

# Démonstration des algorithmes de couplage à travers un graphe

Les algorithmes de Gale-Shapley et Irving

# Introduction

Pour les deux présentations suivantes, on utilise des graphes dirigés et pondérés. Plus le poids d'une arête est élevée, plus le sommet d'origine apprécie le sommet de destination.

Les graphes sont générés à l'aide de la librairie NetworkX de Python et les poids sont choisis à l'aide d'un générateur de nombre aléatoire. Chaque algorithme utilise un type de graphe différent qui sera présenté en détail dans la partie correspondante.

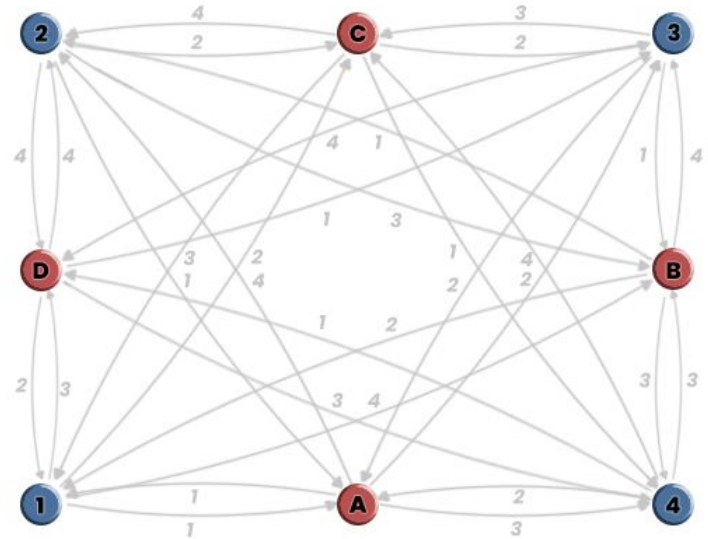
Les programmes utilisés pour générer les graphes de Gale-Shapley et Irving sont disponibles dans le dépôt aux noms `galeshapley.py` et `irving.py` respectivement.

# I. Gale-Shapley

Couplage entre deux ensembles distincts avec un graphe biparti.

Le graphe utilisé est un graphe dirigé biparti pour représenter les ensembles des deux genres utilisés dans cet algorithme. Chaque ensemble a 4 sommets et chaque sommet a une arête vers chaque sommet de l'autre ensemble.

La forme du graphe a été choisie pour réduire la densité des arêtes de la représentation classique d'un graphe biparti et faciliter la lecture de l'algorithme.



Chaque sommet a sa propre couleur de graphe pour faciliter la lecture.

L'algorithme se termine lorsque 4 couples sont créés.

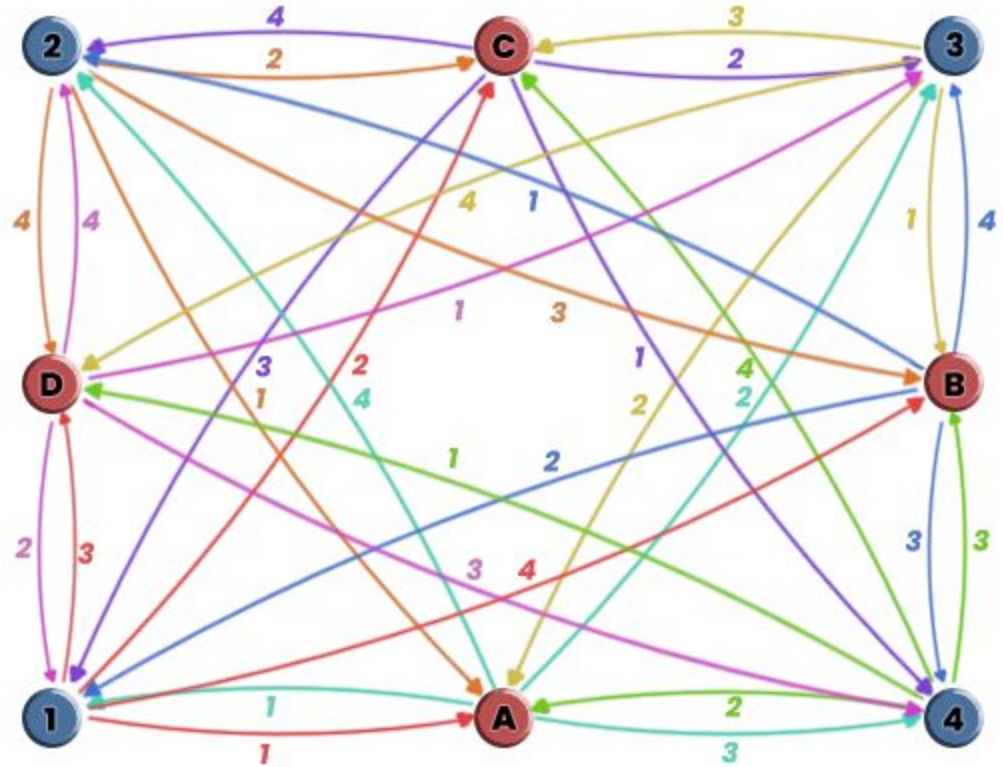
Couplage courant

(1, ?)

(2, ?)

(3, ?)

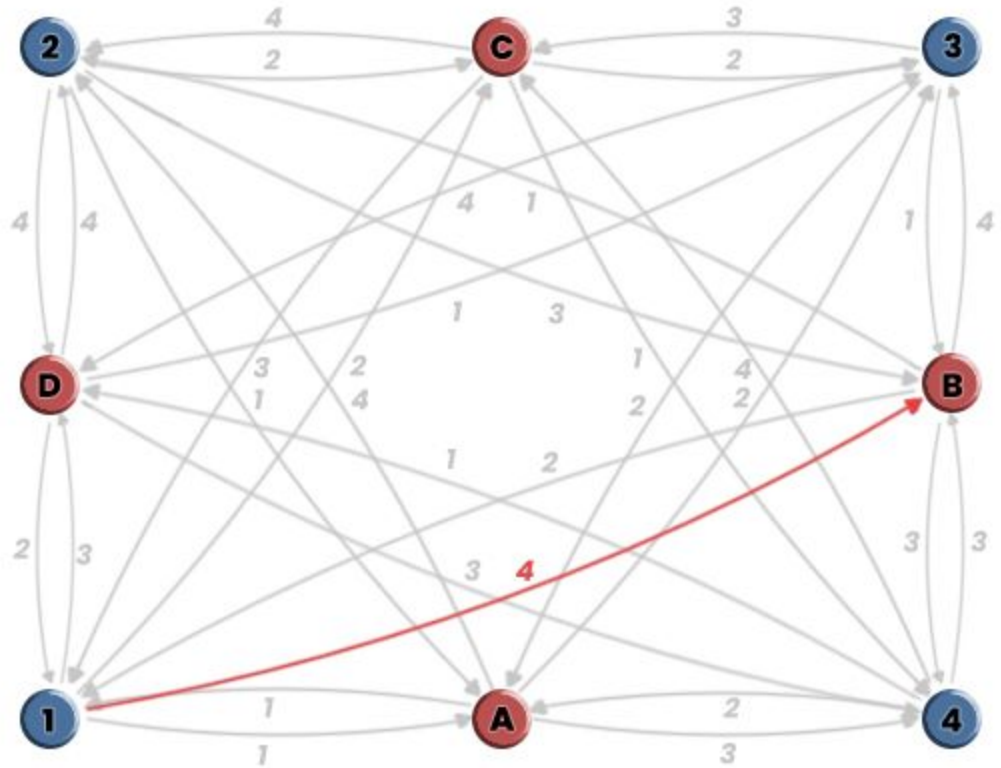
(4, ?)



1 propose à B, qui n'a  
reçu aucune autre  
demande et donc  
accepte.

Couplage courant

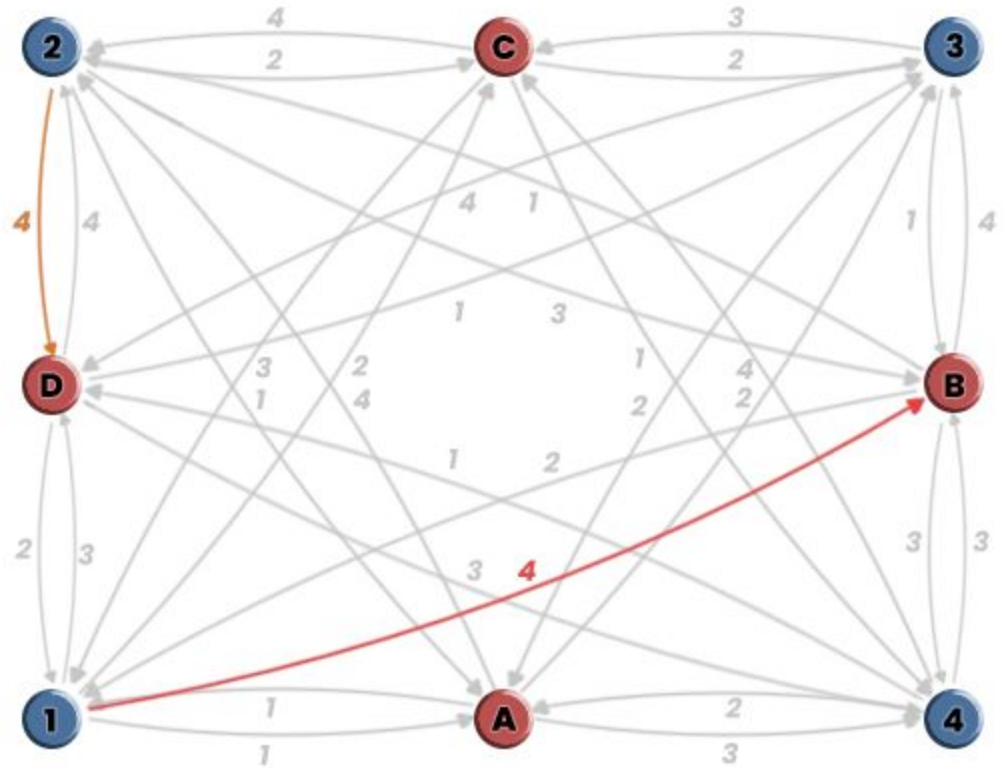
(1, B)  
(2, ?)  
(3, ?)  
(4, ?)



Pareillement, D accepte la proposition de 2.

Couplage courant

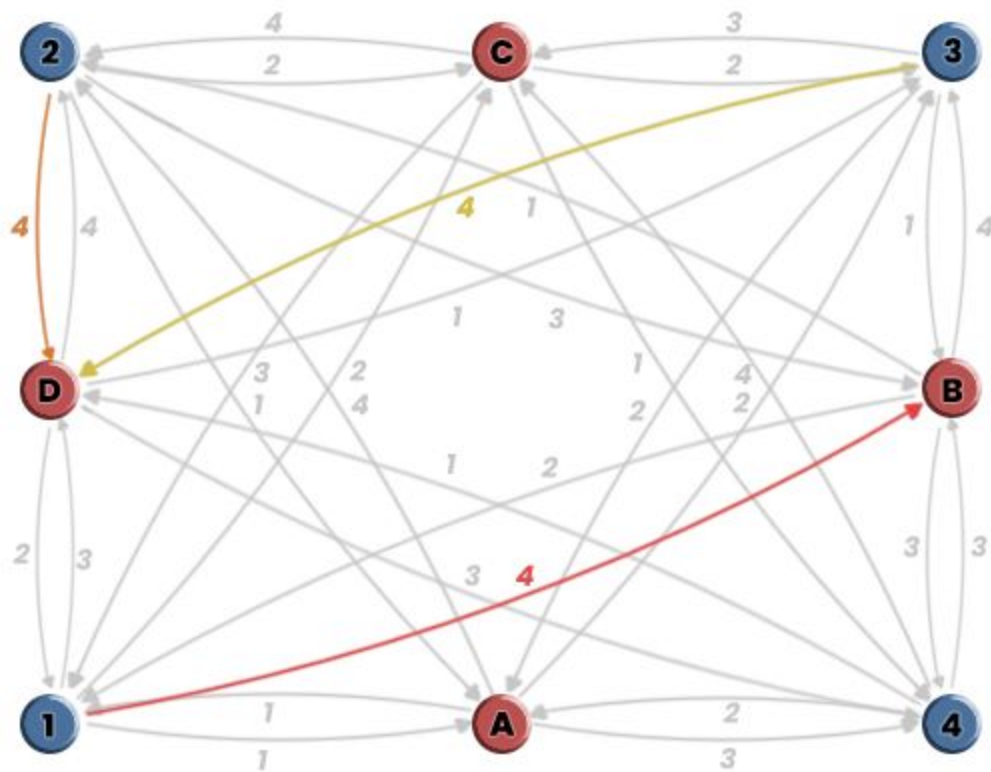
(1, B)  
(2, D)  
(3, ?)  
(4, ?)



3 propose aussi à D, il y  
à ici un conflit.

Couplage courant

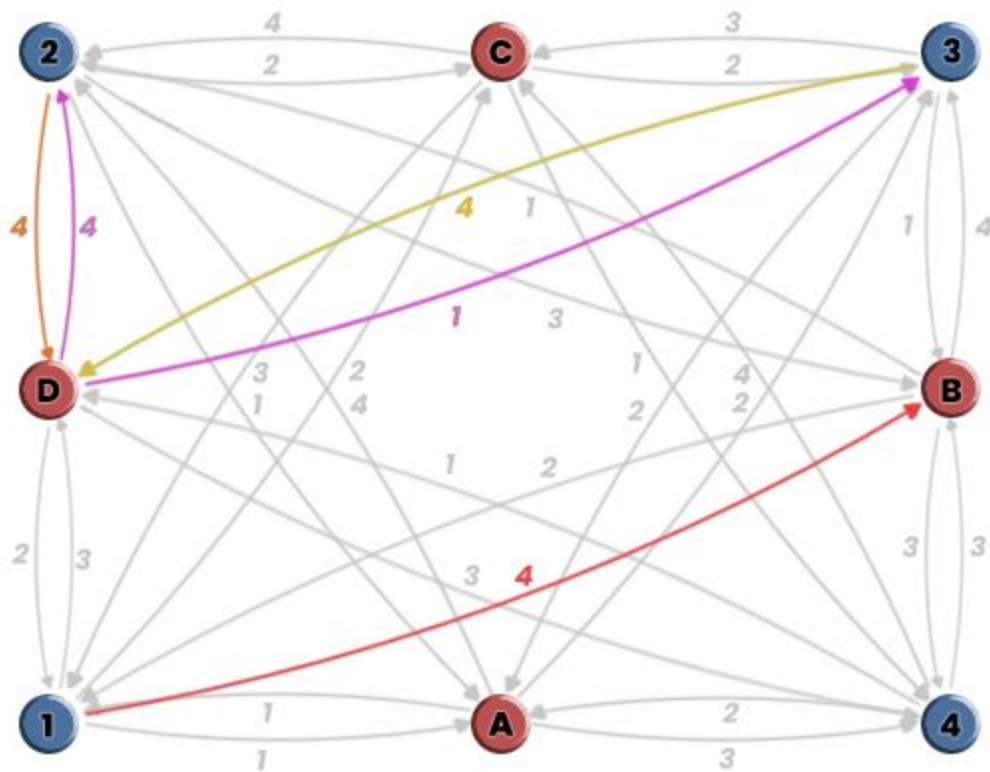
(1, B)  
(2, D)  
(3, D)  
(4, ?)



D préfère 2 à 3, donc le couple (2,D) est conservé et le couple (3, D) est supprimé.

Couplage courant

- (1, B)
- (2, D)
- (3, ?)
- (4, ?)

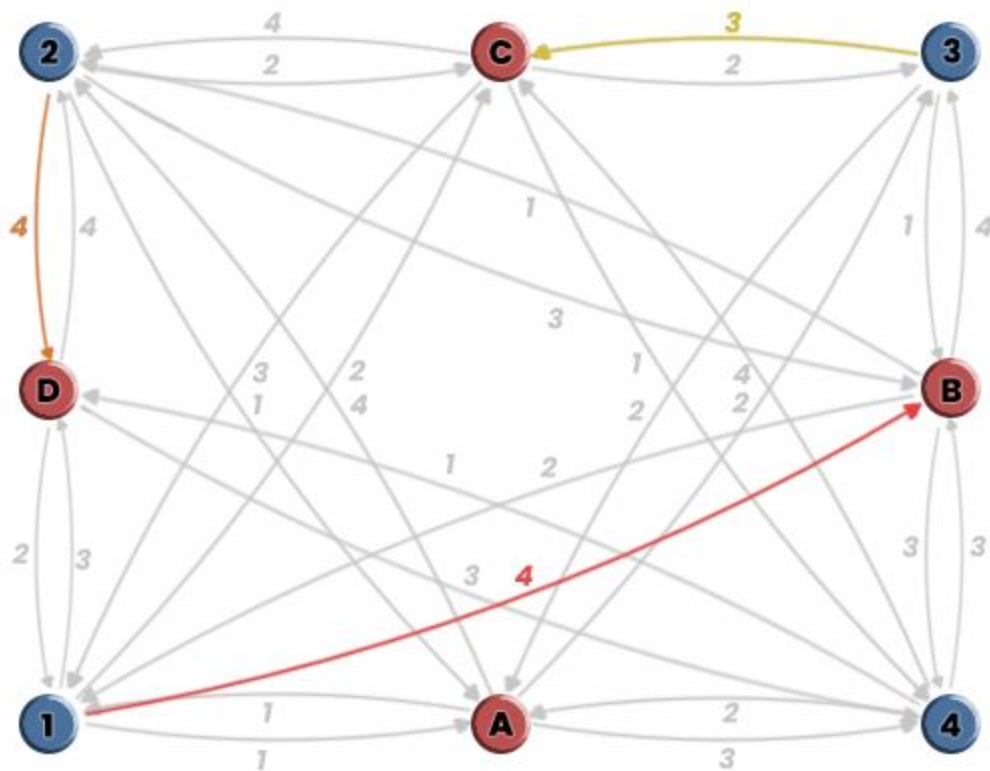




Les arêtes D3 et 3D sont retirés du graphe et 3 propose à sa deuxième préférence : C.

Couplage courant

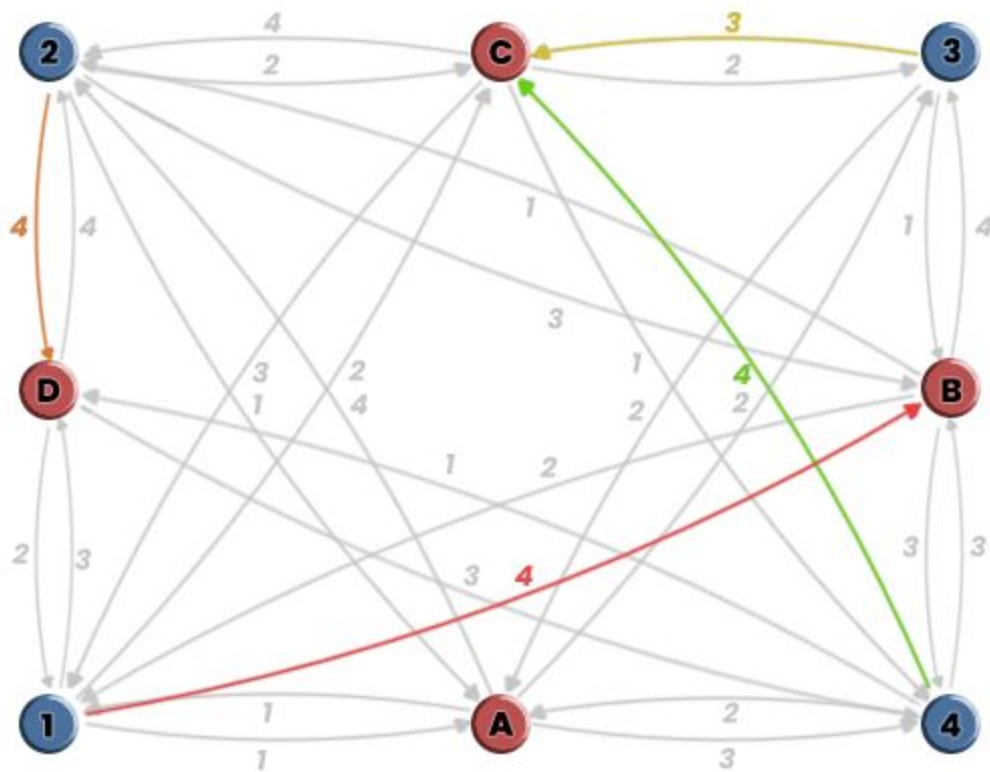
- (1, B)
- (2, D)
- (3, C)
- (4, ?)



Viens le tour de 4 de  
faire sa proposition,  
créant un conflit avec 3.

Couplage courant

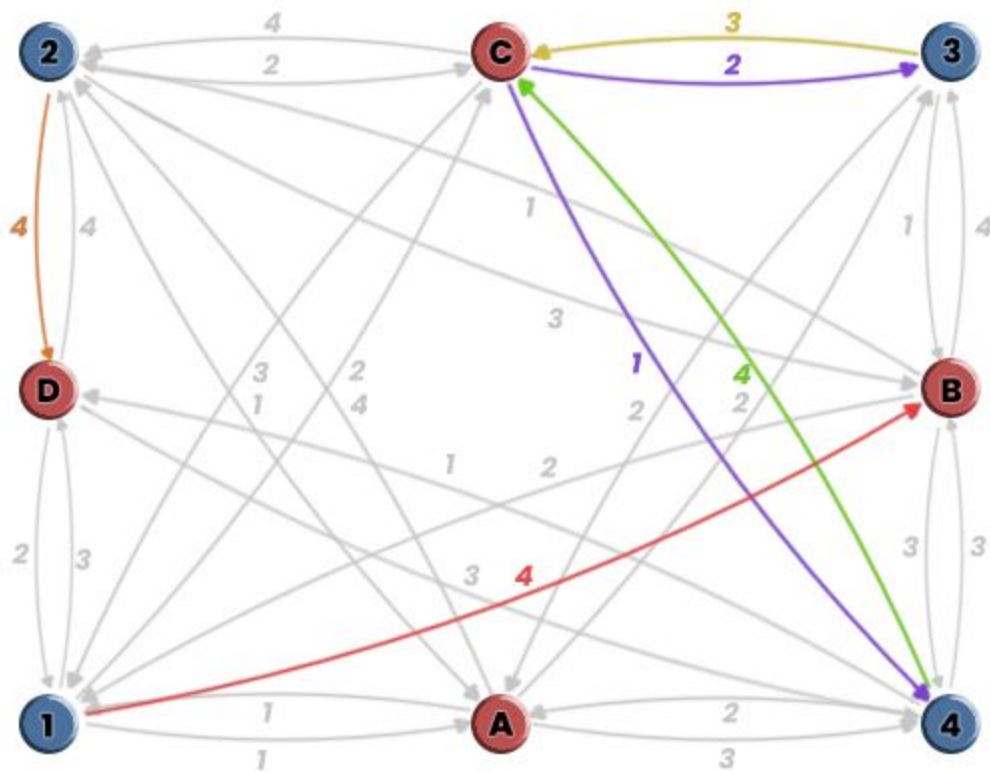
- (1, B)
- (2, D)
- (3, C)
- (4, C)



C préfère 3 donc le couple (3,C) est celui conservé.

Couplage courant

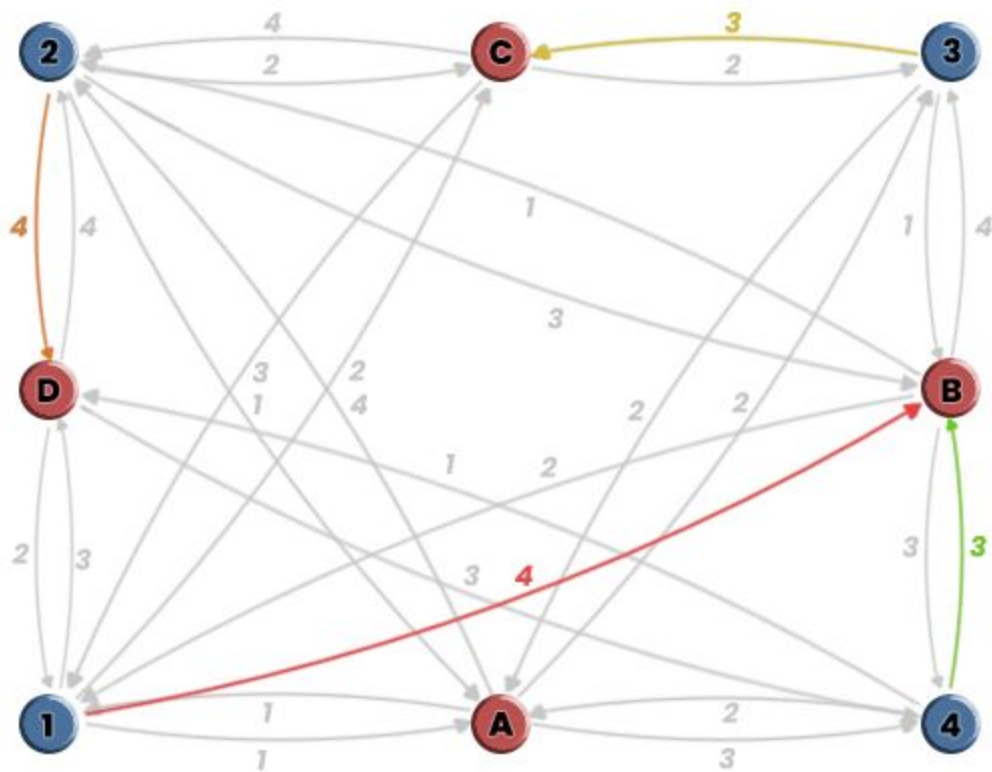
- (1, B)
- (2, D)
- (3, C)
- (4, ?)



4 propose à B, ce qui cause un autre conflit.

Couplage courant

- (1, B)
- (2, D)
- (3, C)
- (4, B)



4 remporte cette fois-ci les faveurs, le couple (1, B) est supprimé et 1 doit faire une nouvelle proposition.

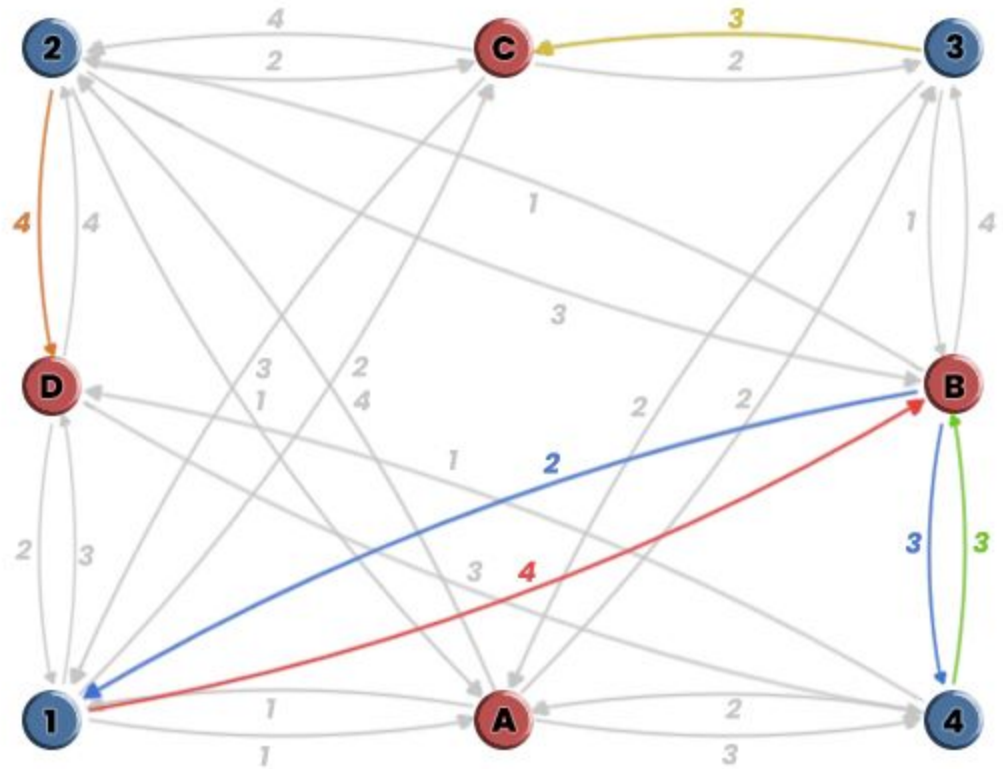
Couplage courant

(1, ?)

(2, D)

(3, C)

(4, B)



1 propose à A, qui n'a eu aucune proposition avant.

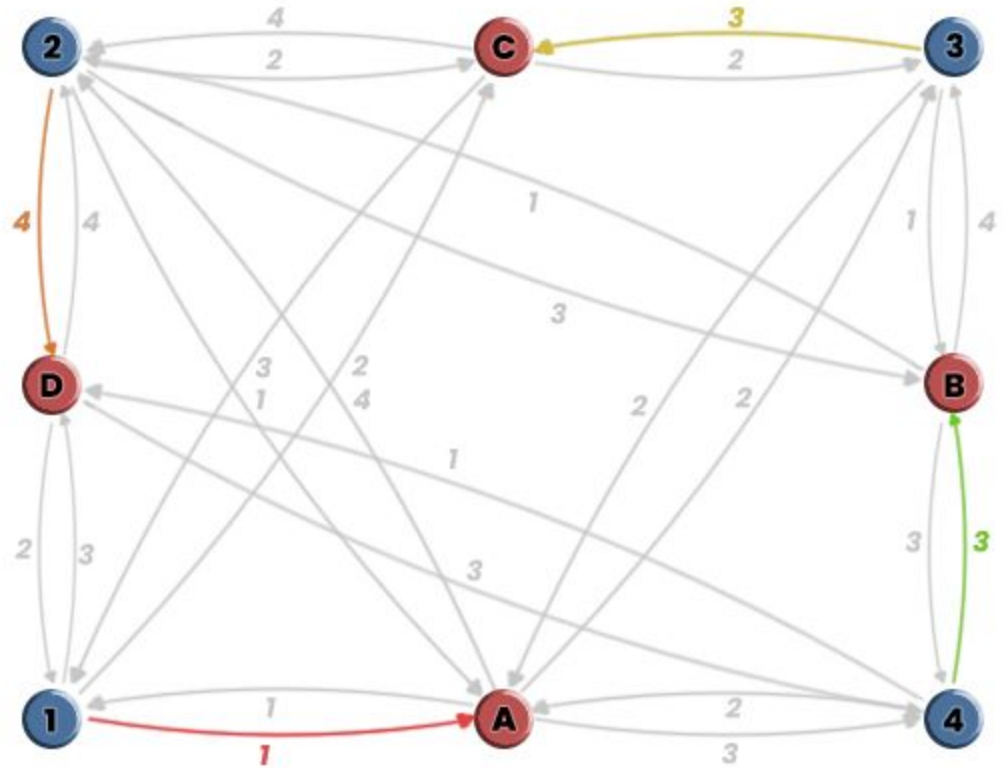
Couplage courant

(1, A)

(2, D)

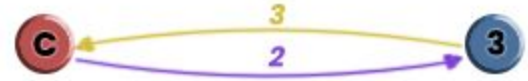
(3, C)

(4, B)



Il y a 4 couples et aucun conflit, le couplage est donc stable et l'algorithme prend fin.

- (1, A)
- (2, D)
- (3, C)
- (4, B)

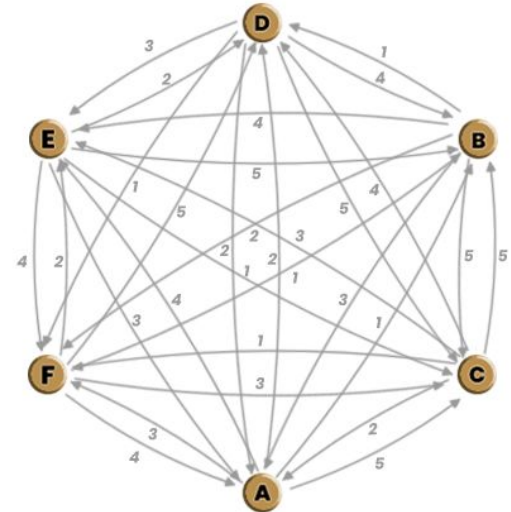


# II. Irving

Couplage dans un même ensemble avec un graphe complet.

Le graphe utilisé est un graphe dirigé complet pour représenter un ensemble unique avec 5 possibles candidats de couplage. L'algorithme d'Irving sur un graphe s'arrête lorsqu'il y a au moins  $n/2$  (ici 3) composantes connexes dans le graphes, à savoir les couples trouvés, ou les cas isolés si un couplage stable n'existe pas.

Les 3 phases de l'algorithme seront présentées normalement mais des libertés ont été prises pour montrer le plus possible les différents conflits que l'algorithme peut rencontrer, ces décisions seront indiquées dans les étapes respectives.

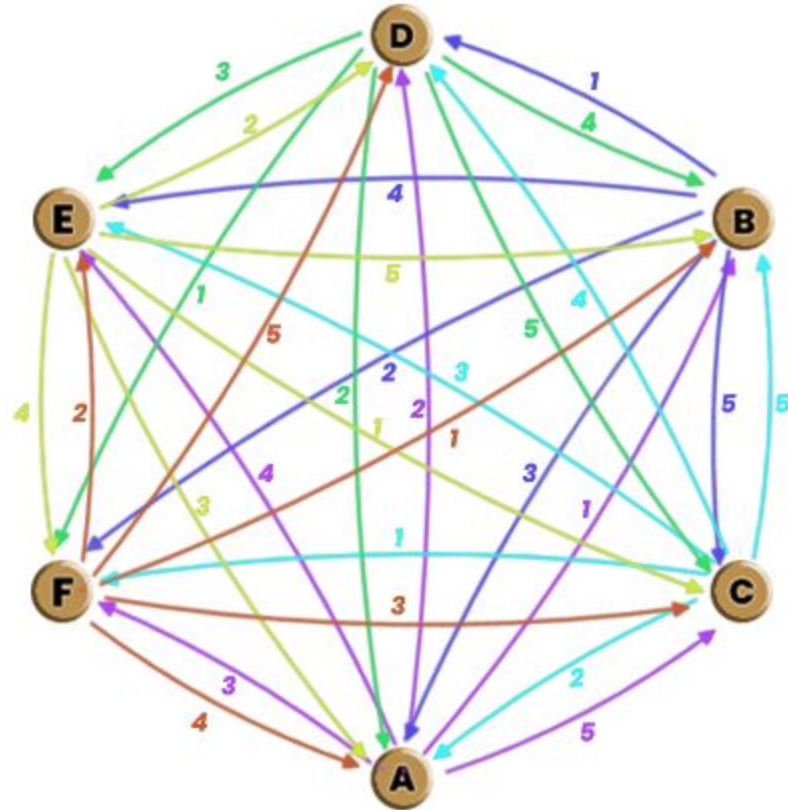




## Phase 1

La phase 1 consiste en une application de l'algorithme de Gale-Shapley qui autorise un cycle.

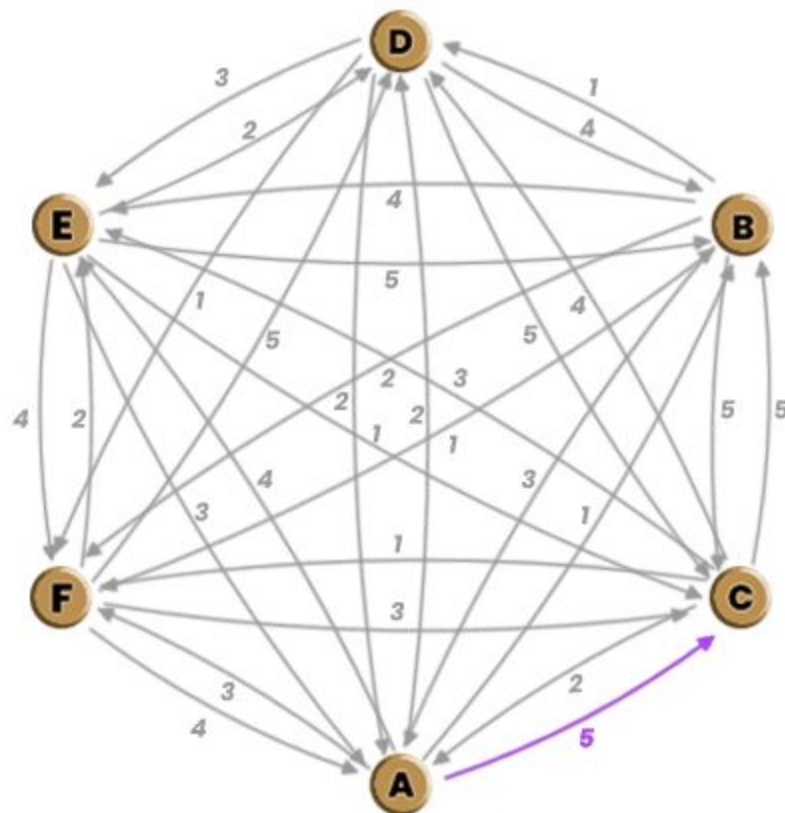
Chaque sommet a sa propre couleur d'arête pour faciliter la lecture.



- (A, ?)
- (B, ?)
- (C, ?)
- (D, ?)
- (E, ?)
- (F, ?)

## Phase 1

A fait une demande à C, acceptée automatiquement.



(A, C)

(B, ?)

(C, ?)

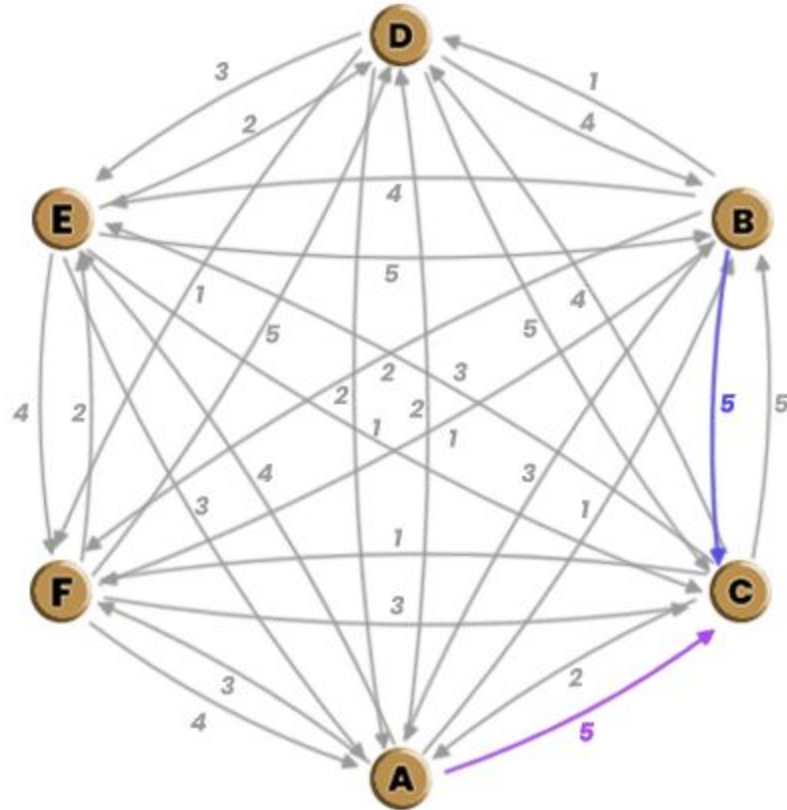
(D, ?)

(E, ?)

(F, ?)

## Phase 1

B fait aussi une demande à C, créant alors un conflit.



(A, C)

(B, C)

(C, ?)

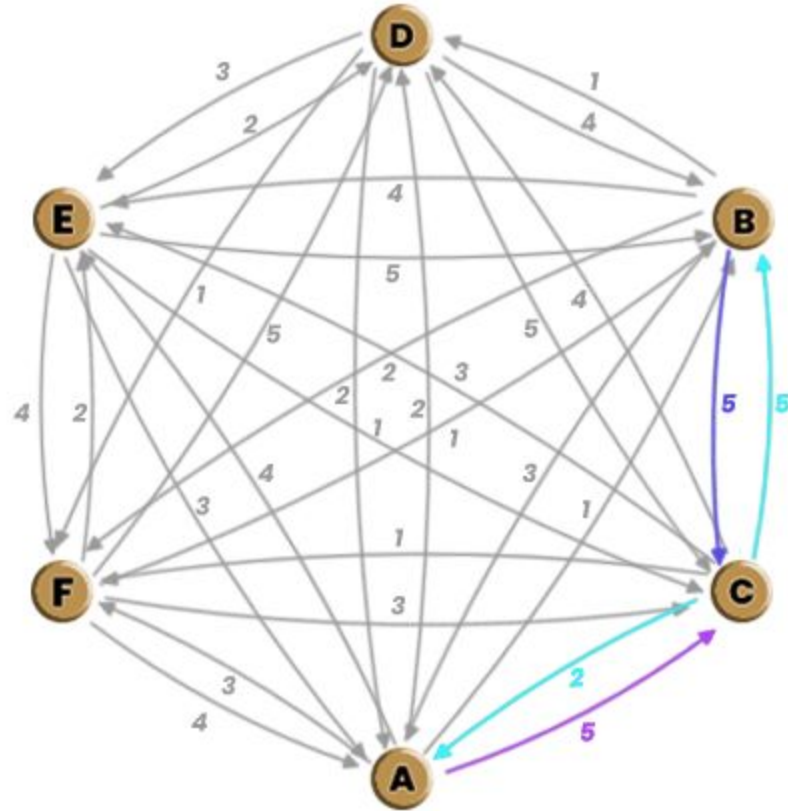
(D, ?)

(E, ?)

(F, ?)

## Phase 1

B remporte le conflit, A doit effectuer une nouvelle demande. On constate que C est aussi la préférence de B donc on réciproque la demande.



(A, ?)

(B, C)

(C, B)

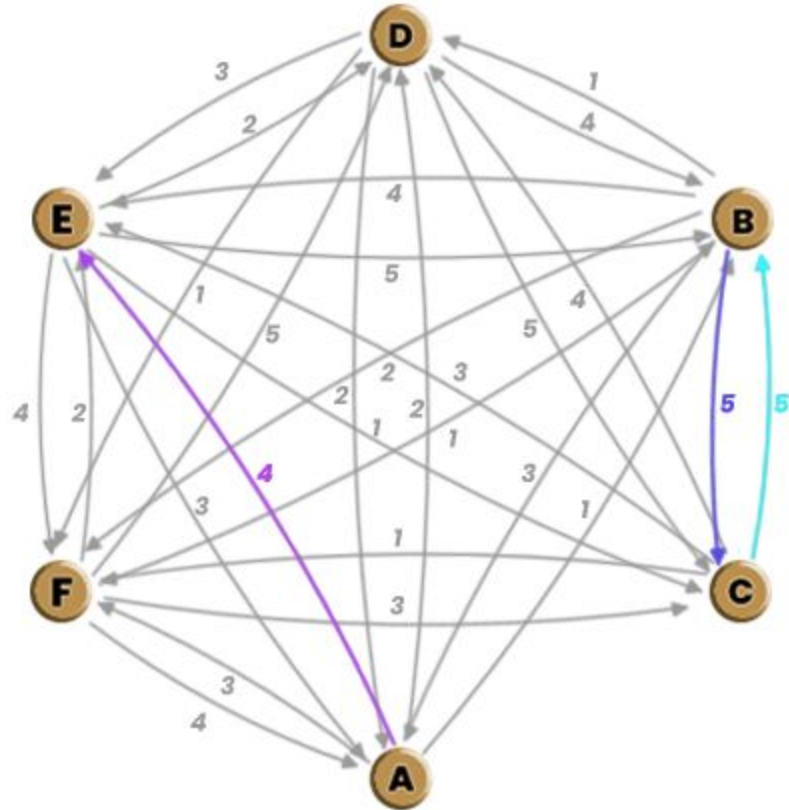
(D, ?)

(E, ?)

(F, ?)

## Phase 1

A fait sa demande à E qui est acceptée.



(A, E)

(B, C)

(C, B)

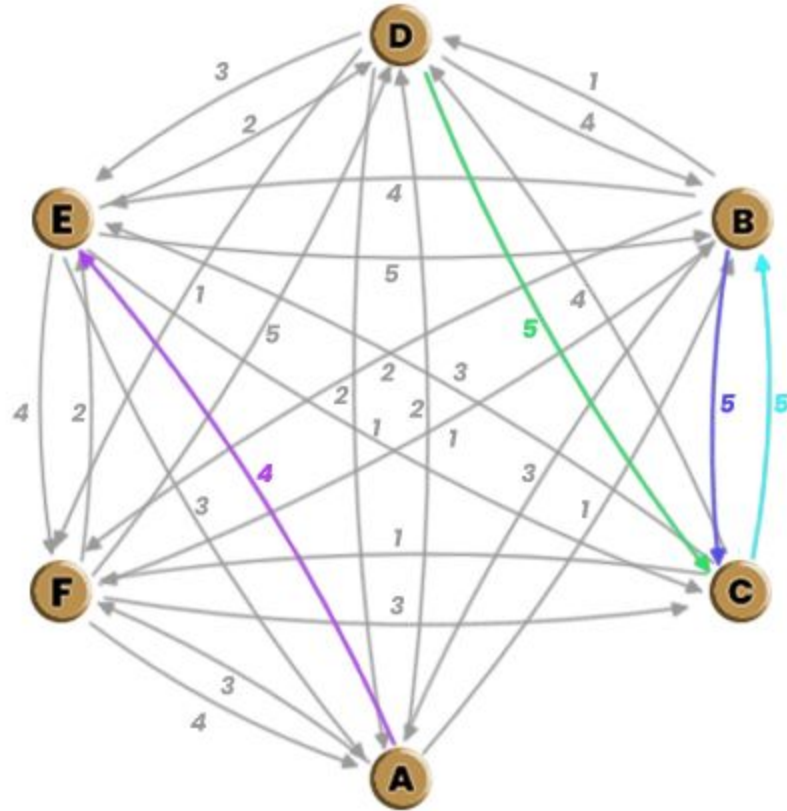
(D, ?)

(E, ?)

(F, ?)

## Phase 1

D fait une nouvelle demande à C, créant un nouveau conflit.



(A, E)

(B, C)

(C, B)

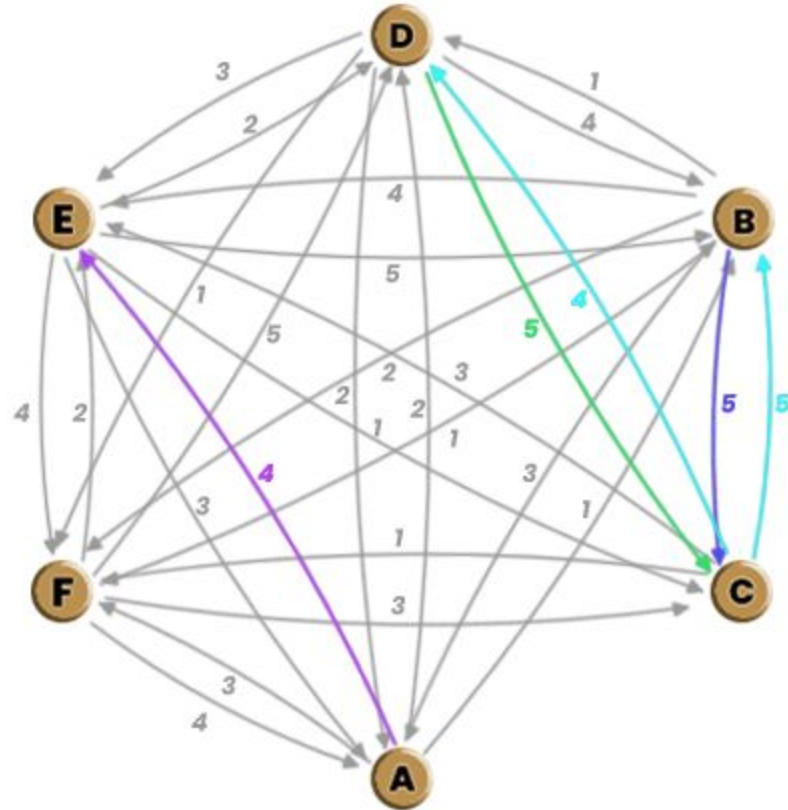
(D, C)

(E, ?)

(F, ?)

## Phase 1

D est naturellement refusé par C et doit donc faire une nouvelle demande.



(A, E)

(B, C)

(C, B)

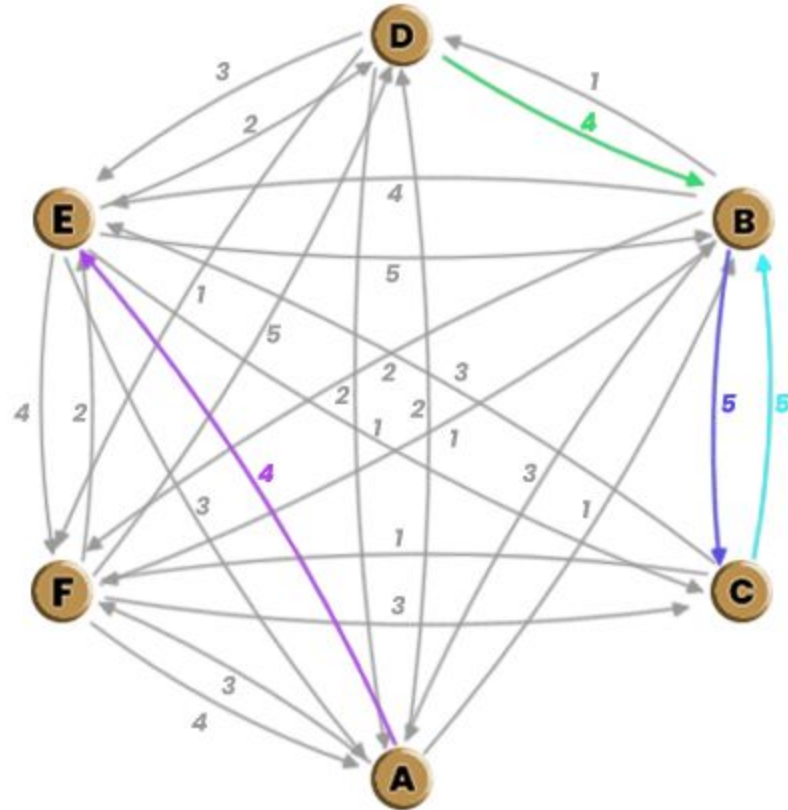
(D, ?)

(E, ?)

(F, ?)

## Phase 1

D fait sa nouvelle demande à B, qui va également provoquer un conflit.



(A, E)

(B, C)

(C, B)

(D, B)

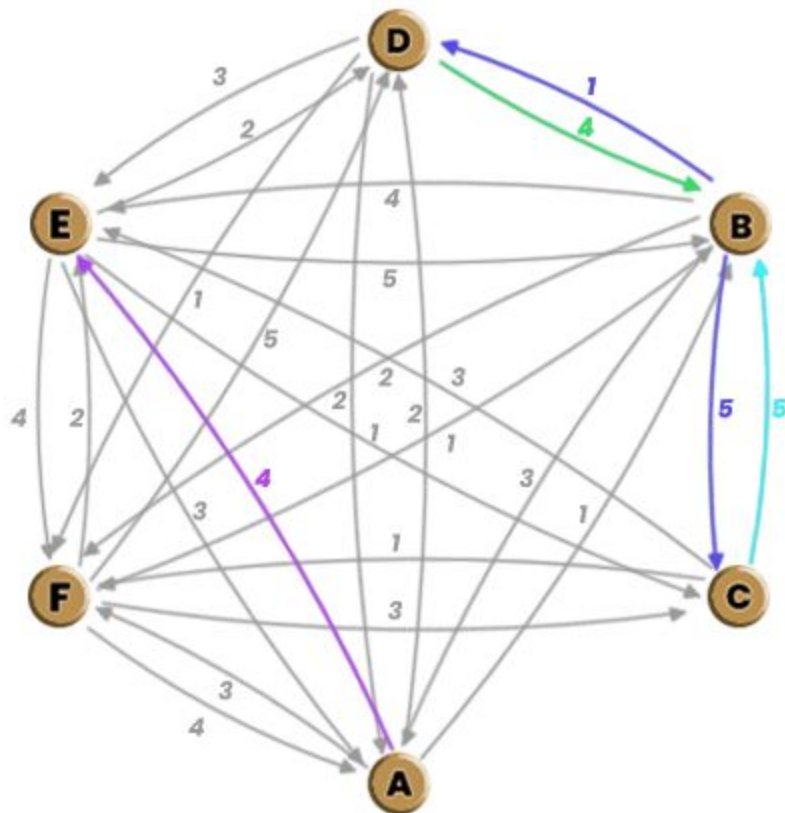
(E, ?)

(F, ?)



## Phase 1

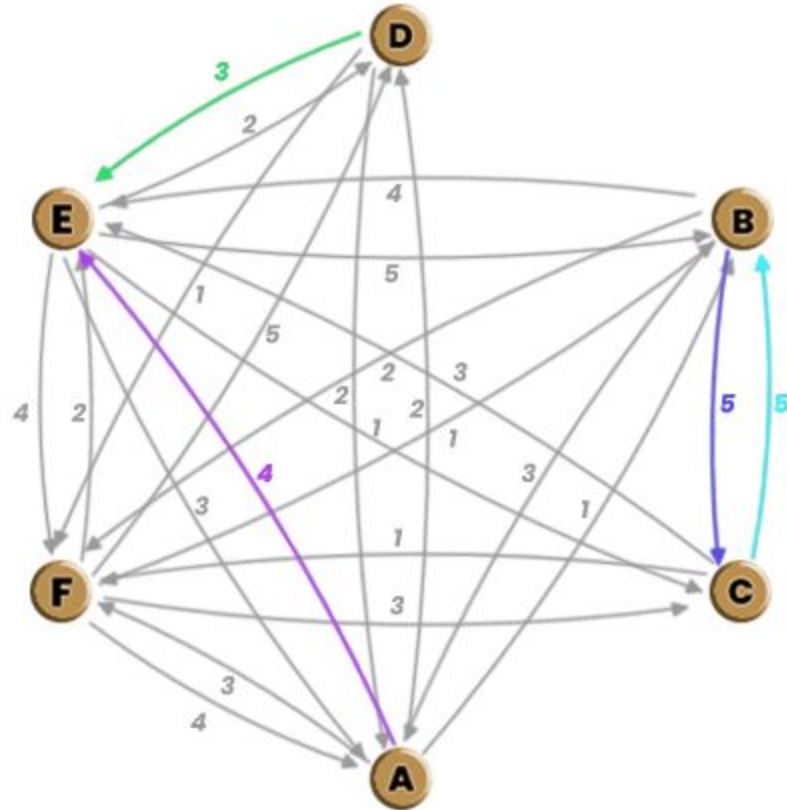
D perd à nouveau le conflit.



- (A, E)
- (B, C)
- (C, B)
- (D, ?)
- (E, ?)
- (F, ?)

## Phase 1

D fait une troisième proposition à E qui provoque un troisième conflit.



(A, E)

(B, C)

(C, B)

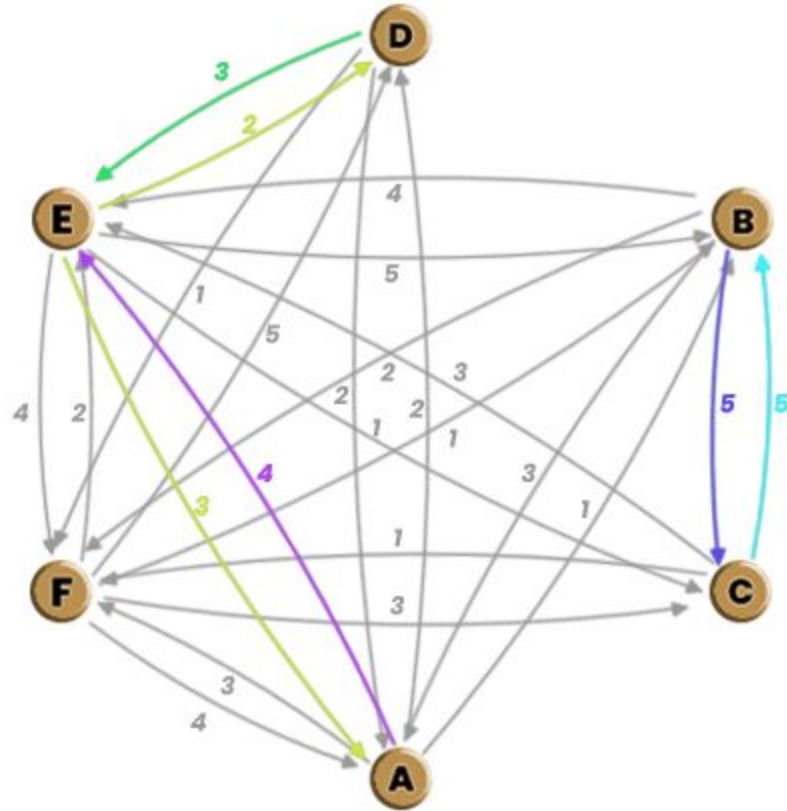
(D, E)

(E, ?)

(F, ?)

## Phase 1

Et à nouveau, D voit sa demande refusée.



(A, E)

(B, C)

(C, B)

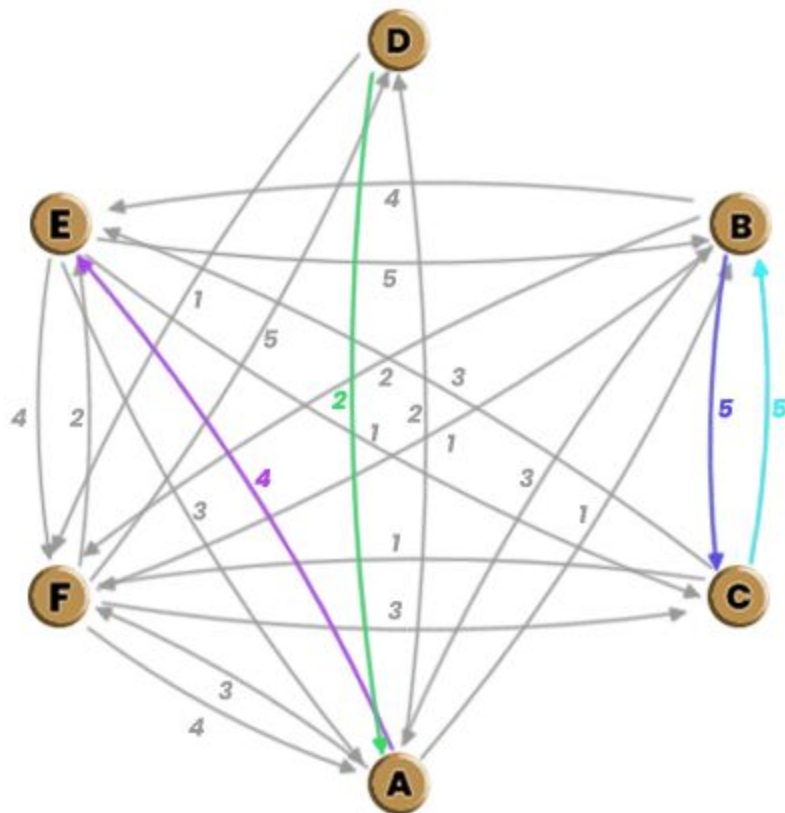
(D, ?)

(E, ?)

(F, ?)

## Phase 1

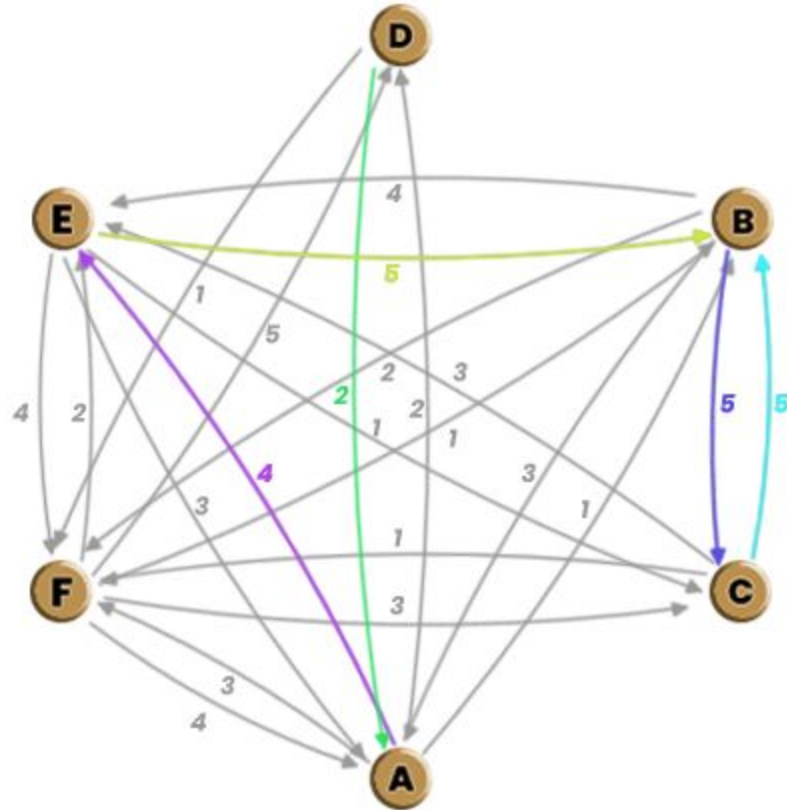
D fait une dernière demande vers A qui accepte.



- (A, E)
- (B, C)
- (C, B)
- (D, A)
- (E, ?)
- (F, ?)

## Phase 1

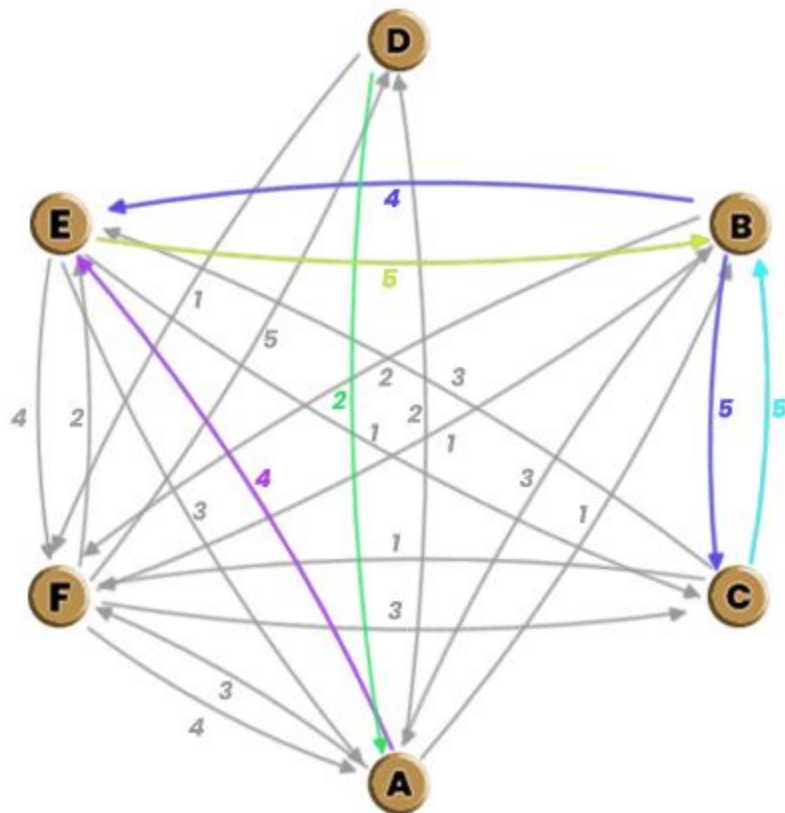
E fait sa demande à B, créant encore un conflit.



- (A, E)
- (B, C)
- (C, B)
- (D, A)
- (E, B)
- (F, ?)

## Phase 1

B préfère C à E.



(A, E)

(B, C)

(C, B)

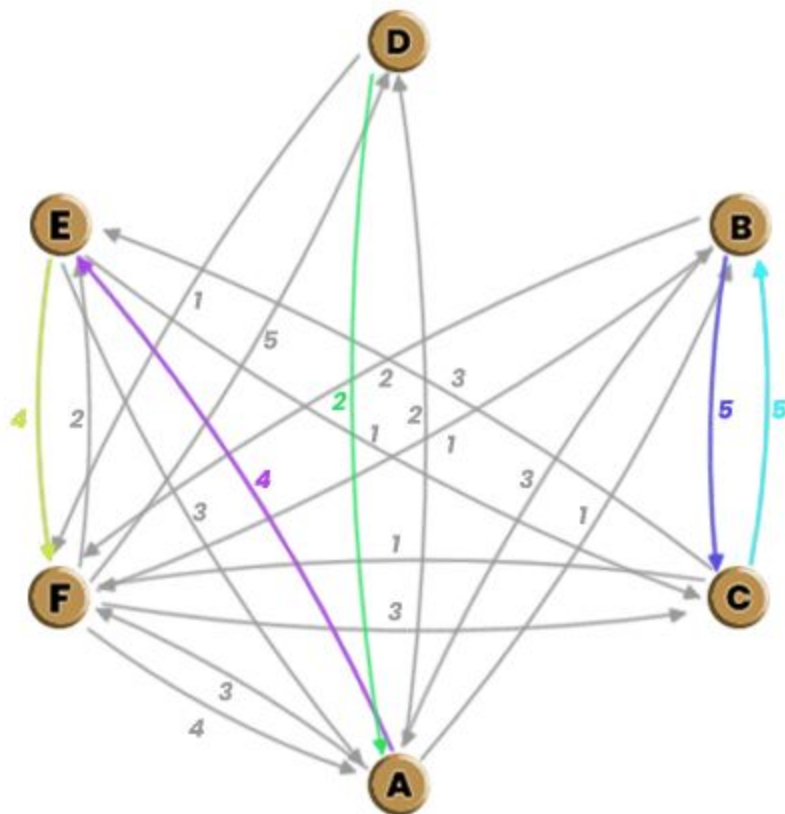
(D, A)

(E, ?)

(F, ?)

## Phase 1

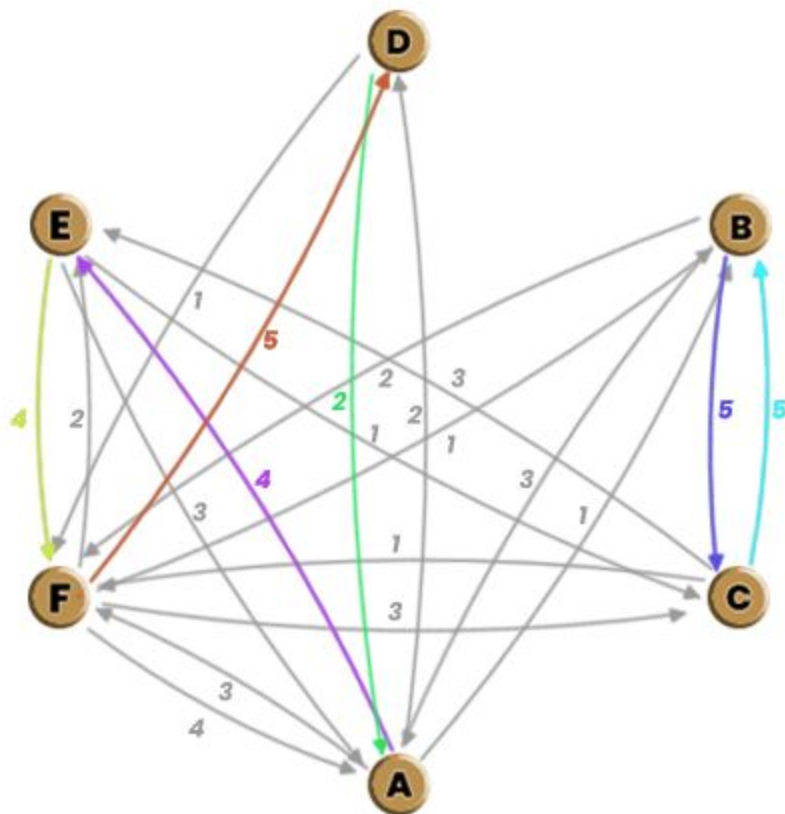
E fait sa proposition à F, qui l'accepte.



- (A, E)
- (B, C)
- (C, B)
- (D, A)
- (E, F)
- (F, ?)

## Phase 1

La phase 1 s'achève avec un couplage temporaire.



(A, E)  
(B, C)  
(C, B)  
(D, A)  
(E, F)  
(F, D)

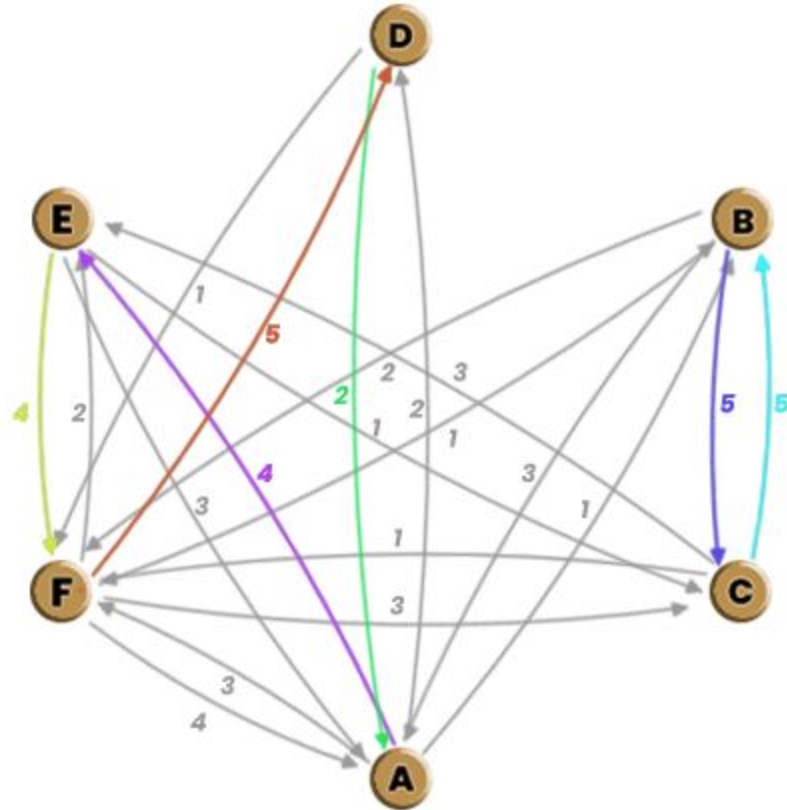


## Phase 2

La phase 2 élimine les couples indésirables, soit tous les couples moins intéressants que ceux déjà reçus par chaque sommets.

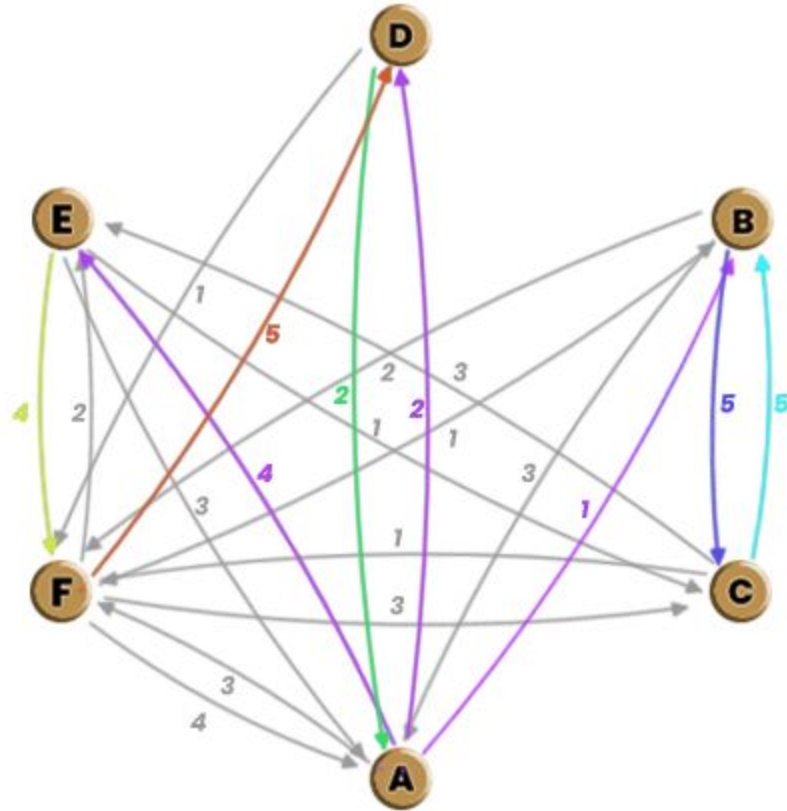
Afin de mieux montrer le fonctionnement de cette étape, les sommets B et C seront ignorés jusqu'à la fin de la phase pour éviter de retirer toutes les options des autres sommets.

Pour représenter les couplages, on utilisera une liste alphabétique, qui ignore les poids des arcs.



## Phase 2

On regarde d'abord le cas de A qui est demandé par D.

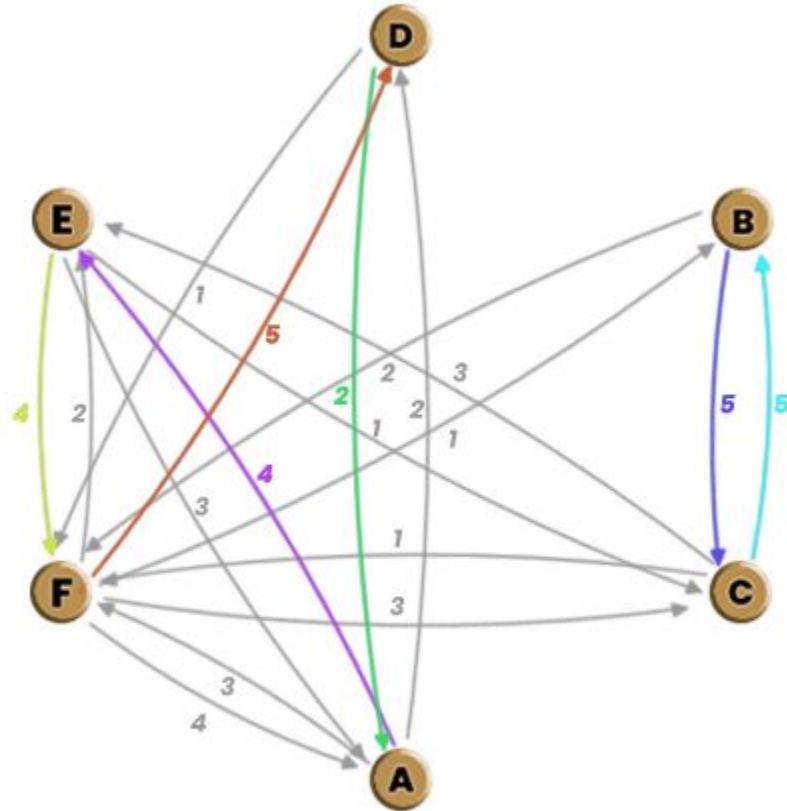


**(A: BDEF)**  
(D: AF)  
(E: ACF)  
(F: ABCDE)

**(B, C)**  
**(C, B)**

## Phase 2

Il existe un seul couplage pire que  $(A, D)$  pour A, le couple  $(A, B)$  qui est retiré du graphe.

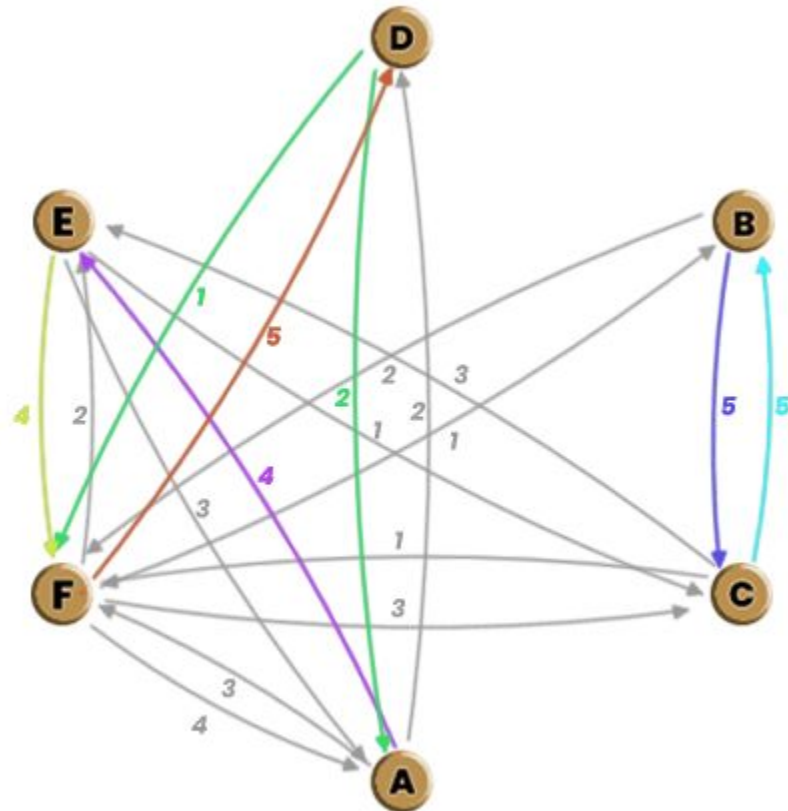


**(A: DEF)**  
(D: AF)  
(E: ACF)  
(F: ABCDE)

**(B, C)**  
**(C, B)**

## Phase 2

On regarde maintenant les options de D. Sa seule demande étant son pire choix, il n'y a rien à faire.

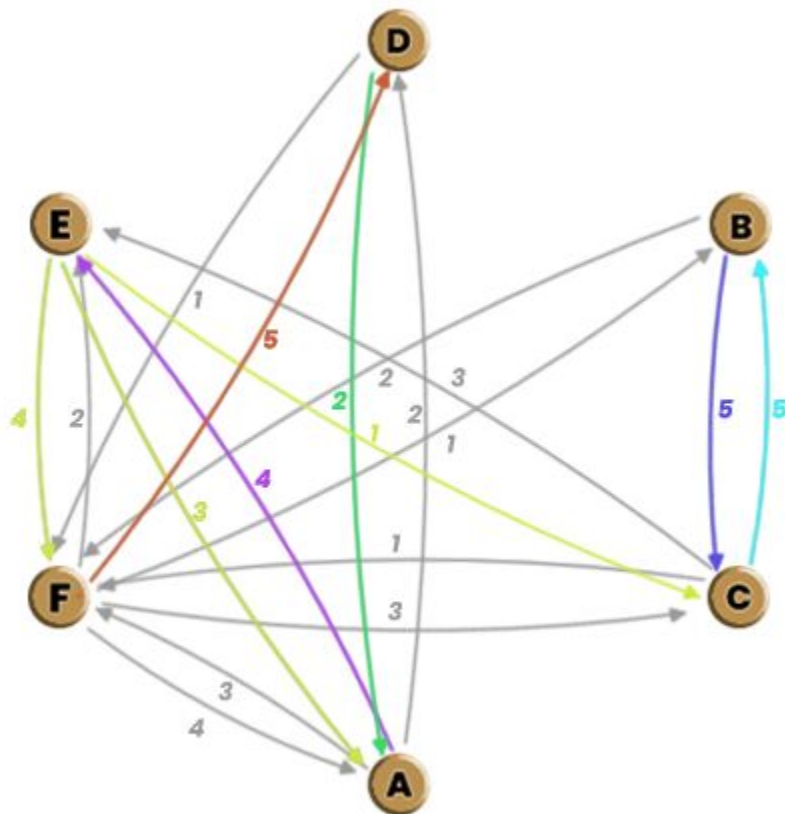


(A: DEF)  
(D: **AF**)  
(E: ACF)  
(F: ABCDE)

(B, C)  
(C, B)

## Phase 2

On passe directement à E.

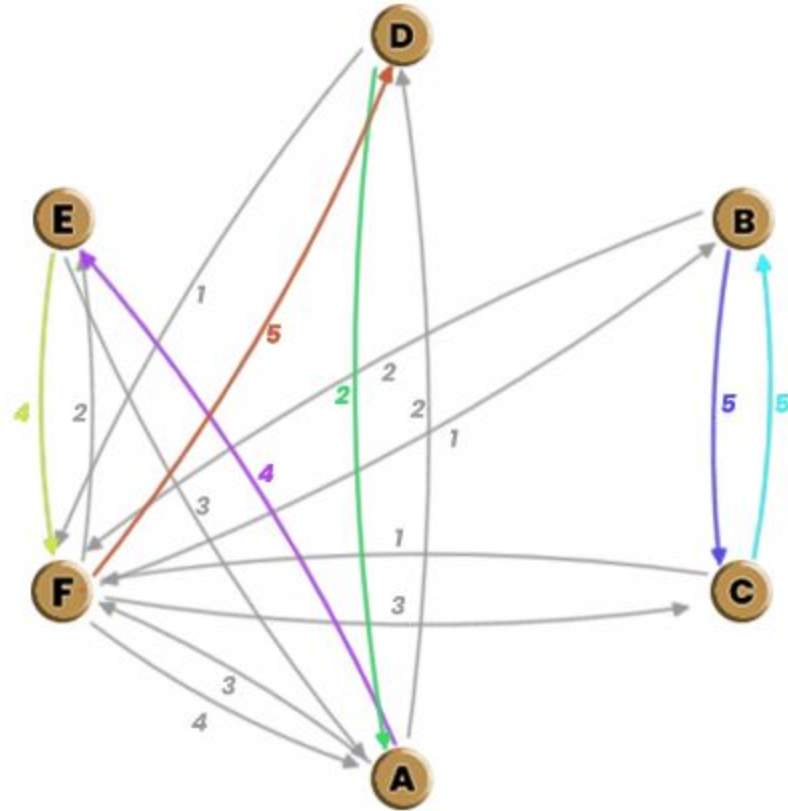


(A: DEF)  
(D: AF)  
**(E: ACF)**  
(F: ABCDE)

(B, C)  
(C, B)

## Phase 2

Sa demande vient de son deuxième choix, (E, C) est retiré.

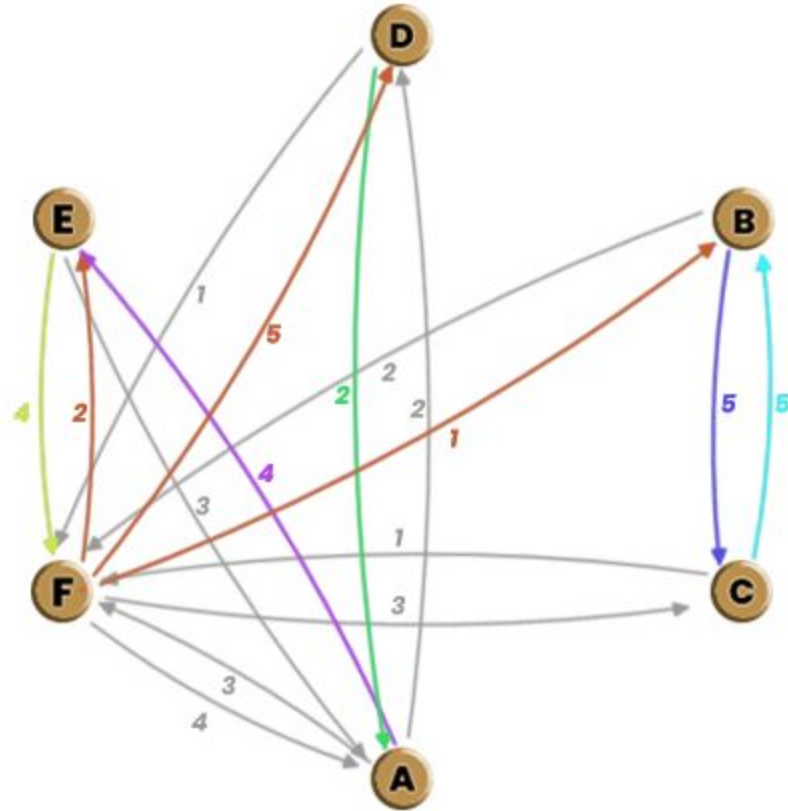


(A: DEF)  
(D: AF)  
**(E: AF)**  
(F: ABCDE)

**(B, C)**  
**(C, B)**

## Phase 2

On fini avec le cas de F. Qui est demandé par son deuxième pire choix.

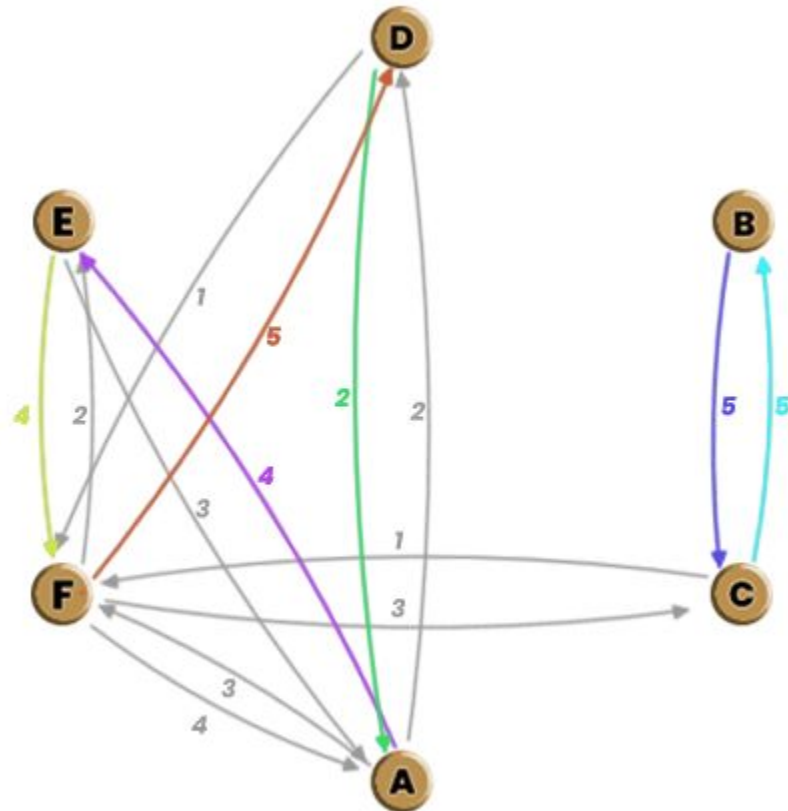


(A: DEF)  
(D: AF)  
(E: AF)  
**(F: ABCDE)**

(B, C)  
(C, B)

## Phase 2

On supprime alors (B, C), la pire option de F.



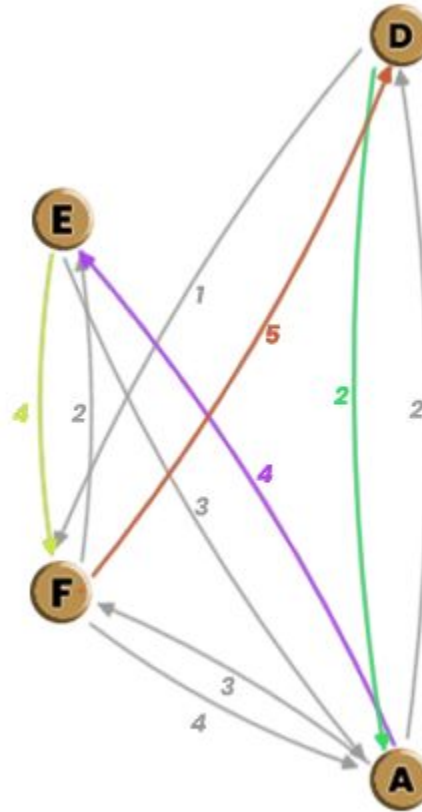
(A: DEF)  
(D: AF)  
(E: AF)  
**(F: ACDE)**

(B, C)  
(C, B)



## Phase 3

On supprime le dernier couple de C qui n'a pas été retiré pour pouvoir procéder à la troisième et dernière phase.

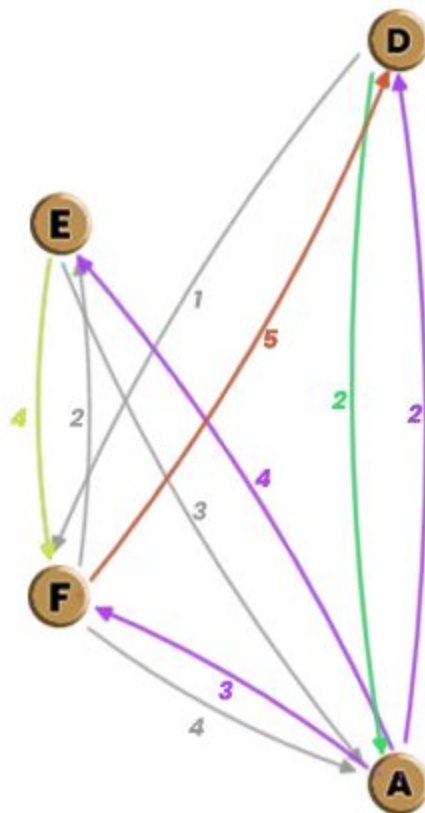


(A: DEF)  
(D: AF)  
(E: AF)  
(F: ADE)

(B, C)  
(C, B)

## Phase 3

On cherche la deuxième préférence de A, ici il s'agit de F.



**(A: DEF)**

(D: AF)

(E: AF)

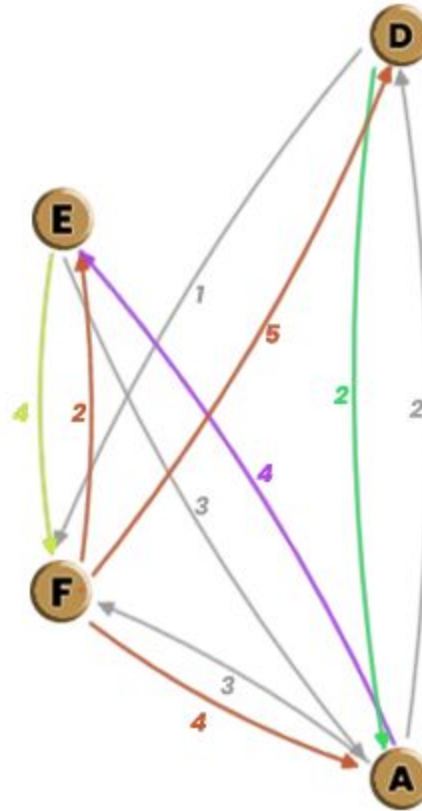
(F: ADE)

(B, C)

(C, B)

## Phase 3

On veut maintenant supprimer le pire choix de F.

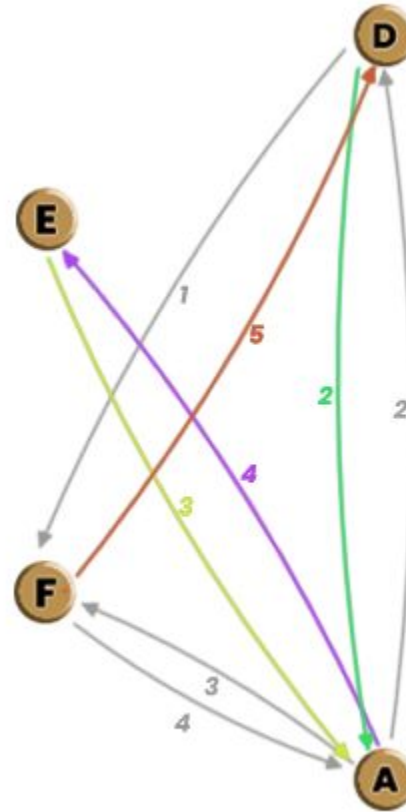


(A: DEF)  
(D: AF)  
(E: AF)  
**(F: ADE)**

**(B, C)**  
**(C, B)**

### Phase 3

Le couple (E, F) est retiré du graphe  
et on va vers le deuxième meilleur,  
et seul, choix de E, qui est ici A.

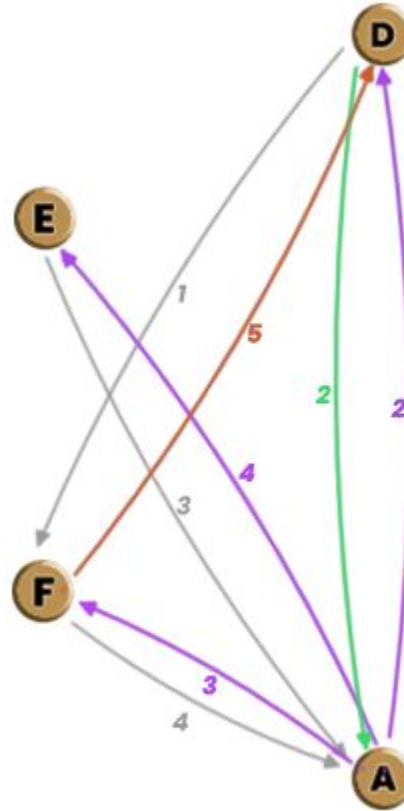


(A: DEF)  
(D: AF)  
(E: A)  
**(F: AD)**

**(B, C)**  
**(C, B)**

## Phase 3

On répète le processus avec A, en supprimant son pire choix.



**(A: DEF)**

**(D: AF)**

**(E: A)**

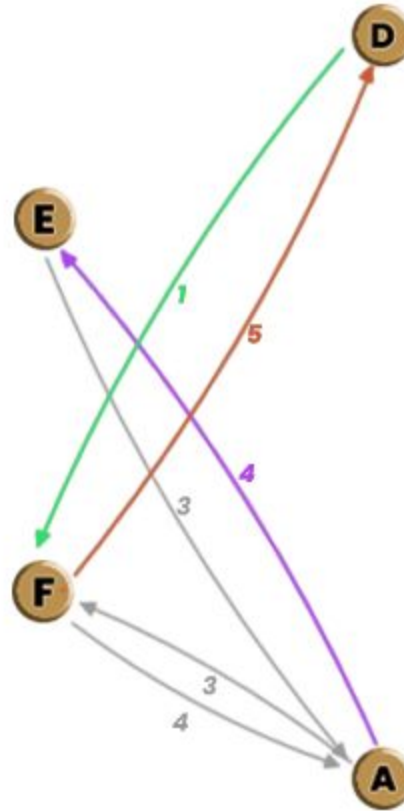
**(F: AD)**

**(B, C)**

**(C, B)**

### Phase 3

On supprime ici le couple (A, D),  
et allons sur le dernier choix de D,  
qui est F.

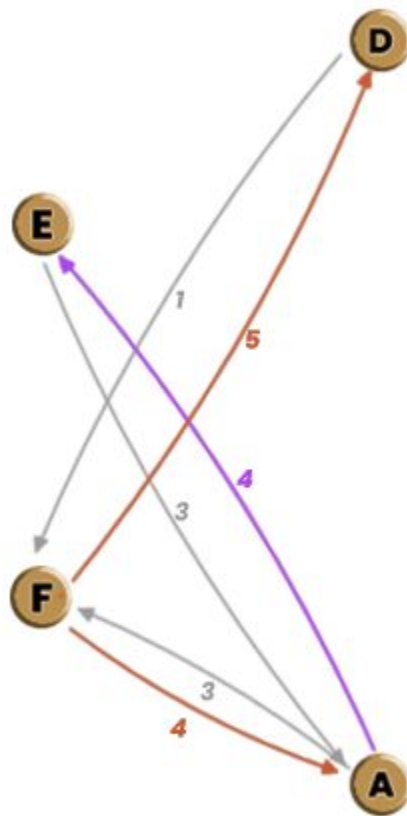


**(A: EF)**  
(D: F)  
(E: A)  
(F: AD)

**(B, C)**  
**(C, B)**

## Phase 3

F cherche alors son dernier pire choix.

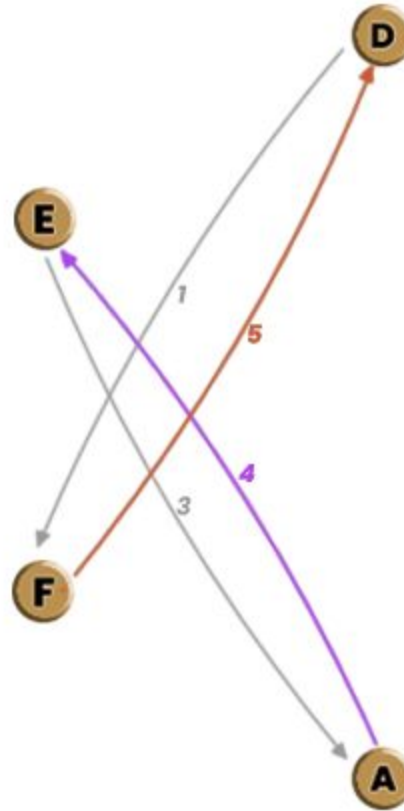


(A: EF)  
(D: F)  
(E: A)  
**(F: AD)**

(B, C)  
(C, B)

### Phase 3

A étant le pire choix restants pour F,  
le couple (A, F) est retiré.



(A: E)  
(D: F)  
(E: A)  
**(F: D)**

**(B, C)**  
**(C, B)**



## Conclusion

On a alors 3 composantes connexes, représentant les 3 couples stables du graphe fourni/

(A, E)

(B, C)

(D, F)

