

Prenez le temps de faire les calculs au brouillon avant de les recopier sur votre copie d'examen. Efforcez-vous de traiter les questions dans l'ordre. Indiquez bien le numéro de chaque question avant la réponse. Les copies mal présentées seront pénalisées.

1 Modélisation d'une ferme (7 points)

Des citoyens en mal de campagne ont acheté une ferme avec 12 ares de terrain. Ils souhaitent y élever des vaches et y cultiver de la betterave. Il faut exactement deux ares pour faire paître une vache. Les ares qui ne sont pas utilisés par les vaches peuvent être mis en culture. Chacun d'eux permet de produire une tonne de betteraves. Tout le terrain ne doit pas obligatoirement être utilisé. Les betteraves produites doivent suffire à alimenter les vaches. Il en faut une tonne pour nourrir une vache. Les tonnes qui ne sont pas consommées par les vaches sont vendues à 50 euros la tonne. Chaque vache fournit du lait. Les produits laitiers ainsi obtenus rapportent 200 euros par vache. On cherche à maximiser le profit.

Question 1 [1 pt]. Modéliser le problème ci-dessus sous la forme d'un programme linéaire en variables réelles. Bien préciser les dimensions des variables, fournir un mot-clé par contrainte.

SOLUTION. Les variables sont x_1 (le nombre de vaches à élever) et x_2 (le nombre de tonnes de betteraves à produire).

$$\begin{cases} 150x_1 + 50x_2 = z & [\text{max}] \\ 2x_1 + x_2 \leq 12 & (\text{surface}) \\ x_1 - x_2 \leq 0 & (\text{autosuffisance}) \\ x_1, x_2 \geq 0 \end{cases}$$

Question 2 [1 pt]. Résoudre le programme linéaire graphiquement. Bien préciser le polygone des solutions réalisables, la solution optimale, l'objectif réalisé à l'optimum. Dessiner au moins deux droites de l'objectif.

Question 3 [2 pts]. Résoudre le programme linéaire par l'algorithme du tableau simplexe. Donner à chaque itération, la solution de base et le sommet correspondant du polygone des solutions réalisables.

SOLUTION. Voici au moins les itérations du tableau simplexe. La solution optimale est

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}.$$

Le profit réalisé à l'optimum est de 800 euros.

	x1	x2	x3	x4	
x3	2	1	1	0	12
x4	1	-1	0	1	0
	150	50	0	0	0 = - z0

	x1	x2	x3	x4	
x3	0	3	1	-2	12
x1	1	-1	0	1	0
	0	200	0	-150	0 = - z0

	x1	x2	x3	x4	
x2	0	1	1/3	-2/3	4
x1	1	0	1/3	1/3	4
	0	0	-200/3	-50/3	-800 = - z0

Question 4 [1 pt]. Écrire le programme linéaire dual du programme précédent.

SOLUTION. La variable y_1 est associée à la contrainte « surface ». La variable y_2 est associée à la contrainte « autosuffisance ». Le programme dual est

$$\begin{cases} 12y_1 = w & [\text{min}] \\ 2y_1 + y_2 \geq 150 & (\text{vaches}) \\ y_1 - y_2 \geq 50 & (\text{betteraves}) \\ y_1, y_2 \geq 0 \end{cases}$$

Question 5 [1 pt]. Donner, sans faire de calcul, la solution optimale du dual.

SOLUTION. Elle se lit dans le tableau simplicial final du primal

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 200/3 \\ 50/3 \end{pmatrix}.$$

Question 6 [1 pt]. Donner les valeurs marginales des contraintes du primal et les interpréter.

SOLUTION. Les valeurs marginales des contraintes sont les valeurs à l'optimum des variables duales des contraintes. Si on augmente d'un are la surface, le profit augmente de $200/3$ d'euros. Si on diminue de 1 le nombre de vaches à nourrir avec la production de la ferme, le profit augmente de $50/3$ d'euros.

2 Modélisation d'une ferme (6 points)

Enthousiasmés par le succès de leur ferme, nos citoyens décident de voir plus grand. Ils achètent une ferme de 200 ares pour y élever des vaches et y cultiver différents types de végétaux.

Il faut au moins deux ares pour faire paître une vache. Les ares qui ne sont pas utilisés par les vaches peuvent mis en culture. Tout le terrain ne doit pas obligatoirement être utilisé. Différents végétaux sont cultivés : betterave, blé, maïs. Ces végétaux doivent suffire à nourrir les vaches. Les végétaux qui ne sont pas consommés par les vaches peuvent être vendus. Chaque vache fournit du lait. Les produits laitiers ainsi obtenus rapportent 200 euros par vache. On cherche à déterminer le nombre de vaches à élever ainsi que les nombres de tonnes de différents végétaux à cultiver pour maximiser le profit. Le tableau suivant donne le rendement des végétaux en tonnes par are, les quantités consommées en tonnes par vache, le nombre de tonnes qu'il est possible de vendre au maximum et les prix de vente à la tonne.

		Végétaux		
	betterave	blé	maïs	
rendement	1	0.6	0.5	
consommation	0.6	0.2	0.2	
vente max	10	20	20	
prix de vente	100	120	90	

Question 7 [4 pts]. Compléter le modèle AMPL suivant. Utiliser toutes les variables. Bien choisir les identificateurs de contraintes. Les solutions trop compliquées seront considérées comme fausses.

```

set VEGETAUX;

param nb_total_ares >= 0;           # ares

param rendement {VEGETAUX} >= 0;    # tonnes / are
param consommation {VEGETAUX} >= 0; # tonnes / vache
param vente_max {VEGETAUX} >= 0;    # tonnes
param prix_vente {VEGETAUX} >= 0;   # euros / tonnes

param nb_ares_par_vache >= 0;       # ares / vache
param prix_vente_pdt_laitiers >= 0;  # euros / vache

var nb_ares_paturage >= 0;           # ares
var nb_ares_culture {VEGETAUX} >= 0; # ares

var nb_vaches >= 0;                 # vaches
var qte_produite {VEGETAUX} >= 0;   # tonnes
var qte_vendue {VEGETAUX} >= 0;     # tonnes
var qte_consommee {VEGETAUX} >= 0;  # tonnes

data;

set VEGETAUX := betterave ble mais;

param nb_total_ares := 200;

param nb_ares_par_vache := 2;
param prix_vente_pdt_laitiers := 200;

```

```

param :          rendement consommation vente_max prix_vente :=
betterave        1          .6          10          100
ble               .6          .2          20          120
mais              .5          .2          20          90;

```

SOLUTION. Voici les contraintes et l'objectif qui manquent.

```

maximize objectif :
    nb_vaches * prix_vente_pdt_laitiers +
    sum {v in VEGETAUX} qte_vendue [v] * prix_vente [v];

subject to loi_de_conservation {v in VEGETAUX} :
    qte_produite [v] = qte_vendue [v] + qte_consommee [v];

subject to relation_production_surface_culture {v in VEGETAUX} :
    qte_produite [v] = nb_ares_culture [v] * rendement [v];

subject to relation_nb_vaches_surface_pature :
    nb_vaches * nb_ares_par_vache = nb_ares_paturage;

subject to contrainte_superficie :
    nb_ares_paturage + sum {v in VEGETAUX} nb_ares_culture [v] <= nb_total_ares;

subject to relation_consommation_nb_vaches {v in VEGETAUX} :
    qte_consommee [v] = nb_vaches * consommation [v];

subject to limitation_ventes {v in VEGETAUX} :
    qte_vendue [v] <= vente_max [v];

```

Question 8 [1 pt]. En supposant le modèle et ses données stockés dans un fichier *fermette.ampl*, donner une ou plusieurs commandes AMPL permettant de déterminer les valeurs des variables *nb_vaches* et *nb_ares_culture* ainsi que l'objectif réalisé à l'optimum.

SOLUTION. Voici les commandes.

```

model fermette.ampl;
solve;
display nb_vaches, nb_ares_culture;

```

Question 9 [1 pt]. On aimerait savoir quel serait l'effet sur l'objectif d'une petite variation sur les quantités maximales de végétaux qu'il est possible de vendre. Quelle commande AMPL permettrait d'obtenir cette information?

SOLUTION. Voici la commande.

```

display limitation_vente;

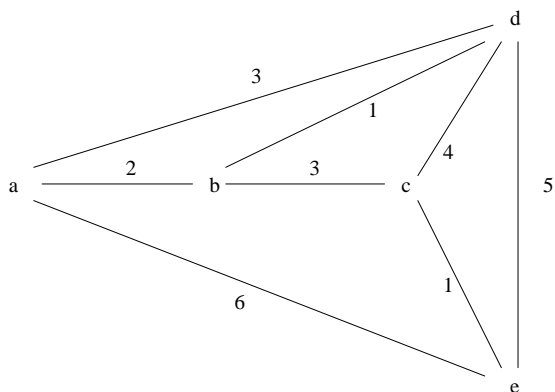
```

3 Phylogénétique (10 points)

Le problème qui suit est une version abstraite et simplifiée d'un problème classique de phylogénétique, qui consiste à déterminer l'histoire des mutations qui ont affecté une espèce animale ou végétale au cours des âges.

Soit $T = (S, A)$ un arbre dont les arêtes sont valuées avec des nombres strictement positifs. On construit un graphe G à partir de T en rajoutant à A une ou plusieurs nouvelles arêtes $(x, y) \in S \times S$. La valeur d'une nouvelle arête (x, y) est la valeur minimale des chaînes d'extrémités x et y dans T .

Question 10 [1 pt]. Le graphe G ci-dessous a été obtenu à partir d'un arbre T en appliquant le mécanisme décrit ci-dessus. Donner l'arbre T sans justifier.



SOLUTION. Les arêtes de l'arbre sont (a,b) , (b,c) , (b,d) , (c,e) .

Question 11 [1 pt]. Démontrer en une ou deux phrases que quelle que soit la nouvelle arête (x, y) , toutes les chaînes d'extrémités x et y dans T ont même valeur.

SOLUTION. Quels que soient les sommets x et y , l'arbre T comporte une unique chaîne élémentaire d'extrémités x et y (thm du cours).

On se pose le problème : à partir de G , retrouver T . Pour cela, une idée consiste à distinguer l'un des sommets s puis à appliquer un algorithme qui détermine un chemin de valeur minimale entre s et tous les autres sommets du graphe G . Dans le cas où plusieurs chemins de valeur minimale sont possibles, l'algorithme en choisit un de longueur maximale (en nombre d'arêtes).

Question 12 [1 pt]. Indiquer le nom du ou des algorithmes vus en cours qui peuvent être appliqués sur le graphe G et le sommet s distingué pour déterminer un chemin de valeur minimale (pas forcément de longueur maximale) entre s et tous les autres sommets. Justifier.

SOLUTION. L'algorithme de Dijkstra (les valeurs sont positives). L'algorithme de Bellman ne s'applique pas parce que le graphe est non orienté.

Question 13 [3 pts]. Choisir l'un des algorithmes cités ci-dessus. L'adapter pour qu'il détermine un chemin de longueur maximale parmi les chemins de valeur minimale de s vers tous les autres sommets. Donner l'algorithme en pseudo-code à la façon du support de cours.

Pour chaque sommet x votre algorithme doit calculer $\text{pi}(x)$ (la valeur minimale des chemins de s vers x), $\text{lg}(x)$ (la longueur maximale des chemins de valeur minimale de s vers x), $\text{pred}(x)$ (le prédécesseur de x dans le chemin choisi par l'algorithme).

SOLUTION. Voici une solution en pseudo-code.

procédure Dijkstra (G, s)

begin

colorier tous les sommets sauf s en bleu

colorier s en vert

$\text{pi}(s) := 0$

$\text{lg}(s) := 0$

$\text{pred}(s)$ est indéfini

while il existe au moins un sommet vert do

soit E l'ensemble des sommets verts tels que pi est minimal

choisir dans E un sommet x tel que $\text{lg}(x)$ est maximal.

colorier x en rouge

for tous les successeurs y de x do

if y est bleu ou (y est vert et $\text{pi}(y) > \text{pi}(x) + v(x,y)$) ou

(y est vert et $\text{pi}(y) = \text{pi}(x) + v(x,y)$ et $\text{lg}(y) < \text{lg}(x) + 1$) then

colorier y en vert

$\text{pi}(y) := \text{pi}(x) + v(x,y)$

$\text{lg}(y) := \text{lg}(x) + 1$

$\text{pred}(y) := x$

fi

od

od

end

Question 14 [1 pt]. Indiquer la ou les structures de données importantes utilisées par votre algorithme.

SOLUTION. La gestion des sommets verts peut se faire grâce à une file avec priorité. Elle peut s'implanter au moyen d'un tas binaire. La propriété de tas maintenue est la suivante : *tout sommet a soit une valeur de pi strictement inférieure à celle de ses deux fils soit la même valeur de pi mais une valeur de lg supérieure ou égale à celle de ses deux fils.*

Pour chacun des n sommets x du graphe, on souhaite stocker les valeurs $\text{pi}(x)$, $\text{lg}(x)$ et $\text{pred}(x)$ dans une table de hachage H à adressage ouvert, utilisant la technique du double hachage.

Question 15 [2 pts]. Indiquer les objets à stocker dans la table H ainsi que leur clef. Exprimer en fonction de la taille du graphe le nombre d'objets qui seront stockés dans H à la fin de l'exécution de votre fonction.

SOLUTION. Les objets stockés sont des triplets (code, sommet, valeur). Le code appartient au type énuméré : pi , lg , pred . Les deux autres champs sont le sommet et la valeur. Les clefs

sont formées des deux premiers champs des triplets. Au total, $3n$ objets seront stockés dans la table.

On suppose que les deux fonctions de hachage répartissent les valeurs de hachage uniformément dans la table H . On souhaite que le nombre moyen de collisions pour la recherche d'un emplacement libre ne dépasse jamais 2, y compris à la fin de l'exécution de l'algorithme, lorsque tous les objets sont stockés dans la table.

Question 16 [1 pt]. Exprimer en fonction de la taille du graphe la taille que devrait alors avoir H . Justifier.

SOLUTION. On cherche le facteur de remplissage α tel que le nombre moyen de collisions $1/(1 - \alpha)$ soit égal à 2. On trouve $\alpha = 1/2$. Comme à la fin de l'exécution de l'algorithme la table contient $3n$ objets, elle doit comporter $6n$ entrées.