L'approximation polynomiale et ses enjeux : une promenade dans le jardin enchanté de l'approximation polynomiale

Vangelis Th. Paschos

paschos@lamsade.dauphine.fr

27 juin 2003

Faire face aux problèmes NP-difficiles

- méthodes exactes : privilégient la qualité de la solution calculée par rapport à la rapidité du calcul ;
- heuristiques : la rapidité du calcul d'une solution prime sur son optimalité
  - méthodes polyhèdrales ;
  - métaheuristiques ;
  - recherche locale ;
  - *algorithmes polynomiaux approchés avec garanties de performance*

Qu'est-ce-que l'appoximation polynomiale ?

C'est l'art d'obtenir des solutions *réalisables*, de valeur *proche* (dans un sens prédéfini) de la valeur optimale et tout cela en temps polynomial

Pourquoi l'approximation polynomiale ?

Parce que :

- il existe des problèmes réels où :
  - sont exigées des solutions *réalisables* obtenues *rapidement, i.e., en temps polynomial* ;
  - l'optimum est mal défini ou il n'a pas de sens ;
  - il est nécessaire d'avoir une estimation *a priori* de la qualité de l'algorithme ;
- elle est le prolongement naturel de la théorie de la complexité dans l'optimisation combinatoire et contribue à l'enrichissement mutuel des deux domaines ;

- elle est passionante !!!

Notions de base et notations

Un problème Π de **NPO** à valeurs entières est un quadruplet $(\mathcal{I}, \mathrm{Sol}, m, \mathrm{opt})$ qui vérifie :

- $\mathcal{I}$ est l'ensemble des instances de Π ; tout $I \in \mathcal{I}$ est reconnaissable en temps polynomial en $|I|$ (sa taille) ;
- $\forall I \in \mathcal{I}$, $\mathrm{Sol}(I)$ est l'ensemble des solutions réalisables de I ; tout $S \in \mathrm{Sol}(I)$ est représentable et décidable en temps polynomial en $|I|$; enfin, $\forall I \in \mathcal{I}$, on sait déterminer une solution $\mathrm{triv}(I) \in \mathrm{Sol}(I)$ en temps polynomial en $|I|$; pour $I \in \mathcal{I}$ et $S \in \mathrm{Sol}(I)$, $m(I, S)$ est la valeur de la solution S ;
- $m : \mathcal{I} \times \mathrm{Sol} \to \mathbb{N}$ est calculable en temps polynomial en $|I|$;
- $\mathrm{opt} \in \{\min, \max\}$

Pour $\Pi = (\mathcal{I}, \mathrm{Sol}, m, \mathrm{opt})$:
$\mathrm{opt}(I)$: la valeur d'une solution optimale
$\omega(I)$: la valeur d'une pire solution, *i.e.*, la solution optimale du problème $\Pi' = (\mathcal{I}, \mathrm{Sol}, m, \mathrm{opt}')$ avec

$$\mathrm{opt}' = \begin{cases} \max & \text{si } \mathrm{opt} = \min \\ \min & \text{si } \mathrm{opt} = \max \end{cases}$$

Pour un algorithme approché $\mathtt{A}$:
$m_{\mathtt{A}}(I, S)$, la valeur de la solution S calculée par $\mathtt{A}$ sur I

Rapport d'approximation classique pour `A` :

$$\rho_{\mathtt{A}}(I) = \frac{m_{\mathtt{A}}(I,S)}{\mathrm{opt}(I)}$$

Rapport d'approximation différentielle pour `A` :

$$\delta_{\mathtt{A}}(I) = \frac{|\omega(I) - m_{\mathtt{A}}(I,S)|}{|\omega(I) - \mathrm{opt}(I)|}$$

Plus les rapports sont proches de 1, meilleure est considérée la performance d'un algorithme approché

Classification absolue des problèmes et des algorithmes approchés

- Algorithmes à rapports dépendant de l'instance :
  - rapport exponentiel en $|I|$ — **Exp-APX**, **Exp-DAPX** ; exemple en **Exp-APX** : MIN TSP ;
  - rapport polynomial en $|I|$ — **Poly-APX**, **Poly-DAPX**
    - exemples en **Poly-APX** : MAX STABLE, MAX CLIQUE, COLORATION, ... ;
    - exemples en **Poly-DAPX** : MAX STABLE, MAX CLIQUE, MIN VERTEX COVER, MIN SET COVER, ... ;
  - rapport logarithmique en $|I|$ — **Log-APX**, **Log-DAPX** ; exemples en **Log-APX** : MIN SET COVER, MIN DOMINATING SET ;

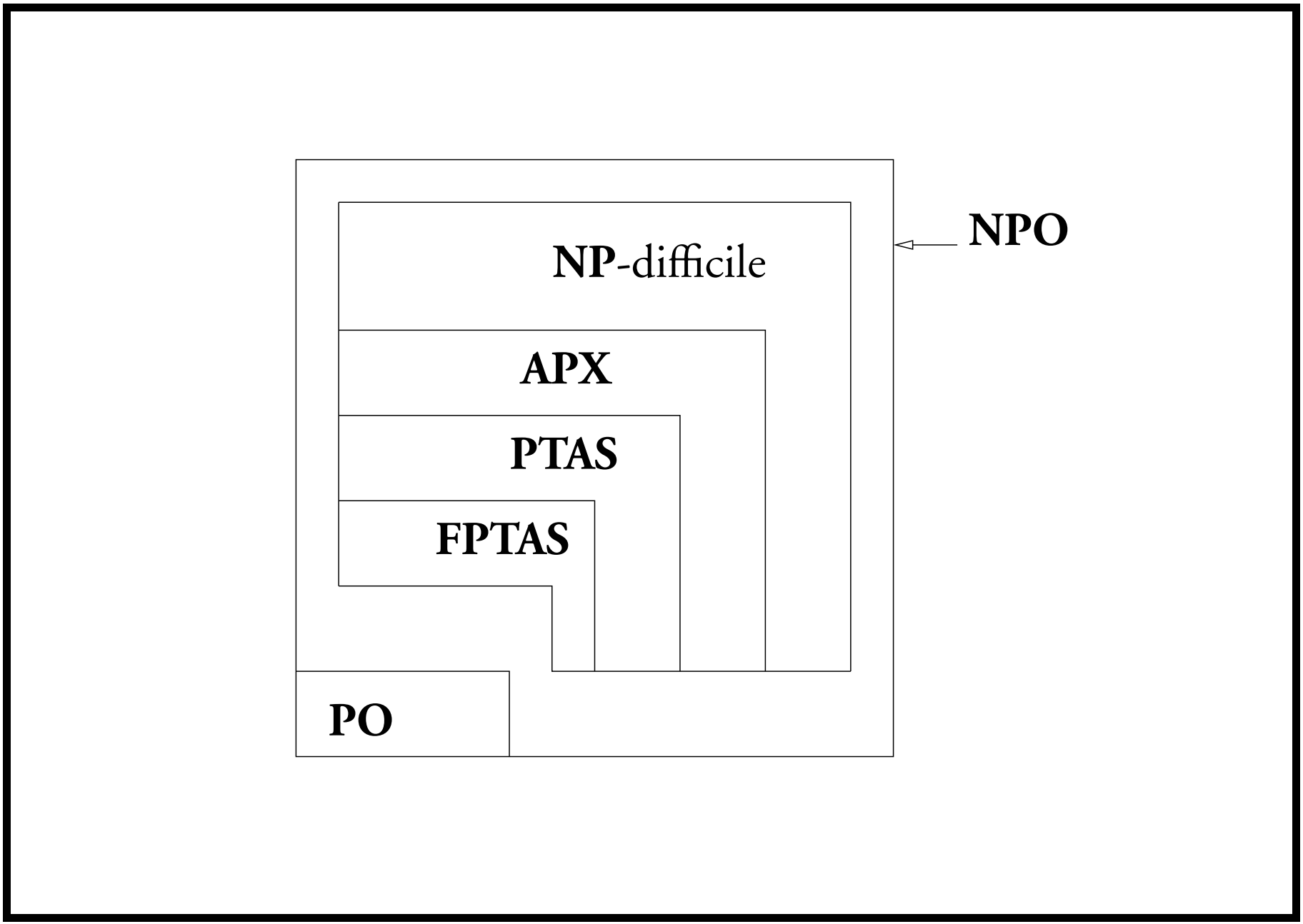
• Algorithmes à rapports constants (indépendant de $|I|$) — **APX**, **DAPX**

- exemples en **APX** : MIN VERTEX COVER, MIN METRIC TSP, BIN PACKING, MAX TSP, ... ;
- exemples en **DAPX** : MIN et MAX TSP, COLORATION, ... ;

• Schémas polynomiaux d'approximation : suites d'algorithmes paramétrés par ϵ à rapports $1+\epsilon$ (si $\mathrm{opt} = \min$), ou $1-\epsilon$ (si $\mathrm{opt} = \max$, ou en différentiel pour tout type d'optimisation), $\forall \epsilon > 0$ et à complexité polynomiale en $|I|$ mais, éventuellement, exponentielle en $1/\epsilon$ — **PTAS**, **DPTAS**

- exemples en **PTAS** : MIN VERTEX COVER ou MAX STABLE dans les graphes planaires, MIN TSP EUCLIDIEN, ... ;
- exemples en **DPTAS** : MIN VERTEX COVER ou MAX STABLE dans les graphes planaires, BIN PACKING, COLORATION PONDÉRÉE dans des graphes bipartis, ... ;

• Schémas complètement polynomiaux d'approximation : suites d'algorithmes paramétrés par ϵ à rapports $1+\epsilon$ (si $\mathrm{opt} = \min$), ou $1-\epsilon$ (si $\mathrm{opt} = \max$ ou en différentiel pour tout type d'optimisation), $\forall \epsilon > 0$ et à complexité polynomiale à la fois en $|I|$ et en $1/\epsilon$ — **FPTAS**, **DFPTAS**
exemple en **FPTAS** et **DFPTAS** : SAC-À-DOS


NPO
NP-difficile
APX
PTAS
FPTAS
PO

Remarques :

- Un problème est classifié par rapport au meilleur rapport d'approximation actuellement connu pour lui
- Il y a en réalité un *continuum* des classes d'approximation

Quelques exemples d'algorithmes approchés

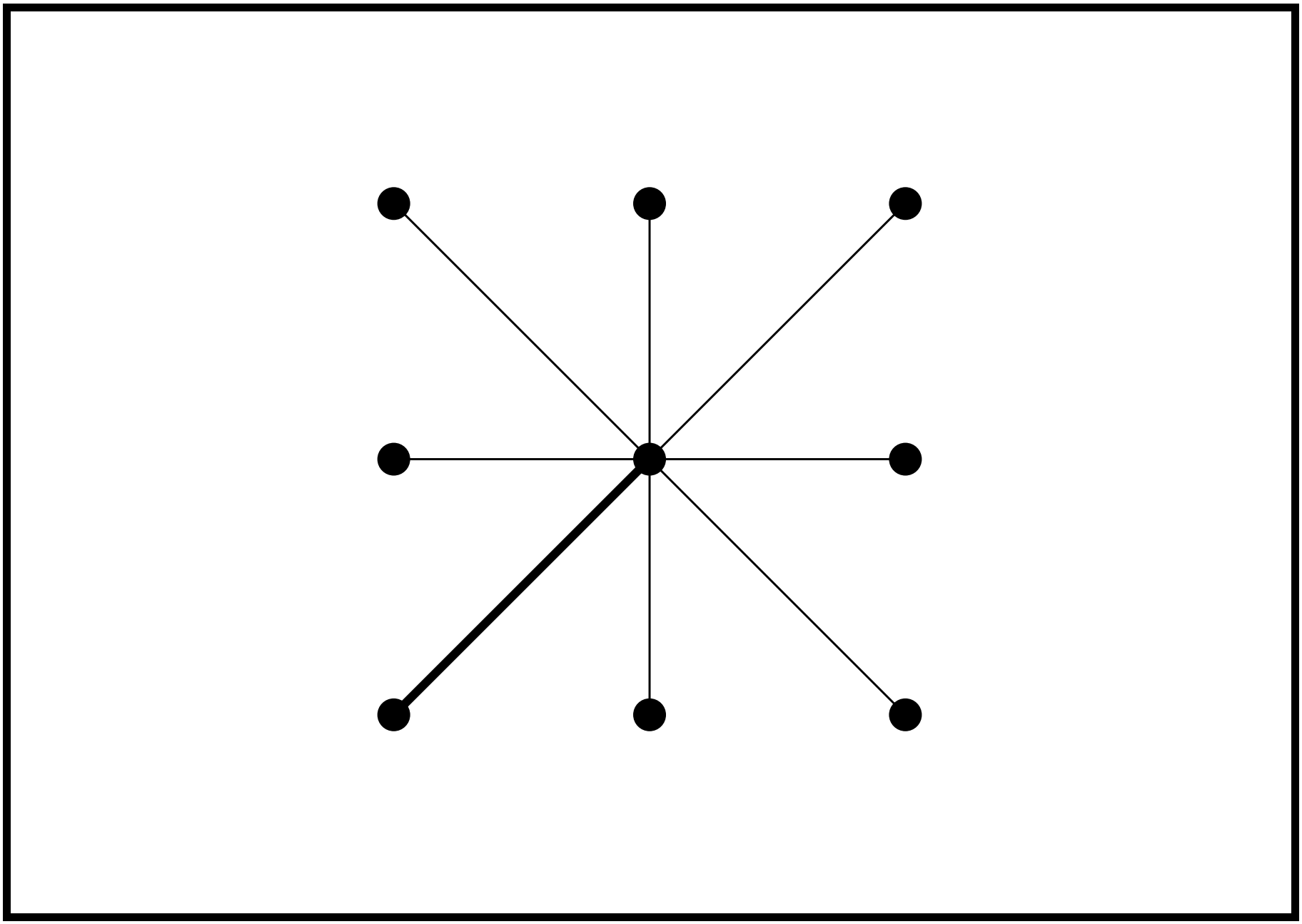
MIN VERTEX COVER

1. choisir une arête ;
2. mettre ses extrémités dans la solution en construction ;
3. mettre à jour G ;
4. répéter les étapes 1 à 3 jusqu'à ce que G soit vide en arêtes ;
5. retourner la solution construite

Le rapport de l'algorithme est majoré par 2 et ce rapport est atteint

Pouvons-nous l'améliorer ?

MIN VERTEX COVER $\in \mathbf{APX} \setminus \mathbf{PTAS}$



COLORATION

1. répéter : trouver un 3-stable S dans G, colorier les sommets de S avec la même nouvelle couleur et mettre à jour G tant que des 3-stables existent ;
2. determiner une famille de 2-stables disjoints de taille maximum dans G (ayant survécu) et colorier les sommets de chacun d'eux avec une nouvelle couleur ; colorier chacun des sommets qui restent avec une couleur distincte

Le rapport différentiel de l'algorithme est borné inférieurement par 2/3 et ce rapport est atteint

COLORATION $\in$ **DAPX** $\setminus$ **DPTAS**

L'étape 2 colorie un graphe G avec $\alpha(G) = 2$ de façon optimale

Réccurence sur n

Vrai pour $n \leqslant k$

Considérons un graphe d'ordre $k + 1$:

- s'il n'existe pas de 3-stable, l'algorithme a trouvé la coloration optimale et le résultat est vrai ;
- supposons qu'un 3-stable S a été trouvé :

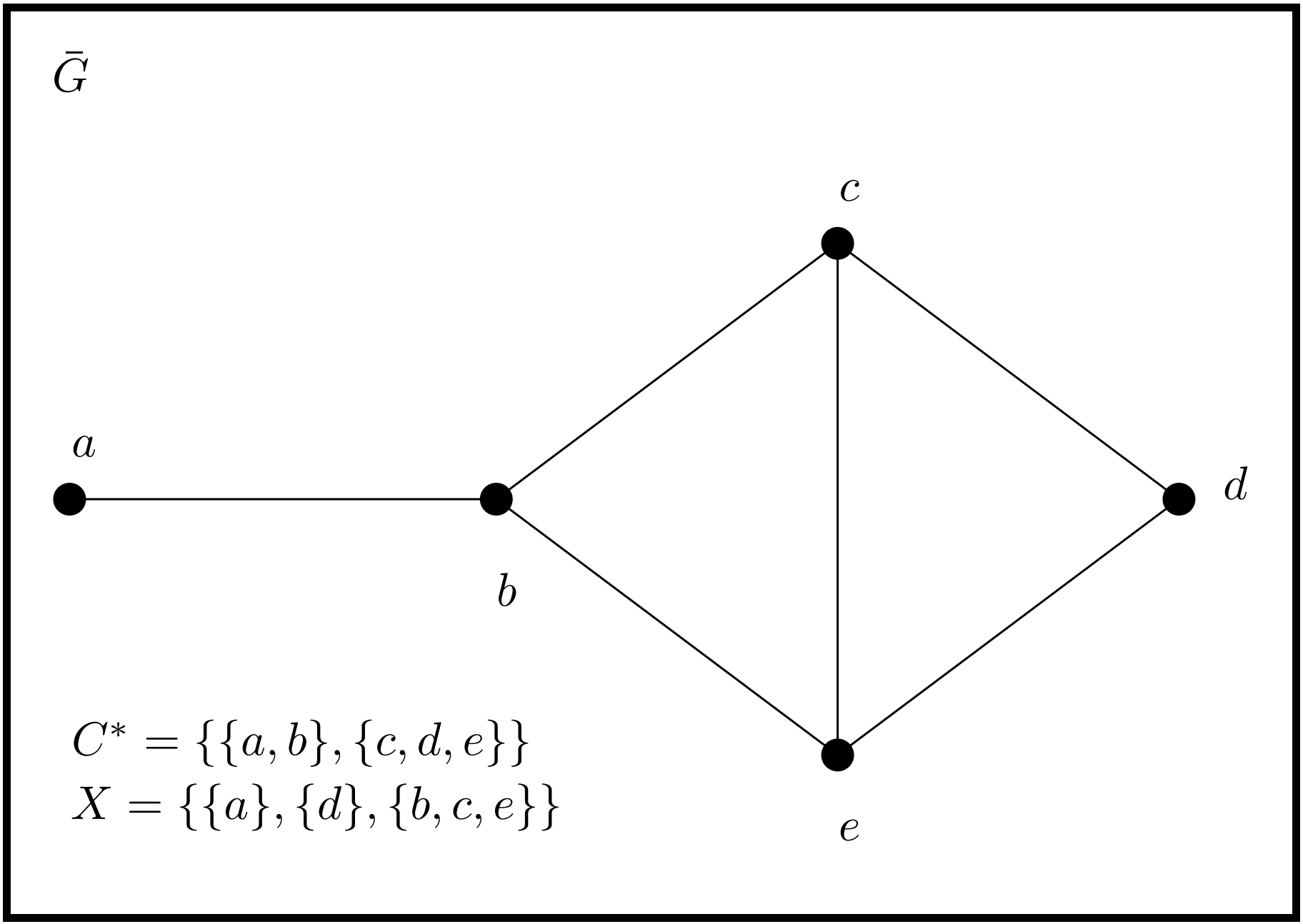
$$\chi(G[V \setminus S]) \leqslant \chi(G) \tag{1}$$

X : la solution retournée par l'algorithme
$X \setminus S$ réalisable pour $G[V \setminus S]$ (et calculable par l'algorithme) :

$$n - 3 - |X \setminus S| \geqslant \frac{2}{3}(n - 3 - \chi(G[V \setminus S])) \qquad (2)$$

De (1) et (2) :

$$\begin{aligned} n - |X| &= n - |X \setminus S| - 1 \\ &\geqslant \frac{2}{3}(n - 3 - \chi(G[V \setminus S])) + 2 \\ &\geqslant \frac{2}{3}(n - \chi(G)) \end{aligned}$$


$\bar{G}$
c
a
d
b
e
$C^* = \{\{a, b\}, \{c, d, e\}\}$
$X = \{\{a\}, \{d\}, \{b, c, e\}\}$

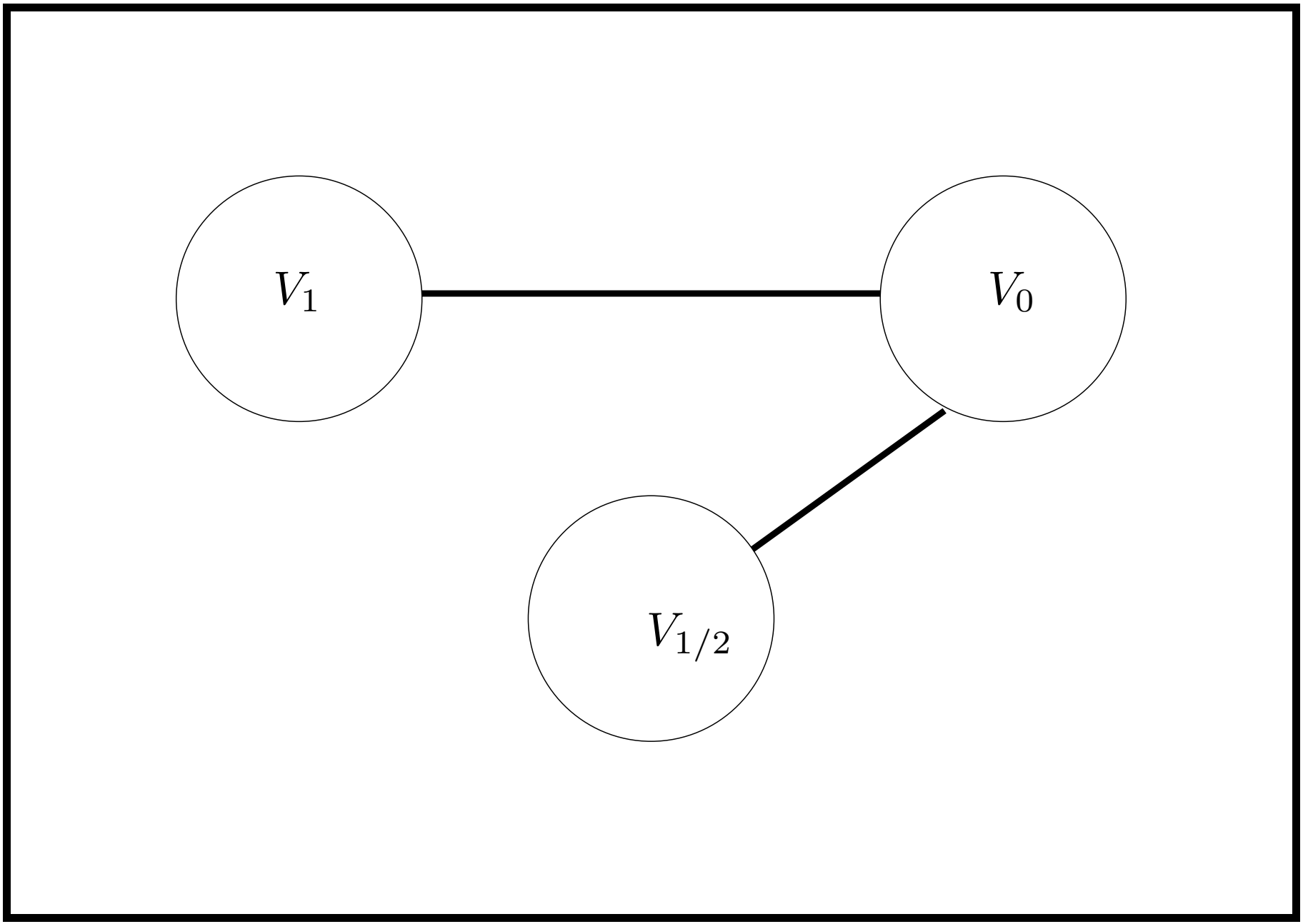
MAX STABLE

$$\text{MAX STABLE} = \begin{cases} \max & \vec{1} \cdot \vec{x} \\ & A \cdot \vec{x} \leqslant \vec{1} \\ & \vec{x} \in \{0,1\}^n \end{cases}$$

$$\text{MAX STABLE-R} = \begin{cases} \max & \vec{1} \cdot \vec{x} \\ & A \cdot \vec{x} \leqslant \vec{1} \\ & \vec{x} \in (\mathbb{Q}^n)^+ \end{cases}$$

Une solution optimale de MAX STABLE-R affecte aux variables des valeurs dans l'ensemble $\{0, 1, 1/2\}$. Si V_0, V_1 et $V_{1/2}$ sont les sous-ensembles de V correspondant aux variables à 0, 1 et 1/2, respectivement, alors, il existe une solution optimale S^* de MAX STABLE telle que :

1. $V_1 \subseteq S^*$;
2. $V_0 \subseteq V \setminus S^*$.


V_1
V_0
$V_{1/2}$

- résoudre MAX STABLE PONDÉRÉ-R pour déterminer les ensembles V_0, V_1 et $V_{1/2}$; $G := G[V_{1/2}]$;
- colorier G avec $\Delta(G)$ couleurs ; soit $\hat{S}$ la plus lourde couleur ;
- retourner $S := V_1 \cup \hat{S}$

Le rapport d'approximation de l'algorithme est minoré par $2/\Delta(G)$

MAX STABLE $\in$ **Poly-APX**

Meilleurs rapports connus :

$O(\log^2 n/n)$

$k/\Delta(G), \forall k > 0$ (asymptotique)

$O(\log n/\Delta(G) \log\log n)$

$S = V_1 \cup \hat{S}$ réalisable
$S^* = V_1 \cup S^*_{1/2}$

$$\left|\hat{S}\right| \geqslant \frac{\left|V_{1/2}\right|}{\Delta(G)}$$

$$m(S,G) = |S| = |V_1| + \left|\hat{S}\right| \geqslant |V_1| + \frac{\left|V_{1/2}\right|}{\Delta(G)}$$

$$\mathrm{opt}(G) = |S^*| = |V_1| + \left|S^*_{1/2}\right| \leqslant |V_1| + \frac{\left|V_{1/2}\right|}{2}$$

MIN SET COVER

- répéter : soit S_0 un ensemble de $\mathcal{S}$ de cardinalité maximum ; mettre S_0 dans la solution $\mathcal{S}'$; mettre à jour $\mathcal{S}$ et C ; jusqu'à ce que $C := \emptyset$;

Le rapport de l'algorithme est majoré par $1 + \ln \Delta$
($\Delta = \max_{S \in \mathcal{S}}\{|S|\}$**) et il peut être asymptotiquement atteint**

MIN SET COVER $\in$ **Log-APX**

$I_i(\mathcal{S}_i, C_i)$ l'instance au premier instant où la cardinalité maximum résiduelle des ensembles de $\mathcal{S}$ est inférieure ou égale à i
$m(I_i, \mathcal{S}')$ le nombre d'ensembles de cardinalité résiduelle i mis dans $\mathcal{S}'$

$$C_\Delta = C \tag{3}$$

$$m(I, \mathcal{S}') = \sum_{i=1}^{\Delta} m(I_i, \mathcal{S}') \tag{4}$$

Pour $i = 1, \ldots, \Delta$:

$$|C_i| = \sum_{k=1}^{i} k \times m(I_k, \mathcal{S}')$$

Multiplier la ligne de C_Δ par $1/\Delta$ et les autres par $1/i(i+1)$ et utiliser (3) et (4) :

$$\sum_{i=1}^{\Delta-1} \frac{|C_i|}{i(i+1)} + \frac{|C|}{\Delta} = \sum_{i=1}^{\Delta} m\left(I_i, \mathcal{S}'\right) = m\left(I, \mathcal{S}'\right) \tag{5}$$

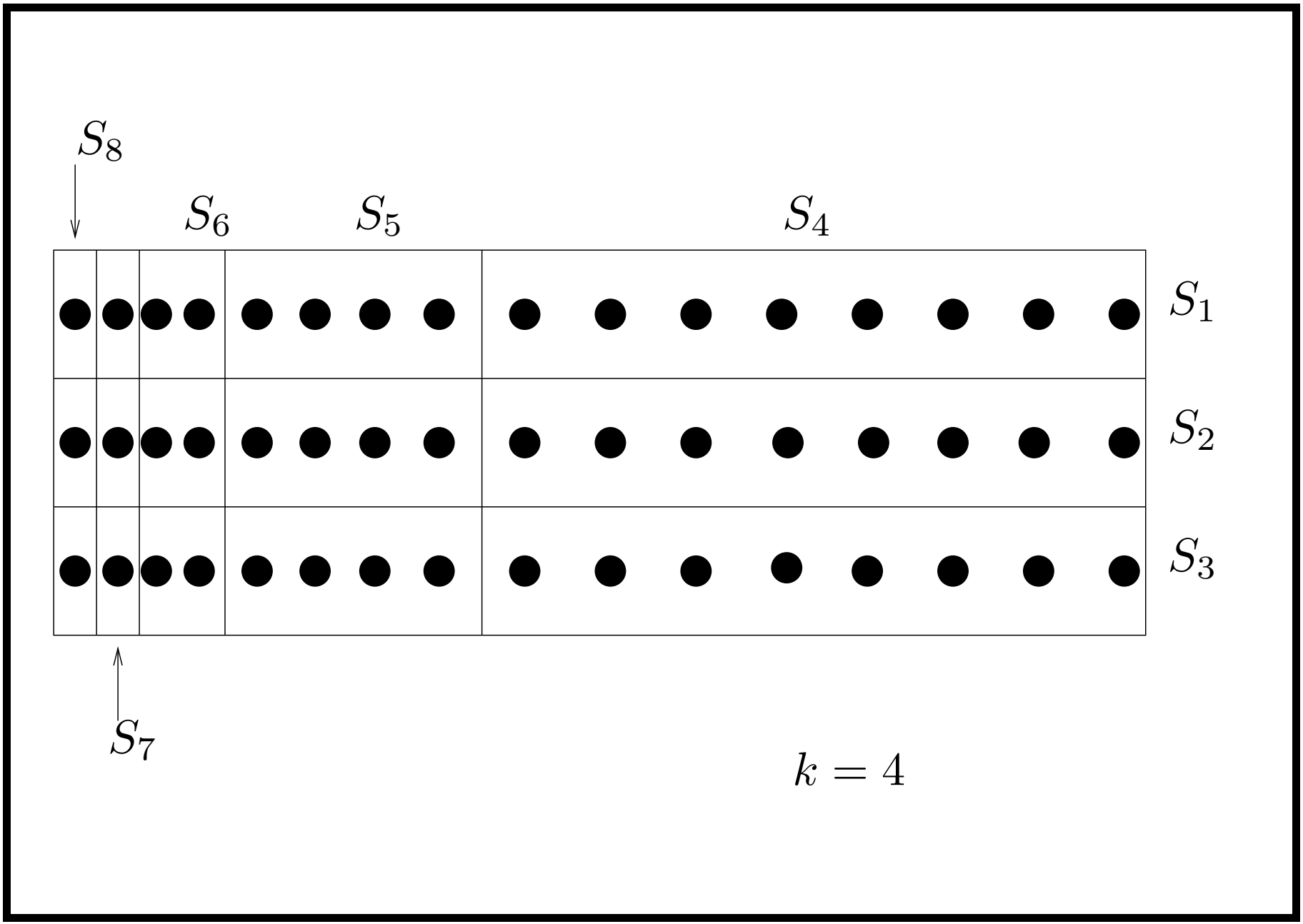
$\mathcal{S}^*$ une solution optimale pour I
$\mathcal{S}_i^*$ une solution optimale pour I_i, $i = 1, \ldots, \Delta$
Les membres de $\mathcal{S}^*$ couvrant C_i sont toujours présents dans I_i et forment une solution réalisable pour I_i

$$\mathrm{opt}\left(I_i\right) = \left|\mathcal{S}_i^*\right| \leqslant \left|\mathcal{S}^*\right| = \mathrm{opt}(I) \tag{6}$$

$$|C_i| \leqslant i \times \mathrm{opt}\left(I_i\right) \tag{7}$$

Par (5), (6) et (7) :

$$m\left(I, \mathcal{S}'\right) \leqslant \mathrm{opt}(I) \times \sum_{i=1}^{\Delta} \frac{1}{i} \leqslant \mathrm{opt}(I)(1 + \ln \Delta)$$


S_8
S_6
S_5
S_4
S_1
S_2
S_3
S_7
$k = 4$

COLORATION PONDÉRÉE dans des graphes bipartis

- ordonner les sommets de $B = (U, D, E, \vec{w})$ en ordre décroissant des poids ;
- fixer un $\epsilon > 0$; $\eta := \lceil 1/\epsilon \rceil$; $S_U := \{v_{4\eta+3}, \ldots, v_n\} \cap U$; $S_D := \{v_{4\eta+3}, \ldots, v_n\} \cap D$;
- calculer la meilleure coloration $\tilde{\mathcal{S}}$ du graphe induit par les sommets $v_1, \ldots, v_{4\eta+2}$;
- retourner $\hat{S} := S_U \cup S_D \cup \tilde{\mathcal{S}}$

Le problème de COLORATION PONDÉRÉE dans des graphes bipartis appartient à DPTAS

$\mathcal{S}^* = (S_1^*, S_2^*, \ldots, S_p^*)$: une solution optimale
$w_{i_1}, w_{i_2}, \ldots, w_{i_p}$: les poids de ses stables
$(w_1 = w_{i_1} \geqslant w_{i_2} \geqslant \ldots \geqslant w_{i_p})$
$B' = B[\{v_1, \ldots, v_{4\eta+2}\}]$

$$\begin{aligned} \omega(B) &= \sum_{v_i \in U \cup D} w(v_i) \\ \mathrm{opt}(B) &= w_{i_1} + w_{i_2} + \ldots + w_{i_p} \\ \omega(B') &\leqslant \omega(B) \end{aligned}$$

- $|\tilde{\mathcal{S}}| \leqslant 2\eta + 2$
  (sinon il existe au moins trois couleurs-singletons dont au moins deux sont dans L ou R ; en les réunissant on diminue la valeur) ;

- $m(B', \tilde{\mathcal{S}}) = \mathrm{opt}(B') \leqslant \mathrm{opt}(B)$
  ($S_1^* \cap V(B'), S_2^* \cap V(B'), \ldots, S_p^* \cap V(B')$ est réalisable pour B')

$$\begin{aligned}
\omega\left(B'\right) - \mathrm{opt}\left(B'\right) &= \sum_{i=1}^{4\eta+2} w_i - \sum_{j=1}^{|\tilde{\mathcal{S}}|} w_{i_j} \\
&\geqslant \left(4\eta + 2 - (2\eta + 2)\right) w_{4\eta+2} \\
&= 2\eta w_{4\eta+2} \quad \geqslant \quad \frac{2}{\epsilon} w_{4\eta+2} \\
w\left(S_U\right) &\leqslant w_{4\eta+2} \\
w\left(S_D\right) &\leqslant w_{4\eta+2}
\end{aligned}$$

$$
\begin{aligned}
m\left(B, \hat{S}\right) &= w\left(S_U\right) + w\left(S_D\right) + \mathrm{opt}\left(B'\right) \\
&= (1-\epsilon)\mathrm{opt}\left(B'\right) \\
&\quad + \epsilon\left(\mathrm{opt}\left(B'\right) + \frac{1}{\epsilon} w\left(S_U\right) + \frac{1}{\epsilon} w\left(S_D\right)\right) \\
&\leqslant (1-\epsilon)\mathrm{opt}\left(B'\right) + \epsilon\left(\mathrm{opt}\left(B'\right) + \frac{2}{\epsilon} w_{4\eta+2}\right) \\
&\leqslant (1-\epsilon)\mathrm{opt}(B) + \epsilon\omega(B)
\end{aligned}
$$

SAC-À-DOS

$$I = \begin{cases} \max & \sum_{i=1}^{n} a_i x_i \\ & \sum_{i=1}^{n} c_i x_i \leqslant b \end{cases}$$

1. construire $I' = ((a'_i, c_i)_{i=1,\ldots,n}, b)$ avec $a'_i = \lfloor a_i n/(a_{\max}\epsilon) \rfloor$;
2. retourner comme solution pour I, la solution S calculée par la programmation dynamique sur l'instance I'

SAC-À-DOS $\in$ **FPTAS**

Complexité de la programmation dynamique : $O(n^2 a_{\max} \log c_{\max})$
Complexité de l'étape 2 : $O(n^2 a'_{\max} \log c_{\max}) = O((n^3 \log c_{\max})/\epsilon)$
S^* : une solution optimale pour $I = ((a_i, c_i)_{i=1,\ldots,n}, b)$
elle est réalisable pour I'
$t = a_{\max}\epsilon/n$
$a'i = \lfloor a_i/t \rfloor$

$$\begin{aligned} \operatorname{opt}(I') &\geqslant \sum_{i \in S^*} a'_i \geqslant \sum_{i \in S^*} \left(\frac{a_i}{t} - 1\right) \\ &\geqslant \frac{\operatorname{opt}(I)}{t} - |S^*| \geqslant \frac{\operatorname{opt}(I)}{t} - n \qquad (8) \\ t \operatorname{opt}(I') &\geqslant \operatorname{opt}(I) - nt \qquad (9) \end{aligned}$$

La solution qui consiste à prendre seulement l'objet i_0 de valeur $a_{\max}$ est réalisable : $\operatorname{opt}(I) = \sum_{i \in S^*} a_i \geqslant a_{\max}$

$$nt = a_{\max}\epsilon \leqslant \epsilon \operatorname{opt}(I) \tag{10}$$

Par (9) et (10) :

$$\begin{aligned} m(I,S) &= \sum_{i\in S} a_i \geqslant t\sum_{i\in S} a'_i = t\operatorname{opt}(I') \\ &\geqslant \operatorname{opt}(I) - nt \geqslant (1-\epsilon)\operatorname{opt}(I) \end{aligned}$$

Réductions et approximabilité

Comment :

- Comparer les problèmes entre eux ?
- Comparer les différentes versions d'un même problème (p.ex., pondérée - non pondérée) ?
- Relier les différents types d'approximation pour un même problème
  (p.ex., classique - différentielle) ?
- Appréhender le rôle des paramètres par rapport auxquels nous analysons un rapport d'approximation
  (p.ex., rapports fonctions des divers paramètres d'une instance) ?

- Transférer des résultats d'un problème à l'autre ?
- Affiner, si possible, la structure de chaque classe d'approximabilité
  (p.ex., existence des problèmes très difficiles (complets) pour des classes d'approximabilité) ?

Réductions

Les exemples les plus classiques :

- MAX STABLE – MAX CLIQUE
- COLORATION – MIN PARTITION EN CLIQUES

Une réduction de Π à Π' ($\Pi \leq \Pi'$) préservant l'approximation est la spécification de trois fonctions f, g, c :

- *f transforme I en $I' = f(I)$;*
- *g transforme S' en $S = g(I, S')$;*
- *c transforme $r'(I, S')$ en $r(I, S) = c(r'(I, S'))$*

Propriété de base d'une réduction préservant l'approximation

- si Π' est approximable à rapport r', alors Π est approximable à rapport $r = c(r')$;
- si Π (sous une hypothèse de complexité) n'est pas approximable à rapport r, alors (pour peut que c soit inversible) Π n'est pas non plus approximable à rapport $c^{-1}(r)$

MAX STABLE $\leq$ MAX SOUS-GRAPHE BIPARTI

$G(V, E) \longrightarrow G'(V', E')$: deux copies de G reliées de façon bipartie complète

$\mathrm{opt}(G) = \mathrm{opt}(G')/2$

$m(S', G') \leqslant 2m(S, G)$

$r'(G', S') \leqslant r(G, S)$

$c : r' \mapsto r = r'$

$$
\begin{array}{rcl}
r'(G', S') \text{ constant} & \longrightarrow & r(G, S) \text{ constant} \\
r'(G', S') = r'(|G'|) & \longrightarrow & r(G, S) = r(|G'|) \\
 & & = r(2|G|) \\
r'(G', S') = r'(\Delta(G')) & \longrightarrow & r(G, S) = r(\Delta(G) + n)
\end{array}
$$

MAX SOUS-GRAPHE BIPARTI n'est pas approximable à mieux que $O(n^{\epsilon-1})$, $\forall \epsilon > 0$

Les réductions ne sont pas universelles !

La plupart des réductions telles qu'elles sont définies ne préservent pas toute forme de rapport d'approximation !

MAX STABLE PONDÉRÉ $\leq$ MAX STABLE

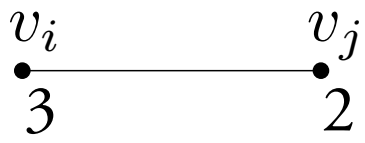
$G \longrightarrow G'$:

Un sommet v pondéré par w_v devient un stable W_v sur w_v sommets ;

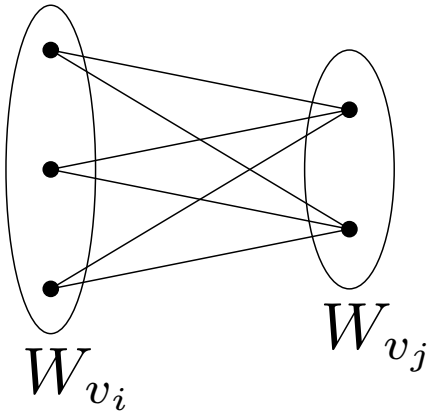
Une arête (u, v) devient un graphe biparti complet K_{w_u, w_v} ;

Toute solution S' dans G' correspond à une reunion des groupes W entier : $S' = W_{v_{i_1}} \cup \ldots \cup W_{v_{i_k}}$;
$S = \{v_{i_1}, \ldots, v_{i_k}\}$ un stable dans G ;

$$m(G', S') = |S'| = \sum_{j \in \{1,\ldots,k\}} w_{v_{i_j}} = m(G, S)$$



(a) Une arête de G



(b) la même arête dans G'

Cette réduction préserve **APX** *mais ni* ***Poly-APX****, ni* **Δ*-APX***

n' peut être en $O(nw_{\max})$ et Δ' en $O(\Delta w_{\max})$

MAX STABLE et MAX CLIQUE

La réduction classique : *stable dans* G *– clique dans* $\bar{G}$ préserve **APX** et **Poly-APX** mais pas **Δ-APX**

$\forall v \in V, G_v = G[\{v\} \cup \Gamma(v)]$
$|G_v| = |\bar{G}_v| \leqslant \Delta(G) + 1$
$\Delta(\bar{G}_v) \leqslant \Delta(G)$

Un des G_v *contient une clique de* G *de taille maximum*

Alors : rapport $r(|\bar{G}_v|) \geqslant r(\Delta(G)+1)$ pour MAX STABLE sur $\bar{G}_v \Longleftrightarrow$ rapport $r(|\bar{G}_v|) = r(|G_v|) \geqslant r(\Delta(G)+1)$ pour MAX CLIQUE dans G

La réduction préserve ***Δ-APX***

MAX CLIQUE est approximable à rapport $O(\log^2 \Delta(G)/\Delta(G))$

Inapproximabilité

Gap technique

L'idée de base (pour un problème Π' de minimisation) :
on conçoit une réduction entre Π, un problème de décision **NP**-complet, et Π', un problème de **NPO**, telle que :
$\exists c, r$ tels que

- si $I \in \mathcal{O}_\Pi$ alors $\mathrm{opt}(I') \leqslant c$;
- $I \in \mathcal{I}_\Pi \setminus \mathcal{O}_\Pi$, alors $\mathrm{opt}(I') > rc$

Alors Π' n'est pas approximable à rapport r

Puis si on conçoit une réduction (f, g, c) préservant l'approximabilité entre Π' et Π'', alors Π'' n'est pas approximable à mieux que $c^{-1}(r)$

Si Π' approximable à rapport r, alors :

- $\forall I \in \mathcal{O}_\Pi, m(f(I), S') \leqslant rc$;
- $\forall I \in \mathcal{I}_\Pi \setminus \mathcal{O}_\Pi, m(f(I), S') \geqslant \mathrm{opt}(f(I)) > rc$

MIN TSP

Si $\mathbf{P} \neq \mathbf{NP}$, alors il n'y a pas d'algorithme polynomial pour MIN TSP garantissant rapport d'approximation (classique) strictement meilleur que $d_{\max}/(nd_{\min})$

Le rapport $d_{\max}/d_{\min}$ est garanti par tout algorithme polynomial

CYCLE HAMILTONIEN se réduit polynomialement à MIN TSP
$G \longrightarrow K_n$; pour $e \in E(K_n)$,

$$d(e) = \begin{cases} 1 & \text{si } e \in E(G) \\ \frac{d_{\max}}{d_{\min}} & \text{si } e \notin E(G) \end{cases}$$

G hamiltonien :

$$\begin{aligned} \operatorname{opt}(K_n) &= n \\ m(K_n, T) &< n\frac{d_{\max}}{n d_{\min}} = \frac{d_{\max}}{d_{\min}} \end{aligned}$$

G non-hamiltonien :

$$\begin{aligned} \operatorname{opt}(K_n) &\geqslant \frac{d_{\max}}{d_{\min}} \\ m(K_n, T) &\geqslant \operatorname{opt}(K_n) \geqslant \frac{d_{\max}}{d_{\min}} \end{aligned}$$

La promenade est finie mais l'aventure a à peine commencé

- Plus de résultats d'approximation, structurels ou opérationnels ;
- Approximation, en temps prédéfini, des problèmes polynomiaux ;
- Approximation et algorithmique *on-line*