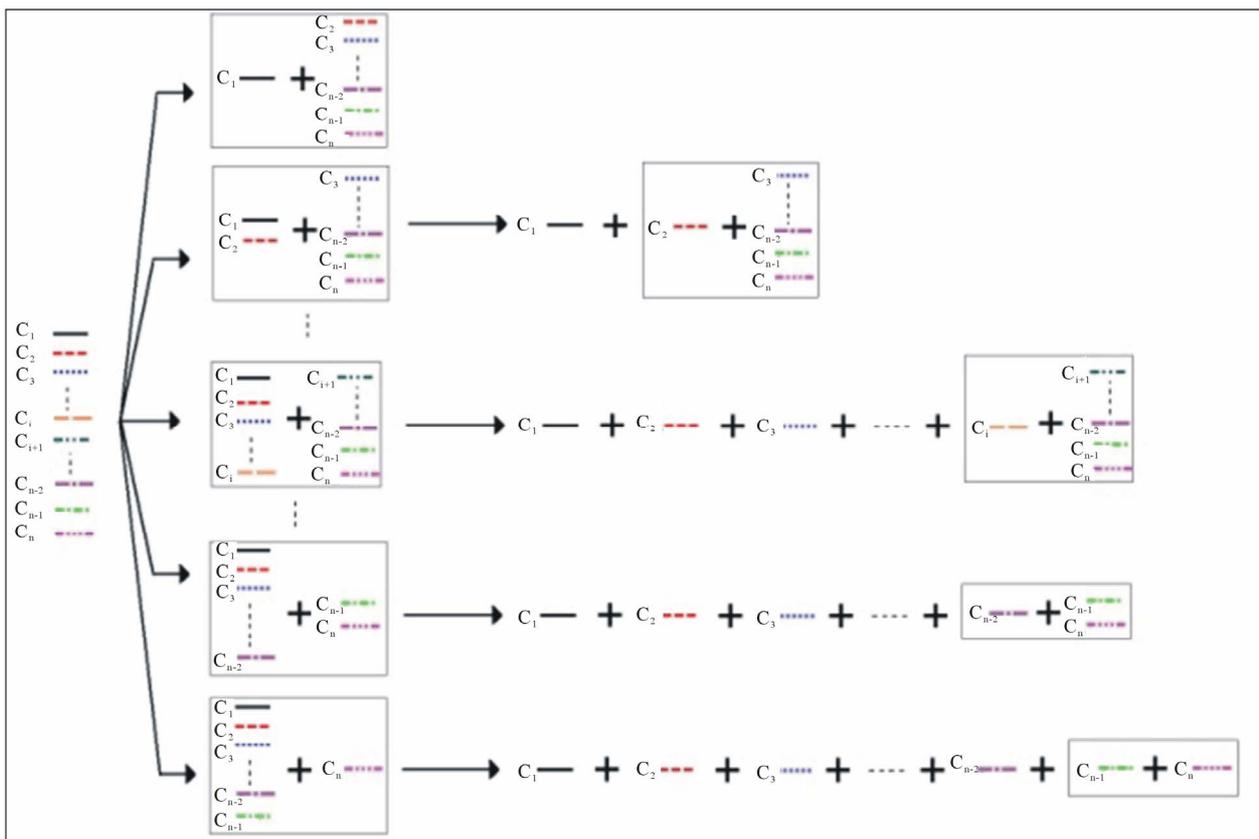
rithm, only one call to the recursive module is necessary to calculate  $\mathcal{M}_0(1,q)$ . At the same time, (q - 1) calls are necessary to calculate  $\mathcal{M}_0(q,1)$ , as presented in **Figure 25**.

In order to determine  $\mathcal{M}_0(p,q)$  the initial algorithm is still used but if  $p > q$ , the two decks are switched before, as presented in **Figure 26**.

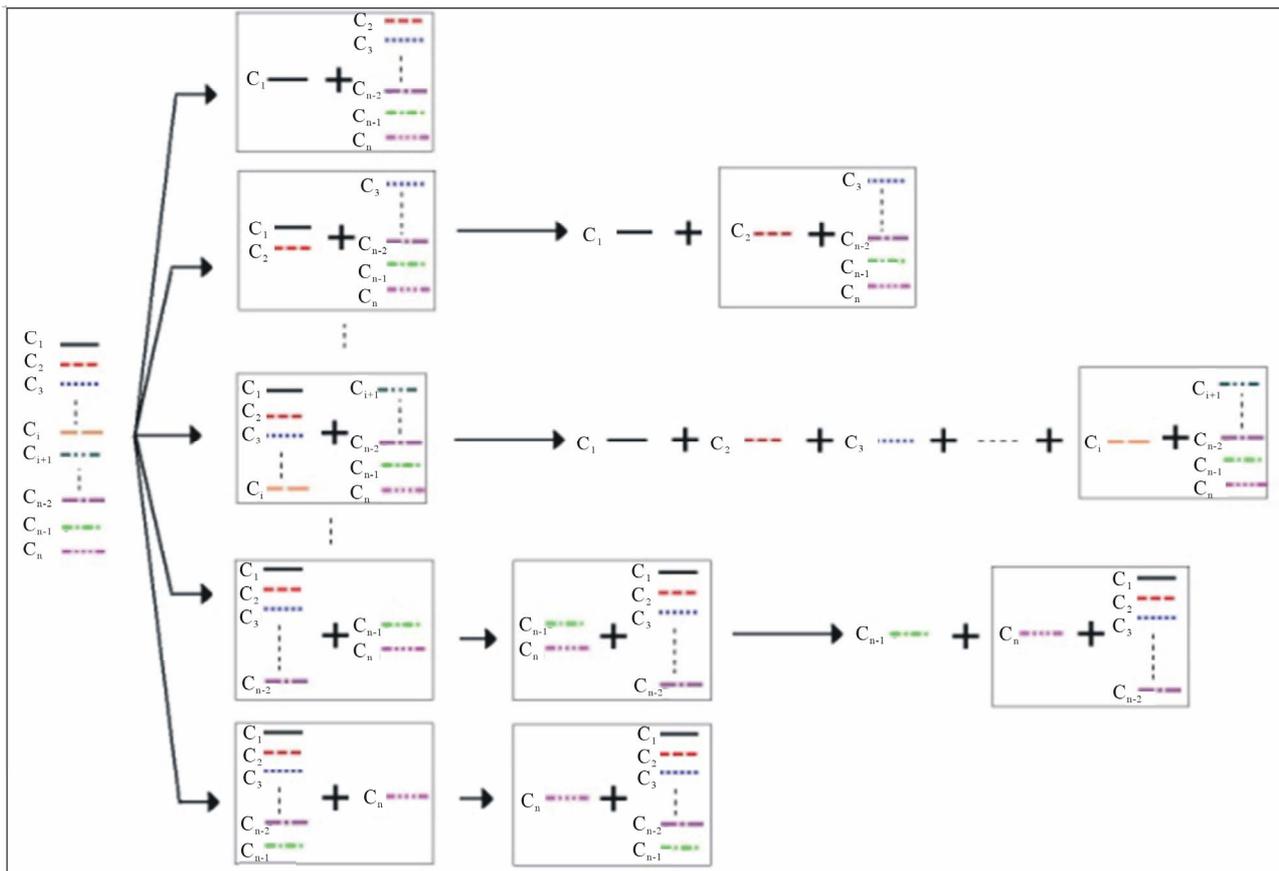
The gains are not significant. In **Figure 27**, let us note that the higher the number of cards, the more these gains decrease—



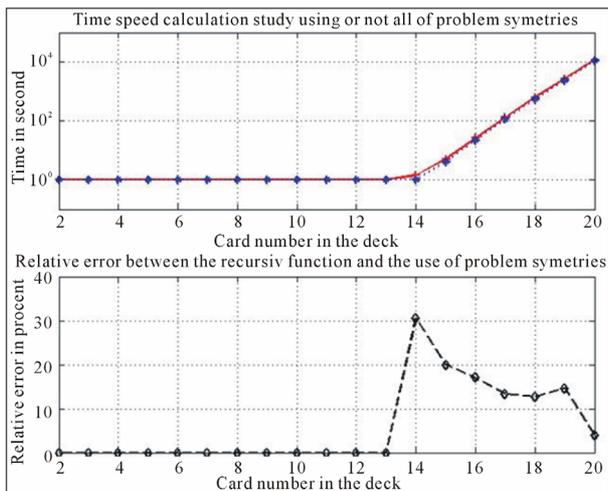
**Figure 24.** Computing time comparison between the recursive module (full red line) and the iterative module using the while loop (dotted blue line).



**Figure 25.** All possible cuts of an n card deck and initial algorithm presentation using the recursive module for each cut.



**Figure 26.** All possible cuts of an  $n$  card deck and initial algorithm presentation using the recursive module and all problem symmetries for each cut.



**Figure 27.** Computing time comparison between the initial algorithm using the recursive module (full red line) versus the implementation of the problem symmetries simplifications (dotted blue line) and relative difference (black dashed line).

which is normal because the saved calls (thanks to the problem symmetries properties) become small relative to the total calls.

So we have to find other ideas to improve significantly and durably the computing time.

**Use of Files: Presentation of the Idea**

During the use of the initial algorithm for a 3 card deck—described in **Figure 4**—we noticed that the algorithm does the same thing twice to obtain the same result! In fact shuffling a 1 card deck into a 2 card deck is similar to shuffling a 2 card deck into a 1 card deck.

During the use of the initial algorithm for a 4 card deck—described in **Figure 6**—we notice that the algorithm does the same thing twice to obtain the same result (shuffling 3 cards into 1). Furthermore, during the initial algorithm process—described in **Figure 8**—we notice that at the end a 2 cards decks must be shuffled into a 1 card deck—which is already evaluated thanks to the riffle shuffle count for a 3 card deck!

These considerations are summarized in **Figure 28**.

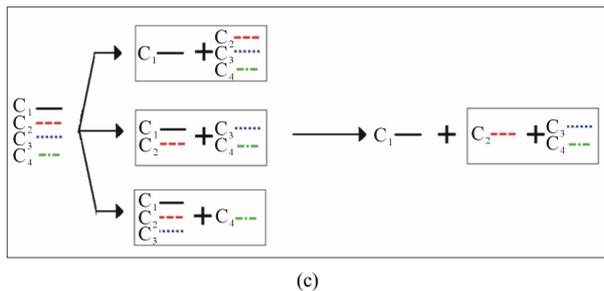
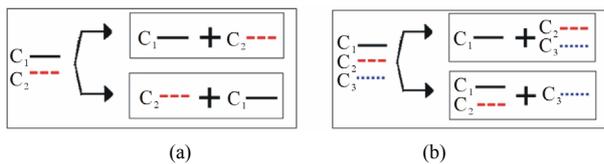
So, it's very sensible to use the previous riffle shuffle results. So, all riffle shuffle results of a  $p$  card deck into a  $q$  card deck must be saved in files. Moreover, at any time, the program has to know which shuffle results are already saved: we use a specific file whose content is similar to the one presented in the **Table 2**.

As we want to reuse the previous results, they must be independent of the card order of the chosen initial deck. That is why we decide to save the results with the agreements presented in **Figure 21**. It is necessary to create two “small” modules thanks to them the program can transform the real deck into a  $(1 \text{ to } p)$  numbered deck and its reciprocity so allowing to generate the riffle shuffle obtained with the chosen deck.

**Table 2.**

Total card number in the deck to find the  $M_0(p,q)$ . When the number is saved, the results are saved in files.

p/q	1	2	3	...	n-2	n-1
1	2	3	4	...	n-1	n
2	3	4	...	n-1	n	
3	4	...	n-1	n		
⋮	⋮	n-1	n			
n-2	n-1	n				
n-1	n					



**Figure 28.** All possible cuts of a 2, 3 and 4 cards deck and presentation of the algorithm that uses files.

**Use of Files: Results without Using the Symmetries**

The gains are not really significant when the card number increases. In **Figure 29**, let us note that the computing time is lower for the initial algorithm up to a 16 card deck, then the use of files saves only 15%. In fact, it takes a long computing time to reach the ROM in comparison of SD-RAM for example.

**Last Improvements of the Algorithm: Using the Files and the Problem Symmetries!**

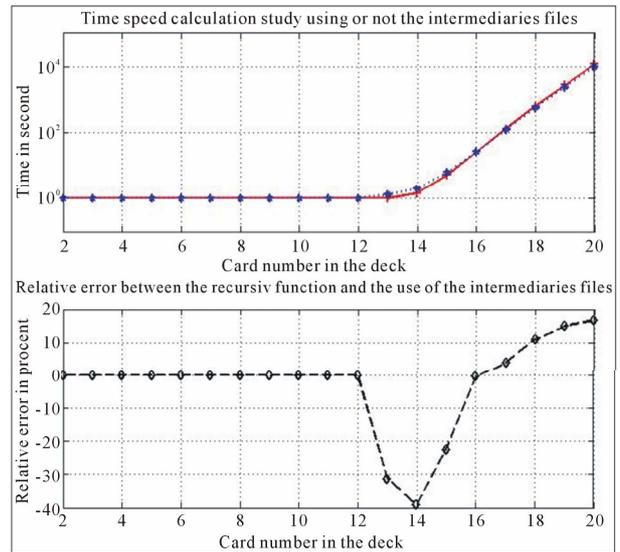
In **Figure 30**, let us note that with the both improvements, the earnings are now really significant (more than 50%).

**Synthesis, Conclusion, Going Further and Prospects**

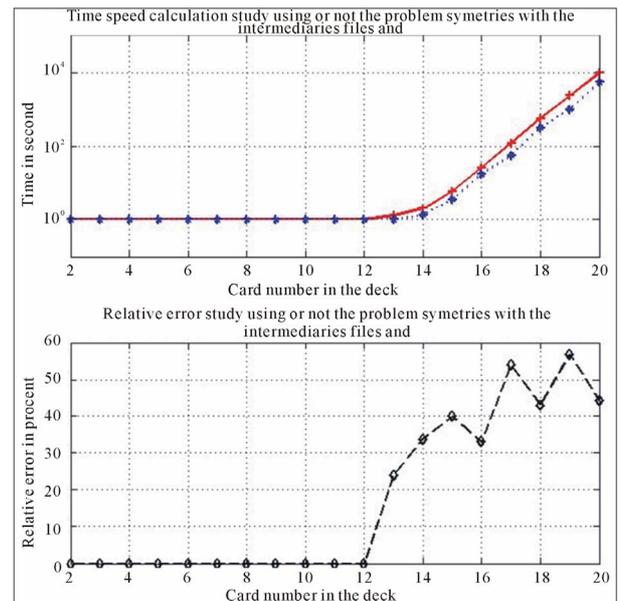
Before asking the question “is it opportune to develop this type of teaching?”, let me make a synthesis of students' work and then a conclusion in relation to my expectations. After this, I can answer the aforementioned question by suggesting ways to go further.

**Synthesis and Conclusion**

The aim of the paper was to not only show the result of a high education project, but to also give ideas for a course or using project pedagogy.



**Figure 29.** Computing time comparison between the initial algorithm using the recursive module (full red line) versus the implementation of using files (dotted blue line) and relative difference (black dashed line).



**Figure 30.** Computing time comparison between the initial algorithm using the recursive module (full red line) versus the both improvements (dotted blue line) and relative difference (black dashed line).

The students have had a 20 hour Matlab™ course before the project. Only 4 groups of 2 students have had this project. The others students have had other projects such as:

- Informatical verification of the Gilbreath’s principles;
- Create a “Sudoku” playable software;
- Determine acceleration and distance of a car from velocity data, etc.;
- No group achieved success for the entire project because of the recursively problem. All of them have achieved;
- The program to find all possible cuts;

- The program to shuffle one card in a q card deck;
- The program to find many riffle shuffles—but not all! Indeed in order to determine ONE possible riffle shuffle, the program determines randomly a card number of the first card deck P1 to go in the final shuffle, then a card number of the second deck P2 to go in the final shuffle, and so on.

It is very interesting to note that one of the groups made a Human Machine Interface under Matlab™, which was not expected (!), presented in **Figure 31** (Xuereb, 2010). They have discovered many things that exceeded my expectations...

I have shown to the students the application of magic tricks thanks to Gilbreath's principles, and these students have to work with students groups who have undertaken the Gilbreath's project in order to demonstrate Gilbreath's principles using informatics. This is why, in my project management, I reminded them repeatedly that the project was primarily scientific, but also applicable to the magic.

This is why I think that this method is efficient and practicable.

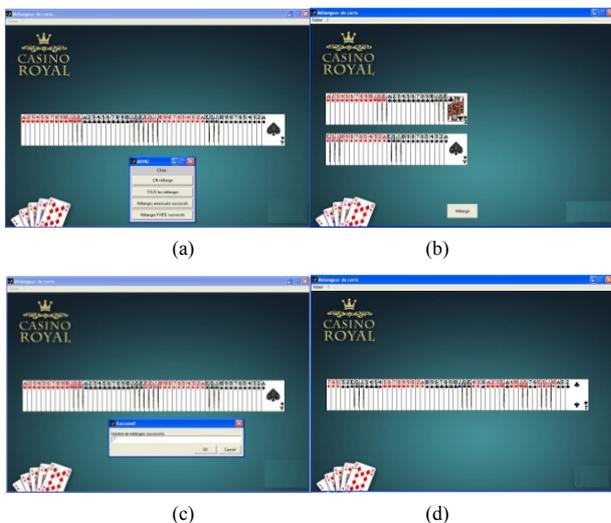
### Going Further and Prospects

This kind of project could be continued by these ideas:

- Find mathematical demonstrations of Gilbreath's principles.
- Find relations for a  $2^p$  card deck; for example answer this question: "Whose suit of IN and OUT Faro must have done to move a card from position 'j' to position 'k' in the card deck?"

The answer is given by (Lachal, 2010), but it must be very interesting to give a new project in order that the students discover and prove this property.

This leads me to think about adapting this to all mathematics teaching, from primary school to higher education (Schott, 2009) is to use magic as an investigation method to lead students to understand the mathematics principles and to come to fond of maths. I participate in the "week of the science" with primary and college classes; the results are promising.



**Figure 31.**

Card deck representation with real cards—final card deck after 7 riffle shuffles—thanks to the Matlab™ GUI interface. (a) Initial 52 new card deck; (b) 2 subdeck (cut at the 26th card); (c) Choice of shuffle numbers; (d) Initial card deck after 7 riffle shuffles.

In the past, I had proposed to two groups of students a first project using magic to discover the optics principles (Schott, 2010). The results were different: one group having focused on the realization of the magic trick, the other one on the physical principles.

That is why I continued to propose these kinds of subjects and after a more formal follow-up, the results improved. One group even suggested to me a different (magic) subject which I reformatted to fit the P.E.S imperatives.

However, the magic/science parallel must be perfect and not felt by the students to be a trick. It is for this reason that I develop at present approaches which seem to me much more adapted than the magic:

- An informatics project: using origami for teaching the C language;
- An electronics project: creation of a robot playing several musical "home made" instruments in order to allow them discover the principles of waves propagation, the programming of an embarked system, etc.

The most important thing for me remains to use my passion for the magic as a teaching vector which also weaves a human bond between the professor and the student.

### Acknowledgements

This paper could not have been written without the encouragements of Bernard SCHOTT, my father, the invaluable help of Anne SERRIE, my mother, Claire LEROUX, the head of ARNUM, Peter WILSON who corrected the English.

A special mention for Ph. Quost, a mathematical colleague of mine: thanks for your invaluable help!

### REFERENCES

- Aldous, D., & Diaconis, P. (1986). Shuffling cards and stopping times. *The American Mathematical Monthly*, 93, 333-348. doi:10.2307/2323590
- Ammar, M. (1998). *The complete cups & balls*. Tahoma, MA: L&L Publishing.
- Assaf, S., Soundararajan, K., & Diaconis, P. (2009). Riffle shuffles of a deck with repeated cards. *21st International Conference on Formal Power Series and Algebraic Combinatorics*, Hagenberg, 20-24 July 2009, 89-102.
- Diaconis, P., Graham, R. L., & Kantor, W. M. (1983). The mathematics of perfect shuffles. *Advances in Applied Mathematics*, 4, 175-196. doi:10.1016/0196-8858(83)90009-X
- Diaconis, P. (1998). From shuffling cards to walking around the building. An introduction to markov chain theory. *Proceedings of the International Congress of Mathematicians, 1*, 187-204.
- Diaconis, P. (2003). Mathematical developments from the analysis of riffle-shuffling. In A. Fuanou, & M. Liebeck, (Eds.), *Groups combinatorics and geometry* (pp.73-97). Hackensack, NJ: World Scientific.
- Erdnase, S. W. (1902). *The expert at the card table*. Mineola, NY: Dover Publication.
- Gardner, M. (1958). *Mathematics, magic and mystery*. Mineola, NY: Dover Publication.
- Gardner, M. (2005). *Martin Gardner's mathematical games: The entire collection of his scientific American columns*. Washington DC: Mathematical Association of America.
- Gilbreath, N. L. (1958). Magnetic colors. *The Linking Ring*, 3, 60.
- Gilbreath, N. L. (1966). Second Gilbreath principle. *Linking Ring*, June 1966.
- Gilbreath, N. L. (1989). Magic for an audience. *Series of 3 Articles in Genii*, 52.
- Huet, G. (1991). The Gilbreath trick: A case study in axiomatisation and proof development in the Coq Proof Assistant. *Proceedings of*

- Second Workshop on Logical Frameworks*, Edinburgh, May 1991.  
[doi:10.1017/CBO9780511569807](https://doi.org/10.1017/CBO9780511569807)
- Lachal, A., & Schott, P. (2012). Cartomagie: Principes de Gilbreath (I). *Quadrature*, 85, 24-35.
- Magid, A. (2005). *Notices*. Washington DC: American Mathematical Society.
- Mayol, H. (2000). *La magie des cordes maestro*. Gassaway, WV: HBM Production.
- Mulcahy, C. (2003). Fitch Cheney's five card trick. *Maths Horizon*, 10.
- Mulcahy, C. (2004). Top 5 reasons to like mathematical card tricks. *American Mathematical Society*, 11.
- Mulcahy, C. (2007). An ESPeriment with Cards. *American Mathematical Society*, 14.
- Poincaré, H. (1912). *Calcul des probabilités*. Paris: Gauthier-Villars.
- Schott, P. (2009). The use of magic in mathematics: From primary school to higher education. *Proceedings of ICERI2009 Conference*, Madrid, 16-18 November 2009, 58-70.
- Schott, P. (2010). The use of magic in optics in higher education. *Creative Education*, 1, 11-17.
- Schott, P. (2011). How to introduce the cyclic group and its properties representation with Matlab? Thanks to Magic using the perfect Faro shuffle. *Creative Education*, 1, 27-40.
- Sheynin, O. B. (1991). H. Poincaré's work on probability. *Archive for History of Exact Sciences*, 42.
- Xuereb, C., Clement, O., & Haddad, D. (2010). *Le mélange américain*. Projet IRI 5, Paris: ESME Sudria.

### Appendix A: Mathematical Demonstration: “How Many Different Riffle Shuffles Exist for an N Card Deck $\mathcal{M}_T(n)$ ?”

We propose the enumeration demonstration riffle shuffles between a p card deck and a q card deck  $\mathcal{M}_0(p,q)$ . Then the demonstration of  $\mathcal{M}_T(n)$  is done.

#### Calculation of $\mathcal{M}_0(p,q)$

Thanks to the study for n = 2, 3 and 4, we think that the combinatorial must be used. Let us define  $C_n^p$ :

$$C_n^p = \frac{n!}{p!(n-p)!} \quad (24)$$

Indeed, we have the following properties:

$$\mathcal{M}_0(p,q) = \mathcal{M}_0(q,p) \text{ et } C_q^{p+q} = C_p^{p+q} \quad (25)$$

Finally, Equation (19) can be written with combinatory as follows:

$$\mathcal{M}_0(1,q) = q + 1 = C_1^{q+1} \quad (26)$$

Then, this equation can be extrapolated and demonstrated by recurrence.

#### Recurrence Hypothesis

The recurrence equation is:

$$\mathcal{M}_0(p,q) = C_p^{p+q} \quad (27)$$

#### Verification of Recurrence Hypothesis q Is Fixed and p Varies

- p = 1, the hypothesis is verified thanks to Equation (26),
- p = 2,  $\mathcal{M}_0(2,q)$  must be calculated. Using the proposed algorithm, we have:

1) Firstly, the last card is put away;

2) Secondly, the number and the composition of all riffle shuffles between a 1 card deck and a q card deck  $\mathcal{M}_0(1,q)$  is worth: (q + 1);

3) Lastly the card that was put away is shuffled with the (q + 1) obtained shuffles. So:

$$\sum_{i=0}^{\mathcal{M}_0(1,q)} M_i(1,q+1) = \sum_{i=0}^{\mathcal{M}_0(1,q)} M_0(1,q+1-i) \quad (28)$$

$$\sum_{i=0}^{\mathcal{M}_0(1,q)} M_i(1,q+1) = 1 + 2 + 3 + \dots + q + (q + 1) \quad (29)$$

$$\sum_{i=0}^{\mathcal{M}_0(1,q)} M_i(1,q+1) = \frac{(q+1)(q+2)}{2} \quad (30)$$

Other hand,  $C_2^{2+q}$  is calculated and worth:

$$C_2^{2+q} = \frac{(q+2)!}{2!(q+2-2)!} = \frac{(q+1)(q+2)}{2} \quad (31)$$

By equalizing the Equations (30) and (31), the recurrence hypothesis is verified for p = 2.

#### The Recurrence Hypothesis Is Considered as True Until a Given p for a Constant Value q

We suppose that Equation (27) is true. Then Equation (18) becomes:

$$\mathcal{M}_0(p,q) = \sum_{k=0}^{q+1} \mathcal{M}_0(p-1,q+1-k) = C_p^{p+q} \quad (32)$$

#### Recurrence Hypothesis Demonstration for “p + 1” with the q Constant Value

Let us calculate  $\mathcal{M}_0(p+1,q)$ . Thanks to Equation (18), we obtain:

$$\mathcal{M}_0(p+1,q) = \sum_{k=1}^{q+1} \mathcal{M}_0(p,q+1-k)$$

$$\mathcal{M}_0(p+1,q) = \sum_{k=0}^{q+1} C_p^{p+q-k} = \sum_{k=0}^q C_p^{p+q+1-k}$$

The telescopic sum is well known:

$$\sum_{i=0}^n C_k^i = C_{k+1}^n$$

Thus:

$$\mathcal{M}_0(p+1,q) = C_{p+1}^{p+1+q} \quad (33)$$

which concludes the demonstration by recurrence.

#### Calculus of $\mathcal{M}_T(n)$

Let us take Equation (14) using Equation (27):

$$\mathcal{M}_T(n) = \sum_{p=1}^{n-1} C_p^n \quad (34)$$

### Appendix B: Informatics Matlab Programs

No representations are given here of the Matlab™ programs. To test the entire software, you have to:

- Save the given programs in 4 files. Be careful, the name of the file must be the same as the function name;
- Enter “tous\_melanges\_americaains” in the Matlab command window.

#### Main Program: “tous\_melanges\_americaains.m”

The Matlab™ program and the header of the main program (Table B1) “tous\_melanges\_americaains” are given below:

```

clear all
close all
clc
n=4;
Pi=1:n;
choix_algo = 1;
for i = 1:n-1
    P1=Pi(1:i);
    P2=Pi(i+1:n);
    if (choix_algo == 1)
        [Mi,place_carte] = melange_p_dans_n(P1,P2);
    else
        [Mi,place_carte] =
            melange_p_dans_n_non_recurisif(P1,P2)
    end
    if (i==1)
        M=M;
    else
        M=[M;Mi];
    end
    clear P1
    clear P2
end

```

Matlab™ program 4.

Main program: “tous\_melanges\_americaains”.

**Table B1.**  
Header of the Matlab™ program “tous\_melanges\_americains.m”.

Synopsis	The program generates all possible cuts with n cards and for each cut, all riffle shuffles are calculated and generated.	
Input	None	
Output	None	
Local	n	
	Choix_algo	=1 if recursif; 0
	Pi, P1, P2	Card deck
	M,Mi	Shuffles Matrix (one horizontal line = one shuffle)

**To Generate All of the Possible Riffle Shuffles between a p Card Deck and a q Card Deck [ $\mathcal{M}_0(p,q)$ ]: The Recursive Function “melange\_p\_dans\_n.m”**

The Matlab™ program and the header of the main program (Table B2) “melange\_p\_dans\_n” are given below:

```
function [M,place_carte] = melange_p_dans_n(P1,P2)
n1 = length(P1);
n2 = length(P2);
if (n1 == 1)
[M,place_carte] = melange_1_dans_n(0,P1(1,1),P2);
else
carte_dessus = P1(1);
P1reste = P1(1,2:n1);
[Mi,place_carte_p_dans_n] = melange_p_dans_n(P1reste,P2);
tab = size(Mi);
nb_ligne_Mi = tab(1);
for i=1:nb_ligne_Mi
[Mii,place_carte_ii] =
melange_1_dans_n(place_carte_p_dans_n(i),carte_dessus,Mi(i,:));
if (i == 1)
M=Mii;
place_carte = place_carte_i;
else
M=vertcat(M,Mii);
place_carte = horzcat(place_carte,place_carte_i)
end
clear Mii;
clear place_carte_i
end
end
```

Matlab™ program 5.  
Recursive function: “melange\_p\_dans\_n”.

**Table B2.**  
Header of the Matlab™ program “mealnge\_p\_dans\_n.m”.

Synopsis	The program generates all possible riffle shuffles between P1 and P2.	
Input	P1, P2	Card deck (the first card in the table is the upper one!)
Output	place_carte	Table of the last inserted card position in the shuffle.
	M	Shuffles Matrix (one horizontal line = one shuffle).
	n1,n2	Card number of deck
Local	XXXi	Intermediare variable XXX
	carte_dessus	The put away card of P1
	Nb_ligne_Mi	Shuffles number of Mi

**To Generate All of the Possible Riffle Shuffles between a p Card Deck and a q Card Deck [ $\mathcal{M}_0(p,q)$ ]: The “WHILE” Loop Function “melange\_p\_dans\_n\_nr.m”**

Only the Matlab™ program of the non-recursive “melange\_p\_dans\_n\_non\_recurisif” is given below. The header is the same as the function header “melange\_p\_dans\_n” (Table B3):

```
function [M,place_carte] = melange_p_dans_n_non_recurisif(P1,P2)
n1 = length(P1);
n2 = length(P2);

nb_cartes_restantes = n1;
nb_tour_while = 0;
while (nb_cartes_restantes > 0)
nb_tour_while = nb_tour_while + 1;
clear M;
clear place_carte;
if (nb_tour_while == 1)
[M,place_carte] = melange_1_dans_n(0,P1(1,n1),P2);
else
carte_dessus = P1(1,n1-nb_tour_while+1);
tab = size(Mi);
nb_ligne_Mi = tab(1);
for i=1:nb_ligne_Mi
[Mii,place_carte_ii] =
melange_1_dans_n(place_carte_i(i),carte_dessus,Mi(i,:));
if (i == 1)
M=Mii;
place_carte = place_carte_ii;
else
M=vertcat(M,Mii);
place_carte = horzcat(place_carte,place_carte_ii);
end
clear Mii;
clear place_carte_ii
end
end
Mi = M;
place_carte_i = place_carte;
nb_cartes_restantes = nb_cartes_restantes - 1;
end
```

Matlab™ program 6.  
Recursive function: “melange\_p\_dans\_n\_non\_recurisif”.

**Table B3.**  
Header of the Matlab™ program “mealnge\_1\_dans\_n.m”.

Synopsis	The program generates all possible riffle shuffles between the card c1 and P2 under the i <sup>th</sup> card of P2.	
Input	P2	Card deck (the first card in the table is the upper one!).
	i	Position limit in P2 for the inserted card c1.
	c1	Inserted card in P2.
Output	place_carte	Table of the last inserted card position in the shuffle.
	M	Shuffles Matrix (one horizontal line = one shuffle).
Local	n2	Card number of deck
	fin	Last possible position for the insertion of c1.

**To Generate All of the Possible Riffle Shuffles  
between an 1 Card Deck and a q Card Deck [ $\mathcal{M}_0(1,q)$ ]:  
The “if-then-else” and “FOR” Loop:  
“melange\_1\_dans\_n.m”**

```
function [M,place_carte] = melange_1_dans_n(i,c1,P2)
    n2 = length(P2);
    if (i == 0)
        fin = n2 + 1;
    elseif (i > n2+1)
        fin = n2 + 1;
    else
        fin = i;
    end
    for k=1:fin
        place_carte(k) = k;
        if (k==1)
            M(k,1) = c1;
            M(k,2:n2+1) = P2(1,:);
        elseif (k==n2+1)
            M(k,1:n2) = P2(1,:);
            M(k,n2+1) = c1;
        else
            M(k,1:k-1) = P2(1,1:k-1);
            M(k,k) = c1;
            M(k,k+1:n2+1) = P2(1,k:n2);
        end
    end
end
```

Matlab™ program 7.  
Function “melange\_1\_dans\_n”.