Programmation linéaire – Problème d'affectation
Algorithme De Kuhn-Munkres (dit algorithme Hongrois)

I. Préambule

Contexte :

- Il s'agit d'un problème binaire.

Les contraintes :

- autant de ressources que de besoins.

L'objectif :

- Maximiser l'ensemble des préférences (ou performance, indice qualité)
- Minimiser l'ensemble des regrets (ou coûts)

Problème identique à celui des "8 reines" qui consiste à placer 8 reines sur un échiquier sans que l'on trouve 2 reines sur la même colonne ou la même ligne.

Exercice fil rouge :

1. Tableau de préférences

Partons du tableau suivant qui peut s'interpréter comme un tableau de préférences pour l'affectation de 3 étudiants à 3 universités

Quelle affectation réaliser pour Maximiser la somme des préférences fournies ci-dessous ?

	U1	U2	U3
E1	0	2	3
E2	3	0	0
E3	3	1	1

2. Initialisation du tableau

Penser à bien initialiser le problème, afin d'obtenir une matrice prête à appliquer la minimisation indispensable pour appliquer l'algorithme hongrois.

Le tableau initial doit donc toujours être un tableau de coûts.

Ainsi, si le problème est un problème de minimisation, vous n'aurez pas à passer par cette phase 2 d'initialisation

Par contre, dans le cas d'une problématique de maximisation, "inversez" le problème (fournissez donc son dual) en soustrayant à chaque coefficient présent dans la fonction à optimiser, le coefficient maximum. Ainsi on obtiendra à la place

- D'une maximisation des préférences, satisfaction, bénéfices
- Une minimisation des regrets, coûts

Le tableau des préférences est donc transformé en tableau des regrets à l'aide de la formule suivante :

$$a'_{ij} = \max(a_{ij}) - a_{ij}$$

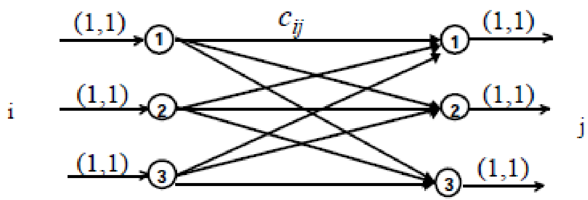
Exercice fil rouge :

$\max(a_{ij}) =$

D'où, le tableau des regrets est le suivant :

	U1	U2	U3
E1			
E2			
E3			

II. Modélisation d'un problème d'affectation en PL :



$$\begin{array}{ll} \multicolumn{2}{c}{\text{Min} \quad \sum\sum c_{ij}x_{ij}} \\ \sum x_{ij} = 1 & i = 1, \ldots, n \\ \sum x_{ij} = 1 & j = 1, \ldots, n \\ \multicolumn{2}{c}{x_{ij} \geq 0} \end{array}$$

Application à l'exercice fil rouge : (applicable sur le Solveur)

(i lignes et j colonnes).

Min $3x_{11}+1x_{12}+0x_{13}+0x_{21}+3x_{22}+3x_{23}+0x_{31}+2x_{32}+2x_{33}$

Sous les contraintes :

$x_{11}+x_{12}+x_{13}=1$

$x_{21}+x_{22}+x_{23}=1$

$x_{31}+x_{32}+x_{33}=1$

$x_{11}+x_{21}+x_{31}=1$

$x_{12}+x_{22}+x_{32}=1$

$x_{13}+x_{23}+x_{33}=1$

avec $\forall i, \forall j, x_{ij} \in \{0;1\}$

Remarque : il s'agit bien d'un graphe (voir théorie des graphes)

III. Algorithme Hongrois : le principe

Initialisation : table des regrets
Chaque itération comprend les phases suivantes (ou devrait-on dire les algorithmes suivants) :

- Préparation
- Test de faisabilité d'affectation
  Si affectation possible :
  - Alors Affectation
  - Sinon Mise à jour du tableau

  Fin Si

Tant que l'affectation n'est pas complète
Finalisation : Retour au problème initialement demandé

1. Préparation du tableau : Réduction (un 0 par ligne et par colonne)

Objectif : L'idée de la préparation est de faire apparaître au moins un 0 par ligne et par colonne (nouveau tableau) :

Pratiquement : Pour cela : à tous les éléments d'une même colonne, enlevez le plus petit élément de la colonne. Faites de même pour chaque ligne.

Plus petit : Plus petit :

	U1	U2	U3
E1			
E2			
E3			

Exercice fil rouge : une seule réduction est nécessaire

2. Test de faisabilité d'affectation

Objectif : Rayer tous les 0 (nouveau tableau)

Pratiquement : Rayez les lignes et colonnes afin de couvrir TOUS les 0 du tableau

Les contraintes :

- Utilisez le moins de traits pour cela.
- Un trait doit couvrir le maximum de 0

Remarque importante : Là est votre choix. Un algorithme supplémentaire pourrait probablement améliorer cette phase. Lorsque vous aurez bien compris les incidences de l'algorithme Hongrois, revenez à cette phase, et pensez à une amélioration

Nb de 0 :

Nb de 0 :

	U1	U2	U3
E1			
E2			
E3			

Exercice fil rouge : 2 traits suffisent

Pour comprendre ce qui va suivre :
A l'issue de cette phase de rayure des 0:

Si le nombre de traits (horizontaux et verticaux) est supérieur ou égal au nombre de lignes/colonnes,
Alors vous avez fini : vous savez ainsi que vous pouvez affecter toutes vos ressources sur les besoins. Passez au b)
Sinon, vous devrez forcément continuer avec la mise à jour du tableau.
Finsi.

3. Affectation : si affectation possible d'aprés 2)

Objectif : Procéder à la totalité des affectations (affectation complète) (nouveau ableau)

- Les affectations possibles se font sur l'emplacement des 0
- Amélioration proposée pour cette affectation :
  Tant que Affectation incomplète
  i. Commencer par les cellules intersection des colonnes/lignes qui présentent le moins de 0 non affectés et/ou non rayé (les choix restants, et les plus contraignants d'abord),
  ii. Rayez la colonne ET la ligne de la cellule que vous venez d'affecter
  iii. Mettez à jour les sommes de 0 non rayés par ligne/colonne
  iv. Réitérez le protocole tant que l'affectation n'est pas complète
  Fin tant que

Exercice fil rouge : vous voyez bien que l'affectation n'est pas complète :

Soit E3 avec U1, Soit E2 avec U1.
et, Soit E1 avec U3, Soit E1 avec U2.

Donc cette tentative d'affectation n'était pas nécessaire

Nb de 0 non rayés:

Nb de 0 non rayés:

	U1	U2	U3
E1			
E2			
E3			

4. Mise à jour du tableau – si non affectation possible d'après 2)

Objectif : Augmenter l'écart d'utilité de chaque case (nouveau tableau)

Pratiquement :

- Sélectionner le coefficient positif (>0) le plus faible du tableau, en dehors des traits
- Ajoutez ce nombre à toutes les intersections
- Enlevez ce nombre aux valeurs qui ne sont pas couvertes par un trait.
- Nous ne touchons pas aux valeurs restantes (couvertes par un trait MAIS sans être une intersection)

	U1	U2	U3
E1			
E2			
E3			

Exercice fil rouge: la valeur la plus faible est 1

Itération suivante :

1) Préparation :

Plus petit :

	U1	U2	U3
E1			
E2			
E3			

Plus petit :

2) Test de faisabilité d'affectation :

Nb de 0 :

	U1	U2	U3
E1			
E2			
E3			

Nb de 0 :

3) Si affectation possible, alors Affectation :

Nb de 0 non rayés:

	U1	U2	U3
E1			
E2			
E3			

Nb de 0 non rayés:

4) Si affectation impossible, alors Mise à jour :

	U1	U2	U3
E1			
E2			
E3			

N'oubliez pas :

Si le nombre de traits est égal au nombre de lignes/colonnes, alors savez que vous pouvez affecter toutes vos ressources sur les besoins.

5. Finalisation (répondez à la demande)

Objectif : N'oubliez pas de revenir au tableau initial, avant modification d'initialisation (nouveau tableau) et donnez le résultat numérique !

Pratiquement : Retour au tableau initial, repassez éventuellement en terme préférences ou restez en termes de regrets.

Calculez la valeur ainsi obtenue pour la fonction objectif (Maximisation des préférences ou Minimisation des regrets)

Exercice fil rouge :

Exercice 2 : Organisation d'atelier

Dans une entreprise, on a un atelier de 5 ouvriers P1, P2, P3, P4 et P5 qui doivent se répartir 5 tâches T1, T2, T3, T4 et T5.

Le critère de polyvalence est établi à l'aide de 3 niveaux de compétence OP1 OP2 et OP3, sachant qu'OP3 (niveau pro de niveau 3) est le plus performant.

Ci-dessous le tableau de polyvalence des ouvriers :

	T1	T2	T3	T4	T5
P1	0	3	2	0	2
P2	2	3	0	1	1
P3	1	1	2	3	1
P4	0	3	2	2	1
P5	1	2	3	0	0

Exercice 3 : Le bon sous-traitant

Lors de l'établissement du budget des achats, une entreprise se penche plus précisément sur les achats de pièces sous-traitées que nous noterons P1, P2, P3, P4 et P5.

Elle décide de comparer les prestations de 5 fournisseurs F1, F2, F3, F4 et F5.

Chacun des sous-traitants n'accepte d'assembler qu'un seul type de pièces pendant l'année pour des raisons diverse : gestion de charge de travail, gestion d'atelier, capacité de l'outil de production, volonté d'indépendance commerciale…

Une étude a permis de définir, à partir de l'expérience antérieure, un indice de satisfaction pour chaque produit et chaque fournisseur.

Cet indice compris entre 0 et 12 est donné dans le tableau suivant :

	F1	F2	F3	F4	F5
P1	0	6	12	12	6
P2	0	4	10	12	7
P3	4	5	6	6	12
P4	12	12	6	0	4
P5	3	10	12	5	0

6. Cas atypiques :

Tout cas atypique modélise une contrainte supplémentaire, non prévue par l'algorithme général

Il faut donc créer une « **perturbation** » dans l'énoncé, afin de pouvoir intégrer le cas atypique, non sans avoir préalablement ajouté un module au sein de l'algorithme général (souvent intégré dans un bloc conditionnel : Si-alors-sinon-finsi).

6.1 cas d'une affectation impossible

Contrainte : tel employé ayant une incompatibilité pour un poste

Perturbation créée : on pourra affecter la cellule d'un coefficient

- M pour un tableau de regrets / coûts
- 0 pour un tableau de préférences / scores (avec valeurs toutes positives)

6.2 La matrice d'affectation n'est pas carrée

Contrainte : le nombre de machine n'est pas égale au nombre d'emplacements

Perturbation créée : ajouter une ligne fictive (ou une colonne) avec un score rendant « neutre » les coefficients artificiellement ajoutés, tout en permettant d'appliquer l'algorithme général

Exercice 4 :

Une usine de transformation doit acheter 3 nouvelles machines ayant des caractéristiques différentes et qui transforment de la matière brute fournie par un secteur différent dans l'usine.

Il existe 4 emplacements où ces machines peuvent être installées. Certains de ces emplacements sont préférables à d'autres, selon le type de machine, en fonction de la proximité du secteur qui doit fournir la matière brute à la machine.

L'objectif est donc d'affecter chacune des machines à un emplacement différent dans l'usine de façon à minimiser le coût d'acheminement de la matière brute à chacune des machines.

La matrice originale des coûts est la suivante :

Machines	Emplacements			
	1	2	3	4
1	13	16	12	11
2	15	---	13	20
3	5	7	10	6

La conjonction des deux perturbations créées fournie le tableau suivant :

Machines	Emplacements			
	1	2	3	4
1	13	16	12	11
2	15	M	13	20
3	5	7	10	6
4	0	0	0	0

L'algorithme de Kühn peut alors être mis en œuvre