Principe Algorithme PERCO.

Son analyse fut faite chez IBM en 1962 pour apporter une solution au problème du Calendrier d'examens, (dossier_LR) reporté ici.

N candidats se présentent à un oral de concours. Ils doivent passer successivement **p** épreuves, mais pas plus d'une par jour. L'oral dure **n** jours : (n= > p). Le professeur **P** peut faire passer le jour **j** un maximum de **Cpj** examens. Un professeur représente une matière.

Ici on prend **n** = 7 (L, M, M, J, V, L, M) pour **P** = 6.

Il faut établir la calendrier des passages d'examens de chaque candidat.

P/j	L	M	M	J	V	L	M
p1	20	19		10		11	
p2	18		18		15		17
p3		23			13	17	20
p4	15		22		18	5	18
p5	8	22		30		11	
p6	8	12	22		18		20

Contraintes de faisabilité, il faut que :

$\sum \mathbf{C_{pj}}$ / **p** >> **N.** 430 / 6 => 71 >> 60.

$\sum \mathbf{C_{p}}$ pour au moins **p** jours => **N**

L'objectif est alors d'obtenir un algorithme qui fasse cela pour chaque candidat. Son numéro d'ordre doit apparaître une seule fois par jour et par professeur, et il doit avoir connu les 6 professeurs sur les 7 jours alloués pour se présenter.

L'algorithme reçoit les données initiales du problème et restitue une liste de la forme $\mathbf{X_{jp}}$

Pour **X** est le numéro de candidat, ici de 1 à 60

j le jour où il doit se présenter

p le professeur avec lequel il doit passer.

Dans l'exemple retenu on a constitué 3 tranches de 20 candidats :

Première tranche : 1-20

Deuxième tranche : 21-40

Troisième tranche : 41-60

Après chaque tranche reste en rouge ce qui n'a pas été utilisé.

P / j	L	M	M	J	V	L	M
p1	1:20	19		10		11	
p2	18		18		1:15		16:20 12
p3		1:20 3			13	17	20
p4	15		1:20 2		18	5	18
p5	8	22		1:20 10		11	
p6	8	12	22		16:20 13		1:15 5

Après première tranche **1 : 20**

P / j	L	M	M	J	V	L	M
p1		21:38 1		39:40 8		11	
p2	21:38		39:40 16				12
p3		39:40 1			13	17	21:38 2
p4	15		2		21:38	5	39:40 16
p5	8	22		21:30		31:40 1	
p6	8	12	21:38 4		39:40 11		5

Après deuxième tranche **21 : 40**

P / j	L	M	M	J	V	L	M
p1		49		41:48		50:60	
p2			41:56				57:60 8
p3		1			48:60	41:47 10	2
p4	57:60 11		2			5	41:56
p5	41:48	50:60 11				49	
p6	49:56	41:48 4	57:60		11		5

Après troisième tranche **41 : 60**

Observations. Ce problème n'a pas de solution si les contraintes initiales ne sont pas respectées, et pas respectées par les professeurs avant tout !

Mais contraintes respectées, l'algorithme peut se borner à devoir numéroter les N candidats de 1 : N, et les distribuer, comme des jetons, sur des cases non vides qui peuvent les accepter.

Faire cela soi-même sur une feuille de papier, comme il a été procédé ci-dessus, s'avère plus amusant que d'écrire un programme pour un ordinateur !

Abandon du projet. Il me parut évident dès 1962, que ce problème cachait quelque chose.

Prenons un écoulement fait d'un fluide de débit **N**, et une grille d'écoulement constituée d'ouvertures **Cpj**. Les **Cpj** sont disposés comme sur le tableau initial ci-dessus, et avec les valeurs et contraintes indiquées : elles donnent le débit acceptable au niveau de cette ouverture.

Le problème devient alors :

- comment découper le débit **N,** pour obtenir des petits "ruisseaux",

- quelles "trajectoires" leur imposer, pour que l'ensemble obtenu, passe à travers la grille, sans engorgements, sans coups de béliers.

Ce devenait un problème de diffusion de gaz ou de fluides !

Je testais l'algorithme sur ordinateur, pour m'assurer que cela fonctionnait, qu'il restituait la bonne liste $\mathbf{X}_{\mathbf{jp},}$ et je ne diffusais jamais la solution.

Je le testais avec X = 100 pour diverses grilles de Cpj.