Recherche opérationnelle
Premier et deuxième projets

Youssef Bendagha
Robin Xambili
Mohamed Smail
Aimad Elhouir
Salem Ben El Cadi

30 mai 2018

Table des matières

1 Algorithme du simplexe **7**

2 Construction automatique d'un emploi du temps **9**

2.1 Introduction 9

2.2 Modélisation sous la forme d'un Programme Linéaire en variables Entières (PLE) 10

2.2.1 Formulation mathématique du problème 10

2.2.2 Ecriture matricielle en vue d'une implémentation sur Matlab et calcul par intlinprog 10

2.3 Résultats 15

3 Algorithme de Ford Fulkerson **19**

4 Répartition des taches **21**

Premier projet

Chapitre 1

Algorithme du simplexe

le rapport de l'algorithme du simplexe sera detaillee sur un document annexe.

Chapitre 2

Construction automatique d'un emploi du temps

2.1 Introduction

L'objectif de cette partie du premier projet est de construire automatiquement l'emploi du temps (EDT) de deux promotions de l'ENSEEIHT (version simplifiée), et ce avec le minimum de trous possible.
Tous les cours durent 1 heure et 45 minutes. Du lundi au vendredi ($d = 5$) , il y a $t = 4$ créneaux par journée : 8h-9h45 , 10h15-12h , 14h- 15h45 , 16h15-18h.
Il y a $c = 2$ promotions d'étudiants. Tous les étudiants d'une même promo suivent exactement les mêmes cours. Les 2 promotions ont les mêmes professeurs sauf pour les cours de maths, d'informatique et de sport. Il y a au total $m = 8$ professeurs : 2 professeurs de maths (Mme Droite pour la 1ère promotion et Mr Ellips pour la seconde promotion), 1 professeur de physique Mme Proton, 2 professeurs d'informatique Mr Pascal et Mme Dell, 1 professeur d'anglais Mr Young et 2 professeurs de sport Melle Gazelle et Mr Bigceps.

- Mme Droite assure 5 cours par semaine
- Mr Ellips, 4 cours
- Mme Proton, 3 cours pour la 1ère promo et 3 cours pour la 2ème
- Mr Pascal, 6 cours pour la promo 1
- Mme Ada, 6 cours pour la promo 2
- Mr Young, 3 cours pour la promo 1 et 3 cours pour la promo 2
- Les cours de sport ont lieu le jeudi après-midi de 14h à 16h.

Le premier créneau du lundi matin est réservé au partiel. Mr Ellips est indisponible le lundi matin. Mme Proton ne peut pas travailler le mercredi. Chaque promo ne doit pas avoir plus d'un cours d'une même matière dans la même journée à l'exception des cours d'informatique où le nombre de cours dans une même journée ne doit pas dépasser deux.

2.2 Modélisation sous la forme d'un Programme Linéaire en variables Entières (PLE)

2.2.1 Formulation mathématique du problème

. Le problème sera modélisé comme suit :

$x_{i,j,k} = 1$ ssi le prof i fait cours à la promo j durant le créneau k
$= 0$ sinon

(1) Min $\sum_{i=1}^{p} \sum_{j=1}^{c} \sum_{l=0}^{d-1} (x_{i,j,lt+1} + x_{i,j,lt+t})$
(2) $\forall$ i=1..p,j=1..c $\sum_{k=1}^{dt} x_{i,j,k} = NbC_{i,j}$; $NbC_{i,j}$ étant le nombre de cours que donne le prof i à la promo j par semaine.
(3) $\forall$ j=1..c,k=1..dt $\sum_{i=1}^{p} x_{i,j,k} <= 1$
(4) $\forall$ i=1..p,k=1..dt $\sum_{j=1}^{c} x_{i,j,k} <= 1$
(5) $\forall$ i=1..p,j=1..c,l=1..d $\sum_{k=(l-1)t+1}^{lt} x_{i,j,k} <= 1$
(6) $x_{8,1,15} = 1$
(7) $x_{7,1,15} = 1$
(8) $\forall$ i=1..p,j=1..c $x_{i,j,1} = 0$
(9) $\forall$ j=1,2,k=9,10,11,12 $x_{2,j,k} = 0$
(10) $\forall$ k=1,2 $x_{4,2,k} = 0$
(11) $\forall$ i=1..p,j=1..c,k=1..dt $x_{i,j,k}$=0 ou $x_{i,j,k}$=1

2.2.2 Ecriture matricielle en vue d'une implémentation sur Matlab et calcul par intlinprog

MATLAB propose différents outils de résolution de problèmes d'optimisation au sein de sa toolbox "Optimisation". Pour ce problème, nous utiliserons intlinprog pour la resolution des systèmes linéaires en variables entières. Cet outil peut etre utilisé soit depuis son interface graphique, soit en faisant directement appel aux solveur tels une fonctions MATLAB.
Intlinprog peut résoudre les problèmes formulés sous la forme suivante :
(1) min f'*x
(2) A*x <= b
(3) Aeq*x = beq
(4) lb <= x <= ub
(5) x(intcon) sont des entiers
avec f, x, b, beq, lb, ub et intcon des vecteurs, tandis que A et Aeq sont des matrices.
appel dierct au solveur :
» x = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)

Le vecteur x : Le vecteur x contient les solutions $x_{i,j,k}$ de notre problème. Une représentation de la solution sous forme d'une matrice tridimensionnelle X serait meilleure (vu qu'on a 3 indices i,j et k) mais intlinprog en impose autrement. on aura alors un vecteur colonne de taille m = p*c*dt (dt=d*t). Pour cela on introduit la fonction indiceEq(i,j,k,p,c) qui retourne l'indice correspondant au triplet d'indices (i,j,k) : on fait en sorte d'incrémenter i puis j, et finalement k. On gardera cette modélisation pour toutes les autres composantes du problème.
(indice = 1 + (i - 1) + (j - 1) * p + (k - 1) * p * c)

Vecteur f : f est la fonction dont on doit rechercher la valeur minimum, elle représente dans notre cas la somme du premier et dernier créneau de chaque journée. Minimiser cette somme revient à minimiser le nombre de trous dans un emploi du temps (version simplifiée).
Algorithme de f :

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% f est la fonction (vecteur) à minimiser
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% initialisation

p = 8;
c = 2;
d = 5;
t = 4;

f = zeros(p*c*d*t,1);


for i = 1 : p
    for j = 1 : c
        for l = 0 : (d - 1)
                f(indiceEq(i,j,l*t+1,p,c))=1;
                f(indiceEq(i,j,l*t+t,p,c))=1;
        end
    end
end
```

Matrice A et vecteur b correspondant : A est la matrice des contraintes d'inégalités A ; on définira par la suite le vecteur b tel que A*x <= b. A comporte trois partie, chacune correspondant à une des 3 inégalités citées lors de la modélisation du problème ((3), (4) et (5) de la partie 2.2.1) ; cela va de meme pour b.
Algorithme de A et de b :

```
% initialisation
p = 8;
c = 2;
d = 5;
t = 4;
dt = d * t;
A = zeros(280, p * c * d*t);
b = zeros(280,1);

% numéro de la contrainte courante d'inégalité
numCin = 1;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1iere partie de la matrice A
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% cette partie de A correspond à la 1ière inégalité
% (voir ligne (3) slide n° 53 poly Recherche Opérationnelle)

for j = 1 : c
    for k = 1 : (dt)
        for i = 1 : p
            A(numCin,indiceEq(i,j,k,p,c))=1;
        end
        b(numCin)=1;
        numCin = numCin+1;
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2ieme partie de la matrice A
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% cette partie de A correspond à la 2ieme inégalité
% (voir ligne (4) slide n° 53 poly Recherche Opérationnelle)

for i = 1 : p
    for k = 1 : (dt)
        for j = 1 : c
            A(numCin,indiceEq(i,j,k,p,c))=1;
        end
        b(numCin) = 1;
        numCin = numCin + 1;
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2ieme partie de la matrice A
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% cette partie de A correspond à la 2ieme inégalité
% (voir ligne (4) slide n° 53 poly Recherche Opérationnelle)

for i = 1 : p
    for j = 1 : c
        for l = 1 : d
            for k = (l-1)*t+1:l*t
                A(numCin, indiceEq(i,j,k,p,c)) = 1;
            end
            if(i==4 || i==5)
                b(numCin)=2;
            else
                b(numCin)=1;
            end
        numCin = numCin + 1;
        end
    end
end
```

paragraphMatrice Aeq et vecteur beq correspondant : A est la matrice des contraintes d'égalités ; on définira par al suite le vecteur beq tel que (Aeq * x = beq).

```
%initialisation
p = 8;
c = 2;
d = 5;
t = 4;
dt = d * t;
Aeq = zeros(44, p * c * d*t);
beq = zeros(44,1);

% le numéro de la contrainte d'égalité courante
numCeq = 1;


% cette partie de beq correspond à l'égalité (2) (voir slide n° 53 poly Recherche Opérationnelle)
% concrètement cela représente le nombre de cours donné par le prof i à la promo j
% (voir aussi "feuille de route")
beq(numCeq:numCeq+p*c-1)=[5;0;0;4;3;3;6;0;0;6;3;3;1;0;0;1];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1iere partie de la matrice Aeq
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:p
    for j=1:c
        for k=1:dt
                Aeq(numCeq,indiceEq(i,j,k,p,c))=1;
        end
        numCeq = numCeq + 1;
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2ieme partie de la matrice Aeq
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Les cours de sport ont lieu le jeudi après-midi de 14h à 16h.
%les profs de sport sont (i=7)Melle Gazelle et (i=8)Mr Bigceps

Aeq(numCeq,indiceEq(7,1,15,p,c))=1;

beq(numCeq)=1;
numCeq = numCeq + 1;

Aeq(numCeq,indiceEq(8,2,15,p,c))=1;

beq(numCeq)=1;
numCeq = numCeq + 1;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3ieme partie des contraintes d'egalites
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% cette partie de Aeq correspond à l'égalité (8)
% (voir slide n° 53 poly Recherche Opérationnelle)

for i=1:p
    for j=1:c
        Aeq(numCeq,indiceEq(i,j,1,p,c))=1;
        beq(numCeq)=0;
        numCeq=numCeq+1;
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 4ieme partie de la matrice Aeq
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% cette partie de Aeq correspond à l'égalité (9)
% (voir slide n° 53 poly Recherche Opérationnelle)

for k=1:2
    Aeq(numCeq,indiceEq(2,2,k,p,c))=1;
    beq(numCeq)=0;
    numCeq = numCeq + 1;
end
```

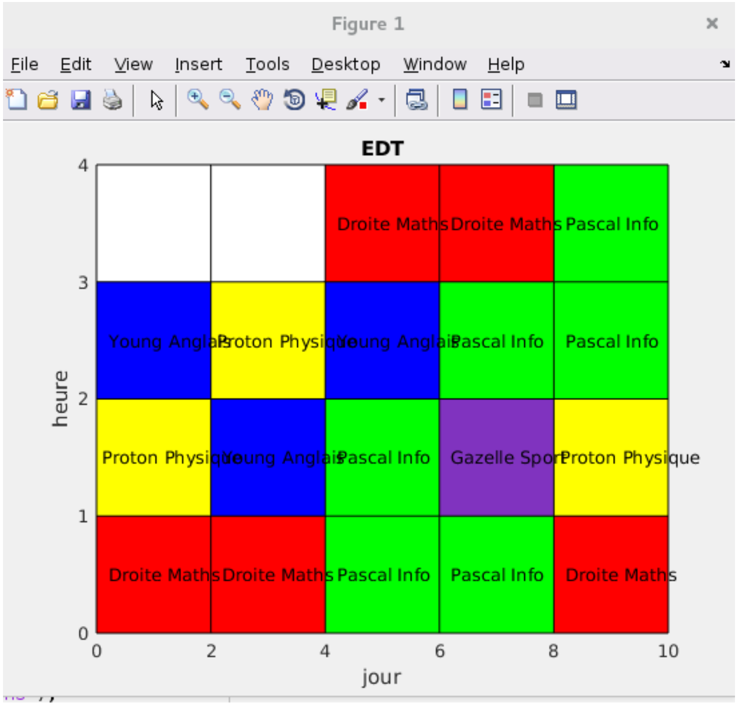
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 5ieme partie de la matrice Aeq
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% cette partie de Aeq correspond à l'égalité (10)
% (voir slide n° 53 poly Recherche Opérationnelle)

for j=1:2
    for k=9:12
        Aeq(numCeq,indiceEq(3,j,k,p,c))=1;
        beq(numCeq) = 0;
        numCeq = numCeq + 1;
    end
end
```

lb,ub et intcon : lb est un vecteur colonne de meme taille que x avec ∀ i=1..p lb_i=0
ub est un vecteur colonne de meme taille que x avec ∀ i=1..p ub_i=1
intcon est un vecteur colonne de meme taille que x avec , les indices des éléments de x.
cela garanti que tout les éléments de x sont des entiers lorsque calculés avec intlinprog, et que (lb <= x <= ub).

2.3 Résultats

la résolution de ce probleme donne en prenant l'exemple de la promo1 :



Deuxième projet

Chapitre 3

Algorithme de Ford Fulkerson

Le travail consiste à :
1. Programmer en Matlab l'algorithme de Ford et Fulkerson (FF) pour les flots puis pour les tensions
2. Modéliser le problème de trafic maritime (cf. poly) sous la forme d'un problème de flot maximum. Le résoudre par l'algorithme FF pour les flots.
3. Modéliser le problème de plus court chemin sous la forme d'un problème de tension maximum (cf. poly). Le résoudre par l'algorithme FF pour les tensions.
4. Modéliser le problème d'ordonnancement (cf. poly) sous la forme d'un problème de tension maximum. Le résoudre par l'algorithme FF pour les tensions.

Chapitre 4

Répartition des taches

Youssef Bendagha :Résponsable qualité ; a traité le problème de l'emploi du temps ainsi que le probleme de l'ordonnancement, et rédigé une partie du rapport.
Robin Xambili :Chef de projet ; a traité le problème du simplexe ainsi que la partie TraficMaritime et FF flots du second projet.
Mohamed Smail : a aidé à la réflexion sur le probleme de l'edt ainsi que sur le Projet2 de maniere generale.
Aimad Elhouir : a aidé à la réflexion sur le probleme de l'edt.
Salem Ben El Cadi : a aidé à la redaction du rapport (la partie algorithme du simplexe).