

# Problème de répartition d'un marché

Valentin Ryckewaert

Marion Noirbent

8 novembre 2015

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Organisation de l'environnement . . . . .	2
1.2	Utilisation du script . . . . .	2
<b>2</b>	<b>Recherche d'une solution réalisable</b>	<b>3</b>
2.1	Fonctionnement . . . . .	3
2.2	Résultat . . . . .	3
<b>3</b>	<b>Recherche avec minimization des valeurs absolues</b>	<b>4</b>
<b>A</b>	<b>repartition_marche</b>	<b>5</b>
<b>B</b>	<b>repartition_marche_realisable.model</b>	<b>6</b>

# Chapitre 1

## Introduction

### 1.1 Organisation de l'environnement

Notre environnement se compose des fichiers suivant :

- Le fichier "repartition\_marche" est un script qui automatise la configuration et la résolution par Ampl.
- Le fichier "repartition\_marche.data" fourni, que nous avons utilisé sans le modifier.
- Le fichier "repartition\_marche\_realisable.model" qui modélise la recherche d'une solution réalisable.
- Le fichier "repartition\_marche\_somme.model" qui modélise la recherche d'une solution minimisant la somme des variations.
- Le fichier "repartition\_marche\_variation.model" qui modélise la recherche d'une solution minimisant la variation maximale.

### 1.2 Utilisation du script

Le script est chargé dans Ampl par la commande : `model repartition_marche`

Sa première action est un reset, afin de s'assurer que le script s'exécute dans un environnement propre.

Ensuite il configure le solveur sur Gurobi, un détaillant ne pouvant pas appartenir à 0.8 à une division.

Puis il charge le fichier de data et le premier modèle (celui de recherche d'une solution réalisable) et lance la résolution.

Ensuite il fera un reset du modèle, rechargera le modèle correspondant et lancera sa résolution pour la minimisation la somme des variations puis pour la minimisation de la variation maximale.

# Chapitre 2

## Recherche d'une solution réalisable

### 2.1 Fonctionnement

N'ayant que deux division à affecter, le problème de se rapprocher du rapport 40/60 est équivalent à se rapprocher de 60% du total pour la 2<sup>e</sup> division.

Nous avons donc fait le choix de leur attribuer la valeurs d'un booléen (0 pour la 1<sup>re</sup> division, 1 pour la 2<sup>e</sup>).

### 2.2 Résultat

Réponse donnée par Ampl lors de la recherche d'une solution réalisable :

```
Gurobi 5.0.2 : optimal solution ; objective 0
div_du_detaillant [*] :=
M1 0 M12 1 M15 1 M18 0 M20 1 M23 1 M5 1 M8 1
M10 1 M13 1 M16 0 M19 1 M21 0 M3 0 M6 1 M9 0
M11 1 M14 1 M17 1 M2 0 M22 0 M4 0 M7 1
;
```

# Chapitre 3

## Recherche avec minimisation des valeurs absolues

# Annexe A

## repartition\_marche

```
reset ;

option solver gurobi ;

/* Question 1 : recherche d'une solution réalisable */
model repartition_marche_realisable.model
data repartition_marche.data
solve ;

display div_du_detaillant ;

/* Question 2 : optimisation */

/* en somme des écart */
reset model ;
model repartition_marche_somme.model
solve ;

/* en variation maximale */
reset model ;
model repartition_marche_variation.model
solve ;
```

# Annexe B

## repartition\_marche\_realisable.model

```
set DETAILLANTS ;
set REGIONS ;
set CATEGORIES ;

/* ----- */
/* Paramètres caractérisant un détaillant */

/* ***** */
/* La région dans laquelle il est implanté */
param region{DETAILLANTS} symbolic in REGIONS ;

/* Son nombre de vente d'huile */
param huile{DETAILLANTS} >= 0 ;

/* Son nombre de points de vente */
param nb_pts_vente{DETAILLANTS} >= 0 ;

/* Son nombre de vente de spiriteux */
param spiritueux{DETAILLANTS} >= 0 ;

/* Sa catégorie */
param categorie{DETAILLANTS} symbolic in CATEGORIES ;

/* ----- */
/* Paramètre calculés */

/* Nombre total de points de ventes */
param pts_vente_totaux :=
sum {m in DETAILLANTS}
nb_pts_vente[m] ;

/* Nombre total de vente de spiriteux */
param spiritueux_totaux :=
sum {m in DETAILLANTS}
spiritueux[m] ;

/* Nombre de points d'huile dans chaque région */
param huile_par_region {r in REGIONS} :=
sum {m in DETAILLANTS : region[m]=r}
```

```

huile[m];

/* Nombre de détaillants de chaque catégorie */
param nb_par_categorie{c in CATEGORIES} :=
sum {m in DETAILLANTS : categorie[m]=c}
1;

/* _____ */
/* Répartition recherchée */

/* Division à laquelle rattaché chaque détaillant
div_du_detaillant[m] vaut 0 si m appartient à la division D1, et vaut 1 sinon */
var div_du_detaillant {m in DETAILLANTS} binary;

/* _____ */
/* Rapprochement de l'objectif */

/* Cas où on cherche une solution réalisable -> seules les contraintes importent */
minimize res : 0;

/* _____ */
/* Contraintes */

/* ***** */
/* Contraintes sur le rapport 40/60 dans chaque division (+/- 5%) */

/* Le nombre total de points de vente */
subject to rapport_pts_vente_min :
sum {m in DETAILLANTS}
div_du_detaillant[m]*nb_pts_vente[m]*100 / pts_vente_totaux >= 55;

subject to rapport_pts_vente_max :
sum {m in DETAILLANTS}
div_du_detaillant[m]*nb_pts_vente[m]*100 / pts_vente_totaux <= 65;

/* Le nombre total de vente de spiritueux */
subject to rapport_spiritueux_min :
sum {m in DETAILLANTS}
div_du_detaillant[m]*spiritueux[m]*100 / spiritueux_totaux >= 55;

subject to rapport_spiritueux_max :
sum {m in DETAILLANTS}
div_du_detaillant[m]*spiritueux[m]*100 / spiritueux_totaux <= 65;

/* Le nombre de points d'huile dans chaque région */
subject to rapport_huile_region_min {r in REGIONS} :
sum {m in DETAILLANTS : region[m]=r}
div_du_detaillant[m]*huile[m]*100 / huile_par_region[r] >= 55;

subject to rapport_huile_region_max {r in REGIONS} :
sum {m in DETAILLANTS : region[m]=r}
div_du_detaillant[m]*huile[m]*100 / huile_par_region[r] <= 65;

```



```

/* Le nombre de détaillants de chaque catégorie */
subject to rapport_categorie_min {c in CATEGORIES} :
sum {m in DETAILLANTS : categorie[m]=c}
div_du_detaillant[m]*100 / nb_par_categorie[c] >= 55;

subject to rapport_categorie_max {c in CATEGORIES} :
sum {m in DETAILLANTS : categorie[m]=c}
div_du_detaillant[m]*100 / nb_par_categorie[c] <= 65;

```