

repartition_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 ayant comme ensemble de données, régions (type de donnée région) */
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 régions */
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 catégories */
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 donnée par l'utilisateur et par la tolérance */
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 représentant le fait que la ligne est validée ou non, en fonction du
```

repartition_marche.mod

```
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
----- */
/* Objectif fixe */
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] *
```

repartition_marche.mod

```
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";
}
```