



SAE S2.01-2.02 – Rapport de Graphes

Roty Clément, Sergheraert Timothée, Lemaire Mano

25 avril 2025 - Version Final

Table des matières

1. Introduction	1
2. Version 1	2
2.1. Choix pour la modélisation	2
2.1.1. Forte affinité	2
2.1.2. Faible affinité	2
2.1.3. Arbitrage entre les critères d'affinité	2
2.1.4. Exemple complet	2
2.2. Score d'affinité	4
2.2.1. Tableau des coûts	4
2.2.2. Pseudo-code de l'algorithme du score d'affinité	4
2.2.3. Retour sur l'exemple complet	5
3. Version 2	6
3.1. Choix pour la modélisation avec appariement total	6
3.2. Choix pour la modélisation sans appariement total	6
3.3. Pseudo-code	6
3.4. Matrices d'adjacences	7
3.5. Robustesse de notre modélisation	7
4. Version 3	8
4.1. Choix de la modélisation	8
4.2. Pseudo-code	8
4.3. Retour sur notre exemple	8



1. Introduction

Dans la suite de ce rapport, nous allons voir comment modéliser et conceptualiser un problème d'affectation. Notre sujet porte, lui, sur la formation de couples de personnes pour un séjour et ceci, basée dans un premier temps sur le préférence pour établir la meilleure compatibilité possible entre toutes ces personnes. Il s'agit donc **d'affecter** chaque individu selon leurs **affinités** respectives.



2. Version 1

2.1. Choix pour la modélisation

On définit un couple d'individus comme une paire (H_n, V_n) où H_n est l'individu qui accueille et V_n est l'individu qui est accueilli. Voyons ensemble quelques exemples pour mieux comprendre ce que nous allons modéliser.

2.1.1. Forte affinité

La paire ($H1, V1$) à une forte affinité car ils possèdent 2 loisirs en commun, de plus ils sont chacun du genre voulu par l'autre et font partie de la même génération.

	HOBBIES	GENDER	PAIR_GENDER	BIRTH_DATE
H1	Lecture, échecs	Homme	Femme	05/12/90
V1	Lecture, échecs	Femme	Homme	25/08/92

Figure 1. Tableau de la forte affinité

2.1.2. Faible affinité

La paire ($H2, V2$) à une faible affinité car ils ne possèdent aucun loisir en commun, ne partagent pas leur préférence de genre et possèdent un écart générationnel.

	HOBBIES	GENDER	PAIR_GENDER	BIRTH_DATE
H2	Football, Jeux vidéo	Homme	Femme	03/11/85
V2	Photographie, Judo	Homme	Femme	15/03/00

Figure 2. Tableau de la faible affinité

2.1.3. Arbitrage entre les critères d'affinité

Les paires ($H3, V3$), ($H4, V4$), ($H5, V5$) possèdent une affinité semblable car chaque paire possède deux préférences partagées mais possède une préférence différente : dans la première paire ($H3, V3$) n'a pas de hobby en commun, la seconde paire ($H4, V4$) ne partage pas leur préférence de genre avec l'autre et la dernière paire ($H5, V5$) a un écart générationnel trop important.

	HOBBIES	GENDER	PAIR_GENDER	BIRTH_DATE
H3	Jeux vidéo, Bricolage	Femme	Homme	17/01/99
V3	Cuisine, Dessin	Homme	Femme	19/02/97
H4	Cinéma, Jeux vidéo	Homme	Femme	09/07/00
V4	Cinéma, Lecture	Homme	Femme	14/06/01
H5	Jardinage, Bricolage	Femme	Femme	10/04/96
V5	Couture, Bricolage	Femme	Femme	30/10/05

Figure 3. Tableau des affinités variées

2.1.4. Exemple complet

Nous avons dans l'exemple ci-dessous, 8 individus, 4 hôtes (Adrien, Béatrice, Cédric, Diane) et 4 visiteurs (Wendy, Xavier, Yasmine et Zack) que l'on représentera par la suite par la 1ère lettre de leur prénom qui possèdent des affinités variées. En fonction de ces affinités nous allons essayer à la main de trouver une affectation optimale :



NAME	HOBBIES 1	HOBBIES 2	GENDER	PAIR_GENDER	BIRTH_DATE
Adrien	Jeux vidéo	Jardinage	Homme	Femme	15/04/1998
Béatrice	Photographie	Lecture	Femme	Homme	29/10/2000
Cédric	Sport	Musique	Homme	Homme	11/02/1997
Diane	Cuisine	Randonnée	Femme	Femme	06/08/1999
Wendy	Lecture	Bricolage	Femme	Homme	21/09/1998
Xavier	Sport	Jeux vidéo	Homme	Femme	04/01/1996
Yasmine	Jardinage	Cuisine	Femme	Femme	19/12/2003
Zack	Photographie	Musique	Homme	Homme	09/07/1999

Figure 4. Tableau des 8 individus de notre exemple

Maintenant s'il devait faire nos choix d'affectation, nous pourrions dire que l'appariement qui nous semble être le meilleur est celui-ci :

Exemple 1. Une affectation possible qui semble optimale :

- **[A avec X]** car ils ont le Jeu-vidéo en hobbie commun, et qu'ils ont un age proche
- **[B avec W]** car ils partagent la lecture comme passion commune, n'ont pas d'écart générationnel important
- **[C avec Z]** car ils ont en commun le hobbie de la musique et que chacun voulait avoir un homme avec eux et leur écart d'age est d'environ 2 ans.
- **[D avec Y]** sont la dernière paire restante, mais elle possède tout de même un hobbie en commun en dépit d'un plus grand écart générationnel.

Conclusion

Nous remarquons ainsi que le plus gros problème dans cet manière de faire est que cela reste **arbitraire** et pourrais changer lors d'une autre affectaion. C'est donc très **peu fiable et représentatif** de la qualité d'une affectation entre deux de ces individus. Enfin nous avons fait ces affectations à la main parce qu'il n'y en avait que 4 à faire. Cependant pour notre projet, nous allons avoir bien plus d'individus à associer donc il nous faut forcément un algorithme pour cela. Hors une machine ça ne comprend qu'une chose, c'est les chiffres. Il nous faut donc **quantifier, par un score**, la qualité de l'affinité entre deux individus.



2.2. Score d'affinité

2.2.1. Tableau des coûts

Dans notre modélisation, nous avons choisi donc que plus le coût est faible, plus l'affinité est forte. Nous avons donc choisi de définir le coût de chaque différence de préférence de la manière suivante :

		ÉCART D'ÂGE ENTRE 2 INDIVIDUS		
		DE MÊME ÂGE	< 2ans	> 2ans & < 5ans
COÛT	-1	0	2	5
		HOBBIES DIFFÉRENTS		
COÛT	3			
		GENRE VOULU DIFFÉRENT		
COÛT	1			

Figure 5. Tableau des coûts

A partir de ce tableau et de ces choix, nous pouvons maintenant établir le coût minimal et maximal d'une paire et définir par la suite ce que représente une affinité forte ou faible.

- Le coût minimal d'une paire est donc de -1
- Le coût maximal d'une paire est donc de 13

Table 1. Tableau de classification des affinités par intervalles de coût

Coût de l'affection d'une paire	[-1 ; 3 [[3 ; 9 [[9 ; 13]
Type d'affinité	Forte affinité	Affinité moyenne	Faible affinité

2.2.2. Pseudo-code de l'algorithme du score d'affinité

Mainenant que nous avons mis des chiffres sur que représente une affinité, on peut alors écrire le pseudo-code de l'algorithme permettant de calculer le score d'affinité d'un couple d'individu :

```
double score_affinité_1(hôte, visiteur):
    score = 0
    si hôte.âge == visiteur.âge:
        score = score - 1
    si abs(hôte.âge - visiteur.âge) > 2 ans && abs(hôte.âge - visiteur.âge) < 5 ans:
        score = score + 2
    sinon:
        score = score + 5
    pour chaque hobbie de hôte:
        si le hobbie du visiteur n'est pas vide et est différent de celui de l'hôte:
            score = score + 3
        si hôte.gender != visiteur.pair_gender:
            score = score + 1
        si hôte.pair_gender != visiteur.gender:
            score = score + 1
    retourner score
```



2.2.3. Retour sur l'exemple complet

Maintenant que nous avons un algorithme qui nous permet de calculer le score d'affinité entre deux individus, nous allons l'appliquer à l'exemple vu avant afin de créer la matrice ci-dessous :

	A	B	C	D
W	6	4	7	7
X	7	8	4	9
Y	9	9	13	5
Z	7	4	5	8

Figure 6. Matrice des scores d'affinité

Maintenant que nous avons notre matrice, nous pouvons effectuer un algorithme qui va calculer l'affectation à cout minimal sur l'ensemble cette matrice. Nous obtenons le résultat suivant :

```
$ java -jar calcul-affectation.jar matrice.csv
Matrice du graphe :
      A          B          C          D
W  6,000    4,000    7,000    7,000
X  7,000    8,000    4,000    9,000
Y  9,000    9,000   13,000   5,000
Z  7,000    4,000    5,000    8,000

Affectation de cout minimal 19,000000 :
(W, A, 6,000000)
(Y, D, 5,000000)
(X, C, 4,000000)
(Z, B, 4,000000)
```

Figure 7. Capture d'écran après exécution de l'algorithme d'affectation de cout minimal.

Finalement, nous avons donc une **affectation à cout minimal de 19** et dont le détail est affiché ci-dessous :

- **[A avec W]** pour un cout de 6,
- **[B avec Z]** pour un cout de 4,
- **[C avec X]** pour un cout de 4,
- **[D avec Y]** pour un cout de 5 et qui était la seule paire que nous avions trouvé avant et qui est restée inchangée.

On remarque alors que dans notre cas, l'algorithme n'a donné aucune associations de **forte affinité** mais a évité toutefois les associations de **faible affinité** à garantie que tout les individus soit moyennement satisfaits de leur paire.



3. Version 2

3.1. Choix pour la modélisation avec appariement total

Name	Hobbies	Gender	Pair_Gender	Birth_Date	Host_Has_Animal	Host_Food
A1	Musique, Jeu vidéo	Homme	Femme	1999-06-20	Oui	Omnivore
B1	Lecture, Yoga	Femme	Homme	2000-03-12	Non	Végétarien
C1	Cinéma, Lecture	Homme	Homme	1994-08-05	Oui	Végétalien
D1	Danse, Cuisine	Femme	Femme	1998-11-30	Non	Omnivore

Name	Hobbies	Gender	Pair_Gender	Birth_Date	Guest_Animal_Allergy	Guest_Food_Constraint
W1	Lecture	Femme	Homme	1999-05-20	Oui	Omnivore
X1	Musique, Cinéma	Homme	Femme	1997-08-10	Non	Omnivore
Y1	Cuisine, Yoga	Femme	Femme	1994-01-05	Non	Végétarien
Z1	Jeux vidéo, Danse	Homme	Femme	2000-07-10	Oui	Végétarien

L'appariement qui nous semble être le meilleur est celui-ci :

Exemple 2. Une affectation possible qui semble optimale :

- [A1 avec X1]
- [B1 avec Z1]
- [C1 avec Y1]
- [D1 avec W1]

Cet appariement permet de respecter les contraintes rédibitoires.

3.2. Choix pour la modélisation sans appariement total

Name	Hobbies	Gender	Pair_Gender	Birth_Date	Host_Has_Animal	Host_Food
A2	Cinéma	Homme	Femme	1995-06-21	Oui	Omnivore
B2	Cuisine, Danse	Femme	Homme	1998-09-12	Non	Végétalien
C2	Yoga, Jeux vidéo	Homme	Femme	2001-01-05	Oui	Carnivore
D2	Musique, Lecture	Femme	Homme	1997-11-30	Non	Végétarien

Name	Hobbies	Gender	Pair_Gender	Birth_Date	Guest_Animal_Allergy	Guest_Food_Constraint
W2	Cinéma, Yoga	Femme	Homme	1996-05-14	Oui	Omnivore
X2	Cuisine, Yoga	Homme	Femme	1997-07-10	Non	Omnivore
Y2	Lecture, Danse	Femme	Homme	1999-04-03	Non	Végétalien
Z2	Musique, Cinéma	Homme	Femme	2000-02-28	Oui	Omnivore

L'appariement qui nous semble être le meilleur est celui-ci :

Exemple 3. Une affectation possible qui semble optimale :

- [A2 avec X2]

Voici le seul appariement possible car toutes les autres combinaisons sont bloquées à causes de contraintes rédibitoires.

3.3. Pseudo-code

```
double score_affinité_2(hôte, visiteur):
    score = 0
    si hôte.animal==vrai and visiteur.allergie==vrai:
        score = score + 1e9
    sinon si hôte.food!=visiteur.food:
        score = score + 1e9
```



```
score = score + score_affinité_1(hôte, visiteur)
retourner score
```

3.4. Matrices d'adjacences

Host \ Guest	W1	X1	Y1	Z1
A1	1e9	1	1e9	5
B1	6	1e9	3	1e9
C1	1e9	4	1e9	1e9
D1	6	1e9	1e9	2

Host \ Guest	W2	X2	Y2	Z2
A2	1e9	5	1e9	1e9
B2	9	1e9	3	5
C2	1e9	4	1e9	1e9
D2	6	1e9	1e9	2

3.5. Robustesse de notre modélisation

Nous utilisons un coût très élevé pour les contraintes rédhibitoires afin de les éviter autant que possible. Cependant, cela ne garantit pas toujours leur respect absolu. Si aucune autre solution valable n'existe, l'algorithme pourrait quand même choisir une paire interdite. Donc mettre un coût très élevé n'est pas forcément la meilleure méthode pour assurer strictement le respect des contraintes.



4. Version 3

4.1. Choix de la modélisation

Name	Hobbies	Genre	Pair_Gender	Birth_Date	Host_Has_Animal	Host_Food
H1	Musique, Jeux vidéo	Homme	Femme	1999-06-20 Oui	Omnivore	
H2	Lecture, Yoga	Femme	Homme	2000-03-12 Non	Végétarien	
H3	Cinéma, Lecture	Homme	Homme	1994-08-05 Oui	Végétalien	
H4	Danse, Cuisine	Femme	Femme	1998-11-30 Non	Omnivore	

Name	Hobbies	Genre	Pair_Gender	Birth_Date	Guest_Animal_Allergy	Guest_Food_Constraint
V1	Musique, Cinéma	Femme	Homme	1998-05-15 Non	Omnivore	
V2	Lecture, Yoga	Homme	Femme	1999-11-10 Oui	Végétarien	
V3	Cinéma	Homme	Homme	1993-12-22 Non	Omnivore	
V4	Danse, Yoga	Femme	Femme	1997-09-18 Oui	Végétarien	

4.2. Pseudo-code

```

double score_affinité_3(hôte, visiteur):
    score = score_affinité_1(hôte, visiteur)
    si hôte.has_animal == vrai et visiteur.animal_allergy == vrai:
        score = score + 25
    si hôte.food != visiteur.food:
        si hôte.food == "Omnivore" or visiteur.food_constraint == "Omnivore":
            score = score + 15
        sinon:
            score = score + 25
    retourner score

```

4.3. Retour sur notre exemple

Exemple 4. Voici les scores obtenus pour les paires :

- [**H1 avec V1**] 28 + 25 pour une incompatibilité animal, Total : 53
- [**H2 avec V2**] 34, sans incompatibilité
- [**H3 avec V3**] 32 + 25 pour un incompatibilité de nourriture, Total : 57
- [**H4 avec V4**] 35, sans incompatibilité

On remarque donc que les scores obtenus sont similaires, voire proches, malgré un écart dû aux incompatibilités, ce qui reste cohérent avec notre volonté de rechercher un équilibre entre affinité et incompatibilité.