

# Projet ZZ1 - Présentation du code source

Hugo Bayoud et Antony Vigouret

13 avril 2019

## 1 Présentation :

Le contenu de ce rapport explique comment s'agance le code source, écrit en C, que nous avons écrit pour la résolution du programme linéaire à l'aide de la bibliothèque Gurobi.

Note : L'ensemble du code est disponible sur : <https://github.com/hugobayoud/ProjetZZ1.git>

## 2 Structure du rapport :

Pour ce projet nous adoptons la structure suivante :

```
|
|   main.c
|   makefile
|   Test_NonG.txt
|
|---include
|   └─ // FICHIERS INCLUDES (.h)
|
|---src
|   └─ // FICHIERS SOURCES (.c)
```

Le fichier "*Text\_NonG.txt*" correspond au fichier d'entrée qui contient l'ensemble des informations sur les choix de projets que chaque binôme et de leur poids respectifs. La structure de ce fichier est explicitée dans le rapport *Projet ZZ1 - Problème de généralisation*.

### 3 Structure de données :

#### 3.1 MATRICE :

Le fichier d'entrée était implémenté de la manière suivante dans "*matrice.h*" :

```
/*Structure qui regroupe l'ensemble
des données d'entrée utiles*/

typedef struct donneesEntree
{
    int    nbBinome;
    int    nbProjet;
    int    nbChoix;
    int    *poids;
    int    **choix;
    int    **res;
} Datas_t;
```

Néanmoins, nous avons remarqué lorsque nous essayons pour la première fois les instances de test que les numéros de projets n'étaient nécessairement des entiers de 1 à 'Nombre\_de\_projet'.

Or **gurobi a besoin d'entiers allant de 1 à 'Nombre\_de\_projet' pour la résolution**. Nous créons donc un **tableau de correspondances** pour récupérer le vrai numéro de projet et donner un fichier de sortie que Mme Chabrol peut lire facilement.

Ainsi, la structure finale est de la forme :

```
/*Structure qui regroupe l'ensemble
des données d'entrée utiles*/

typedef struct donneesEntree
{
    int    nbBinome;
    int    nbProjet;
    int    nbChoix;
    int    *poids;
    int    **choix;
    int    **res;
    int    **correspondance;
} Datas_t;
```

## 4 Utilisation et Commandes :

Les commandes sont les suivantes :

- Lancer la compilation "make".
- Lancer les tests du programme "./prog"
  1. "./prog1" : lancement de gurobi avec le fichier d'entrée "Test\_NonG.txt".
  2. "./prog2" : lancement de la génération d'un fichier "Test\_G.txt" d'entrée en demandant nbBinome, nbProjet, nbChoix, puis lancement de gurobi.

*Note :* Des tests sont faits dans la fonction "main" pour être sûr que l'argument en entrée, lors de l'exécution de "./prog", soit bien un 1 ou un 2.

*Note 2 :* le fichier "main.c" ne contient que les appels de génération de fichier d'entrée test et de lancement de gurobi pour la résolution.

## 5 Fonctions, prototypes et données d'entrée :

Dans cette partie nous allons détailler les fonctions disponibles dans le code.

### 5.1 Fonctions liées aux Matrices :

Voici toutes les fonctions disponibles dans "matrice.c".

*Note :* Les fonctions utilisent des éléments de la structure "Datas\_t" qui a été défini plus haut dans ce rapport.

#### 5.1.1 Fonction - "FichierEnStructureDatas"

La fonction "FichierEnStructureDatas" : Affecter les données contenues dans le fichier d'entrée dans une structure.

**Signature :** Datas\_t\* FichierEnStructureDatas(FILE \*fic);

- Entrée :
  - fil (FILE \*) : Pointeur du fichier d'entrée qui contient l'ensemble des données d'entrée.
- Sortie
  - Datas (Datas\_t \*) : Pointeur vers Datas\_t.

### 5.1.2 Fonction - "*StructureEnMatrice*"

La fonction "*StructureEnMatrice*" : création de la matrice d'entrée (pour gurobi) à partir de la structure "*Datas\_t*".

**Signature :** `int** StructureEnMatrice(Datas_t *Datas);`

- Entrée :
  - *Datas* (*Datas\_t* \*) : Pointeur de la structure de données.
- Sortie :
  - *matrice* (int \*\*) : double pointeur pour regrouper les données de la structure sous forme de matrice.

### 5.1.3 Fonction - "*NumProjet\_BinomeChoix*"

La fonction "*NumProjet\_BinomeChoix*" : donner le N° de projet du  $i^{me}$  choix du  $j^{ime}$  binôme.

**Signature :** `int NumProjet_BinomeChoix(Datas_t *Datas, int numBinome, int numChoix);`

- Entrée :
  - *Datas* (*Datas\_t* \*) : Pointeur de la structure de données.
  - *numBinome* (int) : N° du binôme auquel on veut connaître un choix de projet.
  - *numChoix* (int) : N° choix du projet du binôme auquel on veut connaître le N° de projet associé.
- Sortie :
  - *res* (int) : N° projet de "*numChoix*" choix du "*numBinome*" binôme.

### 5.1.4 Fonction - "*NumChoix\_BinomeProjet*"

La fonction "*NumChoix\_BinomeProjet*" : donner le N° d'un choix d'un projet pour un binôme donné.

**Signature :** `int NumChoix_BinomeProjet(Datas_t *Datas, int numBinome, int numProjet);`

- Entrée :
  - *Datas* (*Datas\_t* \*) : Pointeur de la structure de données.
  - *numBinome* (int) : N° du binôme auquel on veut connaître un N° de choix à partir d'un N° de projet.
  - *numProjet* (int) : N° projet auquel on veut connaître le N° de choix associé pour un binôme donné.
- Sortie :
  - *res* (int) : N° projet de "*numChoix*"<sup>*ieme*</sup> choix du "*numBinome*" binôme.

### 5.1.5 Fonction - "*CreationFichierDeSortie*"

La fonction "*CreationFichierDeSortie*" : Création du fichier de sortie sous forme de tableau [N° de binôme, N° de projet obtenu].

**Signature :** `void CreationFichierDeSortie(double *sol, Datas_t *Datas);`

- Entrée :
  - po (Pile\_t \*) pile dans laquelle ajouter l'elt.
  - elt (Element\_pile) élément à ajouter.
  - code (code\_t\*) code erreur.
- Sortie
  - void.

## 5.2 Générer un fichier d'entrée :

La génération d'un fichier d'entrée pour la résolution est implémentée dans "*generer\_entree.c*".

**La génération de fichier d'entrée permet uniquement de tester notre résolution via Gurobi. Elle n'est pas vouée à rester si les seuls fichiers d'entrée sont ceux donnés par Michelle Chabrol** une fois par an lors de l'affectation des choix de projets aux ZZ1.

Une fonction est disponible pour générer un fichier d'entrée avec le nombre de binôme, projet et choix par binôme souhaités pour ce fichier.

*Note : Les fonctions utilisent des éléments de la structure "Datas\_t" qui a été défini plus haut dans ce rapport.*

### 5.2.1 Fonction - "*GenerateurDonneesEntree*"

La fonction "*GenerateurDonneesEntree*" : génère automatiquement un fichier d'entrée exploitable par la suite pour gurobi.

**Signature :** `void GenerateurDonneesEntree(int nbBinome, int nbProjet, int nbChoix);`

- Entrée :
  - nbBinome (int) : Nombre de binômes souhaités dans le fichier.
  - nbProjet (int) : Nombre de projets souhaités dans le fichier.
  - nbChoix (int) : Nombre de choix laissés à chaque binôme.
- Sortie
  - void.

## 5.3 Lancement de Gurobi :

Deux fonctions sont disponibles pour appeler un fichier d'entrée puis lancer la résolution avec gurobi. Ces fonctions sont implémentées dans : "*GUROBI.c*"

### 5.3.1 Fonction - "*LancementGurobi1*"

La fonction "*LancementGurobi1*" permet de lancer la résolution avec gurobi. Que le fichier d'entrée doive être généré ou non, "*LancementGurobi1*" est appelée.

**Signature :** `void LancementGurobi1(char *Fichier);`

- Entrée :
  - void.
- Sortie
  - void.

### 5.3.2 Fonction - "*LancementGurobi2*"

La fonction "*LancementGurobi2*" : permet de faire appel à la fonction "*GénérateurDonnéesEntrée*" puis de lancer gurobi via "*LancementGurobi1*".

**Signature :** `void LancementGurobi2();`

- Entrée :
  - void.
- Sortie
  - void.

## 6 Notes :

L'ensemble du code source est disponible sur *github* : <https://github.com/hugobayoud/ProjetZZ1.git>.

Chaque fonction est commentée un minimum.