Université de Nantes

M1 Informatique – Graphes
Problèmes de couplage

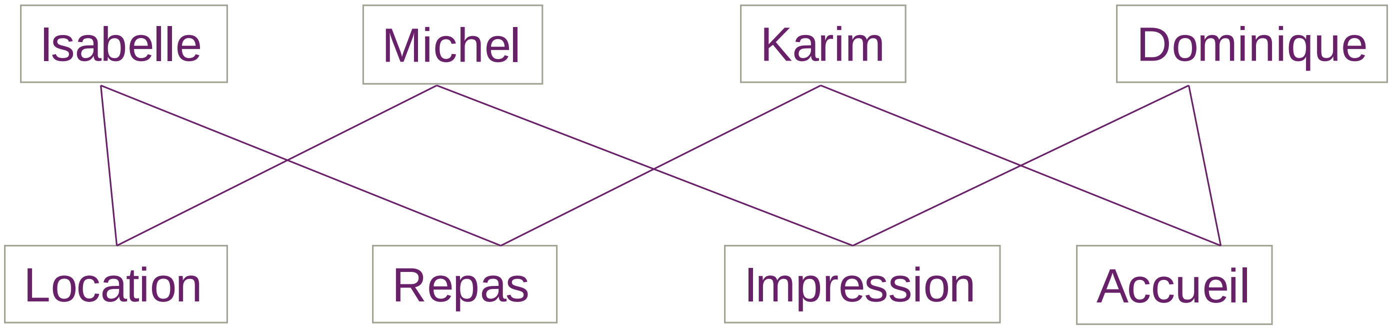
Irena.Rusu@univ-nantes.fr
LS2N, bât. 34, bureau 303
tél. 02.51.12.58.16

- Deux exemples
- Généralités et première tentative d'algorithme
- Algorithme de couplage
  - dans les graphes bipartis
  - dans les graphes arbitraires
- Problème de l'affectation stable

- A. Schrijver – A course in Combinatorial Optimization (2003).

- © Transparents de Hans Bodlaender, Anselm Ringleben, Lap Chi Lau, Kaspar Riesen, I. Rusu

- Deux exemples
- Généralités et première tentative d’algorithme
- Algorithme de couplage
  - dans les graphes bipartis
  - dans les graphes arbitraires
- Problème de l’affectation stable



Problème d’affectation de tâches:

Chaque personne est disposée à effectuer un certain nombre de tâches.

Peut-on trouver une affectation qui assure que chaque tâche sera effectuée ?

- **Soit :** Un réseau de téléphonie mobile
  - Les sommets sont des antennes-relais
  - Chaque arc relie deux antennes-relais qui sont capables de communiquer directement entre elles (deux telles antennes sont dites voisines)

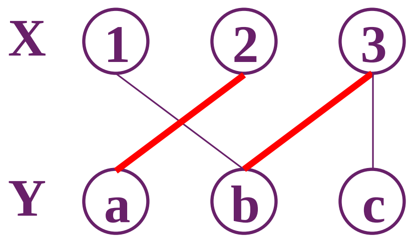
- **A faire :** Une antenne-relais qui ne fonctionne plus doit être identifiée aussi rapidement que possible. Pour cela, il a été décidé de définir des couples d'antennes-relais voisines qui vérifient chacune si l'autre est toujours en fonctionnement (en s'envoyant réciproquement des signaux).

- **Problème:** Etant donné un réseau, trouver un ensemble *maximum* de tels couples.

- Deux exemples
- Généralités et première tentative d'algorithme
- Algorithme de couplage
  - dans les graphes bipartis
  - dans les graphes arbitraires
- Problème de l'affectation stable

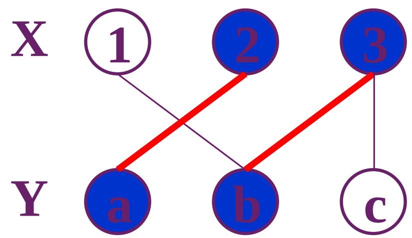
- G= (V, A) un graphe non-orienté avec n=|V| et m=|A|

- Couplage : sous-ensemble M d'arêtes de G tel que tout sommet est incident à au plus une arête.



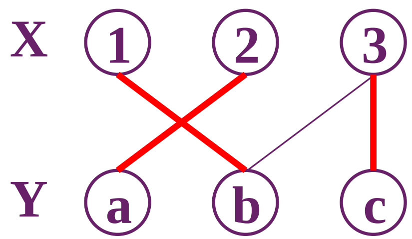
Sommet saturé : sommet v de G qui est incident à une arête de M.
Dans le cas contraire, v est libre ou non-saturé ou insaturé.

Arête du couplage : arête de G qui appartient à M.

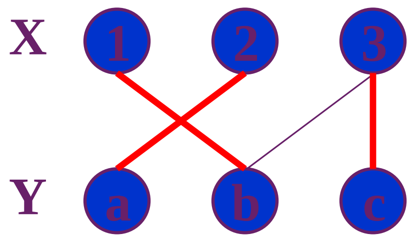


Couplage parfait : Un couplage de cardinalité p/2 dans un graphe G à p sommets.

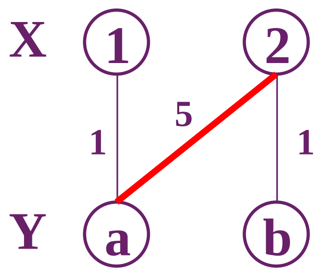
Note. Si un tel couplage existe, p doit être pair (condition nécessaire, pas suffisante).

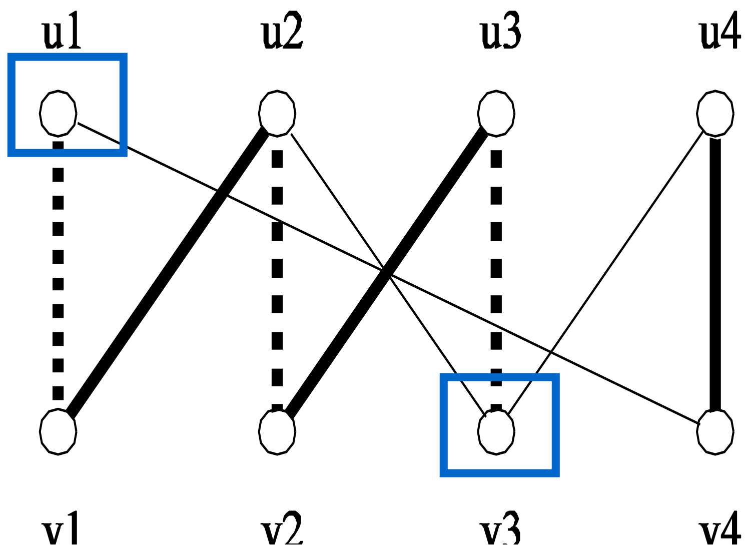


Couplage (de cardinalité) maximum : un couplage dont la cardinalité est aussi grande que possible dans G.



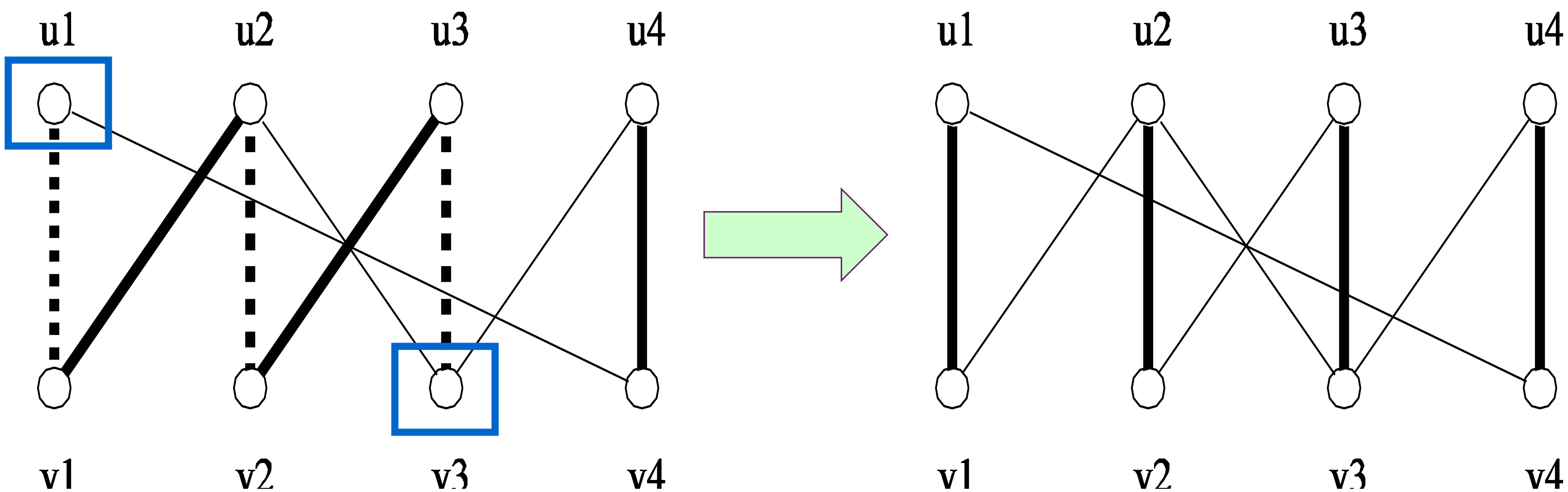
Couplage de poids maximum : Couplage dans un graphe pondéré, tel que la somme des poids de ses arêtes est aussi grande que possible dans G.





Etant donné un couplage M :

- un chemin alternant par rapport à M est un chemin dont les arêtes dans M et en dehors de M alternent.
- un chemin alternant par rapport à M et dont les extrémités sont non-saturées est un chemin de croissance par rapport à M.



M couplage, P chemin de croissance par rapport à M

Echange (ou transfert) dans M par rapport à P : l'operation qui enlève de M toutes les arêtes de M qui sont dans P et ajoute à M les autres arêtes de P.

$\boxed{M^* = M \oplus P}$ le couplage obtenu

M:=Ø
Tant qu'il existe un chemin de croissance P par rapport à M faire
    $M := M \oplus P$
Retourner M

Correction ?

- Partie assez facile :

  si M est un couplage maximum $\Rightarrow$ il n'y a pas de chemin de croissance par rapport à M

- Mais des questions ouvertes restent :
  - S'il n'y a pas de chemin de croissance $\Rightarrow$ M est-il maximum?
  - Comment trouver P efficacement ?

Un couplage M d‘un graphe G est maximum
si et seulement si
il n‘existe pas de chemin de croissance dans G.

Preuve :

$\Rightarrow$: M couplage maximum de G.
S‘il existe un chemin de croissance P, alors en faisant un échange par rapport à P on obtient un

couplage M´ tel que |M´| = |M| +1.

Impossible, puisque M est maximum

$\Rightarrow$ il n‘existe pas de chemin de croissance par rapport à M

Théorème de Berge (2)

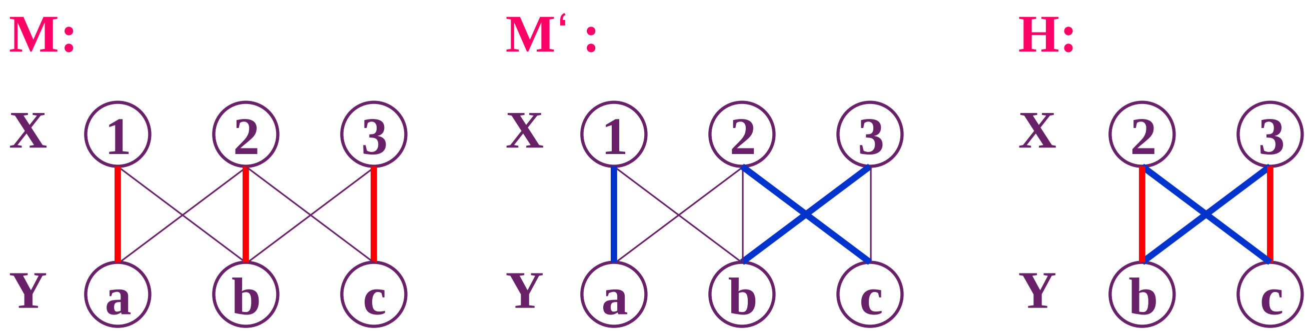
$\Leftarrow$: Soit M un couplage tel qu'il n'y a pas de chemin de croissance p.r. à M.
On veut montrer que M est maximum.
Soit M´ un couplage maximum arbitraire. On montre que |M| = |M´|.

Soit H le sous-graphe de G défini par l'ensemble d'arêtes :

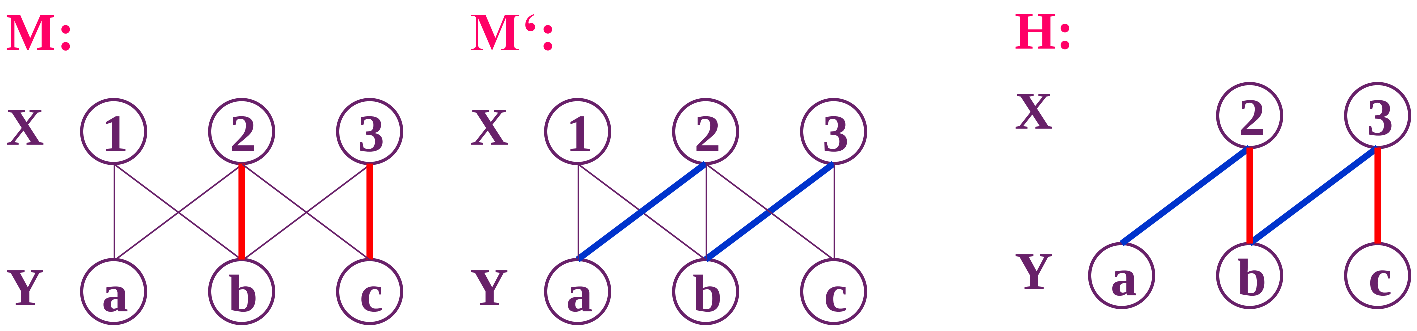
$$M \,\Delta\, M' = (M - M') \cup (M' - M)$$

Alors chaque composante connexe de H est du type 1 ou 2 ci-dessous :

1. Un cycle pair dont les arêtes sont alternativement dans M et dans M'.



2. Un chemin dont les arêtes sont alternativement dans M et M‘, et dont les extrémités sont saturées soit par M soit par M‘ mais pas par les deux.



(Pourquoi pas un autre type ?)

Conclusion: H a le même nombre d’arêtes rouges et bleues

$\Rightarrow$ |M| = |M´|.

Idée : *M* est maximum ⇔ il n'y a pas de chemin de croissance p.r. à *M*.

Algorithme CouplageMax

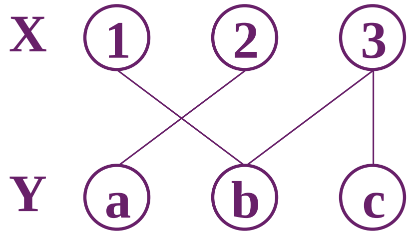
M:=Ø
Tant qu'il existe un <u>chemin de croissance P par rapport à M</u> faire
        M : = M ⊕ P
Retourner M

Comment le trouver efficacement ?

- Deux exemples
- Généralités et première tentative d'algorithme
- Algorithme de couplage
  - dans les graphes bipartis
  - dans les graphes arbitraires
- Problème de l'affectation stable

Graphe biparti :

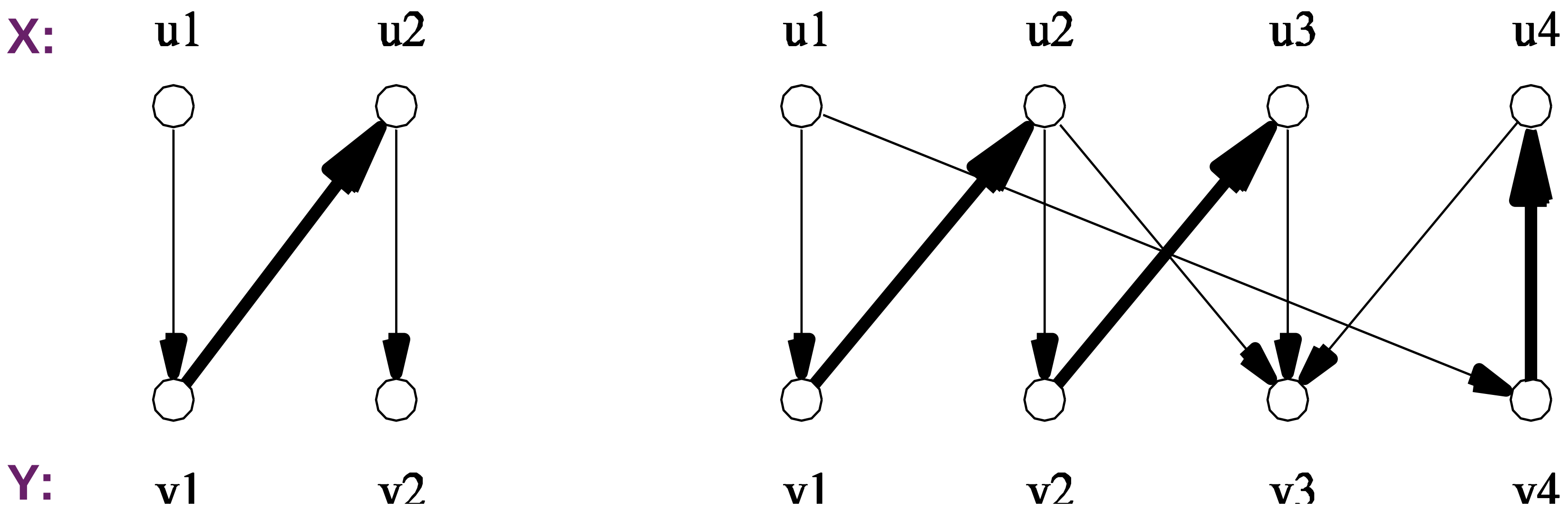
Graphe $G = (X \cup Y, A)$ avec $X \cap Y = \Phi$ et tel que toutes les arêtes ont exactement une extrémité dans chaque ensemble X, Y.



Pourquoi ?

Les graphes bipartis sont beaucoup plus simples que les graphes arbitraires, et font – malgré leur simplicité - l'objet de beaucoup d'applications.

Orienter les arêtes (celles de *M* vers le haut, les autres vers le bas)



Un chemin de croissance par rapport à M correspond à un chemin orienté avec les deux extrémités insaturées par M.

- Chercher un chemin de croissance :
  - Orienter les arêtes : $O(|A|)$
  - Trouver un chemin orienté : $O(|A|)$

  (parcourir à partir d'un sommet insaturé; si chemin pas trouvé, effacer tous les arcs parcourus et recommencer)
- Calculer $M := M \oplus P$: longueur de P, c.à.d. $O(|A|)$
- Nombre d'itérations : $O(|X|)$ (par exemple; $O(|Y|)$ fonctionne aussi)

$\Rightarrow$ $O(mn)$, où m = nombre d'arêtes de G
n = nombre de sommets de G

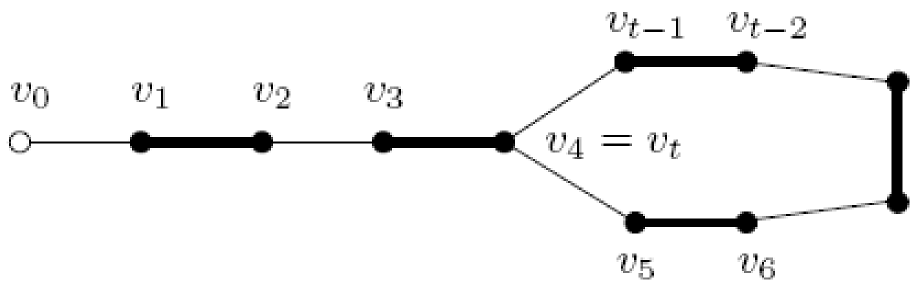
- Deux exemples
- Généralités et première tentative d'algorithme
- Algorithme de couplage
  - dans les graphes bipartis
  - dans les graphes arbitraires
- Problème de l'affectation stable

Idée : *M* est maximum $\Leftrightarrow$ il n'y a pas de chemin de croissance p.r. à *M.*

Algorithme CouplageMax

M:=Ø
Tant qu'il existe un chemin de croissance P par rapport à M faire
    M : = M $\oplus$ P
Retourner M

Comment le trouver efficacement ?

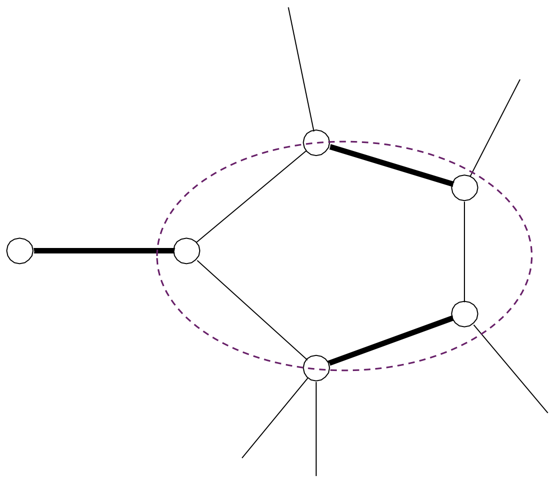


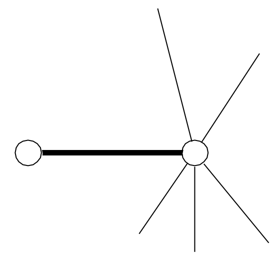
Un chemin alterné P=(v_0, v_1, …, v_t) est appelé une fleur par rapport à M si t est impair, v_0, …, v_{t-1} sont distincts, v_0 est non-saturé et il existe i<t tel que v_t=v_i.

Le cycle impair C=(v_i, v_{i+1}, …, v_t) est appelé une inflorescence par rapport à M (associée à la fleur).

Edmonds 1965
Algorithme CouplageMax pour graphes arbitraires : contracter les inflorescences

G





G/C

Obs. 1 La contraction n'affecte pas la maximalité du couplage.

Obs. 2 La contraction n'affecte pas les chemins de croissance.

Mais : elle aide beaucoup à la recherche des chemins.

- A partir d'un sommet insaturé, construire un chemin d'alternance avec des sommets distincts (fait avec un parcours en profondeur ou en largeur)

- Le résultat est soit un chemin de croissance, soit une inflorescence.

- Dans le deuxième cas, contracter l'inflorescence C.

- Sur le graphe résultant G/C, trouver récursivement un chemin de croissance. Le transformer en un chemin de croissance du graphe G.

M:=∅ ;

Tant que (*G* contient au moins deux sommets insaturés) **faire**

  soit *u* un sommet insaturé

  (*AT,P*) ← *ConstructionArbreAlterné(G, u, M);* // voir page suivante

  si (un chemin de croissance *P* a été trouvé) **alors**

    transformer *P* (si besoin) en un chemin de croissance de *G*

    augmenter *M* utilisant *P* (faire l'échange $M := M \oplus P$)

  sinon

    effacer de *G* les sommets de *AT* (mais se rappeler le couplage)

  finsi

fintantque

ConstructionArbreAlterné(G, u, M) : (chemin, arbre alterné)

S(AT)={u}, A(AT)=Ø; marquer *u* avec un “P”;

répéter

considérer le **premier** cas valide ci-dessous :

cas A. Il existe un sommet étiqueté “P” de *AT* qui est adjacent à un sommet insaturé de *G*. Alors un chemin de croissance *P* est trouvé. *STOP:=Vrai*.

cas B. Il existe dans *AT* un sommet étiqueté “P” que l’on note *x,* ainsi qu’un chemin *xyz* tel que (1) *y,z* n’appartiennent pas à *AT*, (2) $xy \notin M$, (3) $yz \in M$. Alors ajouter *y* (marqué “I”), *z* (marqué “P”) et les arêtes *xy, yz* à *AT*. *STOP:= Faux.*

cas C. Il existe une arête, n’appartenant pas à *AT*, entre deux sommets “P” de *AT*. Alors une inflorescence *C* est trouvée. La contracter dans un pseudo-sommet pour obtenir *G/C.* *STOP:= Faux*

cas D. Aucun des cas A, B, C n’est valide. Alors il n’y a pas de chemin de croissance. *STOP:= Vrai.*

jusqu’à *STOP*

retourner *P* (qui est vide s’il n’a pas été trouvé) et *AT*

- Au plus ***n*** sommets sont utilisés pour commencer la recherche d'un chemin de croissance.
- Chaque recherche effectue au plus ***n*** contractions, et chaque contraction nécessite de trouver un chemin d'alternance (***O(m)***).
- Le temps total est donc en ***$O(mn^2)$.***

Note. Pour le couplage de poids maximum dans un graphe arbitraire, des algorithmes polynomiaux existent également (ils sont bien plus compliqués).

- Deux exemples
- Généralités et première tentative d'algorithme
- Algorithme de couplage
  - dans les graphes bipartis
  - dans les graphes arbitraires
- Problème de l'affectation stable

- **Contexte :** Il y a *n* hôpitaux qui doivent recruter *n* diplômés en médicine. Chaque hôpital a une liste de préférences (décroissante) propre des *n* diplomés, et chaque diplômé à une liste de préférences (décroissante) propre des *n* hôpitaux.

- Est-ce qu'il existe, et peut-on la trouver le cas échéant, une **affectation stable** des diplômés aux hôpitaux ?

  affectation stable : couplage parfait M de type hôpital-diplômé tel qu'aucune paire H, D n'existe avec la propriété (dite *d'instabilité*) que H préfère D à son affectation par M, et D préfère H à son affectation par M.

Note : également connu comme problème du marriage stable.

- H1: 2 1 3
- H2 1 3 2
- H3 1 3 2
- D1 2 1 3
- D2 1 3 2
- D3 2 1 3

- Affectation stable : (1,2), (2,1), (3,3)

- Le couplage (1,1), (2,2), (3,3) n’est pas une affectation stable. Par exemple, H1 et D2 se préfèrent mutuellement à leurs affectations actuelles (autrement dit, ils satisfont la propriété d’instabilité).

- Ordonner les hôpitaux selon un ordre arbitraire.

- Répéter jusqu'à obtenir *n* couples hôpital-diplômé

  - Soit X le premier hôpital sans affectation selon l'ordre défini
  - Trouver un diplômé Y tel que
    - Soit Y est le diplômé le plus souhaité dans la liste de X, tel que Y est non-affecté.
    - Soit Y est actuellement affecté à un hôpital Z, et Y préfère X à Z.
  - Ajouter la paire X, Y au couplage. Eventuellement, Z n'aura plus d'affectation.

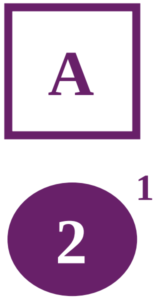
Hôpitaux

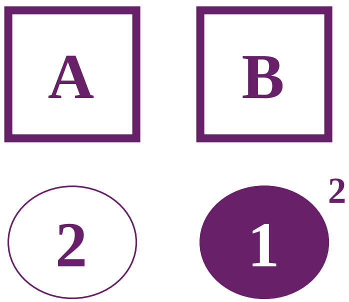
A	B	C	D	E
2	1	2	1	5
5	2	3	3	3
1	3	5	2	2
3	4	4	4	1
4	5	1	5	4

Diplômés

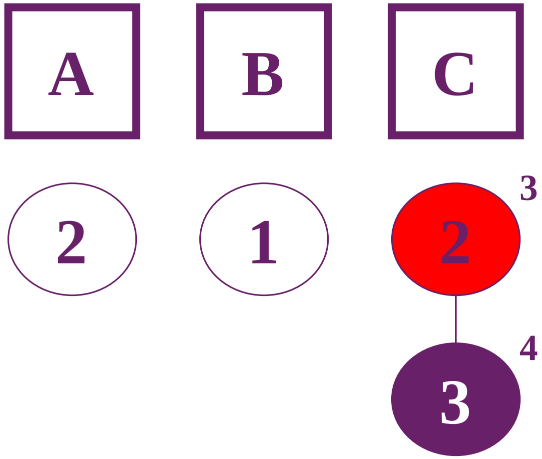
1	2	3	4	5
E	D	A	C	D
A	E	D	B	B
D	B	B	D	C
B	A	C	A	E
C	C	E	E	A

- **L'hôpital A fait une proposition au diplômé 2, qui l'accepte.**

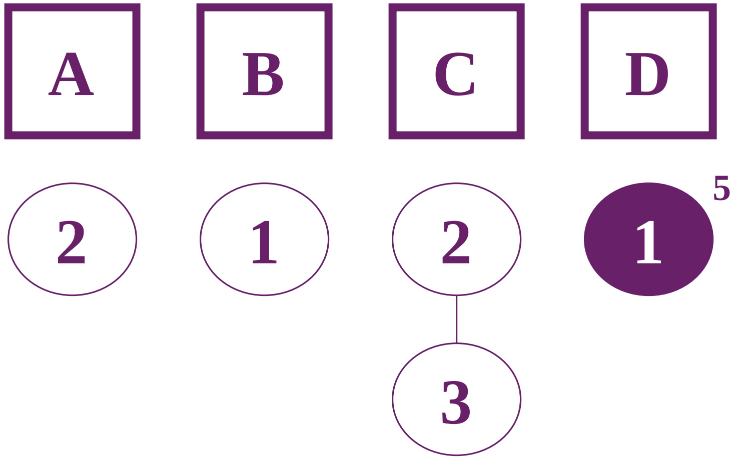




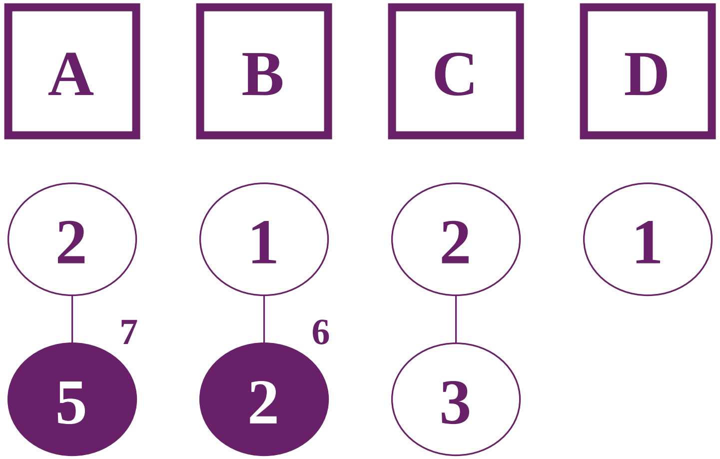
- **L'hôpital B fait une proposition au diplômé 1, qui accepte.**



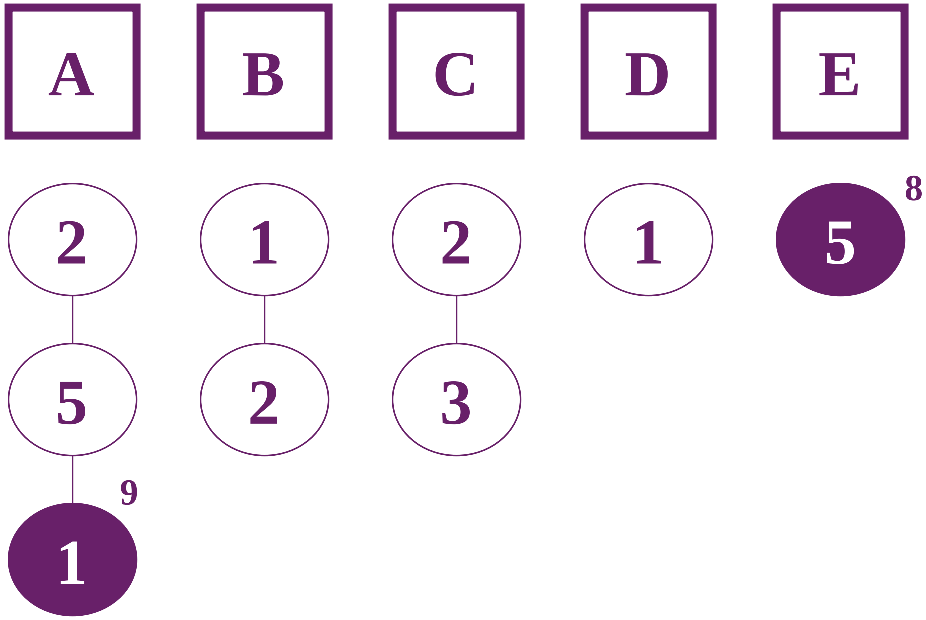
- L'hôpital C fait une proposition au diplômé 2, qui la refuse.
- C fait une proposition au diplômé 3, qui l'accepte.



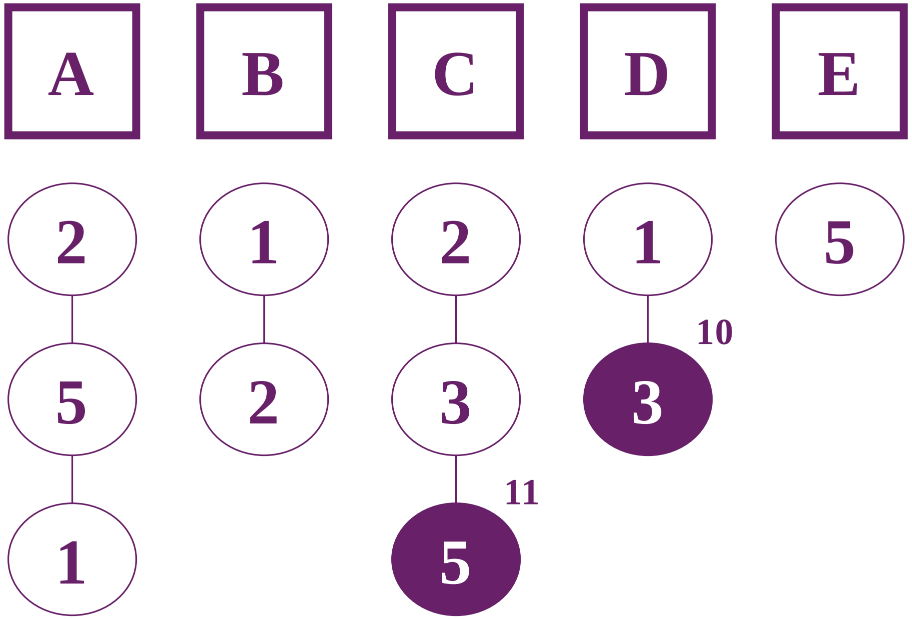
- L’hôpital D fait une proposition au diplômé 1, qui l’accepte.
- L’hôpital B n’a plus de diplômé affecté.



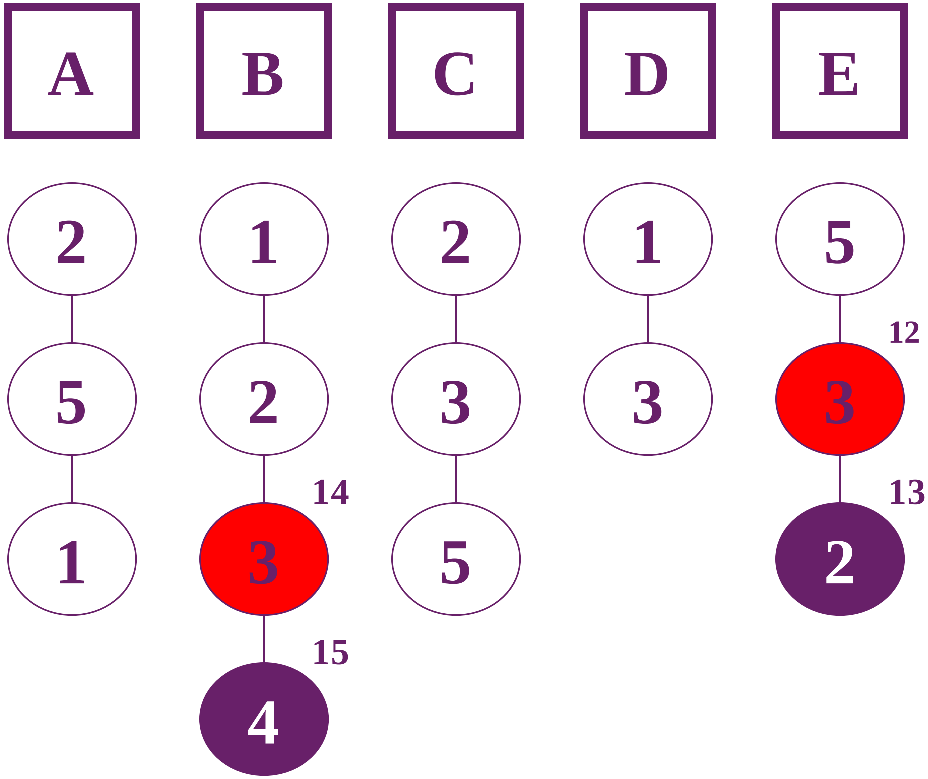
- L’hôpital B fait une proposition au diplômé 2, qui l’accepte.
- L’hôpital A est à la recherche d’un diplômé.
- Il fait une proposition au diplômé 5, qui l’accepte.



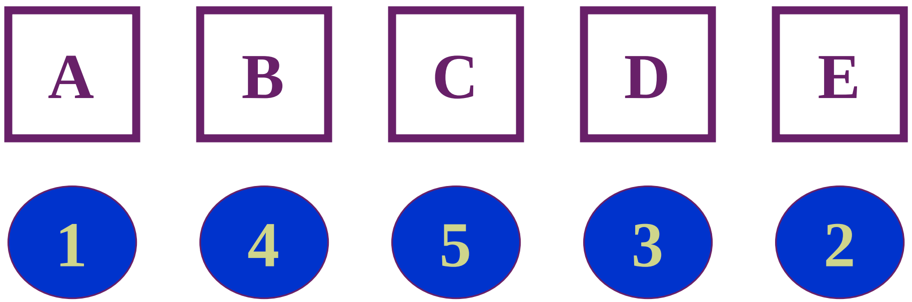
- L'hôpital E fait une proposition au diplômé 5, qui l'accepte.
- L'hôpital A repart à la recherche d'un diplômé. Il fait une proposition au diplômé 1, qui l'accepte.



- L'hôpital D fait une proposition au diplômé 3, qui l'accepte.
- L'hôpital C fait une proposition au diplômé 5, qui l'accepte.



- L'hôpital E fait une proposition au diplômé 3, qui la refuse. Il fait une proposition au diplômé 2, qui l'accepte.
- L'hôpital B fait une proposition au diplômé 3, qui la refuse. Il fait une proposition au diplômé 4, qui l'accepte.



- **Solution trouvée!**

Questions:

Est-ce que l’algorithme finit toujours ? Après combien de temps ?

Est-ce que le résultat est toujours une affectation stable ?

- Une fois qu’un diplômé est affecté, il reste affecté (seul l’hôpital d’affectation peut changer).

- Lorsque l’hôpital d’affectation change pour un diplômé, c’est pour un hôpital situé plus haut dans sa liste de préférences : cela arrive au plus ***n – 1*** fois.

- A chaque étape, soit un diplômé devient affecté (alors qu’il ne l’était pas avant), soit un diplômé change d’hôpital d’affectation : au plus ***n²*** étapes.

Complexité : ***O(n²).***

- Par contradiction, supposons que l’affectation finale n’est pas stable. Alors il existe Hx et Dy qui satisfont la propriété d’instabilité.

- Soient :
  - H*x* est couplé avec D*x*,
  - H*y* est couplé avec D*y*,
  - H*x* préfère D*y* à D*x*,
  - D*y* préfère H*x* à H*y*.

- D*y* est avant D*x* dans la liste de préférences de H*x*, mais H*x* n’est pas couplé avec D*y*. Deux cas :

- Lorsque H*x* considère D*y*, celui-ci est déjà affecté à H*z*, qu’il préfère à H*x*: Dy préfère donc également H*z* à H*y*, mais dans l’algorithme un diplômé ne peut être réaffecté qu’à des hôpitaux plus haut dans sa liste, une contradiction.
- Lorsque H*x* considère D*y*, celui-ci est libre, mais H*x* est remplacé plus tard par un hôpital préférable à H*x*. De nouveau, D*y* ne peut pas finir par être affecté à H*y*.

- Une affectation stable existe toujours, et elle peut être trouvée en temps polynomial.

- Discussion : l’algorithme est-il meilleur pour les hôpitaux ?

- Beaucoup d'applications des couplages dans :
  - L'analyse des structures chimiques
  - La reconnaissance de caractères
  - L'analyse des formes
  - La recherche d'images (dans une base)
  - Planification, emploi du temps …

- Beaucoup de variantes résolues, mais d'autres apparaissent en fonction des applications spécifiques.