IMHOTEP VOL. 4, NO 1 (2003)

THE MIXTURE PROBLEM

L. P. FOTSO & J. P. NZALI

RÉSUMÉ. *Dans ce papier, nous introduisons le problème du mélange qui concerne le placement des étudiants composant simultanément plusieurs épreuves d'examens dans une ou plusieurs salles de classe. Le but visé est de séparer les étudiants composant la même épreuve de sorte qu'il y ait un maximum de sièges entre tout groupe de deux d'entre eux; ces sièges devant être utilisés autant que possible pour placer les étudiants composant d'autres épreuves. Afin d'éviter le problème associé à l'arrangement géométrique des sièges dans les salles, nous supposerons les sièges numérotés consécutivement et placés de façon linéaire. Dans le problème du mélange, nous nous intéressons à un entier non-négatif* p*, appelé degré de mélange, que nous essayerons de maximiser et qui représente le nombre de sièges entre deux étudiants composant le même examen. Pour simplifier la notation, nous introduisons un entier positif* $q = p + 1$*, qui représente la différence entre des numéros de sièges et que nous appelons "la séparation minimum". Nous appelons* m *le nombre de sièges,* s *le nombre d'étudiants,* n *le nombre d'examens et* C_i *le nombre d'étudiants composant l'examen* i*. Nous montrons d'abord que la séparation maximale est le quotient entier* $q^* = \left\lfloor \frac{m-t}{C-1} \right\rfloor$ *où* t *est le nombre d'entrées dans la suite* $C_0, \ldots, C_{n-1}$ *ayant la valeur* $C = max_{n-1 \geq i \geq 0}(C_i)$*. Si* $q^* < p+1$ *il n'y a pas de mélange de degré* p*. En supposant* $q^* \geq p+1$*, nous construisons la fonction de placement* $F : S \to M$*, qui place l'étudiant* k *sur le siège* $F(k)$*.* M *et* S *sont respectivement des ensembles de numéros de sièges et de numéros d'étudiants. L'algorithme dérivé est assez simple pour être exécuté manuellement en mixant un jeu de cartes d'une certaine façon.*

ABSTRACT. *In this paper we introduce the "mixture problem" which concerns seating students taking several different examinations simultaneously in one or more classrooms. The aim of the problem is to separate students taking the same examination with as many intervening seats as possible either left empty or occupied by students taking other examinations. To circumvent problems associated with the geometrical arrangement of seats in classrooms, we assume that seats are numbered consecutively and arranged in a line. In this mixture problem, we want to maximize a non-negative integer* p*, called the degree of mixture, which represents the number of intervening seats between two students taking the same examination. However, to simplify the notation, we use a positive number* $q = p + 1$ *to represent the difference between seat numbers. We call it "the minimum separation". We take* m *to be the number of seats,* s *the numbers of students,* n *the number of exams, and* C_i *the number of students taking exam* i*. We first show that the maximum separation that can be achieved is the integral quotient* $q* = \left\lfloor \frac{m-t}{C-1} \right\rfloor$ *where* t *is the number of entries in sequence* $C_0, \ldots, C_{n-1}$ *having the value* $C = max_{n-1 \geq i \geq 0}(C_i)$*. If* $q^* < p+1$ *there is no p-mixture. Assuming* $q^* \geq p+1$*, we construct a seating function* $F : S \to M$*, that assigns student* k *to seat* $F(k)$*.* M *and* S *are the sets of indices for seats and students respectively. We show by construction that the derived algorithm is simple enough to be executed manually by shuffling a deck of cards according to certain rules.*

1. Introduction

Where classrooms are overcrowded, students may cheat during exams by copying from their neighbors and so they are often spread into several classrooms in such a way that there is at least one empty seat between any two adjacent students taking the same exam. The examination period must thus extend over several weeks. In the worst case, it overlaps with the begining of the next semester. To avoid the "waste" of empty seats we reformulate the seating of students as the mixture problem.

Received by the editors on April 15, 2002 and in revised form on December 12, 2002.
Mots clés: Algorithme, Fonction d'affectation, Fonction de placement, Séparation maximale, Segment .
Keywords: Algorithm, Assignment function, Maximum separation, Seating function, Segment.
AMS Subject Classification 2000:

This problem concerns seating students taking several different exams simultaneously in one or more classrooms. Its aim is to separate students taking the same examination with as many seats as possible, with intervening seats empty or having students taking other examinations. To circumvent problems associated with the spatial arrangement of seats in classrooms, we make the simplifying assumption that the seats are numbered consecutively and are placed in a single line. In addition, students taking the same exam should be separated by as many other seats as possible.

In this problem, we want to maximize a nonegative integer p, called the degree of mixture, which represents the number of intervening seats between students taking the same exam. However, to simplify the notation, we shall use the positive number $q = p+1$, which is the difference between two seat numbers and we call it the minimum separation. Thus, students sitting next to each other have a separation $q = 1$.

Section 2 gives two possible formulations of the problem. Section 3 elaborates a solution to the problem. In this section, we prove the main theorem of this paper and present an algorithm to construct the seating function $F : S \to M$ that assigns to student k seat $F(k)$. We also provide a graph theory oriented algorithm which constructs a colored linear graph such that, given any two vertices of the same color, there are at least p vertices of distinct colors between them. In section 4 we show how the algorithm constructed for the seating function F can be executed manually by shuffling a deck of cards in a certain way. Section 5 is the conclusion and section 6 is the references. In the Appendices, we illustrate a version of the seating function algorithm coded in the BASIC programming language.

2. Two Possible formulations Of The Problem

2.1. A Non Graph Theory Formulation. In the mixture problem we are given the following quantities:

(i) A positive integer m, called the number of seats. The set of indices representing the seats is $M = \{0, \cdots, m-1\}$

(ii) A positive integer $s \leq m$ called the number of students. The set of indices representing the students is $S = \{0, \cdots, s-1\}$.

(iii) A positive integer $n \leq s$, called the number of exams. The set of indices representing the exams is $N = \{0, \cdots, n-1\}$.

(iv) A sequence of positive integers $C_0, \ldots, C_{n-1}$, where C_i is the number of students taking exam i. We require that $\sum_{i=0}^{n-1} C_i = s$. We number the students so that the first C_0 students take exam 0, the next C_1 students take exam 1 and so on, until the last C_{n-1} students who take exam $n-1$. We call this assignment the exam function $E : S \to N$ and it states that each student $k \in S$ takes exam $E(k) \in N$, $E(k)$ is the smallest integer such that $k \leq \sum_{i=0}^{E(k)} C_i$. In this arrangement, the same student taking more than one exam cannot take them both at the same time.

Using these definitions, the function $F : S \to M$, called the seating function, is constructed which assigns student k to seat $F(k)$. We show that this function is injective and so two or more students will never be allocated to the same seat. To make sure that students taking the same exam are separated by at least q seats, the following condition must be satisfied:

$$\bigwedge_{i,j \in S} 0 < |F(i) - F(j)| < q \Longrightarrow E(i) \neq E(j) \qquad (Seating\ Condition)$$

2.2. A Graph Theory Formulation. We construct a graph where vertices are the m available seats. To simplify the problem definition we assume that the seats are linearly arranged, with an edge between any two adjacent seats. It is a linear graph. We want to place students on seats such that two students sitting on adjacent seats do not take the same exam. This is a kind of coloring a linear graph. The number of colors to be used is the number n of the different exams taken. Let color i be attributed to

exam i and C_i be the number of students taking exam i. The seats of students taking exam i have color i.
Although the final result is a colored linear graph, our problem is more complex than the classical graph coloring problem. A coloring of an undirect graph $G = (S, A)$ where S is the set of vertices and A the set of edges is a function $C : S \to N$ such that given any pair of vertices $u, v \in S$, if $C(u) = C(v)$ then the edge (u, v) does not belong to A [Cormen, 1994].

The complexity of our problem originates from the fact that it has two new constraints and one new objective. The added objective is to maximize the number of vertices of different colors that separate two vertices of the same color. The new constraints are:

(i) The number of vertices to color with a given color i is determined by the number C_i of students writing exam i;

(ii) The number of colors used n is determined by the number of exams scheduled to be taken at the same time.

Without these added constraints and objective, the linear graph obtained could be considered as a linked list and thus contains a 2-coloration graph. i.e., we can color the list with two colors: elements with even indices are colored using one color and elements with odd indices use the other color [Cormen, 1994].

The problem, as formulated, does not always have a solution. In the following section, we find the maximum separation q^*. From the definition of the problem there is a solution if $q^* \geq 2$. When this condition is satisfied, we can elaborate two alternative algorithms for constructing a solution.

3. The Solution

Define $C = \max_{i \in N}(C_i)$, and let t be the number of entries in sequence $C_0, \ldots, C_{n-1}$ having this maximum value. Our aim is to find the maximum separation q^* that can be achieved when we are given m and the sequence $C_0, \ldots, C_{n-1}$, and have to construct the seating function $F : S \to M$, in order to achieve this separation.

Theorem 1. *The maximum separation q^* that can be achieved is the integral quotient:*

$$q^* = \left\lfloor \frac{m-t}{C-1} \right\rfloor \qquad (Maximum\ \ Separation)$$

Proof: Without loss of generality, we order the sequence $\{C_i\}$ so that $C_0 \geq \ldots \geq C_{n-1}$. From the definition of t, we see that $C_0 = C_1 = \cdots = C_{t-1} = C$, where $C = \max_{i \in N} C_i$ and if $t < n$, then $C_{t-1} > C_t$ and $C_t \geq \cdots \geq C_{n-1}$. While q^* is the quotient, let r be the remainder obtained when $m - t$ is divided by $C - 1$, so $m - t = q^*(C-1) + r$. Then

$$m = t + q^*(C-1) + r \geq \sum_{i \in N} C_i \geq tC \ , \quad so \ \ q^*(C-1) + r \geq t(C-1).$$

But, q^* and t are integers, and $0 \leq r < C - 1$ because it is the remainder when dividing by $C - 1$, giving $q^* \geq 1$. This fact is needed in the proof below.

The students taking exam 0 must span at least $(C-1)q^* + 1$ seats because adjacent pairs of students taking the same exam have separation q^* or more. The students taking exam 1 must also span at least $(C-1)q^* + 1$ seats, but since they can´t be the same seats, the span of students taking the two exams must be at least $(C-1)q^* + 2$ seats, etc. This can be achieved by interleaving, which allows the students taking exams $0, 1, \ldots, t-1$ to sit in consecutive seats, and producing what is clearly the minimum span of $(C-1)q^* + t$ seats. Since we know that $q^* \geq t$, interleaving the first t exams of size C is clearly possible using $(C-1)q^* + t$ seats, but it is not possible using fewer.

Note that if $q^* < 2$, the problem has no solution. That is, when $q^* < 2$, it is not possible to place students such that any two students writing the same exam are not on adjacent seats.

3.1. **A Non-Graph-Theory Algorithm For Constructing a solution : Algorithm F.** In the mixture problem, the decision maker specifies the degree of mixture $p(p \geq 1)$ and therefore, the minimum separation $q = p + 1$. Using the problem data, we compute the maximum separation $q^* = \left\lfloor \frac{m-t}{C-1} \right\rfloor$. If $q^* \geq q = p + 1$ there is a p-mixture. If $q^* < q = p + 1$, there is a q^*-mixture with at most $q^* - 1$ degree of mixture and the problem of p-mixture has no solution. For example, if $p = 1$ and $q^* < 2$, there is at most 0 degree of mixture, i.e. no mixture.

Here we assume $q^* = q = p + 1$ and our purpose is to define the seating function $F : M \to S$. To do this we partition the line of seats (designated by $0, 1, \ldots, m - 1$) into C segmeents. The first r segments, if any, are each of length $q^* + 1$, the next $C - r - 1$ segments are each of length q^*, and the last segment is of length t, which we know is less than or equal to q^*. Thus, the segment never increases in length as we proceed along the line.

The rules for constructing F are the following:

1. Place the first students in the first seat of the first segment. Thus $F(0) = 0$.
2. After seating any student, seat the next student in the first available seat of the next segment, if possible.
3. If no seat is available in the next segment, place the next student in the first available seat of the first segment.

Since all segments except the last have lengths that are either $q^* + 1$ or q^*, when rule 2 is followed the separation between consecutive students is large enough to allow them take the same exam. The C students taking any one of the first t exams will occupy seats in all C of the segments. These exams are therefore interleaved as described at the begining of the proof. After this the last segment will be full and have the no more seats available.

If there are additional exams $C_t, \ldots, C_n$, they will have size $C - 1$ or less. If $C_t = C - 1$, the students taking this exam will occupy seats in all $C - 1$ segments. The same will be true for any other exams of size $C - 1$. There will always be seats available in the first $C - 1$ as long as $C - 1$ or more students remain to be assigned seats so that consecutive students (whether taking the same exam or not) will either sit in consecutive segments or in the first and last segments and thus have separation of q^* or more. Since $m \leq s$, all students will be assigned seats.

It is important to note in this proof, first that students taking exam $i \in \{0, \ldots, t - 1\}$ will all occupy the ith position in each of the C segments $0, \ldots, C - 1$, and second that students taking any exams $i \geq t$ of size $C - 1$ will occupy the ith position in each of the first $C - 1$ segments $0, \ldots, C - 2$. In this way the proper separation will be maintained. However, students taking any exam i of size less than $C - 1$ may occupy positions that differ by 1 between different segments. This fact causes no difficulty since these positions are separated by at least one segment not containing exam i. The exam does not use all the first $C - 1$ segments.

This algorithm executes in order $n(m/q^*)$ time when it successfully implements F. It has been successfully implemented at the University of Yaounde I(Cameroon) in the Faculty of Arts and Letters and in the Faculty of Science.

Example 1. *We illustrate the algorithm by constructing F. Suppose $m = 11$ and $s = 10$, with $C_0 = 3$, $C_1 = 3$, $C_2 = 2$, $C_3 = 2$. Then we have $n = 4$, $C = 3$, $t = 2$, $q^* = \left\lfloor \frac{11-2}{3-1} \right\rfloor = 4$, and $r = 1$. The 11 seats are thus partitioned into 3 segments of length 5, 4 and 2 as shown in figure 1. Let x_{ij} denote student j of class i.*

Seg.1					*Seg.2*				*Seg.3*	
0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
x_{01}	x_{11}	x_{21}	x_{31}		x_{02}	x_{12}	x_{22}	x_{32}	x_{03}	x_{13}

Figure 1: *arrangement of students on different seats*

Figure 1 shows the set of 3 segments (first line of the table). Seats are numbered from 0 to 10 (second line of the table). Seg.1 has seats number 0 and 4; Seg.2 seats are numbered from 5 to 8; and Seg.3 has seats number 9 and 10. The first student of the class 0*,* x_{01}*, is placed on the first empty seat (seat number 0) of Seg.1 with respect to rule 1. Rule 2 allows us to place* x_{02} *and* x_{03} *on the first empty seats of Seg.2 and Seg.3 respectively. At this stage, all the students of class 0 are seated and all the first seats of the 3 segments are occupied. We then continue with students of class 1 by placing* x_{11} *on the second seat of the first segment which is the first empty seat of the next segment (seat number 1). Next, we place* x_{12} *on the second seat of the second segment and* x_{13} *on the second seat of the third segment. At this level, we have placed all students of class 1. The first student of the class 2,* x_{21}*, is placed on the first empty seat of the next segment, which is seat number 2 of the first segment. The second student of class 2,* x_{22}*, is placed on the first empty seat of the second segment. The first student of the class 3,* x_{31}*, should be placed on the first empty seat of the next segment but it has no more empty seats. Rule 3 is then applied to place the student on seat 3, the first empty seat of the first segment. The second student,* x_{32}*, is placed on seat 8, the first empty seat of the second segment. At this point, all students of the four classes are sited and the algorithm stop. Seat number 4 remains empty.*

Note that to have the minimum separation equals the maximum separation, we must introduce empty seats. We could eliminate these empty seats by compacting the placement so that students are shifted from the tail to the head of the list. Thus, in example 1, x_{02} *should be assigned to seat 4,* x_{12} *to seat 5,* x_{22} *to seat 6,* x_{32} *to seat 7,* x_{03} *to seat 8, and* x_{13} *to seat 9. When we compact the placement:*

$$\textit{minimum separation} = \textit{maximum separation} - 1$$

and the mixture still holds if q^* *were strictly greater than 2.*

3.2. **A Graph Theory Algorithm For Constructing a solution : Algorithm H.** Let L_i be the list of students taking exam i. Color elements of L_i with color i. Without loss of generality, suppose the L_i, $i = 0, \ldots, n-1$, be such that length $(L_i)\leq$length(L_j) for $j \geq i$; length(L_i) is a function that returns the length of list L_j. Now let L be the concatenation of the L_i starting from L_0 to L_{n-1} in decreasing order of length. Note that the length of L_0 is equal to C specified above. Given that the maximum separation $q^* = q = p+1$, we want to construct a linear colour graph where the vertices are the students, each colored with the color of the exam he is writing, and there is an edge between two students if they are sitting next to each other. Furthermore, that graph should be such that, given any two vertices of the same colour, there are q vertices of different colors that separate them.

Let Y_0 denote the set of the first C elements of L (i.e. $Y_0 = \{L(1), \ldots, L(C)\}$). Y_1 the set of the next C elements of L and so on until we cover all the elements of L. If the last set of Y_i's has less than $C-1$ elements add fictive students colored with an unused color and so make up to $C-1$ elements. The necessary condition implies that there are q such sets. Consider the vertex graph $G = (\bigcup_{0\leq i\leq n-1} Y_i, \varnothing)$ where vertices are isolated (i.e. there is no edge between any pair of elements of Y_i's) elements of the union of the Y_i's. We construct a Hamiltonian path for G such that for any given pair of vertices with the same color on the path, there are q vertices of distinct colors between them as folows:

i) From $i = 1$, draw an arc from the ith element of Y_0 to the ith element of Y_1, an arc from the ith element of Y_1 to the ith element of $Y_2, \ldots$, an arc from the ith element of Y_{q-1} to the ith element of Y_q, and an arc from the ith element of Y_q to the (i+1)th element of Y_0.

ii) From this (i+1)th element of Y_0, repeat the process with $i = i+1$ until all the vertices of G are visited.

The resuting Hamiltonian path is the linear colored graph required. Fictive students correspond to empty seats when placing the students. The placement is straightfoward when seats are considered lineary arranged:

Place the student represented by the first vertex of the Hamiltonian path on the first seat.

Place the student represented by the second vertex of the Hamiltonian path on the second seat and so on until the student represented by the second vertex of the Hamiltonian path on the second seat and so on until the student represented by the last vertex on the path is placed on the next available seat.

Although the placement function is linear with respect to s, the total number of student writing exams, this algorithm is more complex because we have to do s colorings, s/C set constructions, and at least s arc drawings to obtain the Hamiltonian path before starting the placement.

Example 2. *We reuse example 1 where* $m = 11$, $n = 4$, $C_0 = 3$, $C_1 = 3$, $C_2 = 2$, $C_3 = 2$, $t = 3$ *and* $q^* = 4$. *As indicated earlier, let* x_{ij} *represent student* j *writing exam* i *and let* a_j *represent fictive student* j. *Then*

$$L_0 = \{x_{01}, x_{02}, x_{03}\},\ L_1 = \{x_{11}, x_{12}, x_{13}\},\ L_2 = \{x_{21}, x_{22}\} \text{ and } L_3 = \{x_{31}, x_{32}\};$$

$$Y_0 = \{x_{01}, x_{02}, x_{03}\},\ Y_1 = \{x_{11}, x_{12}, x_{13}\},\ Y_2 = \{x_{21}, x_{22}, x_{31}\} \text{ and } Y_3 = \{x_{32}, a_1\}.$$

$$G = (\{x_{01}, x_{02}, x_{03}\} \cup \{x_{11}, x_{12}, x_{13}\} \cup \{x_{21}, x_{22}, x_{31}\} \cup \{x_{32}, a_1\}, \varnothing).$$

Figure 2 shows the resulting Hamiltonian path.

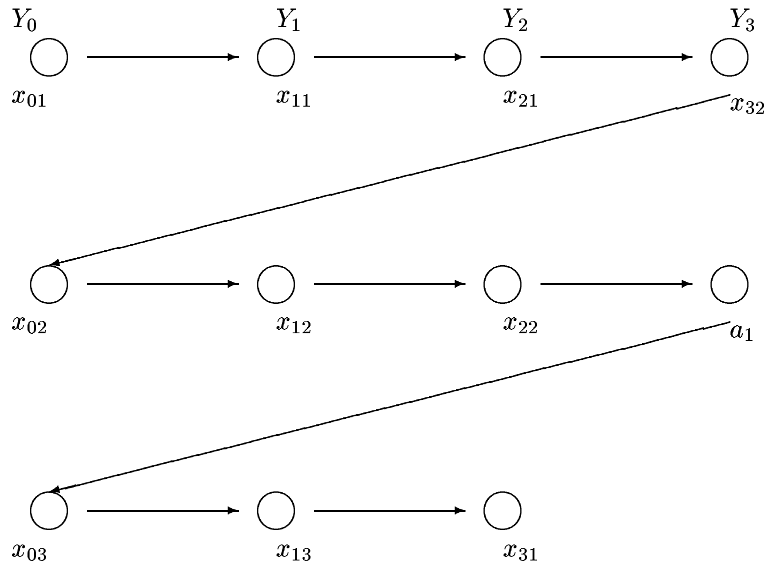

Figure 2: *Hamiltonian path of graph G*

4. Another Viewpoint

The algorithm described in the above proof is simple enough to be carried manually. One can do it by shuffling a deck of cards as described below. Again, we assume that $C_0 \geq \cdots \geq C_{n-1}$ and that $C = \max_{i \in N} C_i = C_0 = \cdots = C_{t-1}$.

1. One begins with a deck of m blank cards placed face down. The first C_0 cards have the number 0 written on their faces, the next C_1 cards have the number 1 written on them, and so on until we have similarly written numbers up to $n-1$ on the first s cards. If there are many additional cards because $m > s$, then those cards will be at the bottom of the deck and have the word "empty"

written on them.

2. The top cards (labeled 0) is removed from the deck, turned over, and placed on the table. Then the next card is removed from the deck, turned over, and placed to the right of the first card. This process is continued until C cards have been placed consecutively on the table. These are the buttom cards of C future piles, each pile corresponding to a segment as described in section 3.1.

3. We now continue removing cards from the top of the deck, turning them over and placing them on consecutive piles until all the C piles contain t cards each.

4. This process is continued using up the rest of the deck. However, cards are only placed on the first $C-1$ piles and and the rightmost pile is not added to.

5. The deck is reassembled by picking up the consecutive piles and placing them; in order, one above the next so the leftmost pile is at the buttom and the rightmost at the top. The entire deck is them turned over so that the deck of cards is face down again.

6. This deck of cards is now taken to the classroom, and the cards are removed from the top of the desk and placed on consecutive seats. The number appearing on the cards indicate which exam should be taken by the student sitting in that seat.

5. Conclusion

In this paper, we study the problem of placement of students writing several exams in one or more classrooms, in such way that seats allocated to any two students writing the same exam are separated by at least one seat. This problem differs from the classical problem of graph coloring problem by the introduction of two news constraints: (1) The number of vertices to color with a given color is imposed, (2) The number of colors to be used is equally imposed. In addition, there is a new goal: The maximization of the number of vertices with different colors that must separate two vertices of the same color.
We first show that the maximum separation q^* is the integer part of the quotient $(m-t)/(C-1)$ where m is the number of available seats, C is the maximum of C_i's where C_i is the number of student writing exam i, and t is the number of C_i's, such that $C_i = C$. Assuming the minimum separation equals q^* the maximum separation, we define a placement function F that allocates seat $F(k)$ to student k. We propose two alternative placement algorithms. The first one constructs the seating function in time $n(m/q^*)$ where n is the number of different exams. The second one is more complex and uses a kind of graph coloring and the Hamiltonian path concept to construct a linear color graph.

References

[Berge, 1973] C. Berge, *Graphs and Hypergraphs*, North Holland, Amsterdam, 1973.

[Bondy, 1971] J.A. Bondy & U.S.R. Murty, *Graph theory and Related Topics*, Academic Press, 1979.

[Brassard, 1988] G. Brassard & P. Bratleyardarin, *Algorithme, conception et analyse*, Masson, 1987.

[Chambadal, 1966] L. Chambadal & J.L. Oveart, *Cours de mathématiques*, Gauthier-Villars, Paris, 1966.

[Cormen, 1994] T. Cormen, C. Leiserson, R. Rivest, *Introduction à l'algorithmique*, Dunod, 1994.

[Fotso, 1985] L. P. Fotso, J. P. Nzali, *Statistique sur les étudiants de l'Université de Yaoundé*, année scolaire 1984-1985, CEPER, 1985.

[Golumbic, 1979] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, 1980.

[Gondran, 1979] M. Gondran, *Des algorithmes linéaires pour les problèmes de partition, de recouvrement et de couplage dans les hypergraphes d'intervalles*, RAIRO 13, 13-21, 1979.

[Gondran, 1986] M. Gondran & M. Minoux, *Graphs and Algorihms.* John Wiley & Sons, 1986

[Levy, 1994] G. Levy, *Algorithmique combinatoire, Méthodes constructives*, Dunod, 19994.

[Marshall, 1971] C. W. Marshall, *Applied Graph Theory*, Wiley-Interscience, 1971.

[Moon, 1961] J.W. MOON & L. MOSER, Hamiltonian Bipartie graphs, Israel J. Math **1**, 163-165, 1963.
[UY, 1993] *La réforme universitaire au Cameroun*, Université de Yaoundé, CEPER, 1993.
[UY, 1997] *Rapport d'évaluation de la réforme de 1993*, Université de Yaoundé I, Juin 1997.

APPENDIX 1: A BASIC implementation algorithm F: Program MIXTURE

```
3    CLS: PRINT " Mixture Program ":PRINT
4    ' Max Number of seats = 200
5    ' Max Number of exams to be written simultaneously at the same time =10
5    ' Max frequency for an exam = 100
7    ' seg(i,1) = beginning of seg i, seg(i,2)= first free seat,
8    ' seg(i,3) = last seat of this segment
9    ' eti$ indicates the label associated to each exam
10   ' Exam 1 is labeled A, exam 2 B etc...
11   ' An empty seat will be indicated by a star (*)
12   '
13   DATA *,A,B,C,D,E,F,G,H,I,J
15   DIM A(200), CL(10), SEG(100,3), ETI$(10)
16   FOR I=0 TO 10:  READ ETI$(I): NEXT I 'label setting
20   INPUT "number of seats:" ;M
30   INPUT "number of classes:" ;N:PRINT
32   s=0 ' number of students
35   FOR I=1 TO N
40   PRINT " class frequency ";I;" :";:INPUT CL(I)
45   S=S+CL(I) ' cumuli of number of students
50   NEXT I
52   IF S>M THEN PRINT " ⟹ - -> Not enough seats for all":  STOP
55   '- - Sort the frequencies of classes - -
60   FOR I=1 TO N-1
70   FOR J=I+1 TO N
80   IF CL(I)<CL(J) THEN
B=CL(I):CL(I)=CL(J):CL(J)=B:X$=ETIQ$(I):ETIQ$(I)=ETIQ$(J) :ETIQ$(J)=X$
90   NEXT J
100  NEXT I
110  C=CL(1) ' class with the highest frequency
120  T=1
130  FOR I=2 TO N
140  IF CL(I)=CL(I-1) THEN T=T+1 ELSE GOTO 160
150  NEXT I
160  Q=FIX((M-T)/(C-1))
170  IF Q<2 THEN PRNT " - -> No solution :":STOP
175  ' - - there is at least one solution - -
180  R = (M − T) − (C − 1)*Q
200  J=0 ' Number of the first seat
205  IF R=0 THEN GOTO 250
207  ' - - Initialize the heading r segments - -
210  FOR I=0 TO R-1
220  SEG(I,1)=j:SEG(I,2)=J:SEG(I,3)=J+Q
230  J=J+Q-1
240  NEXT I
245  ' - - Initialize the intermidiary segments - -
250  FOR I=R TO C-2
260  SEG(I,1)=J:SEG(I,2)=J:SEG(I,3)=J+Q-1
270  J=J+Q
280  NEXT I
285  ' - - Initialize the last segment - -
290  SEG(I,1)=J:SEG(I,2)=J:SEG(I,3)=J+T-1
300  PRINT:PRINT " value of q=:";Q
301  ' - - Print the constitued segments - -
302  PRINT" segmentation"
304  FOR I=0 TO C-1 ' c = number of segments
306  T=SEG(I,3)-SEG(I,2)+1
```

```
307 PRINT "Size of the segment N ";I;" :  ";T; " :";
310 PRINT SEG(I,1);"-";SEG(I,2);"-";SEG(I,3)
315 NEXT I
325 ' - - - - Place students in the segments - - -
330 K=0:  I=0
340 FOR J=1 TO N ' Treatment loop for the exam
350 FOR L=1 TO CL(J) ' Treatment loop for these students
360 A(I)=J 'Place the students
370 SEG(K,2)>SEG(K,3) THEN K=0 ' If it is full, skip to the first segment
400 I=SEG(K,2) ' first free seat in the segment
410 NEXT L
420 NEXT J
490 PRINT: PRINT " - - - - Result - - - -":PRINT
500 FOR I=0 TO M-1:J=A(I):PRINT ETIQ$(J)," - ";:NEXT I
```

APPENDIX 2: Sample executions of Programm MIXTURE

```
        The Mixture Programm
Number  of Seats:?  15
Number  of classes:?  3

Number  of student in class 1 :?  7
Number  of student in class 2 :?  4
Number  of student in class 3 :?  3

        Value of q=:  2
        segmentation
Size    of segment N 0 :  3 :  0 - 0 - 2
Size    of segment N 1 :  3 :  3 - 3 - 5
Size    of segment N 2 :  2 :  6 - 6 - 7
Size    of segment N 3 :  2 :  8 - 8 - 9
Size    of segment N 4 :  2 :  10 - 10 - 11
Size    of segment N 5 :  2 :  12 - 12 - 13
Size    of segment N 6 :  1 :  14 - 14 - 14
        - - - Result - - -
A-B-C-A-B-*-A-B-A-B-A-C-A-C-A
Ok
```

Figure 3: *Students of class 1, 2 and 3 are colored with the letter A, B and C respectively. The star symbol '*' represents an empty seat*

```
        The Mixture Programm
Number  of seats:?  20
Number  of classes:?  2

Number  of student in class 1 :?  15
Number  of student in class 2 :?  4
        - - - No solution :
Break   in 170
Ok
```

Figure 4: *This example illustrates the case where the mixture is not feasible.*

```
        The Mixture Programm
Number  of seats:?  30
Number  of classes:?  4

Number  of student in class 1 :?  21
Number  of student in class 2 :?  15
Number  of student in class 2 :?  13
Number  of student in class 2 :?  7
        - - -> Not enough seats for all
Break   in 52
Ok
```

Figure 5: *This last example shows the situation where there are no enough seats to place all the students.*

Computer Science Department, Faculty of Science, BP.812 Yaoundé, Cameroun
E-mail address: lpfotso@uycdc.uninet.cm, jpnzali@uycdc.uninet.cm