

---

---

## Sujet : Bourse d'échange

---

---

*Le jury attend une présentation de 35 minutes s'appuyant sur ce sujet, en forme de cours, pédagogique, structurée et évitant la paraphrase. Le texte se conclut par des pistes de réflexion facultatives dont vous pouvez vous saisir ; au-delà de ces pistes proposées, toute initiative personnelle pertinente est appréciée. Vous n'êtes pas obligé de traiter le texte dans son intégralité, mais si seule une partie est traitée elle doit l'être de manière particulièrement approfondie.*

*L'exposé doit intégrer une (ou plusieurs) illustrations informatiques. Il doit également contenir une discussion autour d'une dimension éthique, sociétale, environnementale, économique ou juridique en lien avec le texte ; une des pistes de réflexion est spécifiquement conçue à cet effet.*

### 1 Contexte

En France, le recrutement des enseignants-chercheurs par les universités est un processus particulièrement décentralisé. Une des conséquences de cette décentralisation est qu'une fois en poste, il est très difficile pour un enseignant-chercheur de changer d'affectation. Le seul moyen n'impliquant pas de repasser un concours de recrutement est l'échange de postes entre enseignants-chercheurs. Encore faut-il pour cela que la personne qui se trouve dans l'université A et souhaite aller dans une université B connaisse un enseignant-chercheur de l'université B qui souhaite faire le chemin inverse ; voire qu'elle connaisse un enseignant-chercheur de l'université B qui souhaite rejoindre l'université C et un enseignant-chercheur de l'université C qui souhaite rejoindre l'université A, ou un cycle plus grand encore...

En vue de faciliter ces échanges, l'opération postes a mis en place, au début des années 2000, un serveur permettant d'identifier facilement les personnes cherchant à effectuer un échange de postes, le MOUVE (Machine Ouvertes aux Universitaires Voulant Échanger).

C'est le principe de ce serveur et diverses extensions possibles que nous allons étudier ici.

La philosophie de base au moment de la mise en place de MOUVE a été la suivante :

- avoir un site facile d'accès (pour qu'il soit utilisé),
- avoir un site qui nécessite le moins de maintenance possible.

#### 1.1 Caractéristiques d'un poste

Un poste d'enseignant-chercheur possède les caractéristiques suivantes :

- le grade : Maître de Conférences ou Professeur des Universités,

- la section CNU qui identifie le domaine auquel le poste est rattaché ; il s’agit d’un entier entre 1 et 87 (par exemple l’informatique correspond à la section 27) ; certains postes ont une section principale et une ou deux sections secondaires,
- l’établissement de rattachement,
- l’unité de formation au sein de cet établissement.

Dans les textes, rien n’interdit de faire un échange entre deux personnes occupant des postes sans caractéristique commune, mais comme l’échange doit être approuvé par diverses instances dont les unités de formation concernées, il est souvent plus aisé d’obtenir un échange entre des postes de même section CNU, voire de même grade.

## 1.2 Granularité géographique

Bien sûr, une personne qui désire partir occupe un poste dans une université spécifique, mais elle peut être ouverte à des postes dans plusieurs autres universités dans le but de se rapprocher d’une région donnée, voire dans plusieurs autres régions s’il s’agit non pas de se rapprocher d’un endroit, mais de s’éloigner d’un autre.

# 2 Gestion des localisations

Pour éviter d’avoir à générer les localisations (Université X, Ville Y, Région Z ou autre), le site MOUVE s’appuie sur les localisations déjà existantes : si l’utilisateur ne trouve pas la localisation qui lui convient, il peut alors en ajouter une nouvelle qui à son tour sera proposée aux utilisateurs suivants.

## 2.1 Une seule orthographe

Le bon fonctionnement de cette fonctionnalité repose sur la bonne volonté des utilisateurs du site, et plus il y a de localisations présentes, plus le risque qu’un utilisateur ne prenne pas le temps de les parcourir toutes et en ajoute directement une nouvelle existe. Si l’utilisateur n’utilise pas exactement la même suite de caractères, il y a alors risque de doublons.

Pour minimiser les risques d’orthographe multiples de localisations lorsqu’un nouvel utilisateur souhaite s’inscrire, on peut calculer la distance de Levenshtein entre la localisation proposée et les localisations existantes, en prenant soin au préalable de supprimer tout ce qui n’est pas une lettre dans les chaînes de caractères et en passant toutes les lettres en minuscules non accentuées. Dès la proposition du nouvel utilisateur entrée, si une localisation existante semble proche, on la lui propose. Libre à cette personne ensuite de conserver sa proposition ou de se rallier à la proposition existante.

La distance de Levenshtein (ou distance d’édition) représente le coût minimal de transformation d’un mot en un autre, en considérant comme opérations élémentaires :

- les substitutions de lettres,
- les suppressions de lettres,
- les ajouts de lettres.

Elle est donnée par l’algorithme suivant :

---

**Algorithme 1** Distance de Levenshtein

---

**entrée** : deux mots  $u = u_1u_2\dots u_m$  et  $v = v_1v_2\dots v_n$  (les  $u_i$  et les  $v_j$  sont des lettres).

**sortie** : distance de Levenshtein entre  $u$  et  $v$ .

```
1 : si  $\min(m, n) = 0$  alors
2 :   renvoyer  $\max(m, n)$ 
3 : fin si
4 : si  $u_1 = v_1$  alors
5 :   renvoyer  $\text{Levenshtein}(u_2\dots u_m, v_2\dots v_n)$ 
6 : fin si
7 : renvoyer  $1 + \min(\text{Levenshtein}(u_2\dots u_m, v), \text{Levenshtein}(u, v_2\dots v_n),$ 
    $\text{Levenshtein}(u_2\dots u_m, v_2\dots v_n))$ 
```

---

## 2.2 Une hiérarchie de localisations

Dans l’idéal, un utilisateur qui cherche une université en Île-de-France devrait se voir proposer tous les postes mis sur le marché en Île-de-France, pas uniquement ceux estampillés “Île-de-France”.

Le site MOUVE a été mis en place avant que des données géographiques ouvertes soient facilement accessibles, et ce lien Université → Ville → Département → Région doit donc être fait par l’utilisateur. On trouve actuellement sur <https://www.data.gouv.fr/fr/datasets/communes-de-france-base-des-codes-postaux/> un fichier csv contenant les colonnes suivantes :

- le code commune INSEE,
- le nom de la commune en MAJUSCULE,
- le code postal,
- le libellé d’acheminement,
- la ligne 5 de l’adresse (précision de l’ancienne commune ou du lieu-dit),
- la longitude GPS,
- la latitude GPS,
- le code de la commune,
- l’article,
- le nom de la commune en Minuscule,
- le code département,
- le nom du département,
- le code région,
- le nom de la région,

Par ailleurs, on peut aussi accéder au contour des communes via <https://www.data.gouv.fr/fr/datasets/contours-calculés-des-zones-codes-postaux/>, au format geojson qui est un format de type json qui permet de décrire des coordonnées spatiales. Voici le début de ce fichier :

```
{
  "type": "FeatureCollection",
  "features": [
    {
```

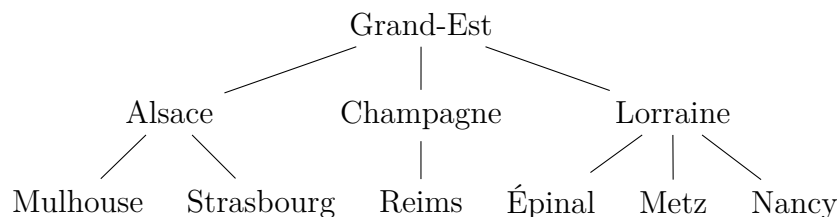
```

    "type": "Feature",
    "properties": { "codePostal": "01000", "nbNumeros": 11581 },
    "geometry":
    {
      "type": "Polygon",
      "coordinates": [[[5.174001, 46.199663],
        [5.155733, 46.224252], [5.1642, 46.228147], [5.168588, 46.228188],
        [5.227866, 46.22774], [5.246482, 46.227133], [5.2759, 46.220754],
        [5.277809, 46.196595], [5.277736, 46.19652], [5.254028, 46.178472],
        [5.195573, 46.188139], [5.174001, 46.199663]]]]
    }
  },
  "feature": ...
]
}

```

Grâce à la donnée de ces deux fichiers, on pourrait ajouter un ordre partiel sur les différentes destinations proposées ou demandées par les inscrits sur le site MOUVE, de sorte à créer une hiérarchie des localisations.

Voici un exemple partiel de ce que l'on pourrait obtenir :



Ainsi, quelqu'un qui déclarerait vouloir aller dans le Grand-Est se verrait automatiquement proposer les échanges possibles pour rejoindre l'Université de Reims, sans avoir à choisir cette destination à la main.

Il faudrait sûrement modifier le schéma actuel de la base de données de l'opération postes qui ne contient que les noms écrits par les utilisateurs, sans liens entre ces noms, pour que ce type de recherche puisse se faire en temps raisonnable.

Dans un deuxième temps, on pourrait imaginer une interface graphique agréable pour désigner un fragment du territoire à la main qui ne serait plus forcément un fragment portant un nom administratif, mais un fragment choisi par l'utilisateur, par exemple sous forme de polygone sur une carte.

### 3 Recherche de cycles

Le but de l'ensemble des données recueillies sur le site MOUVE, est de proposer des cycles existants afin de faciliter les échanges entre plusieurs enseignants-chercheurs. On considère un graphe dont les sommets seraient les localisations présentes dans la base et les arcs seraient étiquetés par les personnes : l'origine est la localisation que la personne veut quitter, et l'arrivée une des destinations qu'elle souhaite rejoindre.

Supposons qu'un utilisateur veuille aller d'un point A à un point B, il s'agit de trouver des chemins allant de B à A, avec les contraintes supplémentaires suivantes :

- les chemins doivent être assez courts (l'utilisateur doit pouvoir donner une borne sur la longueur des chemins recherchés),
- les chemins doivent privilégier les échanges dans la même section CNU, ou à défaut limiter le nombre de sections en jeu.
- les chemins doivent privilégier les échanges dans le même corps.

Un cycle dans le graphe des données de la base, contenant un point fixé, peut être repéré par un parcours classique de graphe (en profondeur ou en largeur). Mais il s'agit ici de trouver un cycle passant par deux points fixé (l'origine du demandeur et la destination qu'il souhaite atteindre).

On pourrait imaginer que certains utilisateurs sont prêts à s'éloigner un peu de la destination souhaitée, il y aurait alors deux solutions pour élargir la recherche :

- remonter dans l'ordre partiel présenté en section 2.2,
- autoriser les destinations éloignées d'au plus une certaine distance de la destination d'origine.

Pour trouver des cycles tenant compte de cette relaxation de contraintes, on peut imaginer ajouter des coûts sur les arcs du graphe : plus une destination est éloignée de la destination désirée, plus le coût qui lui est associé est élevé. Il s'agirait alors de trouver des cycles minimisant le coût du cycle, ou le coût moyen par personne impliquée dans le cycle.

On rappelle l'algorithme de Dijkstra qui permet de trouver un chemin de coût minimal dans un graphe pondéré avec des poids positifs :

---

**Algorithme 2** Algorithme de Dijkstra

---

**entrée :** un graphe  $G = (S, A)$ , un sommet particulier  $s_0 \in S$  et une pondération des arcs  $\rho : A \rightarrow \mathbb{R}_+$ .

**sortie :** deux tableaux `pred` et `dist` ayant  $|S|$  cases ; `pred[i]` donne le prédécesseur du sommet  $i$  dans le chemin de poids minimal de  $s_0$  à  $i$ , et `dist[i]` donne le poids de ce chemin.

```

1 : pour tout  $s \in S$  faire dist[s]  $\leftarrow +\infty$ 
2 : fin pour
3 : dist[s0]  $\leftarrow 0$ 
4 :  $P \leftarrow S \setminus \{s_0\}$ 
5 : tant que  $P$  est non vide faire
6 :    $s \leftarrow$  un sommet de  $P$  qui minimise dist[s]
7 :    $P \leftarrow P \setminus \{s\}$ 
8 :   pour tout  $t \in P$  voisin de  $s$  faire
9 :     si dist[s] +  $\rho(s, t) <$  dist[t] alors
10 :       dist[t]  $\leftarrow$  dist[s] +  $\rho(s, t)$ 
11 :       pred[t]  $\leftarrow s$ 
12 :     fin si
13 :   fin pour
14 : fin tant que

```

---

## 4 Un MOUVE institutionnalisé et automatisé ?

On pourrait imaginer que le ministère de l'Enseignement supérieur et de la Recherche se saisisse de la problématique des échanges de postes et propose tous les ans

aux candidats intéressés de se déclarer jusqu'à une certaine date. Il faudrait alors non plus trouver un cycle d'échanges pour un utilisateur particulier, mais trouver le plus de cycles possibles en maximisant la satisfaction des inscrits.

On pourrait par exemple demander à chaque utilisateur ses destinations préférées et éventuellement un deuxième ensemble de destinations qu'il aime moins mais qui lui conviendrait aussi, et en attribuant des poids selon les situations (aucun échange / déplacement vers une destination préférée / déplacement vers une autre destination), trouver le meilleur ensemble de cycles.

## 5 Pistes de réflexion

1. Donner une implémentation efficace du calcul de la distance de Levenshtein.
2. Comment utiliser les données ouvertes mentionnées en Section 2.2 pour produire un ordre partiel sur les destinations existantes dans MOUVE? Comment mettre cet ordre à jour en cas d'ajout de destination et comment l'utiliser dans la problématique de MOUVE?
3. Proposer un schéma de bases de données qui incluerait les données de MOUVE et l'ordre partiel sur les localisations.
4. Proposer un algorithme qui permettrait de trouver toutes les localisations dans un polygone (convexe ou pas) qui aurait été sélectionné à la main sur une carte. Évaluer la complexité de cet algorithme en fonction des données du problème (localisations et/ou polygone).
5. Proposer un algorithme pour trouver un cycle passant par deux sommets fixés dans un graphe. Évaluer la complexité de cet algorithme en espace et en temps. Peut-on faire mieux si on tient compte du fait qu'on impose une borne à la longueur du cycle?
6. Proposer un algorithme pour trouver un chemin élémentaire (*ie* ne passant pas deux fois par le même sommet) entre deux sommets d'un graphe pondéré avec des poids positifs, qui minimise le poids moyen des arcs sur le chemin.
7. Comment utiliser un algorithme qui permet de trouver un chemin avec des contraintes dans un graphe pour trouver un cycle avec ces mêmes contraintes?
8. Proposer un algorithme pour trouver des cycles disjoints dans le graphe de MOUVE.
9. La base de MOUVE contient des données sensibles. Que préconiseriez-vous pour éviter la perte et la fuite de telles données?