Couplage

Johanne Cohen[1]

[1]LRI-CNRS, Université Paris-Sud, Université Paris-Saclay, France.

Référence

Plan

1. Couplage
2. Couplage
   - Définition
   - Un chemin augmentant/alternant
3. L'algorithme calculant un couplage maximum
   - Graphe contracté et chemin augmentant
   - Algorithme calculant un chemin augmentant de M
   - L'algorithme d'Edmonds (1965)

Plan

1 Couplage

2 Couplage
- Définition
- Un chemin augmentant/alternant

3 L'algorithme calculant un couplage maximum
- Graphe contracté et chemin augmentant
- Algorithme calculant un chemin augmentant de M
- L'algorithme d'Edmonds (1965)

Définition d'un couplage

Soit un graphe simple non orienté $G = (V, E)$
Un couplage M est un ensemble d'arêtes deux à deux non adjacentes :

$$\forall (a, a') \in M^2, \qquad a \neq a' \Rightarrow a \cap a' = \varnothing \ .$$

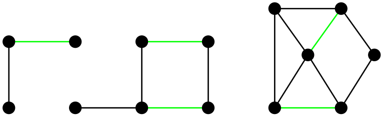
Terminologies :

- Les sommets incidents à une arête de G sont les sommets couplés, apparié ou couverts par M.
- Le couplage M sature U, si U est un ensemble des sommets couplés par M.

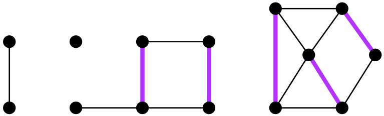
Les différents couplages

- Un couplage maximal est un couplage M ayant la propriété que si une arête e est ajoutée, alors $M \cup \{e\}$ n'en est pas.
- Un couplage maximum est un couplage contenant le plus grand nombre possible d'arêtes.

Un graphe peut posséder plusieurs couplages maximum.



Couplage maximal



Couplage maximum

Un chemin augmentant/alternant

Soit un graphe simple non orienté $G = (V, E)$.
Soit M un couplage de G.

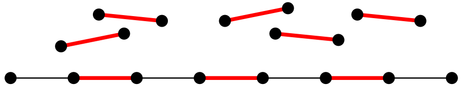
- Un chemin alternant est un chemin dans lequel les arêtes appartiennent alternativement à M et à $E \setminus M$.

- Un chemin augmentant est un chemin alternant qui commence et se finit par un sommet non-apparié.

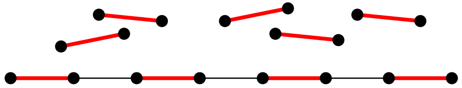
Les arêtes en rouge sont dans M.

Propriété

Si M un couplage (arêtes en rouge) ayant un chemin augmentant P,



alors le couplage $M' = M \Delta P$ posséde plus d'arêtes



Remarque : S'il y a un chemin augmentant pour M,

alors M n'est pas maximum.

Chemin augmentant et couplage maximun

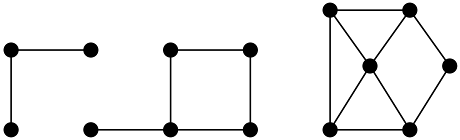
Theorem [Berge 1957]

Soit M un couplage dans un graphe G. M est maximum si et seulement si il n'existe pas de chemin augmentant

Preuve :

- Soit M et M' des couplages non-maximun et maximun resp.
- Soit H le sous-graphe de G induit par les arêtes de $M \Delta M'$.

Les composantes connexes de H sont des cycles ou des chemins où les arêtes de M et M' alternent. Donc



le graphe G

Chemin augmentant et couplage maximun

Theorem [Berge 1957]

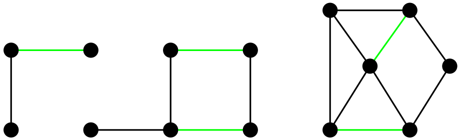
Soit M un couplage dans un graphe G. M est maximum si et seulement si il n'existe pas de chemin augmentant

Preuve :

- Soit M et M' des couplages non-maximun et maximun resp.
- Soit H le sous-graphe de G induit par les arêtes de $M \Delta M'$.

Les composantes connexes de H sont des cycles ou des chemins où les arêtes de M et M' alternent. Donc

$M =$ arêtes vertes



le graphe G

Chemin augmentant et couplage maximun

Theorem [Berge 1957]

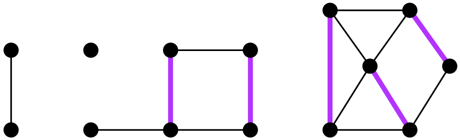
Soit M un couplage dans un graphe G. M est maximum si et seulement si il n'existe pas de chemin augmentant

Preuve :

- Soit M et M' des couplages non-maximun et maximun resp.
- Soit H le sous-graphe de G induit par les arêtes de $M \Delta M'$.

Les composantes connexes de H sont des cycles ou des chemins où les arêtes de M et M' alternent. Donc

$M' =$ arêtes violettes



le graphe G

Chemin augmentant et couplage maximun

Theorem [Berge 1957]

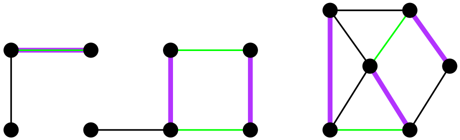
Soit M un couplage dans un graphe G. M est maximum si et seulement si il n'existe pas de chemin augmentant

Preuve :

- Soit M et M' des couplages non-maximun et maximun resp.
- Soit H le sous-graphe de G induit par les arêtes de $M \Delta M'$.

Les composantes connexes de H sont des cycles ou des chemins où les arêtes de M et M' alternent. Donc

$M' =$ arêtes violettes
$M =$ arêtes vertes



le graphe G

Chemin augmentant et couplage maximun

Theorem [Berge 1957]

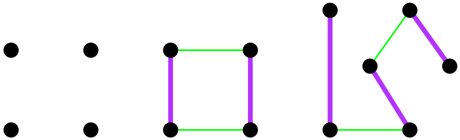
Soit M un couplage dans un graphe G. M est maximum si et seulement si il n'existe pas de chemin augmentant

Preuve :

- Soit M et M' des couplages non-maximun et maximun resp.
- Soit H le sous-graphe de G induit par les arêtes de $M \Delta M'$.

Les composantes connexes de H sont des cycles ou des chemins où les arêtes de M et M' alternent. Donc

M' = arêtes violettes
M = arêtes vertes



le graphe H

Chemin augmentant et couplage maximun

Theorem [Berge 1957]

Soit M un couplage dans un graphe G. M est maximum si et seulement si il n'existe pas de chemin augmentant

Preuve :

- Soit M et M' des couplages non-maximun et maximun resp.
- Soit H le sous-graphe de G induit par les arêtes de $M \Delta M'$.

Les composantes connexes de H sont des cycles ou des chemins où les arêtes de M et M' alternent. Donc

- pour chaque cycle : $\#$ d'arêtes de $M' = \#$ d'arêtes de M.

  car les cycles sont de longueur paire.
- pour un des chemins P :$\#$ d'arêtes de $M' > \#$ d'arêtes de M.

  M' a plus d'arêtes que M

P commence et termine par deux arêtes de M'.

Donc P est un chemin augmentant pour M.

Plan

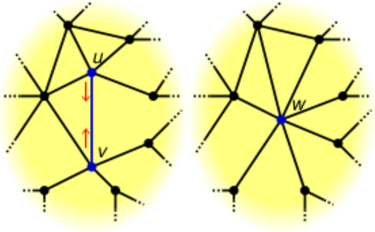
1. Couplage

2. Couplage
   - Définition
   - Un chemin augmentant/alternant

3. L'algorithme calculant un couplage maximum
   - Graphe contracté et chemin augmentant
   - Algorithme calculant un chemin augmentant de M
   - L'algorithme d'Edmonds (1965)

Graphe contracté

Soit un graphe simple non orienté $G = (V, E)$.
La contraction G de l'arête e est le graphe obtenu en fusionnant ses deux extrémités.



La figure a été prise sur le site de wikipedia.

La contraction d'un chemin est le graphe obtenu en effectuant les contractions de toutes ses arêtes.

Blossom

Soit M un couplage dans un graphe G.
Un cycle B est un blossom si

- il possède $2k+1$ arêtes
- k de ses arêtes sont dans M

Couplage et contraction

Theorem [Edmonds 1965]

Soit M un couplage dans un graphe G.

Supposons que B est un blossom de G.

Soit G' le graphe obtenu par la contraction de B dans G.

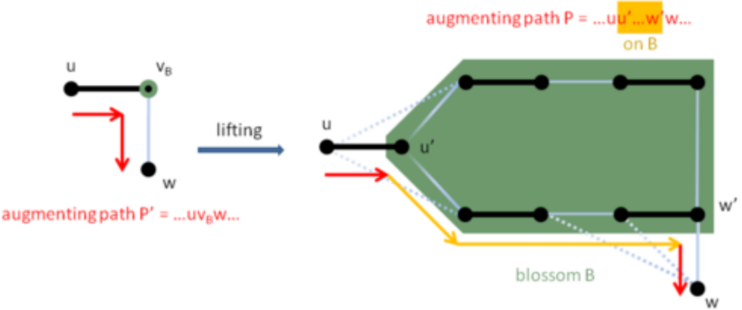
Soit M' le couplage de M projeté dans G'.

G' a un chemin augmentant pour M'
si et seulement si
G a un chemin augmentant pour M

Preuve

Soit v_M le sommet de G' représentant la contraction de B.
Supposons que le chemin P' augmentant pour M' contient v_B.

1. Si v_B est couvert par M', alors

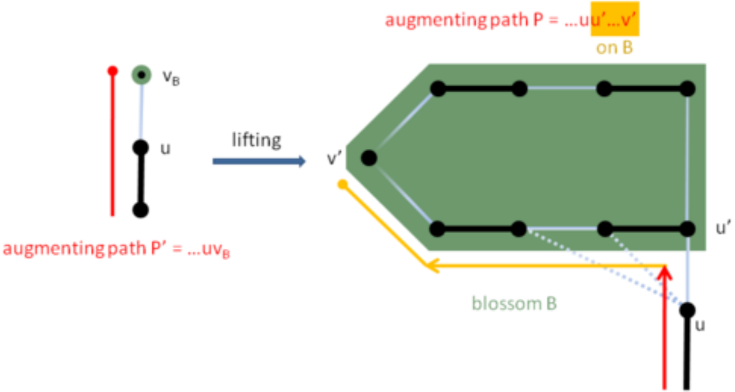


Les figures ont été prises sur le site de wikipedia.

Preuve

Soit v_M le sommet de G' représentant la contraction de B.
Supposons que le chemin P' augmentant pour M' contient v_B.

1. Si v_B n'est pas couvert par M', alors



Les figures ont été prises sur le site de wikipedia.

Preuve

Soit v_M le sommet de G' représentant la contraction de B.
Supposons que le chemin P' augmentant pour M' contient v_B.

Si P' est un chemin augmentant pour M' and G, alors
il existe un chemin augmentant pour M dans G.

Et réciproquement.

Les figures ont été prises sur le site de wikipedia.

arbre alterné

Soit M un couplage du graphe G
Un arbre T dans G est dit alterné si

1. il existe un unique sommet r non-couvert par M (par convention : c'est la racine)
2. pour tout sommet v couvert par l'arbre T,

   le chemin de la racine vers v est un chemin alternant

Par convention

- un sommet v est interne si sa distance de la racine est impaire
- un sommet v est externe si sa distance de la racine est paire

Remarque sur les arbres alternés

Soit M un couplage du graphe G. Soit T un arbre alterné.

Remarque 1 : Soit v un sommet de T.
Si v est adjacent à un sommet w dans G de même type (externe, ou interne), alors le chemin de v vers w dans T plus l'arête (v, w) forment un blossom.

Remarque 2 : Soit v un sommet de T.
Si v est adjacent à un sommet w n'appartenant pas à T dans G et non-couvert par M, alors le chemin de r vers v dans T plus l'arête (v, w) forment une chaine augmentante.

Algorithme **Construction_chemin**(un graphe G,un couplage M)

1. $F \leftarrow \emptyset$; $\forall v \in V$ non-marqué ; seules les arêtes $\in M$ marquées.

   Initialisation des variables

2. retourner $\emptyset$;

Algorithme **Construction_chemin**(un graphe G,un couplage M)

1. $F \leftarrow \emptyset$; $\forall v \in V$ non-marqué ; seules les arêtes $\in M$ marquées.
2. pour tout sommet v non-couvert par M faire

   créer un arbre T_v composé uniquement de v ; $F \leftarrow F \cup T_v$

   Initialisation des arbres alternés (un par sommet non-couvert)

3. retourner $\emptyset$;

Algorithme **Construction_chemin**(un graphe G,un couplage M)

1. $F \leftarrow \emptyset$; $\forall v \in V$ non-marqué ; seules les arêtes $\in M$ marquées.
2. pour tout sommet v non-couvert par M faire

   créer un arbre T_v composé uniquement de v ; $F \leftarrow F \cup T_v$
3. Tant que $\exists v \in F$ non-marqué externe faire
   1. Tant qu'il existe $e = (v, w)$ non-marqué de F faire

      Traitement pour augmenter la forêt
      pour trouver un chemin augmentant
4. retourner $\emptyset$;

   il n'existe aucun chemin augmentant

Algorithme **Construction_chemin**(un graphe G,un couplage M)

1. $F \leftarrow \emptyset$; $\forall v \in V$ non-marqué ; seules les arêtes $\in M$ marquées.
2. pour tout sommet v non-couvert par M faire

   créer un arbre T_v composé uniquement de v ; $F \leftarrow F \cup T_v$
3. Tant que $\exists v \in F$ non-marqué externe faire
   1. Tant qu'il existe $e = (v, w)$ non-marqué de F faire
      1. si $w \notin F$ alors soit x le sommet apparié à w

         ajouter (v, w) et (w, x) à l'arbre contenant v

         Rencontre pour la première fois w, insertion de w dans F
4. retourner $\emptyset$;

Algorithme **Construction_chemin**(un graphe G,un couplage M)

1. $F \leftarrow \emptyset$; $\forall v \in V$ non-marqué ; seules les arêtes $\in M$ marquées.
2. pour tout sommet v non-couvert par M faire

   créer un arbre T_v composé uniquement de v ; $F \leftarrow F \cup T_v$
3. Tant que $\exists v \in F$ non-marqué externe faire
   1. Tant qu'il existe $e = (v, w)$ non-marqué de F faire
      1. si $w \notin F$ alors soit x le sommet apparié à w

         ajouter (v, w) et (w, x) à l'arbre contenant v
      2. sinon

         si w est interne alors
         (i) si $r(v) \neq r(w)$ alors retourner $\{r(v) \to v, w, \to r(w)\}$

Détection d'un chemin entre deux arbres (remarque 2)

4. retourner $\emptyset$;

Algorithme **Construction_chemin**(un graphe G,un couplage M)

1. $F \leftarrow \emptyset$; $\forall v \in V$ non-marqué ; seules les arêtes $\in M$ marquées.
2. pour tout sommet v non-couvert par M faire
   créer un arbre T_v composé uniquement de v ; $F \leftarrow F \cup T_v$
3. Tant que $\exists v \in F$ non-marqué externe faire
   1. Tant qu'il existe $e = (v, w)$ non-marqué de F faire
      1. si $w \notin F$ alors soit x le sommet apparié à w
         ajouter (v, w) et (w, x) à l'arbre contenant v
      2. sinon
         si w est interne alors
         (i) si $r(v) \neq r(w)$ alors retourner $\{r(v) \to v, w, \to r(w)\}$
         (ii) sinon
         $B \leftarrow$ blossom formé par e et par $v \to w$ dans T
         $G', M' \leftarrow$ contacter B dans G et M
         $P' \leftarrow$ CONSTRUCTION_CHEMIN(G', M')
         retourner P' projeté dans G

Détection d'un blossom (traitement comme la remarque 1)

4. retourner $\emptyset$;

Algorithme **Construction_chemin**(un graphe G,un couplage M)

1. $F \leftarrow \emptyset$; $\forall v \in V$ non-marqué ; seules les arêtes $\in M$ marquées.
2. pour tout sommet v non-couvert par M faire

   créer un arbre T_v composé uniquement de v ; $F \leftarrow F \cup T_v$
3. Tant que $\exists v \in F$ non-marqué externe faire
   1. Tant qu'il existe $e = (v, w)$ non-marqué de F faire
      1. si $w \notin F$ alors soit x le sommet apparié à w

         ajouter (v, w) et (w, x) à l'arbre contenant v
      2. sinon

         si w est interne alors

         (i) si $r(v) \neq r(w)$ alors retourner $\{r(v) \rightarrow v, w, \rightarrow r(w)\}$

         (ii) sinon

         $B \leftarrow$ blossom formé par e et par $v \rightarrow w$ dans T

         $G', M' \leftarrow$ contacter B dans G et M

         $P' \leftarrow \textsc{Construction\_chemin}(G', M')$

         retourner P' projeté dans G
      3. marquer e ; Ne plus traiter e
4. retourner $\emptyset$;

Algorithme **Construction_chemin**(un graphe G,un couplage M)

1. $F \leftarrow \emptyset$; $\forall v \in V$ non-marqué ; seules les arêtes $\in M$ marquées.
2. pour tout sommet v non-couvert par M faire
   créer un arbre T_v composé uniquement de v ; $F \leftarrow F \cup T_v$
3. Tant que $\exists v \in F$ non-marqué externe faire
   1. Tant qu'il existe $e = (v, w)$ non-marqué de F faire
      1. si $w \notin F$ alors soit x le sommet apparié à w
         ajouter (v, w) et (w, x) à l'arbre contenant v
      2. sinon
         si w est interne alors
         (i) si $r(v) \neq r(w)$ alors retourner $\{r(v) \to v, w, \to r(w)\}$
         (ii) sinon
         $B \leftarrow$ blossom formé par e et par $v \to w$ dans T
         $G', M' \leftarrow$ contacter B dans G et M
         $P' \leftarrow$ CONSTRUCTION_CHEMIN(G', M')
         retourner P' projeté dans G
      3. marquer e ; Ne plus traiter e
   2. marquer v ; Ne plus traiter v
4. retourner $\emptyset$;

Algorithme **Construction_chemin**(un graphe G,un couplage M)

1. $F \leftarrow \emptyset$; $\forall v \in V$ non-marqué ; seules les arêtes $\in M$ marquées.
2. pour tout sommet v non-couvert par M faire

   créer un arbre T_v composé uniquement de v ; $F \leftarrow F \cup T_v$
3. Tant que $\exists v \in F$ non-marqué externe faire
   1. Tant qu'il existe $e = (v, w)$ non-marqué de F faire
      1. si $w \notin F$ alors soit x le sommet apparié à w

         ajouter (v, w) et (w, x) à l'arbre contenant v
      2. sinon

         si w est interne alors

         (i) si $r(v) \neq r(w)$ alors retourner $\{r(v) \rightarrow v, w, \rightarrow r(w)\}$

         (ii) sinon

         $B \leftarrow$ blossom formé par e et par $v \rightarrow w$ dans T

         $G', M' \leftarrow$ contacter B dans G et M

         $P' \leftarrow$ Construction_chemin(G', M')

         retourner P' projeté dans G
      3. marquer e ; Ne plus traiter e
   2. marquer v ; Ne plus traiter v
4. retourner $\emptyset$;

L'algorithme d'Edmonds (1965)

Algorithme $\textsc{Couplage\_MAX}(G, M)$

Entrée : un graphe $G = (V, E)$, et un couplage de M.

Sortie : un couplage M

1. Trouver un chemin P augmentant de M
2. Si il n'en existe pas alors retourner M
3. sinon retourner $\textsc{Couplage\_MAX}(G, M$ modifié par $P)$

Complexité : $\mathcal{O}(|V|^4)$ car au pire

- on construit $\mathcal{O}(|V|)$ chemins augmentants
- construire un chemin augmentant nécéssite $\mathcal{O}(|V| \times |E|)$ opérations

Il existe des améliorations d'algorithmes de construction de chemins en $\mathcal{O}(|E|)$ opérations.