Optimisation

Algorithmes d'approximation

Louis-Claude Canon
louis-claude.canon@univ-fcomte.fr

Master 2 Informatique – Semestre 9

Plan

Plan

Contexte

De nombreux problèmes sont NP-difficiles mais leur résolution est critique. Différentes approches pour les résoudre :

- De façon exacte si la taille des entrées, n, est petite.
- On identifie une propriété qui permet de concevoir un algorithme polynomial.
- On cherche une heuristique qui fournit un résultat proche de l'optimal. Il s'agit d'un *algorithme d'approximation*.

Algorithme d'approximation

- Un algorithme a un *facteur d'approximation* $\rho(n)$ si pour toute entrée de taille n :
  - C est le coût de la solution produite par l'algorithme ;
  - C^* est le coût de la solution optimale ;
  - $\rho(n) \geq \max\left(\frac{C}{C^*}, \frac{C^*}{C}\right)$.
- On dit alors qu'il s'agit d'un *algorithme d'approximation* $\rho(n)$.
- Exemples :
  - l'algorithme de Dijkstra est un algorithme d'approximation 1 pour le plus court chemin ;
  - LPT est un algorithme d'approximation $4/3$ pour $P||C_{\max}$.

Schéma d'approximation

- Un *schéma d'approximation* est un algorithme d'approximation dont le facteur d'approximation est $1 + \varepsilon$ (avec $\varepsilon \geq 0$) et la complexité dépend de ε.
- Si on diminue ε, le résultat s'améliore mais la complexité augmente.
- Un *schéma d'approximation polynomial*[1] est un algorithme dont la complexité est polynomiale en n, la taille de l'entrée.
- Un *schéma d'approximation entièrement polynomial*[2] est un algorithme dont la complexité est polynomial en n et en $1/\varepsilon$.

1. PTAS (Polynomial Time Approximation Scheme).
2. FPTAS (Fully Polynomial Time Approximation Scheme).

Question

Lequel de ces algorithmes est un schéma d'approximation polynomial ?

1. LPT pour $P||C_{\max}$
2. l'algorithme de Dijkstra pour le plus court chemin
3. Un schéma d'approximation entièrement polynomial
4. Un schéma d'approximation

Question

Lequel de ces algorithmes est un schéma d'approximation polynomial ?

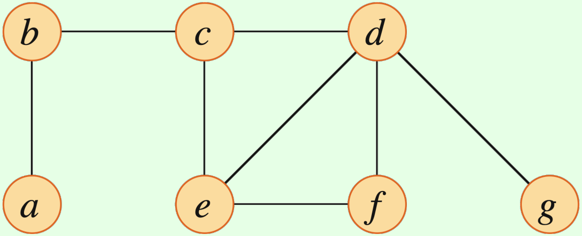
1. LPT pour $P||C_{\max}$
2. l'algorithme de Dijkstra pour le plus court chemin ✓(optimal)
3. Un schéma d'approximation entièrement polynomial ✓
4. Un schéma d'approximation

Plan

Couverture de sommets

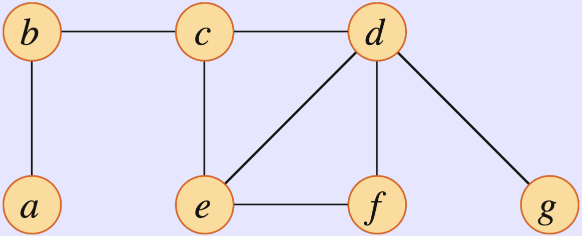
Notion algorithmique

- Un graphe non-orienté $G = (V, E)$.
- Une *couverture de sommets* est un sous-ensemble $C \subseteq V$ tel que pour toute arête $(u, v) \in E$, soit $u \in C$, soit $v \in C$.
- Exemple : $V' = \{a, c, d, f, g\}$ est une couverture de sommets du graphe suivant.



Question

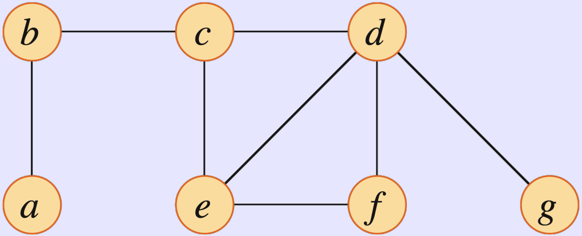
Lequel de ces ensembles n'est pas une couverture ?



1. $\{a, b, d, f, g\}$
2. $\{b, c, e, f, g\}$
3. $\{a, c, d, e\}$
4. $\{a, b, e, d, g\}$

Question

Lequel de ces ensembles n'est pas une couverture ?



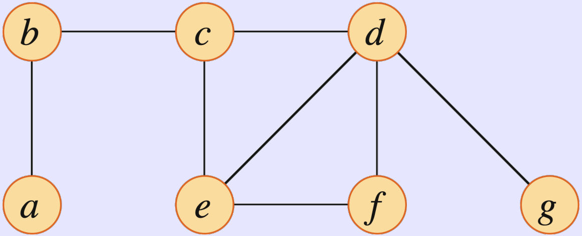
1. $\{a, b, d, f, g\}$ ✓(c, e)
2. $\{b, c, e, f, g\}$
3. $\{a, c, d, e\}$
4. $\{a, b, e, d, g\}$

Problème de la couverture de sommets

- Le *problème de la couverture de sommets* consiste à trouver la couverture de sommets de taille minimum.
- Ce problème est NP-difficile.
- On peut utiliser COUVERTURE-SOMMET-APPROCHÉE, un algorithme d'approximation 2.

Question

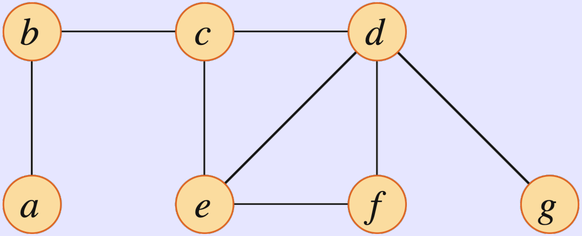
Quelle est la taille de la couverture minimum ?



- 2
- 3
- 4
- 5

Question

Quelle est la taille de la couverture minimum ?



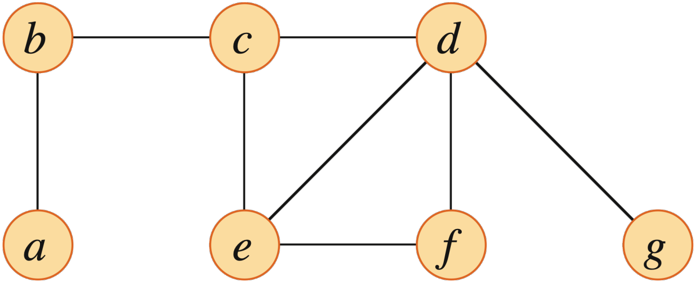
- 2
- 3 ✓$\{b, d, e\}$
- 4
- 5

Couverture-Sommet-Approchée

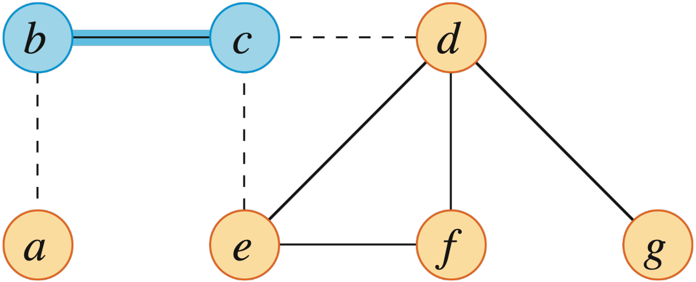
Couverture-Sommet-Approchée(G)

$C \leftarrow \emptyset$
$E' \leftarrow G.E$
tant que $E' \neq \emptyset$ **faire**
  soit (u, v) une arête quelconque de E'
  $C \leftarrow C \cup \{u, v\}$
  supprimer de E' toutes les arêtes incidentes à u ou à v
retourner C

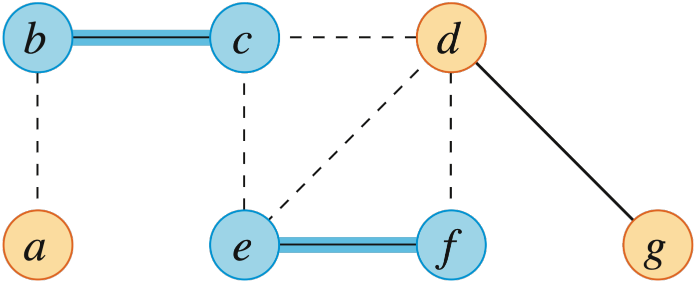
Exemple



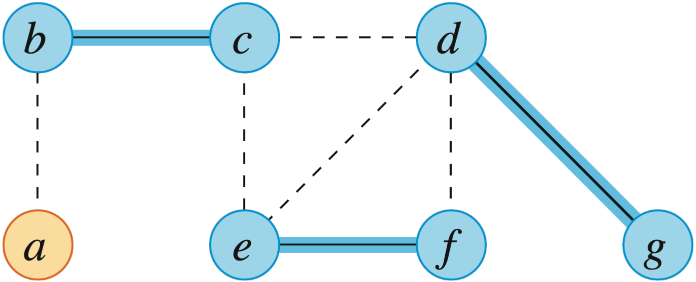
Exemple



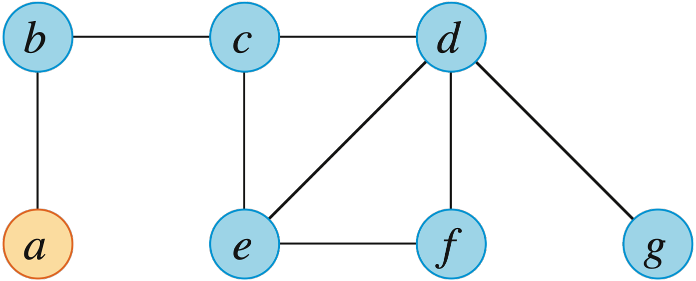
Exemple



Exemple

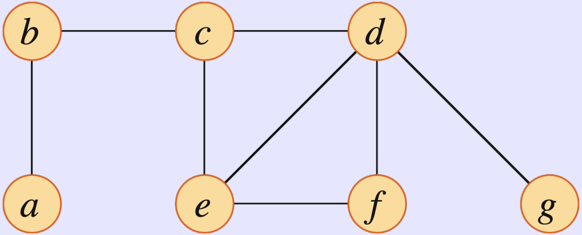


Exemple



Question

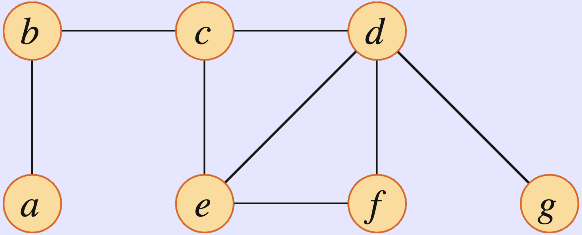
Laquelle de ces couvertures n'aurait pas pu être produite par COUVERTURE-SOMMET-APPROCHÉE ?



1. $\{a, b, c, d, e, f\}$
2. $\{a, b, c, d, e\}$
3. $\{a, b, d, e\}$
4. $\{a, b, d, e, f, g\}$

Question

Laquelle de ces couvertures n'aurait pas pu être produite par COUVERTURE-SOMMET-APPROCHÉE ?



1. $\{a, b, c, d, e, f\}$
2. $\{a, b, c, d, e\}$ ✓(nombre de sommets pair)
3. $\{a, b, d, e\}$
4. $\{a, b, d, e, f, g\}$

Résultat formel

Théorème

Couverture-Sommet-Approchée *est un algorithme d'approximation 2 à temps polynomial pour le problème de la couverture de sommets.*

Preuve de la complexité en temps

Démonstration.

- L'initialisation prend un temps $O(E)$.
- La boucle s'exécute au plus $|V|/2$ fois car on supprime 2 sommets à chaque étape.
- On peut naïvement réaliser chaque itération en temps $O(V + E)$.
- Couverture-Sommet-Approchée est bien un algorithme à temps polynomial.

$\square$

Preuve de la validité

Démonstration.

Pour chaque arête $(u, v) \in E$:

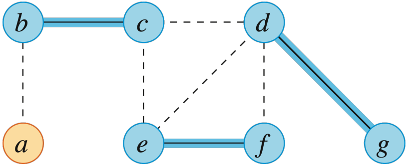
- soit l'arête a été sélectionnée et les 2 sommets sont couverts ;
- soit l'arête n'a pas été sélectionnée car elle était incidente à un autre sommet sélectionné.

$\square$

Facteur d'approximation

Démonstration.

- Soit A les arêtes sélectionnées par l'algorithme.
- La couverture optimale, C^* contient au moins l'un des sommets extrémités de chacune de ces arêtes.
- Comme ces sommets sont tous distincts, $|A| \leq |C^*|$. C'est une **borne inférieure** de la valeur de l'objectif optimale.
- Par construction, la taille de la couverture produite, C, est le double du nombre d'arêtes : $|C| = 2|A|$.
- Donc $|C| \leq 2|C^*|$.



$\square$

Optimalité du facteur d'approximation

S'agit-il d'un *facteur d'approximation optimal* (ou fin) ? Peut-on trouver un pire cas qui l'atteigne et si oui lequel ?

Optimalité du facteur d'approximation

S'agit-il d'un *facteur d'approximation optimal* (ou fin) ? Peut-on trouver un pire cas qui l'atteigne et si oui lequel ?
Deux sommets relié par une arête.

Plan

Tournée

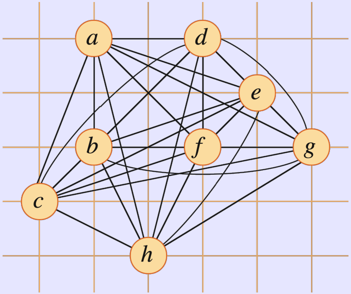
Notion algorithmique

- Un graphe complet non-orienté $G = (V, E)$ avec une pondération sur les arêtes $c : E \to \mathbb{N}$.
- Une *tournée* $A \subseteq E$ est un sous-ensemble d'arêtes constituant un cycle hamiltonien [3] de G de coût $c(A) = \sum_{(u,v)\in A} c(u, v)$.

3. Les arêtes permettent de revenir au point de départ et passent par chaque sommet une seule fois.

Question

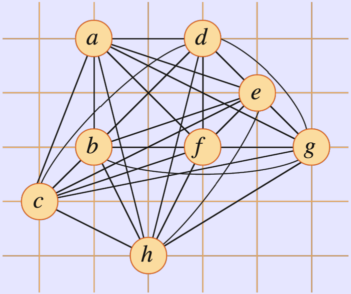
Quel est le coût de la tournée a, b, c, d, e, f, g, h avec la distance de Manhattan ?



- 16
- 21
- 26
- 31

Question

Quel est le coût de la tournée a, b, c, d, e, f, g, h avec la distance de Manhattan ?



- 16
- 21
- 26 ✓$(2+2+6+2+2+2+5+5)$
- 31

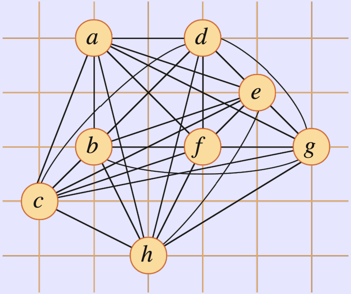
Problème du voyageur de commerce

Notion algorithmique

- On cherche une tournée de coût minimum.
- C'est un problème NP-difficile.
- On suppose l'inégalité triangulaire pour tout triplet de sommets $u, v, w \in V$: $c(u, w) \leq c(u, v) + c(v, w)$ (on ne peut pas réduire le coût avec un arrêt intermédiaire).
- C'est le cas de la distance euclidienne ou de Manhattan.
- C'est encore un problème NP-difficile.

Question

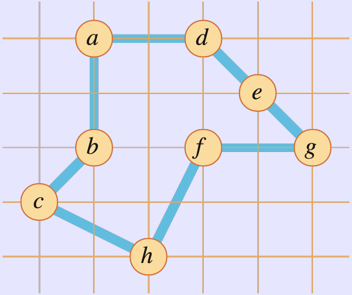
Quelle est la tournée de coût minimum avec la distance de Manhattan ?



1. a, b, c, d, e, f, g, h
2. a, d, e, g, h, c, b, f
3. a, d, e, g, f, b, c, h
4. a, b, c, h, f, g, e, d

Question

Quelle est la tournée de coût minimum avec la distance de Manhattan ?



1. a, b, c, d, e, f, g, h
2. a, d, e, g, h, c, b, f
3. a, d, e, g, f, b, c, h
4. a, b, c, h, f, g, e, d ✓(18)

Tournée-VC-Approchée

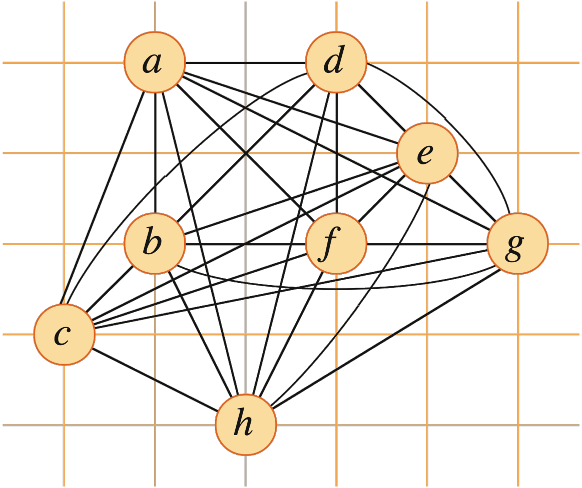
Heuristique du point le plus proche :

Tournée-VC-Approchée(G)

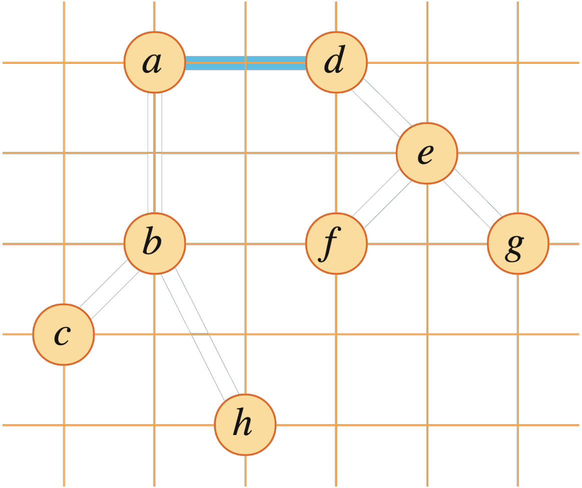
---

sélectionner un sommet quelconque $u \in V$
$H \leftarrow [u]$
tant que $|H| \neq |V|$ **faire**
  choisir le sommet $u \notin H$ dont la distance à un sommet $v \in H$, $c(u, v)$, est minimum
  insérer u après v dans le cycle H
retourner le cycle H

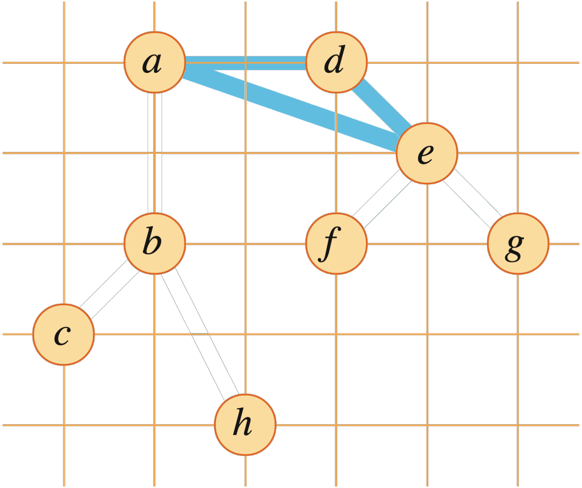
Exemple



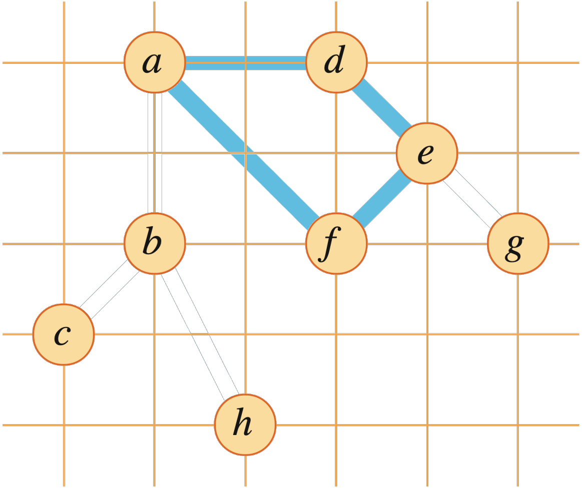
Exemple



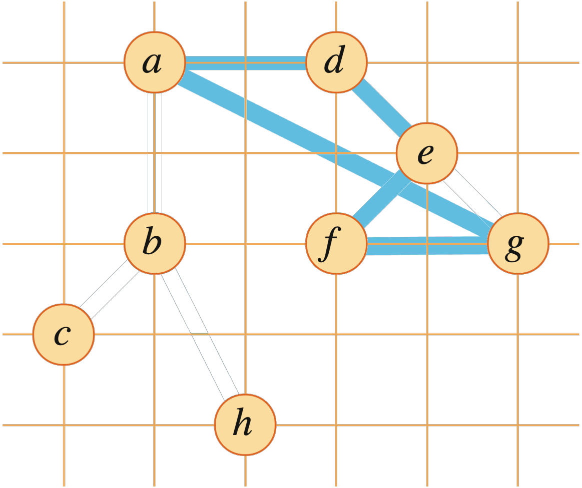
Exemple



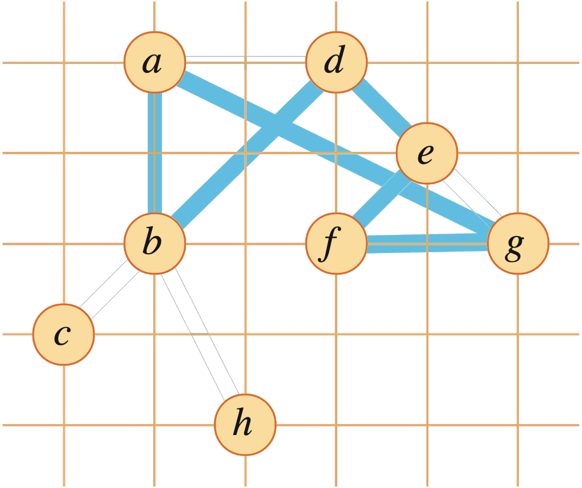
Exemple



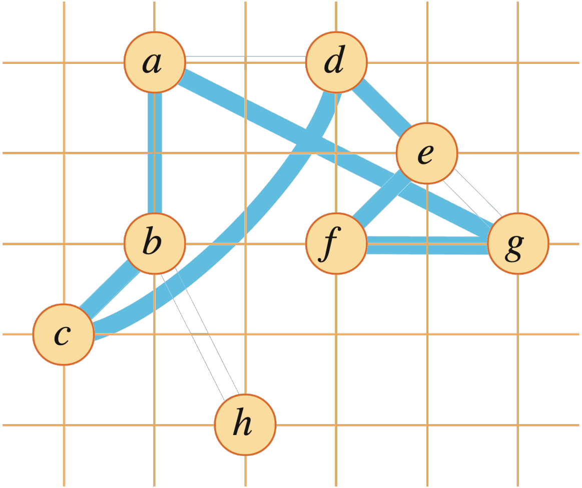
Exemple



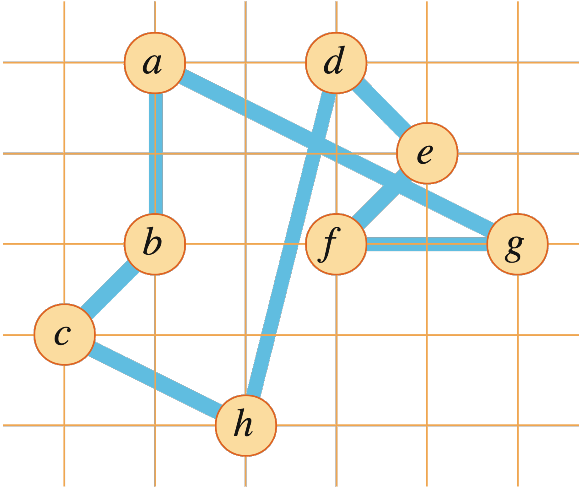
Exemple



Exemple

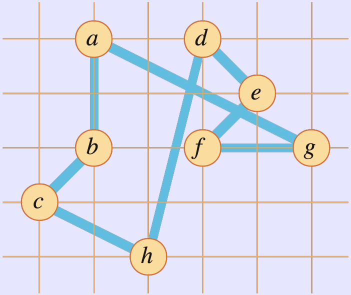


Exemple



Question

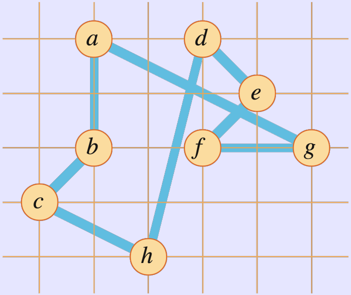
Quelle est le facteur d'approximation de Tournée-VC-Approchée dans cet exemple ?



1. 5/4
2. 4/3
3. 3/2
4. 2

Question

Quelle est le facteur d'approximation de TOURNÉE-VC-APPROCHÉE dans cet exemple ?



1. 5/4
2. 4/3 ✓(24/18)
3. 3/2
4. 2

Résultat formel

Théorème

TOURNÉE-VC-APPROCHÉE *est un algorithme d'approximation 2 à temps polynomial pour le problème du voyageur de commerce avec inégalité triangulaire.*

Preuve de la complexité en temps

Démonstration.

- L'initialisation se fait en temps constant.
- Il y a $|V| - 1$ itérations.
- Chaque itération s'implémente naïvement en $O(E)$.
- Tournée-VC-Approchée est bien un algorithme à temps polynomial.

$\square$

Preuve de la validité

Démonstration.

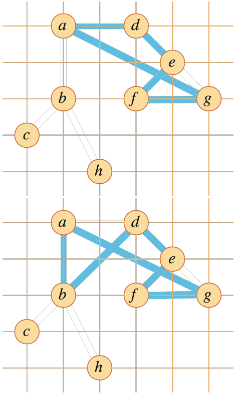
- Initialement, il y a un seul sommet dans H.
- À chaque itération, il existe au moins un sommet dans H et un sommet qui n'est pas dans H.
- Comme le graphe est complet, un couple (u, v) est forcément considéré et un nouveau sommet est ajouté à H.
- L'algorithme termine lorsque H passe par chaque sommet, il s'agit donc bien d'une tournée.

$\square$

Facteur d'approximation

Démonstration.

- Soit H_i la tournée générée à l'étape i et H_i^* une tournée optimale des sommets parcourus par H_i.
- Pour $0 \leq n \leq 3$, $c(H_i) = c(H_i^*)$ et il reste à faire la récurrence en supposant que $c(H_i) \leq 2c(H_i^*)$ pour $i \geq 1$.
- Pour la tournée optimale, $c(H_{i+1}^*) \geq c(H_i^*) + c(u, v)$ car il faut au moins atteindre le nouveau sommet et v en est le plus proche. C'est une **borne inférieure** de la valeur de l'objectif optimale.
- Pour la tournée générée, $c(H_{i+1}) \leq c(H_i) + 2c(u, v)$ grâce à l'inégalité triangulaire et ainsi, $c(H_{i+1}) \leq 2c(H_{i+1}^*)$.



□

Résultats existants

Curiosité

L'algorithme de Christofides (1976) est un algorithme d'approximation $3/2$.
En 2020, ce facteur d'approximation est amélioré à $3/2 - 10^{-36}$.

Sans l'hypothèse de l'inégalité triangulaire, on n'a à priori pas de garantie pour ce problème.

Théorème

Si $P \neq NP$, alors pour toute constante $r \geq 1$, il n'existe aucun algorithme d'approximation r à temps polynomial pour le problème général du voyageur de commerce.

Résultats existants

Curiosité

L'algorithme de Christofides (1976) est un algorithme d'approximation $3/2$.
En 2020, ce facteur d'approximation est amélioré à $3/2 - 10^{-36}$.

Sans l'hypothèse de l'inégalité triangulaire, on n'a à priori pas de garantie pour ce problème.

Théorème

Si $P \neq NP$, alors pour toute constante $r \geq 1$, il n'existe aucun algorithme d'approximation r à temps polynomial pour le problème général du voyageur de commerce.

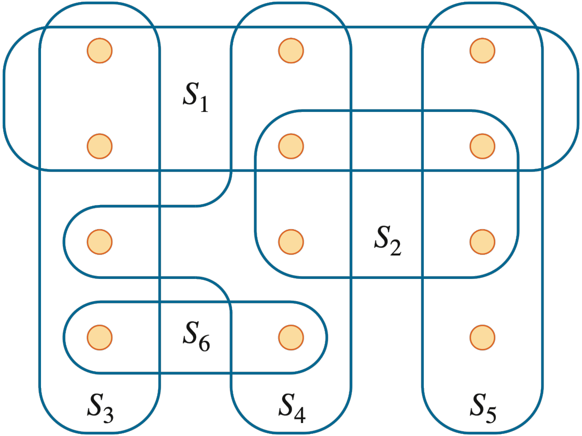
Plan

Couverture d'ensemble

Notion algorithmique

- Un ensemble X et une famille $\mathcal{F}$ de sous-ensembles de X : $\forall S \in \mathcal{F}, S \subseteq X$.
- Chaque élément de X appartient au moins à un sous-ensemble de $\mathcal{F}$: $X = \bigcup_{S \in \mathcal{F}} S$.
- Un sous-ensemble $S \in \mathcal{F}$ *couvre* ses éléments.

Exemple d'instance avec $|X| = 12$ et $\mathcal{F} = \{S_1, S_2, S_3, S_4, S_5, S_6\}$



Exemple de contexte pratique

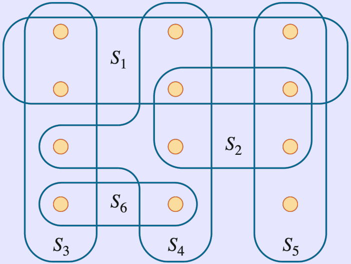
- X représente des compétences nécessaires pour résoudre un problème.
- Un sous-ensemble S représente les compétences liées à une personne.
- $\mathcal{F}$ représente plusieurs personnes.
- On souhaite créer un comité, composé du moins de personnes possible, tel qu'il y ait au moins une personne ayant chacune des compétences.

Problème de la couverture d'ensemble

- Il faut déterminer quelle sous-famille $\mathcal{C} \subseteq \mathcal{F}$ permet de couvrir tous les éléments de X : $X = \bigcup_{S \in \mathcal{C}} S$.
- On cherche à minimiser le nombre des sous-ensembles sélectionnés $|\mathcal{C}|$.
- Ce problème généralise le problème de la couverture des sommets et est donc aussi NP-difficile.

Question

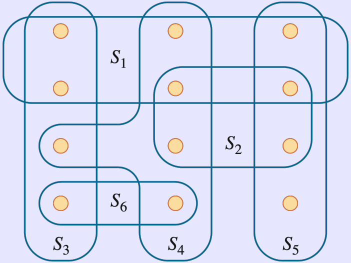
Quelle sous-famille $\mathcal{C} \subseteq \mathcal{F}$ permet de couvrir tous les éléments ?



1. $\mathcal{C} = \{S_2, S_3, S_4, S_5\}$
2. $\mathcal{C} = \{S_1, S_2, S_5, S_6\}$
3. $\mathcal{C} = \{S_1, S_4, S_5, S_6\}$
4. $\mathcal{C} = \{S_1, S_2, S_3, S_4\}$

Question

Quelle sous-famille $\mathcal{C} \subseteq \mathcal{F}$ permet de couvrir tous les éléments ?



1. $\mathcal{C} = \{S_2, S_3, S_4, S_5\}$ ✓
2. $\mathcal{C} = \{S_1, S_2, S_5, S_6\}$
3. $\mathcal{C} = \{S_1, S_4, S_5, S_6\}$ ✓
4. $\mathcal{C} = \{S_1, S_2, S_3, S_4\}$

Couverture-Ensemble-Glouton

Quel principe glouton pourrait être mis en place ?

Couverture-Ensemble-Glouton

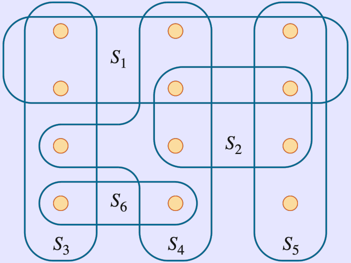
Quel principe glouton pourrait être mis en place ?

Couverture-Ensemble-Glouton$(X, \mathcal{F})$

$U \leftarrow X$
$\mathcal{C} \leftarrow \emptyset$
tant que $U \neq \emptyset$ **faire**
  choisir un sous-ensemble $S \in \mathcal{F}$ qui maximise $|S \cap U|$
  $U \leftarrow U \setminus S$
  $\mathcal{C} \leftarrow \mathcal{C} \cup S$
retourner $\mathcal{C}$

Question

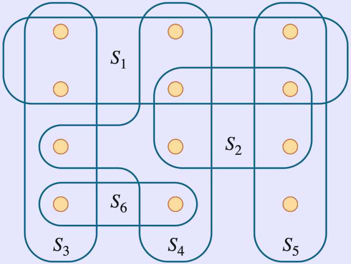
Quelle sous-famille peut être retournée par COUVERTURE-ENSEMBLE-GLOUTON ?



1. $\mathcal{C} = \{S_1, S_4, S_5, S_3\}$
2. $\mathcal{C} = \{S_1, S_4, S_5, S_6\}$
3. $\mathcal{C} = \{S_2, S_3, S_4, S_5\}$
4. $\mathcal{C} = \{S_3, S_4, S_5\}$

Question

Quelle sous-famille peut être retournée par COUVERTURE-ENSEMBLE-GLOUTON ?



1. $\mathcal{C} = \{S_1, S_4, S_5, S_3\}$ ✓
2. $\mathcal{C} = \{S_1, S_4, S_5, S_6\}$ ✓
3. $\mathcal{C} = \{S_2, S_3, S_4, S_5\}$
4. $\mathcal{C} = \{S_3, S_4, S_5\}$

Résultat formel

Théorème

COUVERTURE-ENSEMBLE-GLOUTON *est un algorithme d'approximation* $\log(n)$ *à temps polynomial.*

Preuve de la complexité en temps

Démonstration.

- L'initialisation se fait en temps constant.
- Il y a au plus $|X|$ itérations.
- Chaque itération s'implémente naïvement en $O(X \times \mathcal{F})$.
- Couverture-Ensemble-Glouton est bien un algorithme à temps polynomial.

$\square$

Preuve de la validité

Démonstration.

- Initialement, aucun sommet n'est couvert et U représente tous les sommets non-couverts.
- À chaque itération, on sélectionne un sous-ensemble S qui permet de couvrir le plus d'éléments non-couverts $S \cap U$ ($|S \cap U| \geq 1$ car chaque élément est dans au moins un sous-ensemble).
- L'ensemble des éléments non-couverts U diminue donc à chaque itération.
- L'algorithme termine lorsque tous les éléments sont couverts.

$\square$

Plan

3-SAT

Notion algorithmique

- Il s'agit de la restriction du problème SAT aux formes normales conjonctives avec au plus 3 littéraux par clause.
- On a donc un ensemble de littéraux (variables booléennes) organisés en conjonction (ET logique) de clauses (OU logique).
- Exemple : $(v_1 \vee v_2 \vee v_4) \wedge (\neg v_1 \vee v_3 \vee \neg v_5) \wedge (\neg v_2 \vee \neg v_3 \vee v_4) \dots$
- Le problème de décision consiste à déterminer si toutes les clauses peuvent être satisfaites.
- Le problème d'optimisation consiste à déterminer le nombre maximum de clauses qui peuvent être satisfaites.

Approche aléatoire

- On assigne une valeur aléatoire à chaque littéral (50 % de chance de mettre vrai, 50 % de mettre faux).
- En moyenne, le nombre de clauses satisfaites dépasse $7/8$ du nombre optimal.
- Il s'agit donc d'un algorithme d'approximation $8/7$ en moyenne à temps polynomial.

Plan

Résumé

- Un algorithme d'approximation produit des solutions dont le coût est borné par rapport à l'optimal.
- C'est une approche pour rester en temps polynomial pour des problèmes NP-difficiles.
- Le problème de la couverture de sommets admet un algorithme d'approximation 2 qui consiste à trouver un appariement maximal.
- Le problème du voyageur de commerce admet un algorithme de même facteur d'approximation qui rajoute le point le plus proche à chaque étape.
- Le problème de la couverture d'ensemble admet un algorithme avec un facteur d'approximation $\log(n)$ qui s'appuie sur un critère glouton.
- Le problème 3-SAT admet un algorithme probabiliste avec un facteur d'approximation $8/7$ en moyenne qui s'appuie sur une affectation aléatoire des littéraux.

Prochaines échéances

- Début du projet-tournoi : aujourd'hui !
- Premier rendu intermédiaire du projet-tournoi le 6/10.
- Second rendu intermédiaire du projet-tournoi le 20/10.
- Épreuve sur table le 23/10.
- Rendu final pour le projet-tournoi le 4/11.
- Épreuve de TP le 5/11.
- Restitution du projet-tournoi le 6/11.