

Rendu n°6

Lucas Gonçalves, Fatima Machhour, Raphael Luciano

Partie 1 : premier rendu

Dans ce premier rendu, nous souhaitons partager notre idée de réponse au problème de la répartition entre un ensemble d'élèves au sein de différents groupes. Notre problème se base sur le système de notation proposé dans "Le jugement majoritaire" de Michel Balinski et Rida Laraki. Ce système se base sur le principe que chaque participant vote pour chacun des "représentants" disponibles.

1. Enoncé de notre problème:

Dans notre cas, nous souhaitons affecter un ensemble d'élèves sur un ensemble de projets (maximum 18 projets). Si notre groupe d'élèves est composé de plus de 36 élèves, nous devons constituer des groupes de 3 personnes. Dans notre cas, les données fournies sont constituées de plus de 36 élèves, nous allons donc devoir créer des groupes constitués de trois personnes.

Partie 2 : Démonstration de notre idée - deuxième rendu

Chaque étape supprime les arêtes du plus bas niveau atteint. Idée générale de l'algorithme : L'algorithme améliore à chaque étape la qualité des liens entre les élèves puisqu'on supprime les arêtes en commençant par I ($I < P < AR < AB < B < TB$). De plus on ne crée un groupe que lorsqu'on ne peut pas faire mieux ie lorsqu'un élève S à un degré entrant ≤ 2 . L'algorithme garantit qu'il ne peut y avoir de meilleur groupe pour l'élève S.

Plus formellement : Etape i, Arc de niveau a A l'initialisation, le niveau a est « I » (Insuffisant). L'algorithme déroule les niveaux selon l'ordre suivant : I, P, AR, AB, B, TB. 1 *Invariant* : Au début de l'étape il n'y a aucun sommet ayant un degré entrant ≤ 2 . On supprime toutes les arêtes de niveau a pour améliorer la qualité de nos liens. Si aucun sommet n'a un degré entrant ≤ 2 on

passer alors à l'étape Sinon : on se retrouve avec des sommets ayant un degré ≤ 2 . Il faut alors rétablir l'invariant. 3 cas de figure :

- si un sommet S a un degré entrant $= 2$, on forme un groupe avec S et les 2 sommets extrémités $S1$ et $S2$ des arcs entrants.
- si un sommet S a un degré entrant $= 1$, on forme un groupe avec S , l'extrémité de l'arc entrant $S1$ et on complète avec un le sommet $S2$ ayant le degré sortant maximum.
- si un sommet S a un degré entrant $= 0$, on forme un groupe avec S et les 2 sommets $S1$ et $S2$ ayant les degrés sortants maximum. En supprimant les sommets S , $S1$ et $S2$ du graphe, on rétablit l'invariant et on peut passer à l'étape. Ainsi, on a garanti notre invariant à la fin de notre étape. On peut passer à l'étape $i+1$ en considérant les arcs de niveau $a+1$ et recommencer. On s'arrête des lors que notre graphe ne possède plus aucun sommet. Tous nos groupes auront été alors formé. Lemme1: Au tour de boucle actuel, il n'existe pas de sommet v appartenant à V tel que $\deg_{\text{In}}(v) \leq 2$.*

1 Introduction généralités et notations

Notre problème se base sur le système de notation proposé dans “Le jugement majoritaire” de Michel Balinski et Rida Laraki. Ce système se base sur le principe que chaque participant vote pour chacun des “représentants” disponibles (qui sont ici représentés par les étudiants). Nous souhaitons affecter un ensemble d'étudiants sur un ensemble de projets (maximum 18 projets). Dans notre cas, chaque étudiant de la promotion note les autres étudiants en lui attribuant l'une des notes suivantes : AR, I, P, AB, B, TB. Si notre groupe d'élèves est composé de plus de 36 élèves, nous devons constituer des groupes de 3 personnes en plus des binomes. Dans notre cas, les données fournies sont constituées de plus de 36 élèves, nous allons donc devoir créer des groupes constitués de deux et de trois personnes.

2 Méthode MGL

Afin de résoudre le problème demandé, nous souhaitons mettre en place un système d'algorithme en transformant notre problème par un problème de graphe :

Prenons quatre élèves $e1$, $e2$, $e3$, $e4$:

Donnons les informations suivantes avec les notations suivantes:

- TB : “Très bien”
- B : “Bien”
- AB : “Assez bien”
- P : “Passable”
- I : “Insuffisant”

- *AR* : “A rejeter” notation supposée en l’absence d’avis sur l’élève en question.

La valeur -1 est présente sur les diagonales car un élève ne s’évalue pas lui-meme

Eleves	e1	e2	e3	e4
e1	-1	TB	B	AB
e2	AR	-1	I	I
e3	AR	B	-1	TB
e4	P	B	AB	-1

Table 1 : exemple d’instanciation de notre matrice de préférences.

Nous souhaitons modéliser notre problème par un graphe orienté. **Chaque sommet représente un élève** et **chaque arc représente une notation de l’élève source vers l’élève destination**.

A chaque étape, on supprime toutes les aretes du plus bas niveau ($I < P < AR < AB < B < TB$). Pour l’étape 1, on supprimerait par exemple toutes les aretes qui ont la valeur “I” (I de plus bas niveau a l’étape 1. A l’étape suivante, le niveau le plus bas sera P...).

Si un sommet S a un degré entrant < 2 a la fin de cette opération, on conserve les arcs en provenance des sommets vers lesquels l’arc sortant de S est maximum. On forme un groupe avec les sommets ayant un arc de destination S.

Dans l’exemple, si on supprime tous les arcs ayant la valeur “I” (qui est la valeur la plus mauvaise dans notre graphe), on remarque que le sommet se retrouve avec un seul arc entrant (l’arc venant de e4). Il a donc un degré entrant < 2 ($\text{degIn}(e1) = 1$), il faut alors garder un des deux arcs précédemment supprimés pour compléter (on veut un degré entrant = 2). On choisira alors de conserver l’arc allant de e2 vers e1, car e1 a une préférence supérieure pour e2. On considère donc que e1 est dans le groupe de e2 et e4, car ce sont les deux seul sommets pour lesquels e1 possède un arc entrant. On affecte donc ces 3 sommets dans un groupe et on les retire de notre graphe pour qu’ils ne soient plus pris en compte dans les affectations suivantes.

Eleves	e1	e2	e3	e4
e1	-1	TB	B	AB
e2	I	-1	AR	AR
e3	\emptyset	B	-1	TB
e4	P	B	AB	-1

Table 2 : Version de l’instance a la fin de l’étape 1

On a ici créé le groupe $\{e1, e2, e3\}$.

On passe a l'étape suivante et on recommence jusqu'a supprimer toutes les aretes du graphe. On aura ainsi créé nos groupes en maximisant les préférences.

2.1 Propriétés de la méthode

Notre méthode permet d'augmenter la valeur du minimum en augmentant a chaque étape la qualité des relations entre les élèves. Ainsi, le seuil de satisfaction minimal augmente sans cesse. On cherche a ce que la pire relation soit la meilleure possible.

Propriété mathématique : Etape après étape, la médiane des relations entre les membres ne fait qu'augmenter.

Il est en effet possible de déterminer une médiane pour un ensemble de valeurs qualitatives si on choisit un critère d'ordonnement de ces valeurs ce qui est le cas ici. $[AR < I < P < AB < B < TB]$

En effet, en supprimant les aretes les moins bonnes a l'étape i, la médiane a l'étape i+1 ne peut etre que plus grande ou égale que celle de l'étape i. Ainsi, l'augmentation de la médiane est signe qu'on se rapproche de la meilleure qualité de relation (a savoir TB) et qu'on améliore notre seuil de satisfaction.

2.2 Description détaillée de la méthode optionnel

2.3 Exemples

Prenons neuf élèves e1, e2, e3, e4, ..., e8, e9:

Donnons les informations suivantes avec les notations suivantes:

- *TB* : "Très bien"
- *B* : "Bien"
- *AB* : "Assez bien"
- *AR* : "A rejeter" notation supposée en l'absence d'avis sur l'élève en question.
- *P* : "Passable"
- *I* : "Insuffisant"

La valeur -1 est présente sur les diagonales car un élève ne s'évalue pas lui-meme

Dans la matrice, chaque arc sort de l'élève en ordonné, et entre dans l'élève en abscisse.

Table 3: initialisation

Eleves	e1	e2	e3	e4	e5	e6	e7	e8	e9
e1	-1	TB	B	AB	B	AB	TB	B	AR
e2	I	-1	AR	AR	TB	I	P	TB	AB

Eleves	e1	e2	e3	e4	e5	e6	e7	e8	e9
e3	AB	B	-1	TB	AB	B	B	AB	TB
e4	P	P	AB	-1	B	I	AB	TB	P
e5	I	B	B	TB	-1	AR	TB	I	B
e6	P	AR	AB	B	B	-1	B	AB	TB
e7	I	B	I	TB	AB	TB	-1	AR	I
e8	P	P	AB	AR	P	B	I	-1	TB
e9	TB	B	AB	B	I	TB	AB	I	-1

Table 4: sans les arcs insuffisants

Eleves	e1	e2	e3	e4	e5	e6	e7	e8	e9
e1	-1	TB	B	AB	B	AB	TB	B	AR
e2		-1	AR	AR	TB		P	TB	AB
e3	AB	B	-1	TB	AB	B	B	AB	TB
e4	P	P	AB	-1	B		AB	TB	P
e5		B	B	TB	-1	AR	TB		B
e6	P	AR	AB	B	B	-1	B	AB	TB
e7		B		TB	AB	TB	-1	AR	
e8	P	P	AB	AR	P	B		-1	TB
e9	TB	B	AB	B		TB	AB		-1

A cette étape, nous avons retiré tous les liens notés “insuffisants”. Rien d’autre ne se passe car tous les élèves ont encore un degré entrant >2 .

Table 5: sans les arcs passables

Eleves	e1	e2	e3	e4	e5	e6	e7	e8	e9
e1	-1	TB	B	AB	B	AB	TB	B	AR
e2		-1	AR	AR	TB			TB	AB
e3	AB	B	-1	TB	AB	B	B	AB	TB
e4			AB	-1	B		AB	TB	
e5		B	B	TB	-1	AR	TB		B
e6		AR	AB	B	B	-1	B	AB	TB
e7		B		TB	AB	TB	-1	AR	
e8			AB	AR		B		-1	TB
e9	TB	B	AB	B		TB	AB		-1

A cette étape, nous avons retiré tous les liens notés “passable”. On remarque que l’élève e1 n’a plus que 2 arc entrant. On affecte donc e1 avec e3 et e9, les deux personnes avec lesquelles il s’entendait le mieux donc.

Il faut maintenant enlever tous les arcs qui partent de ces personnes, car elles ne peuvent plus être associées à d'autres élèves:

Table 6: tableau sans les arcs sortants

Eleves	e2	e4	e5	e6	e7	e8
e2	-1	AR	TB			TB
e4		-1	B		AB	TB
e5	B	TB	-1	AR	TB	
e6	AR	B	B	-1	B	AB
e7	B	TB	AB	TB	-1	AR
e8		AR		B		-1

Une fois les élèves retirés, les élèves restant ont encore un degré entrant >2 , on peut alors continuer l'algorithme.

Table 7: tableau sans les valeurs à rejeter

Eleves	e2	e4	e5	e6	e7	e8
e2	-1		TB			TB
e4		-1	B		AB	TB
e5	B	TB	-1		TB	
e6		B	B	-1	B	AB
e7	B	TB	AB	TB	-1	
e8				B		-1

À cette étape, nous avons retiré tous les liens notés “à rejeter”. Deux élèves ont un degré entrant ≤ 2 : e2 et e6. On remarque aussi que ces deux élèves devraient être affectés avec e7. Comme e2 est dans la plus “mauvaise” des situations, c’est lui qui aura la priorité pour s’associer avec lui. Le groupe e2, e5, e7 est donc formé. Le groupe restant est alors formé avec les trois élèves restant.

Nous obtenons donc à la fin de notre algorithme les groupes suivants : - e1, e3, e9 - e2, e5, e7 - e4, e6, e8

3 Correctifs et modifications

Lors de notre premier rendu, une erreur de récupération du fichier source était présente. Désormais corrigée, cette version devrait (sauf erreur de chemin) pouvoir proposer quelques répartitions en fonction des matrices données. Néanmoins, nous savons que notre méthode n’est pas infaillible et connaît quelques erreurs en fonction des répartitions proposées en entrée.

4 Algorithme

4.1 Description de l'algorithme

Tant qu'il reste des sommets dans V: Pour tout sommet u appartenant E tq $u == i$ (avec i la valeur que l'on étudie) $E = E$ privé de v Si il existe un sommet e tel que $\text{degIn}(e) \leq 2$ alors: On associe e avec deux autres sommets dont les valeurs sont maximales on incrémente la valeur des relations Fin Pour Fin tant que fonction `repartitionGroupe`(matrice de notes , liste des niveaux, index de la liste des niveaux) : liste d'affectations de groupes Si tous les sommets ont été affecté : on retourne une liste vide Sinon Si il n'existe pas d'élève avec un degré entrant inférieur ou égal à deux : On supprime les arêtes du niveau correspond au parametre index On relance `repartitionGroupe` en augmentant l'index de niveau (appel récursif) Si il existe des élèves avec un degré entrant égal à 0: On récupère les élèves pour qui il a attribué les meilleures notes On lance notre algorithme sur toutes les combinaisons possible et on récupère la meilleure Si il existe des élèves avec un degré entrant égal à 1: On crée le groupe avec cette personne.

5. Dernières modifications:

Cette partie du rendu est consacrée aux dernières modifications apportées au programme. Nous avons amélioré notre algorithme mais notre méthode n'est toujours pas complète. Nous sommes conscients que nous rencontrons des problèmes sur des "*cas critiques*".

Parmi les cas critiques rencontrés, nous avons notés: - Matrice TB/TB ou TB/X: nous ne parvenons pas à créer une répartition malgré les différentes combinaisons que nous avons. - Retour de répartitions vides. Nous avons des situations où, en fonction de la répartition, notre méthode de résolution ne retourne rien sur la matrice de préférence normale (répartition = [])

Néanmoins, nous avons essayé d'apporter quelques solutions à ces problèmes. Dans le cas d'une matrice remplie avec une seule et unique note, nous avons essayé de créer une méthode "random" qui ressort un ensemble de répartitions composées de 18 groupes (le nombre de groupe idéal pour les projets) => méthode dans le fichier `special.py`. Cette solution nous semblait intéressante car, sachant que tout le monde souhaite travailler avec tout le monde dans cette configuration là, cela est aussi pertinent que de sortir des répartitions aléatoires. Seule une distribution par projet (dans ce cas précis) permettrait d'établir un nouvel ordre de préférence mais ce n'est pas le cas.

L'équipe en charge du script de modification s'est décidée à mettre en place ce dernier la veille du rendu à 23h. Face à cette absence de décision, nous n'avons pas eu le temps de d'adapter l'ensemble des fonctions de notre méthode. C'est pourquoi il est possible que le script actuel ne fonctionne toujours pas.

5.1 Gestion des égalités :

En cas d'égalité dans une matrice TB/TB nous avons décidé de retirer un nombre x de distributions. x étant un paramètre inscrit dans le code qui peut être changé dans la fonction `specialGroupRepartition()`. Pour cela, les premières répartitions créées sont celles qui sont renvoyées car nous n'avons pas d'autres critères discriminants permettant de favoriser une distribution par rapport à une autre. Pour ce qui est des autres types de matrices, nous avons renvoyé l'ensemble des répartitions qui sont de même niveau de qualité (médiane) que la note de la meilleure répartition parmi les x premières répartitions (x étant un paramètre).

6. Post Mortem:

6.1 problemes :

Plusieurs problemes ont été rencontré durant le projet. Le plus gros problème auquel on a du faire face durant ce projet fut la gestion du temps. En effet, ce dernier était limité et partagé entre plusieurs projets. Egalement, l'implémentation de notre solution a demandé un investissement constant. Malgré cela, nous n'avons pas réussi à terminer notre algorithme. De plus, nos nombreux problemes de chemin d'exécution ne nous ont pas permis d'avoir un retour sur les problemes liés à notre implémentation.

Nous avons beaucoup investi dans ce projet, autant en temps qu'en énergie et en bonne volonté.

6.2 Solutions et points positifs :

Ce projet nous a permis de développer un algorithme basé sur un problème mathématique concret et complexe. On a ainsi pu implémenter une solution à ce dernier en ayant travaillé sur l'aspect mathématique. Malgré la complexité de la demande qui nous a été soumise, nous n'avons pas modifié notre méthode pour des solutions plus simples (brute-force) et nous avons essayés de nous dépasser pour trouver une solution.

Heures de travail effectives : environs 75 heures. Heures travail délégué : 75 heures + 10 heures environ.