

**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 polyédrales ;
 - métaheuristiques ;
 - recherche locale ;
 - *algorithmes polynomiaux approchés avec garanties de performance*

Qu'est-ce-que l'approximation 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 $(\mathcal{I}, \text{Sol}, m, \text{opt})$ 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) ;
- $\forall I \in \mathcal{I}, \text{Sol}(I)$ est l'ensemble des solutions réalisables de I ; tout $S \in \text{Sol}(I)$ est représentable et décidable en temps polynomial en $|I|$; enfin, $\forall I \in \mathcal{I}$, on sait déterminer une solution $\text{triv}(I) \in \text{Sol}(I)$ en temps polynomial en $|I|$; pour $I \in \mathcal{I}$ et $S \in \text{Sol}(I)$, $m(I, S)$ est la valeur de la solution S ;
- $m : \mathcal{I} \times \text{Sol} \rightarrow \mathbb{N}$ est calculable en temps polynomial en $|I|$;
- $\text{opt} \in \{\text{min}, \text{max}\}$

Pour $\Pi = (\mathcal{I}, \text{Sol}, m, \text{opt})$:

$\text{opt}(I)$: la valeur d'une solution optimale

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

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

Pour un algorithme approché A :

$m_A(I, S)$, la valeur de la solution S calculée par A sur I

Rapport d'approximation classique pour A :

$$\rho_A(I) = \frac{m_A(I, S)}{\text{opt}(I)}$$

Rapport d'approximation différentielle pour A :

$$\delta_A(I) = \frac{|\omega(I) - m_A(I, S)|}{|\omega(I) - \text{opt}(I)|}$$

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

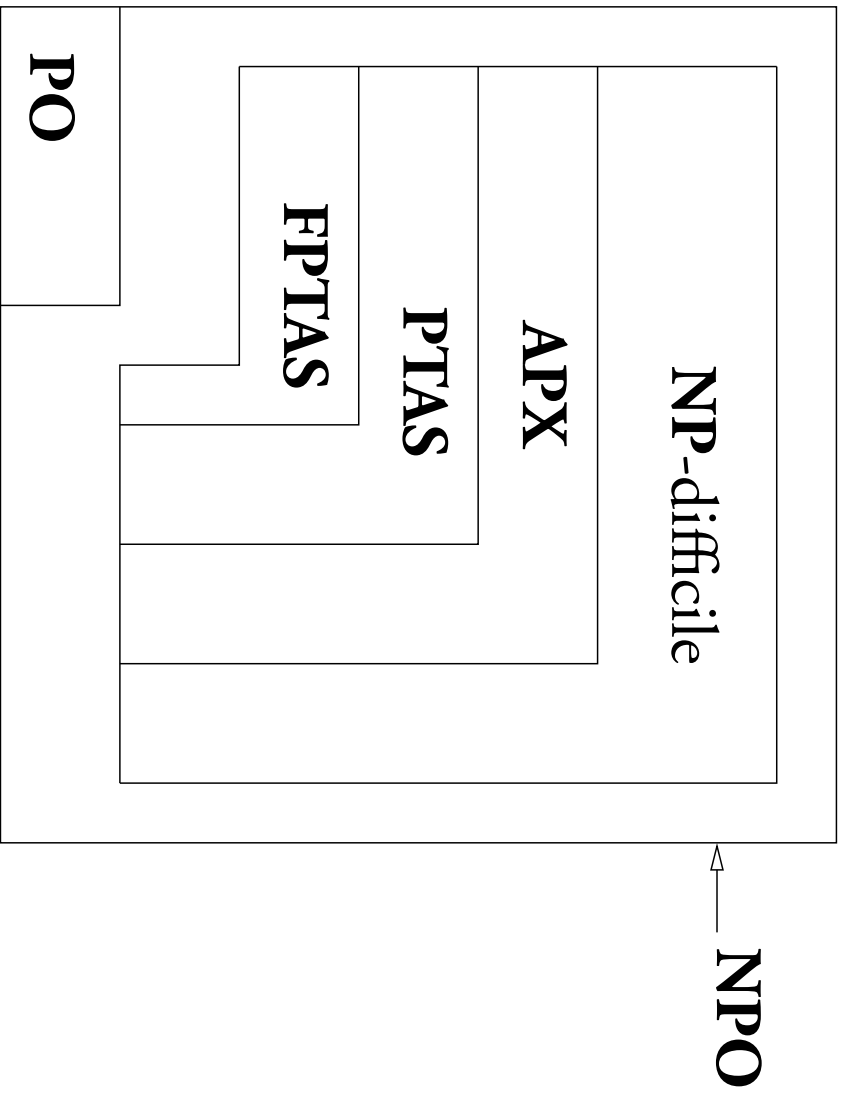
Classification absolue des problèmes et des algorithmes approchés

- 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 $\text{opt} = \text{min}$), ou $1 - \epsilon$ (si $\text{opt} = \text{max}$, 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 $\text{opt} = \min$), ou $1 - \epsilon$ (si $\text{opt} = \max$ 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



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

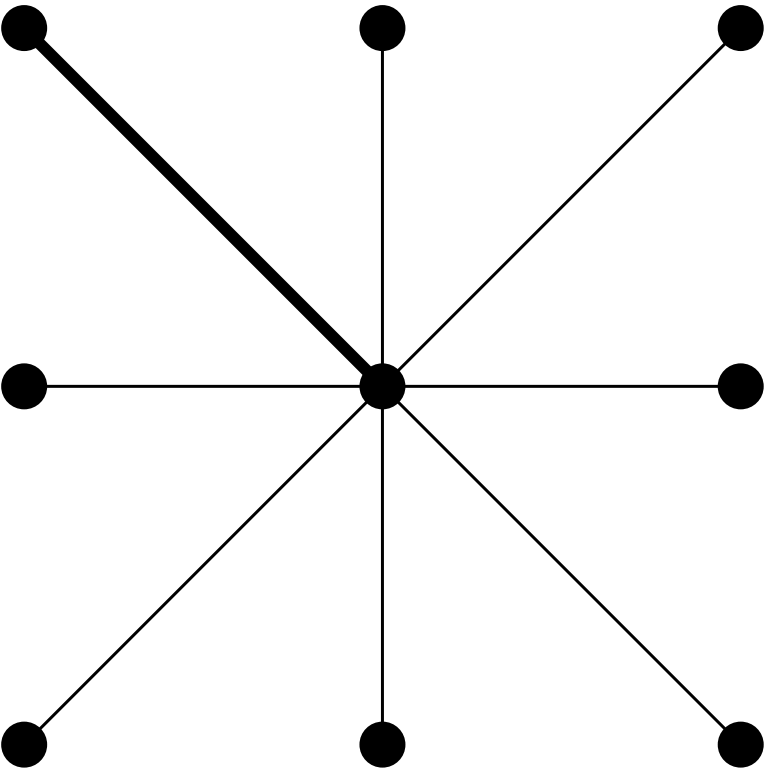
MIN VERTEX COVER

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 APX \ 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. déterminer 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

Réurrence sur n

Vrai pour $n \leq k$

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 :

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

X : la solution retournée par l'algorithme

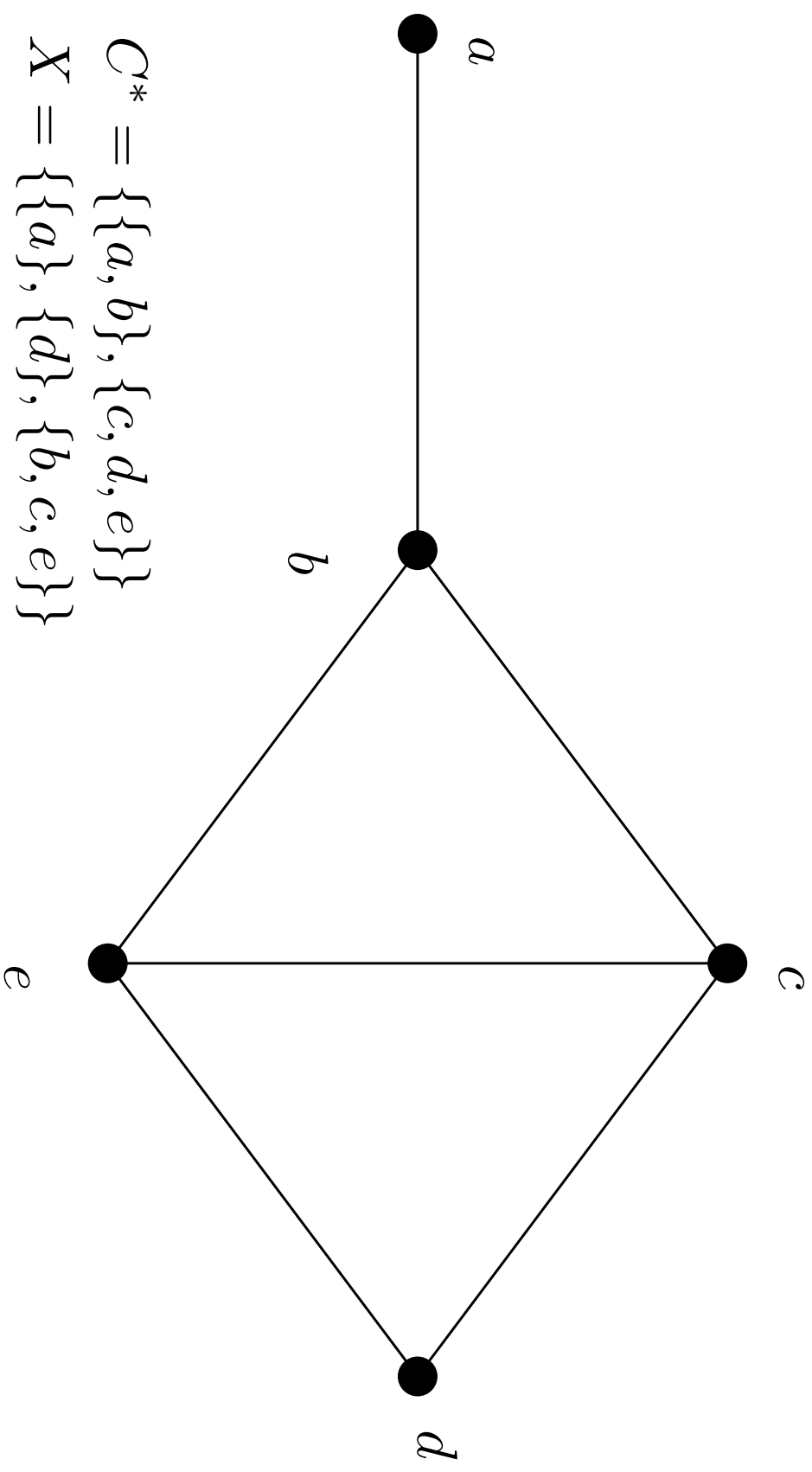
$X \setminus S$ réalisable pour $G[V \setminus S]$ (et calculable par l'algorithme) :

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

De (1) et (2) :

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

\bar{G}



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

$$X = \{\{a\}, \{d\}, \{b, c, e\}\}$$

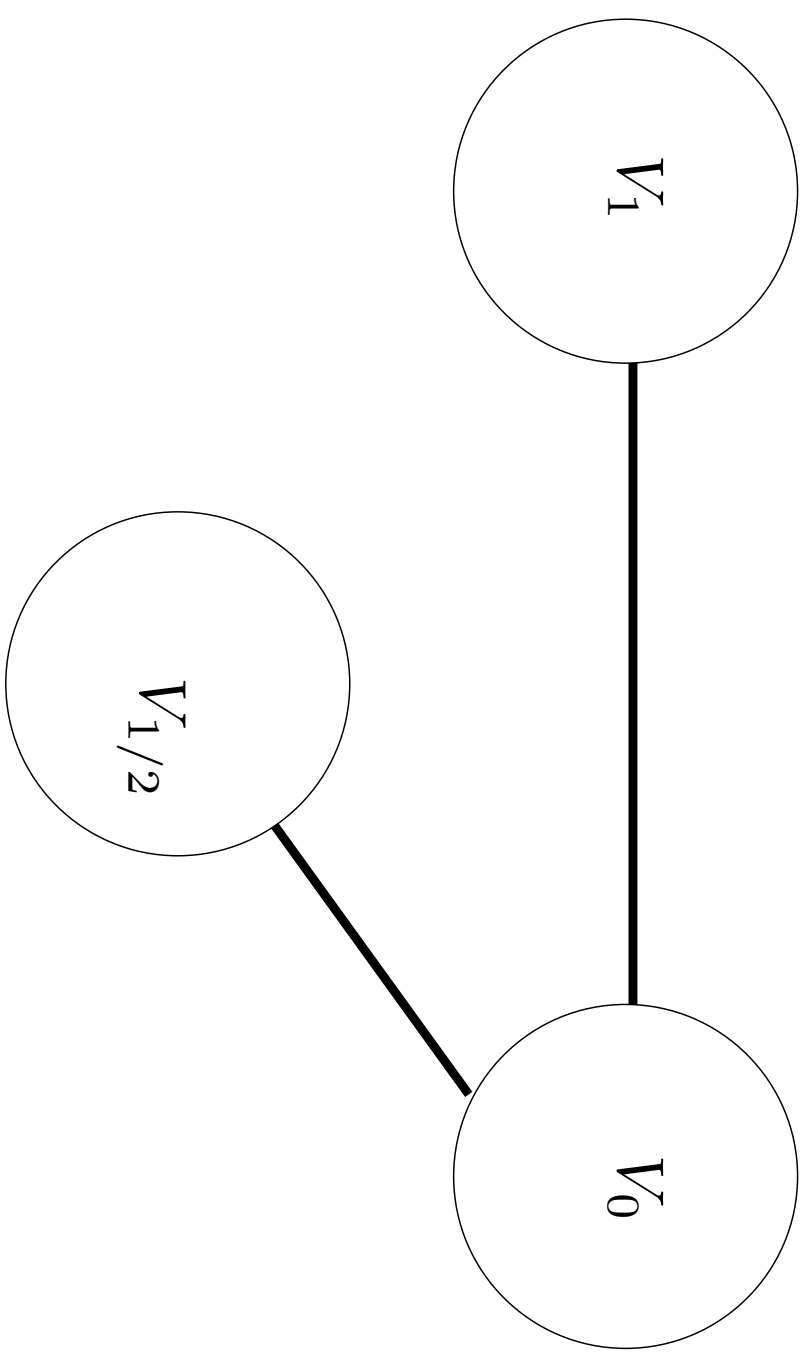
MAX STABLE

$$\text{MAX STABLE} = \left\{ \begin{array}{l} \max \quad \vec{1} \cdot \vec{x} \\ A \cdot \vec{x} \leq \vec{1} \\ \vec{x} \in \{0, 1\}^n \end{array} \right.$$

$$\text{MAX STABLE-R} = \left\{ \begin{array}{l} \max \quad \vec{1} \cdot \vec{x} \\ A \cdot \vec{x} \leq \vec{1} \\ \vec{x} \in (\mathbb{Q}^n)^+ \end{array} \right.$$

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. $V_0 \subseteq V \setminus S^*$.



- 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

$S^* = V_1 \cup S_{1/2}^*$

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

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

$$\text{opt}(G) = |S^*| = |V_1| + |S_{1/2}^*| \leq |V_1| + \frac{|V_{1/2}|}{2}$$

MIN SET COVER

- répéter : soit S_0 un ensemble de S de cardinalité maximum ;
mettre S_0 dans la solution S' ; mettre à jour 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

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

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

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

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

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

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

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

S^* une solution optimale pour I

S_i^* une solution optimale pour I_i , $i = 1, \dots, \Delta$

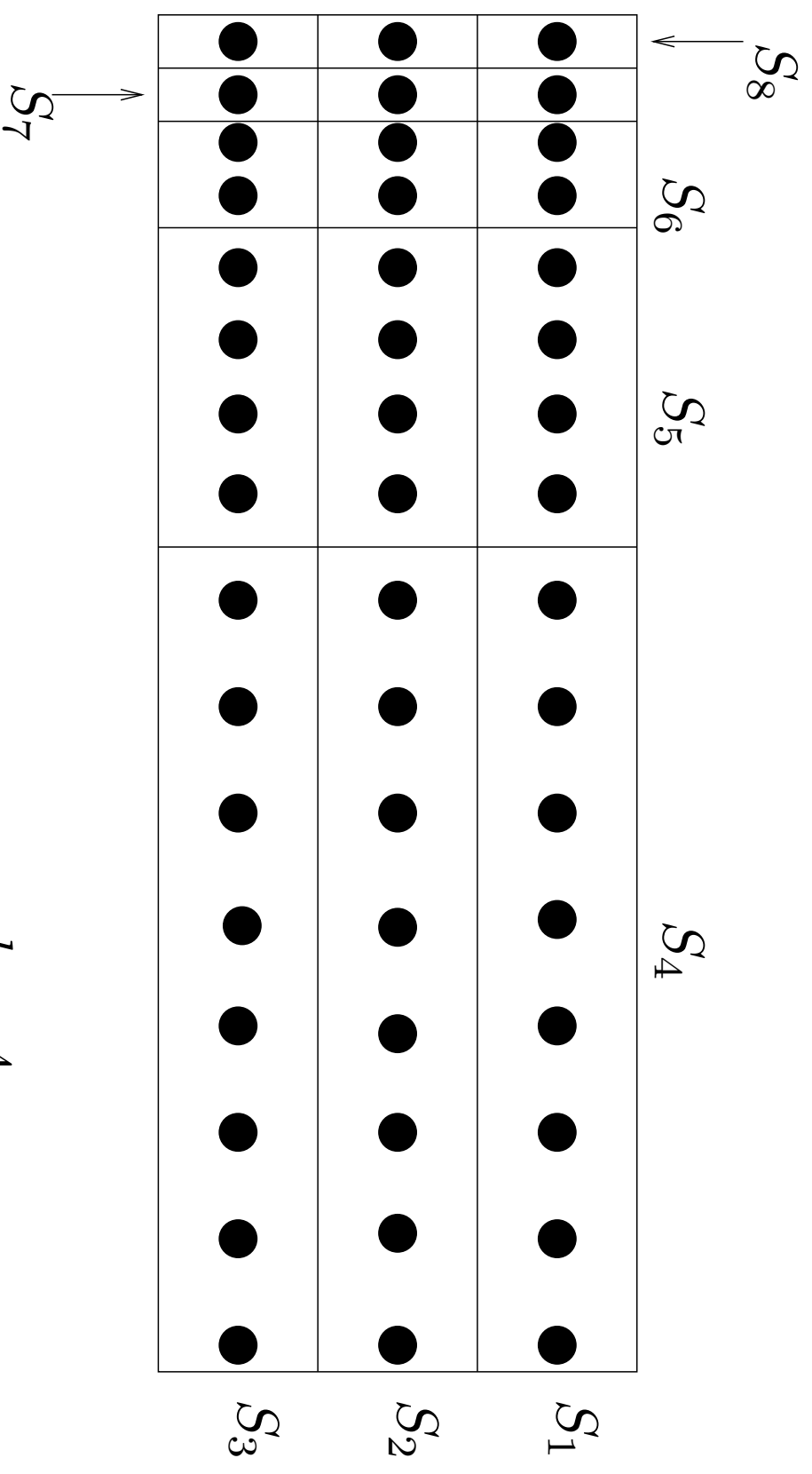
Les membres de S^* couvrant C_i sont toujours présents dans I_i et forment une solution réalisable pour I_i

$$\text{opt}(I_i) = |S_i^*| \leq |S^*| = \text{opt}(I) \quad (6)$$

$$|C_i| \leq i \times \text{opt}(I_i) \quad (7)$$

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

$$m(I, S') \leq \text{opt}(I) \times \sum_{i=1}^{\Delta} \frac{1}{i} \leq \text{opt}(I)(1 + \ln \Delta)$$



$$k = 4$$

COLORATION PONDÉRÉE dans des graphes bipartis

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

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

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

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

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

- $m(B', \tilde{S}) = \text{opt}(B') \leq \text{opt}(B)$
 $(S_1^* \cap V(B'), S_2^* \cap V(B'), \dots, S_p^* \cap V(B'))$ est réalisable pour B'

$$\begin{aligned}
 w(B') - \text{opt}(B') &= \sum_{i=1}^{4\eta+2} w_i - \sum_{j=1}^{|\tilde{S}|} w_{i_j} \\
 &\geq (4\eta + 2 - (2\eta + 2)) w_{4\eta+2} \\
 &= 2\eta w_{4\eta+2} \geq \frac{2}{\epsilon} w_{4\eta+2} \\
 w(S_U) &\leq w_{4\eta+2} \\
 w(S_D) &\leq w_{4\eta+2}
 \end{aligned}$$

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

SAC-À-DOS

$$I = \left\{ \begin{array}{l} \max \\ \sum_{i=1}^n a_i x_i \\ \sum_{i=1}^n c_i x_i \leq b \end{array} \right.$$

1. construire $I' = ((a'_i, c_i)_{i=1, \dots, n}, b)$ 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 $I = ((a_i, c_i)_{i=1, \dots, n}, b)$

elle est réalisable pour I'

$$t = a_{\max} \epsilon / n$$

$$a'_i = \lfloor a_i / t \rfloor$$

$$\text{opt}(I') \geq \sum_{i \in S^*} a'_i \geq \sum_{i \in S^*} \left(\frac{a_i}{t} - 1 \right)$$

$$\geq \frac{\text{opt}(I)}{t} - |S^*| \geq \frac{\text{opt}(I)}{t} - n \quad (8)$$

$$t \text{opt}(I') \geq \text{opt}(I) - nt \quad (9)$$

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

$$nt = a_{\max} \epsilon \leq \epsilon \operatorname{opt}(I) \quad (10)$$

Par (9) et (10) :

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

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

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

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

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

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

$$r'(G', S') \text{ constant} \longrightarrow r(G, S) \text{ constant}$$

$$\begin{aligned} r'(G', S') = r'(|G'|) &\longrightarrow r(G, S) = r(|G'|) \\ &= r(2|G'|) \end{aligned}$$

$$r'(G', S') = r'(\Delta(G')) \longrightarrow r(G, S) = r(\Delta(G) + n)$$

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 ;

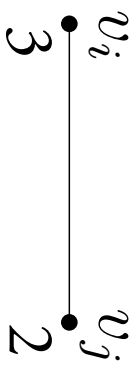
Une arête (u, v) devient un graphe biparti complet K_{w_u, w_v} ;

Toute solution S' dans G' correspond à une réunion des

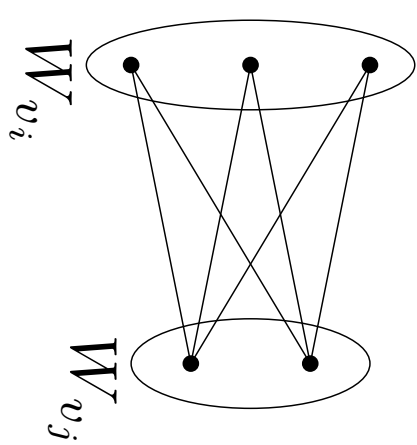
groupes W entier : $S' = W_{v_{i_1}} \cup \dots \cup W_{v_{i_k}}$;

$S = \{v_{i_1}, \dots, v_{i_k}\}$ un stable dans G ;

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



(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 $O(mw_{\max})$ 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**

$$\forall v \in V, G_v = G[\{v\} \cup \Gamma(v)]$$

$$|G_v| = |\bar{G}_v| \leq \Delta(G) + 1$$

$$\Delta(\bar{G}_v) \leq \Delta(G)$$

Un des G_v contient une clique de G de taille maximum

Alors : rapport $r(|\bar{G}_v|) \geq r(\Delta(G) + 1)$ pour MAX STABLE
 sur $\bar{G}_v \iff$ rapport $r(|\bar{G}_v|) = r(|G_v|) \geq r(\Delta(G) + 1)$ 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 $\text{opt}(I') \leq c$;
- $I \in \mathcal{I}_\Pi \setminus \mathcal{O}_\Pi$, alors $\text{opt}(I') > rc$

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 :

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

MIN TSP

Si $P \neq NP$, 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 :

$$\text{opt}(K_n) = n$$

$$m(K_n, T) < n \frac{d_{\max}}{nd_{\min}} = \frac{d_{\max}}{d_{\min}}$$

G non-hamiltonien :

$$\text{opt}(K_n) \geq \frac{d_{\max}}{d_{\min}}$$

$$m(K_n, T) \geq \text{opt}(K_n) \geq \frac{d_{\max}}{d_{\min}}$$

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*