

RAPPORT DM ARO

Répartition d'un marché

Université Lille 1
Année 2015-2016

Introduction

But du travail

Le but de ce travail était de trouver une solution pour répartir l'ensemble des détaillants dans des divisions tout en respectant une liste de contraintes définie au préalable, dont la principale, qui était d'avoir un rapport 40/60, avec une tolérance de +/- 5%, sur tous les ensembles.

Problème à résoudre

Le problème d'affectation est un problème classique de recherche opérationnelle et d'optimisation combinatoire. De manière non-formelle, ce problème consiste à attribuer au mieux des tâches à des machines. Dans ce cas, les "tâches" sont les détaillants et les "machines" sont les divisions.

Solution du problème

La solution pour ce type de problème consiste généralement à introduire des variables binaires (soit un, soit zéro) permettant de représenter l'appartenance entre "les tâches" et les "machines". Ainsi, les contraintes peuvent être plus facilement exprimées, même si elles paraissent contradictoires. Dans notre situation, c'est exactement comme cela que l'on procède.

Outils utilisés

L'outil principal qui a été utilisé est le langage AMPL, il a permis d'exprimer le modèle ainsi que de trouver sa résolution.

Le solveur spécifié a été **gurobi**, car il a permis de résoudre des problèmes linéaires comportant des données entières

Ensembles

Il y avait 4 ensembles, à savoir :

- DIVISIONS
- DETAILLANTS
- REGIONS
- CATEGORIES

Divisions

L'ensemble DIVISIONS permet de représenter les différentes divisions. Dans le sujet, on a "D1" et "D2". Cela nous permet de regrouper différentes notions, comme les rapports et ses paramètres calculés. De plus, on peut généraliser le problème de l'énoncer, et ajouté plus

de deux divisions. Cela implique par ailleurs, une contrainte garantissant une unicité dans les affectations (voir Contraintes -> Contrainte d'unicité).

Détaillants

L'ensemble DÉTAILLANTS, est défini de la manière suivante : À chaque détaillant est associé une estimation de marché. Les détaillants sont définis par les paramètres suivants : (voir Paramètres)

`region, categorie, nb_huile, nb_pts_vente, nb_spiritueux`

Régions

L'ensemble RÉGIONS, chaque détaillant appartient à une région. Dans le sujet, trois régions étaient présentes R1, R2 et R3. Le paramètre contenant cette donnée est `region` indexé par `DÉTAILLANTS`

Catégories

L'ensemble CATÉGORIES représente une perspective d'évolution pour chaque détaillant, dans le sujet sont présentes les catégories A et B. Le paramètre contenant cette donnée est `categorie` indexé par `DÉTAILLANTS`

Paramètres

Dans la modélisation du problème, on peut distinguer, deux type de paramètres :

- **Paramètres classiques** : Ces paramètres sont déterminés par la section **data**, ils sont variables. Ce sont les données que l'utilisateur indique. Ils permettent de rendre le modèle plus générique.
- **Paramètres calculés** : Ils sont calculés à partir des paramètres classiques. Cela permet de simplifier la résolution du problème. Le modèle n'en est que plus visible, car il y a une réduction de code. De plus, cela optimise les opérations, car elles ne sont pas répétées plusieurs fois.

Paramètres classique

Exemple de déclaration :

```
19 | param region {DÉTAILLANTS} symbolic in REGIONS;
```

- `region` Ce paramètre représente la région du détaillant, il est de type **REGIONS** (grâce au mot clé "**symbolic in**").
- `categorie` Ce paramètre représente la catégorie du détaillant, il est de type `categorie`.
- `nb_huile` Ce paramètre représente le nombre d'huiles, il est de type **integer**. La borne minimal est 0

- `nb_pts_vente` Ce paramètre représente le nombre de points de vente, il est de type **integer**. La borne minimal est 0.
- `nb_spiritueux` Ce paramètre représente le nombre de spiritueux, il est de type **integer**. La borne minimal est 0.
- `tolerance` Ce paramètre représente la tolérance qui sera appliquée au rapport renseigné. La borne minimal est 0, la borne maximal est 100.
- `rapport_marche` Ce paramètre représente le rapport qui sera appliqué aux différentes divisions.
-

Paramètres calculés

Exemple de déclaration :

```
33 param nb_global_pts_vente := sum {dt in DETAILLANTS} nb_pts_vente [dt];
```

- `nb_global_pts_vente` Ce paramètre est calculé en faisant une somme sur tous les détaillants avec la donnée `nb_pts_vente`.
- `nb_global_spiritueux` Ce paramètre est calculé en faisant une somme sur tous les détaillants avec la donnée `nb_spiritueux`.
- `nb_global_huile_region` Ce paramètre est indexé par les régions. Il est calculé en faisant une somme sur tous les détaillants. Suivant le fait qu'il corresponde à la région courante `rg`, la somme s'appliquera avec la donnée `nb_huile` sinon 0.
- `nb_global_detaillant_categorie` Ce paramètre est indexé par les catégories. Il est calculé en faisant une somme sur tous les détaillants. Suivant le fait qu'il corresponde à la région courante `ct`, la somme s'appliquera avec 1 sinon 0.
- `nhuile_region_detaillant` Ce paramètre est de même nature que sa version globale à l'exception du fait qu'il ne fait pas la somme et qu'il est donc indexé par détaillant.
Nombre d'huiles par région et par détaillant.
- `ndetail_categorie_detaillant` Idem que ci dessus
Nombre de détaillants par région et par détaillant. (Soit 0, soit 1)

Variables

Dans le modèle proposé, les variables utilisées sont au nombre de deux

La variable `affectation` représente le choix d'affectation. Elles lient une division à un détaillant. On peut la représenter comme un tableau `DIVISIONS * DETAILLANTS`. De ce fait, cette variable est de type binaire, si elle est égale à 1, elle indique que le détaillant est

affecté à la division courante. En conséquence, c'est grâce à cette dernière que les choix sont exprimés par rapport aux contraintes.

Représentation :

	D1	D2
M1	0	1
M2	1	0
M3	1	0

La variable `ammo_tol` représente le pourcentage de tolérance que nous avons réussi à réduire, par exemple, si le seuil de tolérance est de 5% et que `ammo_tol` est égale à 1%, alors la tolérance appliquée lors de la résolution saura de $5 - 1 = 4\%$. Cette variable sert donc à essayer de réduire la tolérance au plus proche de 0 voire à la rendre nulle.

Les bornes de cette variable sont de zéro à `tolerance/100`.

Objectif

L'objectif du modèle est de maximiser la variable `ammo_tol`. Plus cette valeur est forte, plus elle amortit la tolérance renseignée et peut même l'annuler. Ainsi, la solution trouvée à une tolérance qui tend vers 0.

Contraintes

Les contraintes permettent d'exprimer le problème. Elle lie et donnent du sens aux variables et paramètres du modèle. Ainsi bien exprimées, elles permettent de trouver une solution si elle existe.

Contraintes générales sur les variables

- affectation :

Dans notre solution proposée, les contraintes exprimées avec cette variable permettent de donner les conditions d'affectation liées au rapport, entre un détaillant et une division.

De façon générale, dans la partie gauche de l'inéquation, on retrouve la variable multipliée par un élément qui définit le détaillant, la partie de droite appliquant le rapport concerné avec la somme globale de l'élément utilisé.

Exemple :

```
(sum {dt in DETAILLANTS} affectation [dt, dv] * nb_pts_vente [dt]) - ammo_tol *
nb_global_pts_vente >= rpm_tol_min [dv] * nb_global_pts_vente;
```

- `ammo_tol` :

De façon générale, cette variable est utilisée dans la partie gauche de l'inéquation pour amortir la tolérance appliqué de l'autre coté. On la multiplie donc avec l'élément courant. Suivant le fait que la contrainte traite de la borne inférieur ou supérieur, elle sera soustrait ou additionné.

Exemple :

Contrainte sur la borne inférieure traitant du rapport sur le nombre de points de vente
`(sum {dt in DETAILLANTS} affectation [dt, dv] * nb_pts_vente [dt]) - ammo_tol * nb_global_pts_vente >= rpm_tol_min [dv] * nb_global_pts_vente;`

Même contrainte sur la borne supérieure

`(sum {dt in DETAILLANTS} affectation [dt, dv] * nb_pts_vente [dt]) + ammo_tol * nb_global_pts_vente <= rpm_tol_max [dv] * nb_global_pts_vente;`

Contraintes sur les bornes

Les contraintes sur les bornes concernent le domaine du paramètre ou de la variable, (supérieur à zéro, inférieur à une valeur donnée). Elles permettent d'exclure des situations illogiques.

Exemple :

- Nombre d'huiles négatif
- Tolérance supérieur au rapport

La plus par du temps elles sont exprimés directement à la déclarations

```
param nb_huile {DETAILLANTS} integer >= 0;
var ammo_tol >= 0, <= (tolerance/100);
```

On peut l'exprimer de façon plus classique

```
subject to tolerance_max {dv in DIVISIONS} :
    tolerance <= rapport_marche [dv];
```

Contrainte d'unicité

Cette contrainte est très importante, du fait que la variable `affectation` étant indexé d'une part par `DIVISIONS`, elle permet de garantir le fait qu'un détaillant ne sera pas affecté dans plusieurs divisions. Pour faire cela, pour chaque détaillant, on fait une somme indexée par ses divisions sur ses affectations. Celle-ci, ne doit pas être supérieure 1. Celle-ci, ne doit pas être supérieure 1.

Une petite ambiguïté dans le sujet, nous a contraint à faire un choix. En effet vu que l'ont à une tolérance possible, une solution qui respecte un rapport de 40/58 peut

être une réponse pertinente, malgré le fait qu'ils y ait des divisions qui ne sont pas réparties.

Nous avons choisi d'exclure ce cas, ce qui signifie que l'on doit forcément être égale à 1 dans la contrainte. Cela a pour conséquence d'obliger un détaillant à être affecté à une division.

Représentation

aff	D1	D2	Commentaire
M1	0	1	SUM DIVISION in d aff ["M1, d] == 1
M2	1	0	Idem avec M2
M3	1	0	Idem avec M3

```
subject to affectation_unique_detaillant {dt in DETAILLANTS} :
    sum {dv in DIVISIONS} affectation [dt, dv] == 1;
```

Problèmes rencontrés

Les problèmes rencontrés durant la réalisation du travail était principalement de l'ordre du manque de pratique de certains cas, vus en théorie, mais pas en pratique. Par exemple :

- L'utilisation concrète de variables binaires
- Bien cerner (séparer) les différentes variables tout en évitant les doublons sur les points communs (l'utilisation d'un seul paramètre de tolérance pour l'ensemble des variables devant respecter cette contrainte par exemple).

Conclusion

Ce projet nous aura sensibilisé sur les différents problèmes que l'on peut rencontrer dans la recherche opérationnelle. Il nous a permis d'explorer et de découvrir différentes approches, notamment les variables binaires, les paramètres calculés, les contraintes bornés, ainsi que les subtilités du langage AMPL. D'une manière générale, cela aura contribué à nous introduire une nouvelle façon d'aborder des problèmes algorithmiques.

Code AMPL

rapport_marche.mod

```

/* Commande script AMPL */
reset;

/* Solveur à utiliser : GUROBI, (permet de résoudre des modèles linéaires contenant des
nombres entiers */
option solver gurobi;

/* Ensembles ----- */
set DIVISIONS;
set REGIONS;
set DETAILLANTS;
set CATEGORIES;

/* Paramètres ----- */
/* Paramètres représentant les rapports ainsi que la tolérance qui doivent être
appliqués au modèle */
param rapport_marche {DIVISIONS} >= 0, <= 100;
param tolerance >= 0, <= 100;

/* Paramètres indicés par les détaillants */
param region {DETAILLANTS} symbolic in REGIONS;
param categorie {DETAILLANTS} symbolic in CATEGORIES;
param nb_huile {DETAILLANTS} integer >= 0;
param nb_pts_vente {DETAILLANTS} integer >= 0;
param nb_spiritueux {DETAILLANTS} integer >= 0;

/* Paramètres calculés */
/* Paramètres calculés, nombre d'huile par région et pour chaque détaillant */
param nhuile_region_detaillant {rg in REGIONS, dt in DETAILLANTS} :=
    if region [dt] = rg then nb_huile [dt] else 0;

/* Paramètres calculés, nombre de détaillant par catégorie et pour chaque détaillant */
param ndetail_categorie_detaillant {ct in CATEGORIES, dt in DETAILLANTS} :=
    if categorie [dt] = ct then 1 else 0;

/* Paramètres calculés, nombre global de points de ventes */

```

```

param nb_global_pts_vente := sum {dt in DETAILLANTS} nb_pts_vente [dt];

/* Paramètres calculés, nombre global des spiritueux */
param nb_global_spiritueux := sum {dt in DETAILLANTS} nb_spiritueux [dt];

/* Paramètres calculés, nombre d'huile global par regions */
param nb_global_huile_region {rg in REGIONS} :=
    sum {dt in DETAILLANTS} if region [dt] = rg then nb_huile [dt] else 0;

/* Paramètres calculés, nombre de détaillants global par categories */
param nb_global_detaillant_categorie {ct in CATEGORIES} :=
    sum {dt in DETAILLANTS} if categorie [dt] = ct then 1 else 0;

/* Paramètres calculés par rapport au borne donné par l'utilisateur et par la tolerance
*/
param rpm_tol_min {dv in DIVISIONS} := (rapport_marche [dv] - tolerance) / 100;
param rpm_tol_max {dv in DIVISIONS} := (rapport_marche [dv] + tolerance) / 100;

/* Variables ----- */
/* Variables binaire representant le fait que la ligne est validé ou non, en fonction
du rapport concernant le nombre de points de vente */
var affectation {dt in DETAILLANTS, dv in DIVISIONS} binary;

/* Variable flottante representant un amortissement à la tolerance appliqué */
var ammo_tol >= 0, <= (tolerance/100);

/* Objectif ----- */
maximize
obj_ammortissement_tolerance :
    ammo_tol;

/* Contraintes ----- */
/* Contraintes d'unicité */
subject to
affectation_unique_detaillant {dt in DETAILLANTS} :
    sum {dv in DIVISIONS} affectation [dt, dv] == 1;

subject to
tolerance_max {dv in DIVISIONS} :
    tolerance <= rapport_marche [dv];

/* Contraintes sur les differents rapports */

```

```

/* Rapport sur le nombre de points de ventes ... */
subject to
rapport_nb_pts_vente_global_min {dv in DIVISIONS}:
    (sum {dt in DETAILLANTS} affectation [dt, dv] * nb_pts_vente [dt]) - ammo_tol *
nb_global_pts_vente >= rpm_tol_min [dv] * nb_global_pts_vente;

subject to
rapport_nb_pts_vente_global_max {dv in DIVISIONS}:
    (sum {dt in DETAILLANTS} affectation [dt, dv] * nb_pts_vente [dt]) + ammo_tol *
nb_global_pts_vente <= rpm_tol_max [dv] * nb_global_pts_vente;

/* Rapport sur le marche du spiritueux ... */
subject to
rapport_marche_spiritueux_global_min {dv in DIVISIONS}:
    (sum {dt in DETAILLANTS} affectation [dt, dv] * nb_spiritueux [dt]) - ammo_tol *
nb_global_spiritueux >= rpm_tol_min [dv] * nb_global_spiritueux;

subject to
rapport_marche_spiritueux_global_max {dv in DIVISIONS}:
    (sum {dt in DETAILLANTS} affectation [dt, dv] * nb_spiritueux [dt]) + ammo_tol *
nb_global_spiritueux <= rpm_tol_max [dv] * nb_global_spiritueux;

/* Rapport sur le marche de l'huile dans chaque regions ... */
subject to
rapport_marche_huile_region_min {dv in DIVISIONS, rg in REGIONS}:
    (sum {dt in DETAILLANTS} affectation [dt, dv] * nhuile_region_detaillant [rg,
dt]) - ammo_tol * nb_global_huile_region [rg] >= rpm_tol_min [dv] *
nb_global_huile_region [rg];

subject to
rapport_marche_huile_region_max {dv in DIVISIONS, rg in REGIONS}:
    (sum {dt in DETAILLANTS} affectation [dt, dv] * nhuile_region_detaillant [rg,
dt]) + ammo_tol * nb_global_huile_region [rg] <= rpm_tol_max [dv] *
nb_global_huile_region [rg];

/* Rapport sur le nombre de detaillant dans chaque categorie ... */
subject to
rapport_nb_detaillant_categorie_min {dv in DIVISIONS, ct in CATEGORIES}:
    (sum {dt in DETAILLANTS} affectation [dt, dv] * ndetail_categorie_detaillant
[ct, dt]) - ammo_tol * nb_global_detaillant_categorie [ct] >= rpm_tol_min [dv] *
nb_global_detaillant_categorie [ct];

```

```

subject to
rapport_nb_detaillant_categorie_max {dv in DIVISIONS, ct in CATEGORIES}:
    (sum {dt in DETAILLANTS} affectation [dt, dv] * ndetail_categorie_detaillant
[ct, dt]) + ammo_tol * nb_global_detaillant_categorie [ct] <= rpm_tol_max [dv] *
nb_global_detaillant_categorie [ct];

/* Inclusion des données */
data "repartition_marche.dat";

/* Commande Script AMPL */
/* Resolution du model */
solve;

/* Affichage des données du model */
display affectation;

for {dv in DIVISIONS} {
    printf "%s : ", dv;
    for {dt in DETAILLANTS} {
        if affectation [dt, dv] == 1 then {
            printf "%s ", dt;
        }
    }
    printf "\n";
}

```

repartition_marche.dat

```
data;

set DIVISIONS := D1 D2;

set DETAILLANTS := M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 M15 M16 M17 M18
M19 M20 M21 M22 M23;

set REGIONS := R1 R2 R3;

set CATEGORIES := A B;


param tolerance := 5;


param rapport_marche :=
D1 40
D2 60;


param : region  nb_huile nb_pts_vente nb_spiritueux categorie :=
M1      R1      9      11      34      A
M2      R1      13     47     411     A
M3      R1      14     44     82      A
M4      R1      17     25     157     B
M5      R1      18     10      5      A
M6      R1      19     26     183     A
M7      R1      23     26     14      B
M8      R1      21     54     215     B
M9      R2      9      18     102     B
M10     R2      11     51     21      A
M11     R2      17     20     54      B
M12     R2      18    105      0      B
M13     R2      18      7      6      B
M14     R2      17     16     96      B
M15     R2      22     34     118     A
M16     R2      24    100     112     B
M17     R2      36     50     535     B
M18     R2      43     21      8      B
```

M19	R3	6	11	53	B
M20	R3	15	19	28	A
M21	R3	15	14	69	B
M22	R3	25	10	65	B
M23	R3	39	11	27	B;