Algorithme pour la création de colloscope

31 janvier 2024

1 Cadre et notations

On souhaite créer un colloscope en associant différents groupes (d'élèves) à des créneaux disponibles. On considère ici que les créneaux sont définis à l'avance et on se concentre sur la tâche de répartition des groupes. On considère aussi que les groupes sont définis et fixes.

Plus précisément, on propose les définitions suivantes :

Matière Peut être obligatoire ou optionnelle. On suppose que les créneaux sont compatibles avec les matières *obligatoires*. Les matières optionnelles sont regroupées par alternative (ex : Allemand/Espagnol ou HGG/ESH) et définissent leurs créneaux de cours. On suppose qu'un même groupe ne peut assister qu'à une seule matière par alternative.

Colleur Enseignant encadrant l'heure de colle. On considère qu'un colleur est associé à une unique matière.

Créneau Plage attribuée à une colle, définit par un colleur et une date (semaine et horaire). On suppose les plages cohérentes (par exemple, on suppose qu'un même colleur ne colle que sur des plages disjointes).

Groupe Ensemble d'élèves assistant aux même colles. On associe à chaque groupe 0, une, ou plusieurs options, ce qui permet de prendre en compte les contraintes additionelles d'emploi du temps.

Colloscope Associe chaque créneau à un groupe. On représente un colloscope par un ensemble de couple (g, c) où g est un groupe et c un créneau. Un colloscope est donc aussi une fonction de l'ensemble des créneaux vers l'ensemble des groupes. On note $\mathcal{C}_o$ l'ensemble des colloscopes.

Dans ce cadre, un colloscope peut être valide ou invalide et plus ou moins optimal. L'enjeu est donc de proposer un algorithme pour générer un colloscope valide et optimisé.

2 Structures de données

Le cadre précédent peut se traduire informatiquement par les types suivants. On utilise `[]Type` pour désigner une liste d'éléments de type `Type`.

```
struct Matiere {
    id: int;
    name: string;
    periode: int; /* 3 pour 1 colle toutes les 3 semaines */

    /* vide pour les matieres principales, sinon, par exemple, LV2 ou HorE */
    optionName: string;
    creneauxHebdomadaires: []Horaire; /* vide pour les matieres principales */
}

struct Horaire {
    weekDay: int /* from 1 to 7 */
    hour: int
    minute: int
    duration: int /* in minutes */
}

struct Colleur {
    id: int;
    name: string;
    matiere: Matiere;
}

struct Creneau {
    id: int;
    semaine: int;
    horaire: Horaire;
    colleur: Colleur;
    salle: string;
}

struct Groupe {
    id: int;
    options: []Matiere;
}

/* Mapping from Creneau to Groupe */
type Colloscope = {Creneau : Groupe};
```

3 Validité et caractéristiques d'un colloscope

3.1 Définitions

On définit la validité d'un colloscope par les propriétés suivantes :

(Rec) : Chaque groupe a des créneaux horaires sans recoupement (colles et options)

(Per) : Le nombre de colle de chaque groupe par matière respecte la période de la matière.

Remarque : La propriété (Per) n'est possible que si le nombre de semaines du colloscope est un multiple de chaque période.

On s'intéresse aussi aux caractéristiques suivantes.

(Eta) : L'étalement des colles pour chaque groupe dans chaque matière : on souhaite éviter deux colles (d'une même matière) trop rapprochées.

(Max) : Le nombre maximum de colles par semaine pour un groupe : idéalement, il devrait être proche du nombre moyen pour éviter les semaines trop lourdes.

(Rot) : La rotation des groupes par rapport aux colleurs : on souhaite éviter qu'un colleur voit toujours les mêmes groupes.

3.2 Implémentation

TODO : décrire une implémentation utilisant les structures de données de la section 2

4 Algorithme

On propose de regrouper les caractéristiques précédentes dans une fonction de coût $\phi : \mathcal{C}_o \to \mathbb{N}$, et d'explorer l'espace des colloscopes *valides* de manière itérative, pour minimiser ϕ.

TODO : Comment faire en pratique ? Algorithme génétique ? Transition aléatoire ?