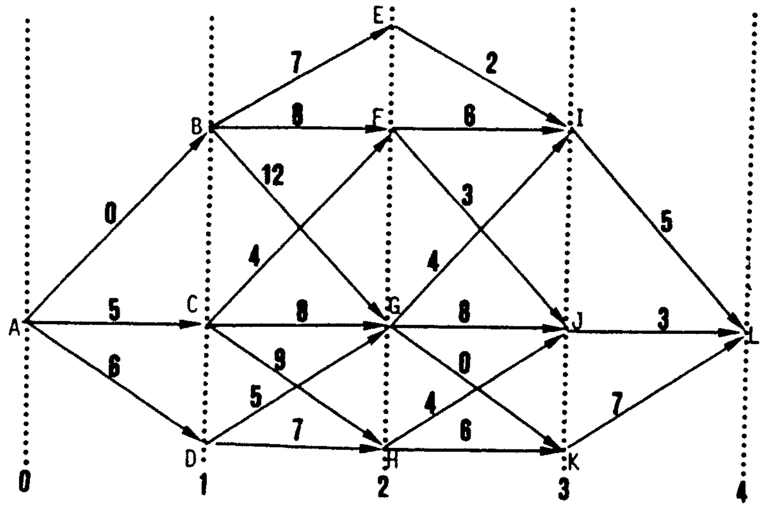
(•) « ET J'ENTENDS SIFFLER LE TRAIN »

(introduction simple à la programmation dynamique).

Une voie de chemin de fer doit être construite entre deux villes : A et L. Les villes A et B, G et K sont déjà reliées et si l'on emprunte un de ces trajets, le coût de construction est **négligeable**.

Trouver les villes intermédiaires telles que le coût total de construction soit minimal. On utilisera une méthode de programmation dynamique.



On notera $v_i(x_i, x_{i+1})$ le coût de construction d'un tronçon de voie ferrée entre une ville x_i du niveau i et une ville x_{i+1} du niveau i + 1. On remarque que les sommets sont rangés en niveaux et que les arcs ont tous leurs extrémités sur des niveaux d'indices consécutifs.

(•) UNE HISTOIRE QUI NE MANQUE PAS DE SEL

(application de l'algorithme de Bellman-Kalaba).

En l'an de grâce 1479, le sire Gwendal, paludier à Guérande (ville notée 0), dési **aller** vendre sa récolte de sel à l'une des grandes foire du Duché. Il connaît les **gains** qu'il pourra réaliser dans chacune des foires, mais ceux-ci seront diminués **des** octrois qu'il devra acquitter le long du chemin emprunté pour s'y rendre. A **quelle** foire et par quel chemin le paludier doit-il se rendre de façon à réaliser **le** plus grand bénéfice possible.

foires	Rennes (11)	Loudéac (12)	Pontivy (13)	Lorient (14)
gains (en écus)	550	580	590	600

Fig. 1. Tableau des gains (en écus) dans les différentes foires.

villes	1	2	3	4	5	6	7	8	9	10	11	12	13	14
octrois (en écus)	10	12	15	5	15	10	3	10	5	20	4	5	20	7

Fig. 2. Tableau des octrois (en écus).

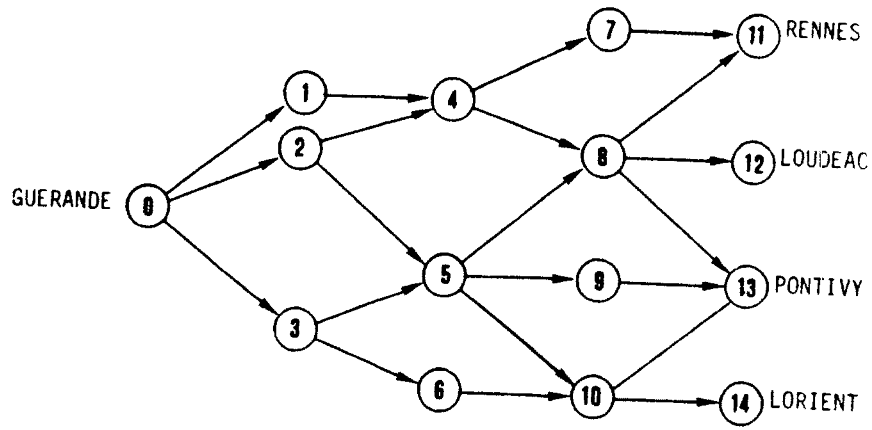


Fig. 3. Graphe des chemins possibles du domicile du paludier aux différentes foires.

(*) L'ARGENT DE POCHE

(Graphe de décisions, formules d'optimisation séquentielle).

Un étudiant désire travailler en dehors de ses heures de cours pour gagner de l'argent de poche. Afin de ne pas compromettre ses études, il a décidé de consacrer un maximum de T heures par semaine à ses activités rémunératrices. Après de minutieuses recherches, il a trouvé n emplois possibles.

Les salaires qui lui sont offerts ne sont pas proportionnels aux nombres d'heures de travail, et sont rassemblés dans le tableau ci-dessous (T et n sont pris égaux à quatre).

heures de travail: x \ salaire : g (francs)	emploi I	emploi II	emploi III	emploi IV
0	0	0	0	0
1	26	23	16	19
2	39	36	32	36
3	48	44	48	47
4	54	49	64	56

1. Pouvez-vous aider cet étudiant à déterminer à quels emplois il doit se consacrer et pour quelle durée (hebdomadaire) afin de maximiser ses rémunérations ?
2. En période d'examen son temps disponible peut être réduit à trois, ou deux, ou une ou zéro heures. Résoudre alors la même question qu'au 1.
3. On formalisera la démarche qui permet de résoudre ce problème.
4. Peut-on résoudre le problème, directement, à l'aide d'un raisonnement économique marginaliste ?

(**) LE PROBLEME DU SAC A DOS ("Knapsack").

Un randonneur, partant pour une longue excursion, détermine avec soin le contenu d[e] son sac à dos. Compte tenu des équipements indispensables déjà chargés, le poids total de nourriture emportée ne devra pas excéder 16 kilogrammes. Il dispose, en quantités limitées, de trois types d'aliments, de valeur nutritive variable, dont les poids unitaires sont différents ; les aliments sont conditionnés par unités no[n] fractionnables.

Aliments	I	II	III
poids unitaire (en kg)	7	5	2
quantités disponibles	4	3	4
valeurs nutritives	15	10	4

Le randonneur cherche la quantité de chaque aliment à emporter, de façon à maximise[r] la valeur nutritive totale, tout en tenant compte de la limite de 16 Kg qu'il s'est fixée.

Il désire ... utiliser la programmation dynamique pour résoudre ce problème dit du "knapsack". D'une manière générale, soient :

n, le nombre d'aliments différents ;
p_i, le poids unitaire de l'aliment i $(1 \leqslant i \leqslant n)$;
q_i, le nombre d'unités disponibles de l'aliment i ;
c_i, sa valeur nutritive ;
x_i, la quantité de l'aliment i emportée (en nombre d'unités) ;
et P, le poids maximal de nourriture qu'il est possible d'emporter.

1. Donner une formalisation du problème en tant que programme linéaire en nombres entiers.
2. Tracer le graphe des décisions associé au problème. Valuer ses arcs.
3. Noter $f_i(Q)$ la valeur nutritive maximale du sac à dos chargé (optimalement) avec uniquement les i premiers produits, et de poids total Q $(Q \leqslant P)$.
Donner les formules d'optimisation séquentielle.
4. Résoudre numériquement.