Affectation des élèves de 1A – Rapport

Février – Mars 2017

Rapport du projet de département « *Affectation des élèves de 1A* »réalisé par Hervé Andrés, Marc-Antoine Augé, Reda Bahi Slaoui, Raphaël Ginoulhac, Michaël Karpe et Imad Lakim sous la direction de Étienne Gaillard de Saint-Germain.

Table des matières

1 Présentation générale du projet

1.1 Problème posé par le département de première année

Dans le cadre de la gestion de l'affectation des élèves aux cours et projets du second semestre, il nous a été demandé de concevoir un programme informatique qui gère l'affectation des élèves aux cours. Pour qu'il réponde aux exigences du département de première année, ce programme doit respecter un certain nombre de contraintes.

Notre objectif est d'optimiser le processus d'affectation des élèves aux cours et aux projets, en proposant une répartition des élèves qui respecte au maximum le classement de leurs vœux.

En effet, nous disposons de trois grands ensembles de cours et de projets :
- les projets de département : ils sont au nombre de 20 en 2017 et se déroulent le lundi après-midi.
- les cours d'ouverture : ils se répartissent sur 4 jours, chaque jour disposant d'un ensemble bien spécifié de cours que les élèves doivent classer.
- les projets d'initiation à la recherche : ils sont associés aux cours d'ouverture. Chaque cours d'ouverture dispose d'un projet qui porte sur le même thème. De ce fait, tout élève doit nécessairement être affecté à un projet qui est associé à l'un des cours qu'il suit.

D'autre part, chaque cours et chaque projet possèdent un effectif maximal d'élèves et un effectif minimal nécessaire à l'ouverture du cours. Ces effectifs peuvent varier d'un cours à un autre. Enfin, certains cours ou projets peuvent être supprimés si la demande n'est pas assez forte.

1.2 Déroulement du projet

Le projet s'étendant sur 5 séances de 4 heures, nous avons dû nous organiser de façon efficace pour pouvoir réaliser le projet à temps. Lors de la première séance, nous avons accueilli François Chevoir qui nous a présenté l'algorithme actuellement utilisé et ses limites. François Chevoir doit procéder à des corrections manuelles, notamment pour les cours d'ouverture et les projets de recherche. L'algorithme effectue en effet une optimisation jour par jour, et non pas sur l'ensemble des jours. Pour pallier ce problème, nous nous sommes fixé comme objectif de réaliser un programme qui traite l'ensemble des affectations, à savoir : les projets de département, les cours d'ouverture et les projets de recherche.

Lors de la première séance, nous avons réalisé la modélisation théorique du problème d'affectation sous la forme d'un problème d'optimisation linéaire, pour qu'il puisse être facilement résolu par un solveur. Cette modélisation nous a pris les 4 heures de la première séance. Cela peut paraître long, mais cela nous a permis d'avoir un programme fonctionnel plus rapidement que s'il avait fallu trouver dans le code informatique des erreurs liées à cette modélisation.

Lors de la deuxième séance, nous nous sommes réparti les différentes tâches à réaliser entre et pendant les différentes séances. Ainsi, pendant ce projet, deux personnes se sont chargées du traitement des données fournies par le département de Première Année, concernant les différents cours et projets et les vœux des élèves. Deux autres se sont principalement concentrées sur la réalisation du code informatique qui résout le problème d'optimisation. Les deux personnes restants se sont occupées de l'interface et du code qui permet au département de Première Année de disposer d'un logiciel clé en main.

En parallèle de la réalisation pratique du logiciel, chaque personne du groupe a participé à la rédaction de ce rapport, de façon à ce que chaque partie de ce rapport puisse être la plus claire possible.

2 Modélisation sous la forme d'un programme d'optimisation linéaire

2.1 Programme d'optimisation pour un seul choix de projet/cours

Afin de rendre compte du fait qu'un élève a ou n'a pas reçu son premier vœu, nous avons introduit une fonction de regret dont la valeur est d'autant plus élevée que les premiers vœux ne sont pas respectés. Notre modélisation vise à minimiser la fonction de regret total. Pour un projet donné [1], la fonction de regret d'un élève correspond à la fonction de poids [2] évaluée au numéro de voeu du projet. Afin de savoir à quel projet un élève est affecté, on utilise une matrice $A = (a_{e,p})_{e,p}$ dont le coefficient $a_{e,p}$ vaut 1 si l'élève e est affecté au projet p et 0 sinon. Compte tenu de cela, la fonction de regret d'un élève pour un projet donné est $\sum_{p \in P} a_{e,p} w(c_{e,p})$. Ce problème de minimisation s'accompagne de plusieurs contraintes :

- L'effectif du projet numéroté p, noté n_p et défini comme la somme sur les élèves des affectations $a_{e,p}$, doit être compris entre l'effectif minimum m_p et l'effectif maximum M_p associés à ce projet. Ces bornes sont multipliées par une variable binaire δ_p qui est égale à 0 si le projet est fermé et 1 s'il est ouvert. Ainsi, s'il n'y a pas assez d'élèves qui ont demandé un projet donné, c'est-à-dire qu'il ne figure pas dans les premiers voeux d'un nombre suffisant d'élèves, alors le projet en question aura un effectif nul.
- Chaque élève doit être affecté à un seul et unique projet. À cet effet, la somme sur les projets des affectations $a_{e,p}$ de l'élève est égale à 1.

Mathématiquement, le problème ainsi modélisé s'écrit :

$$\begin{cases} \text{Min} & \displaystyle\sum_{e \in E} \sum_{p \in P} a_{e,p}.w(c_{e,p}) + K r_{max} & & & \\ \text{s.c.} & n_p & = & \sum_{e \in E} a_{e,p} & \forall p \in P \\ & \delta_p.m_p & \leq & n_p & \forall p \in P \\ & n_p & \leq & \delta_p.M_p & \forall p \in P \\ & \sum_{p \in P} a_{e,p} & = & 1 & \forall e \in E \\ & \delta_p & \in & \{0,1\} & \forall p \in P \\ & a_{e,p} & \in & \{0,1\} & \forall p \in P, \forall e \in E \end{cases}$$

2.2 Paramètres

- J : Ensemble des *« jours »*, deux cours ou projets le même jour ne peuvent pas être cumulés.
- E : Ensemble des élèves.
- P : Ensemble de tous les projets / cours à choisir partitionnés selon les jours $P = \bigcup P_j$, $j \in J$.
- C : Matrice des choix des élèves tq. $c_{e,p}$ est le rang du projet p pour l'élève e.
- m_p et M_p : Effectifs minimal et maximal pour le projet p.
- w : Fonction de poids.
- pr : Fonction qui, à un cours d'ouverture, associe un projet d'initiation à la recherche.
- K : Constante majorant le regret total.

2.3 Variables

- $\delta_p = 1$ si le projet p est ouvert, 0 sinon.
- A : Matrice d'affectation des élèves tq $a_{e,p} = 1$ si l'élève e est affecté au projet p, 0 sinon.
- n_p : Effectif dans le projet p.
- r_{max} : Regret par élève maximal. Éventuellement un paramètre.

1. On considère ici des projets mais la modélisation est strictement identique pour des cours d'ouverture.
2. La fonction de poids est une fonction qui, à un numéro de voeu, associe un nombre d'autant plus élevé que le numéro du voeu est élevé, et qui vaut 0 s'il s'agit du voeu 1 de l'élève.

2.4 Programme d'optimisation pour plusieurs choix de projets/cours

Lorsqu'il y a plusieurs projets ou cours à choisir, il faut que les résultats d'affectation tiennent compte les uns des autres. Ainsi, si par exemple un étudiant est affecté à un projet placé en voeu 3, ses autres affectations doivent correspondre, dans la mesure du possible, à des cours placés en voeu 1. Pour cela, on ajoute une variable r_{max} qui correspond au regret maximal par élève. Le regret de chaque élève ($\sum_{p\in P} a_{e,p}.w(c_{e,p})$) est donc majoré par r_{max}. De façon à minimiser ce regret maximum, on ajoute cette variable au regret total dans la fonction à minimiser avec un coefficient multiplicatif K. Ce coefficient est un majorant du regret total, donc un nombre très grand, de façon à ce que le regret maximal par élève n'augmente que s'il n'y a pas d'autre possibilité. Enfin, il faut faire en sorte que si un élève est attribué à un projet de recherche, alors il est nécessairement attribué au cours d'ouverture associé. En conséquence, la variable d'affectation $a_{e,p}$ à un cours d'ouverture donné p majore la variable d'affectation $a_{e,pr(p)}$ du projet de recherche associé. D'où la reformulation du problème précédent sous la forme :

$$\begin{cases} \text{Min} & \displaystyle\sum_{e\in E}\sum_{p\in P} a_{e,p}.w(c_{e,p}) + Kr_{max} & & & \\ \text{s.c.} & n_p & = & \sum_{e_\in E} a_{e,p} & \forall p \in P \\ & \delta_p.m_p & \leq & n_p & \forall p \in P \\ & n_p & \leq & \delta_p.M_p & \forall p \in P \\ & \sum_{p\in P} a_{e,p}.w(c_{e,p}) & \leq & r_{max} & \forall e \in E \\ & \sum_{p\in P_j} a_{e,p} & = & 1 & \forall j \in J \\ & a_{e,pr(p)} & \leq & a_{e,p} & \forall e \in E, \forall p \in P_{\text{c.o.}} \\ & \delta_p & \in & \{0,1\} & \forall p \in P \\ & a_{e,p} & \in & \{0,1\} & \forall p \in P, \forall e \in E \end{cases}$$

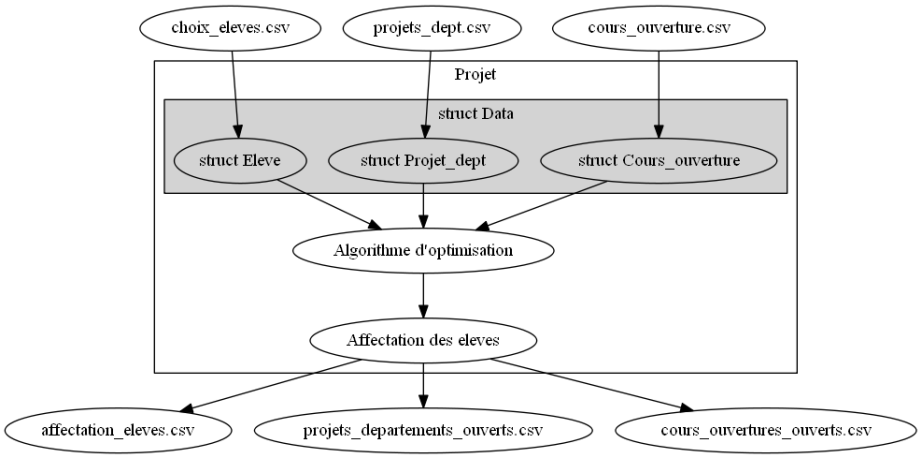
3 Le code

3.1 Utilisation d'un solveur de résolution linéaire pour résoudre le problème

Pour résoudre ce problème linéaire, nous avons choisi d'utiliser le solveur **GLPK : The GNU Linear Programming Kit** , qui est libre et open source, sous la licence **GPL** et qui est également le solveur que nous avions utilisé pour le projet d'optimisation au premier semestre. Il contient de plus une bibliothèque pouvant être utilisée par un programme C++. Compte tenu du faible nombre de variables entières du problème (inférieur à 1000), GLPK fournit une solution optimale en quelques minutes dans les cas non-dégénérés et en quelques dizaines de minutes dans les pires cas, ce qui est tout à fait satisfaisant.

3.2 Structure globale du code

Après avoir réalisé la formalisation de notre problème linéaire comme expliqué précédemment, nous avons procédé au codage informatique du problème. Le principe global est schématisé ci-dessous. Notre code prend en argument des données brutes au format CSV (données séparées par des virgules). Il lit ces données pour les affecter dans des structures telles que `Eleve`, `Projet_dept` et `Cours_ouverture`, ce qui rend les données bien plus facilement exploitables. L'algorithme d'optimisation correspond à la formalisation informatique du problème linéaire présenté en partie 2.4. Ce code résout le problème d'optimisation et renvoie de nouvelles données. Ces données en sortie de programme correspondent à l'affectation des élèves aux différents cours et projets, en fonction des vœux des élèves présents dans les fichiers en entrée. Enfin, les données en sortie de l'algorithme d'optimisation sont mises sous la forme de fichiers CSV similaires à ceux d'entrée.



4 Comparaison de nos résultats avec les résultats du programme précédent

Nous avons comparé notre programme aux résultats d'affectation obtenus en 2017.

4.1 Mise en forme des données de 2017

Avant de pouvoir tester notre programme avec les données de l'année 2017, il a fallu rassembler les données obtenues dans les différents questionnaires, compléter les colonnes manquantes et changer la numérotation des cours d'ouverture afin d'obtenir les trois fichiers que notre programme attend en entrée. Nous avons fait tout cela par l'intermédiaire de quelques scripts Python.

4.2 Des très bons résultats

Nous vous transmettons en plus de notre logiciel un fichier Excel qui comptabilise le nombre de premiers, deuxièmes et troisièmes choix par projet et par élève. Ce fichier sert ainsi à valider les résultats du logiciel.

Nous avons pu voir dans la modélisation (Section 2.4) que le paramètre fondamental de l'optimisation est la fonction de poids w. En jouant sur les poids, on peut changer entièrement le résultat de l'optimisation. Dans le cas de notre problème, on souhaite donner massivement des premiers ou des deuxièmes choix, tout en autorisant exceptionnellement les troisièmes choix.
Pour cela, nous avons travaillé avec les deux fonctions de poids (1) et (2).

Rang	Poids
1	0
2	5
3	**15**
4	1000
5	100000
6	...

TABLE 1 – Fonction de poids où l'on considère qu'il est équivalent d'avoir trois vœux 2 ou un vœu 3.

Rang	Poids
1	0
2	5
3	**50**
4	1000
5	100000
6	...

TABLE 2 – Fonction de poids où le troisième choix est déjà très fortement pénalisé

Comparons les résultats obtenus pour ces deux fonctions de poids avec les résultats de l'optimisation actuelle. Les résultats sont synthétisés sur le tableau (3). Remarquons que nous obtenons les mêmes résultats que ceux de cette année : ils correspondaient donc déjà à un optimum. Notre logiciel offre donc surtout un gain de temps puis-ce qu'il tourne en moins de dix minutes [3]. La méthode précédemment employée obligeait à reprendre manuellement les résultats de l'algorithme et durait donc environ 15h. Le gain de temps est donc considérable.
Toutefois, afin d'obtenir des résultats rapidement, il convient de contraindre le programme en prenant une fonction de poids qui croît très vite : la fonction de poids (2) nous semble à privilégier.

Remarques :
- La satisfaction pour un élève est la somme des rangs des cours où il est affecté.
- Les données initiales contenaient parfois 143 et parfois 144 élèves, d'où des totaux non cohérents avec notre optimisation réalisée sur 142 éléves.

Projet		Actuellement	Avec (1)	Avec (2)
Synthèse	Temps de résolution	$\approx$ 15 h	10 min	8 min
	Satisfaction moyenne	?	1, 15	1, 21
Projets département	Nb vœux 1	110	107	107
	Nb vœux 2	30	35	35
	Nb vœux 3	2	0	0
Cours lundi	Nb vœux 1	137	137	137
	Nb vœux 2	6	5	5
	Nb vœux 3	0	0	0
Cours mardi	Nb vœux 1	127	126	126
	Nb vœux 2	16	16	16
	Nb vœux 3	0	0	0
Cours mercredi	Nb vœux 1	113	114	113
	Nb vœux 2	30	28	29
	Nb vœux 3	0	0	0
Cours vendredi	Nb vœux 1	109	111	111
	Nb vœux 2	34	31	31
	Nb vœux 3	0	0	0
Projets recherche	Nb vœux 1	99	92	90
	Nb vœux 2	37	45	48
	Nb vœux 3	6	5	4

Comparaison des résultats entre l'algorithme actuel et notre logiciel pour deux fonctions de poids.

3. Nous avons choisi d'arrêter l'optimisation au bout de dix minutes, quitte à ce que ce ne soit pas la solution optimale.