

SAE S2.02 – Rapport graphes

Équipe NOM_EQUIPE

Prénom1 Nom1, Prénom2 Nom2, Prénom3 Nom3,

2025

Ci-dessus, remplacer NOM_EQUIPE par le nom de votre équipe, et les Prénom Nom par vos prénoms et noms.

Effacez les indications en italique avant de rendre le rapport.

Le fichier source Markdown de ce rapport se trouve sur Moodle.

1 Version 1

1.1 Choix pour la modélisation

*Ci-dessous, H1, H2, etc. désignent des noms d'hôtes ; V1, V2, etc. désignent des noms de visiteurs. Pour chacun et chacune d'entre iels, vous devrez donner des valeurs pour les colonnes HOBBIES, GENDER, PAIR_GENDER et BIRTH_DATE. Vous pouvez réutiliser les mêmes valeurs plusieurs fois. La présentation des données doit être **lisible** (par ex. tableau, capture d'écran de tableur avec résolution suffisante).*

Forte affinité

Donnez une paire (H1, V1) qui présente une forte affinité. Expliquez pourquoi.

Faible affinité

Donnez une paire (H2, V2) qui présente une faible affinité. Expliquez pourquoi.

Arbitrage entre les critères d'affinité

Donnez trois paires hôte-visiteur (H3, V3), (H4, V4), (H5, V5) d'affinités à peu près équivalentes. Ces paires doivent illustrer comment vous arbitrez entre les différents critères d'affinité (passe-temps, préférences de genre, différence d'âge). Donc, idéalement, les raisons d'affinité seraient différentes dans les trois paires.

1.2 Exemple complet

Donnez un exemple de quatre hôtes A, B, C, D et quatre visiteurs W, X, Y, Z. Puis, donnez l'appariement qui vous considérez le meilleur entre ces hôtes et visiteurs.

1.3 Score d'affinité

Lisez les conseils pour l'écriture de pseudo-code se trouvant sur Moodle.

Donner le pseudo-code de la fonction `score_affinité_1(hôte, visiteur)` qui retourne un nombre représentant le degré d'affinité entre un hôte et un visiteur.

```
double score_affinité_1(hôte, visiteur)
// compléter le code ici
// ...
// ...
```

1.4 Retour sur l'exemple

Donnez la matrice d'adjacence du graphe biparti complet entre les hôtes A, B, C, D et les visiteurs W, X, Y, Z que vous avez introduit plus haut. Les poids des arêtes sont donnés par la fonction `score_affinité_1`.

Calculez l'appariement de poids minimal pour ce graphe. Obtenez-vous l'appariement que vous aviez identifié comme le meilleur ?

2 Version 2

Ci-dessous, vous définirez des hôtes ayant des noms $A1, A2, B1, B2$, etc., et des visiteurs ayant des noms $W1, W2, X1, X2$, etc. Pour chacun et chacune d'entre eux, vous devrez donner :

- la valeur pour la colonne `NAME` parmi $A1, A2, B1, \dots, W1, W2, X1, \dots$;
- des valeurs pour les colonnes `HOBBIES`, `GENDER`, `PAIR_GENDER`, `BIRTH_DATE` pour tout le monde ;
- des valeurs pour les colonnes `HOST_HAS_ANIMAL`, `HOST_FOOD` pour les hôtes ;
- des valeurs pour les colonnes `GUEST_ANIMAL_ALLERGY`, `GUEST_FOOD_CONSTRAINT` pour les visiteurs.

2.1 Exemple avec appariement total

Donnez un exemple de quatre hôtes $A1, B1, C1, D1$ et quatre visiteurs $W1, X1, Y1, Z1$ pour lesquels il existe des incompatibilités entre certains hôtes et certains visiteurs, mais il est possible de trouver un appariement qui respecte les contraintes rédhitoires.

Donnez également l'appariement que vous considérez le meilleur pour cet exemple. Expliquez pourquoi.

2.2 Exemple sans appariement total

Donnez un exemple de quatre hôtes $A2, B2, C2, D2$ et quatre visiteurs $W2, X2, Y2, Z2$ pour lesquels il n'est pas possible de former quatre paires hôte-visiteur à cause d'incompatibilités.

Pour cet exemple, quel est le plus grand nombre de paires qu'on peut former ?

Donnez l'appariement que vous considérez le meilleur. Expliquez pourquoi.

2.3 Score d'affinité

Donner le pseudo-code de la fonction `score_affinité_2(hôte, visiteur)` qui retourne un nombre représentant le degré d'affinité entre un hôte et un visiteur. Vous pouvez réutiliser la fonction `score_affinité_1` (l'appeler ou copier du code).

```
double score_affinité_2(hôte, visiteur)
// compléter le code ici
// ...
// ...
```

2.4 Retour sur l'exemple

Donnez les matrices d'adjacence pour les deux exemples de la Version 2 ($A1, B1, C1, D1/W1, X1, Y1, Z1$ et $A2, B2, C2, D2/W2, X2, Y2, Z2$). Les poids des arêtes sont déterminés par la fonction `score_affinité_2`. Pensez à nommer les lignes et les colonnes.

Calculez l'appariement de poids minimal pour chacun des graphes. Obtenez-vous l'appariement que vous aviez identifié comme le meilleur ?

2.5 Robustesse de la modélisation (question difficile)

Est-ce que votre fonction `score_affinité_2` garantit que les contraintes rédhitoires seront toujours respectées, quel que soit le jeu de données ? Justifiez votre réponse.

Indications : Cherchez un exemple de **grande taille** pour lequel la fonction `score_affinité_2` pourrait ne pas garantir le respect des contraintes. Dans cet exemple, vous auriez beaucoup d'adolescents compatibles sans affinité, et quelques adolescents incompatibles avec beaucoup d'affinité.

Il est possible que votre fonction garantisse le respect des contraintes quel que soit l'exemple. Si vous pensez que c'est le cas, donnez des arguments pour convaincre.

3 Version 3

Ci-dessous, $H1$, $H2$, etc. désignent des noms d'hôtes et $V1$, $V2$, etc désignent des noms de visiteurs. Pour chacun et chacune d'entre iels, vous devrez donner des valeurs pour toutes les colonnes pertinentes en fonction de leur rôle, hôte ou visiteur.

3.1 Équilibrage entre affinité / incompatibilité

Donnez au moins quatre paires hôte-visiteur ($H1$, $V1$), ($H2$, $V2$), ($H3$, $V3$), ($H4$, $V4$), ... que vous considérez quasi équivalents pour l'affectation. Certaines de ces paires doivent ne pas respecter les contraintes considérées rédhitoires dans la Version 2, d'autres doivent les respecter. Ces exemples doivent illustrer l'équilibrage que vous faites entre l'incompatibilité d'une part et l'affinité d'autre part : combien et quel type d'affinité permet de compenser combien et quel type d'incompatibilité. Les exemples seront accompagnés de commentaires expliquant vos choix.

3.2 Score d'affinité

Donner le pseudo-code de la fonction `score_affinité_3(hôte, visiteur)` qui retourne un nombre représentant le degré d'affinité entre un hôte et un visiteur. Vous pouvez réutiliser les fonctions `score_affinité_1` et `score_affinité_2`.

```
double score_affinité_3(hôte, visiteur)
    // compléter le code ici
    // ...
    // ...
```

3.3 Retour sur l'exemple

Donnez le résultat de la fonction `score_affinité_3` pour les exemples d'équilibrage ($H1$, $V1$), ($H2$, $V2$), etc. ci-dessus. Est-ce que vous obtenez des scores proches ?

Remarque: Deux scores ne sont pas proches ou éloignés dans l'absolu ; cela dépend de la valeur minimale et la valeur maximale que peut prendre le score. Par exemple, les nombres 10 et 20 sont "proches" à l'échelle de l'intervalle de 0 à 1000, mais ne sont pas "proches" à l'échelle de l'intervalle 0 à 30.