

Approx

Soit P un pb de minimisation

↳ trouver une solution faisable (valide) qui minimise un certain critère

Pour une instance X de P , on écrit $OPT(X)$ pour désigner la valeur d'une sol. opt. pour X .

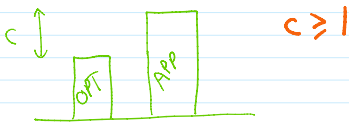
Soit A un algo pour P , i.e. A retourne \forall instance X de P une sol. faisable

On désigne par $APP(A, X)$, la valeur de la sol. retournée par A sur entrée X .

On dit que A est une c -approximation à P si, \forall instance X de P

① A s'exécute en temps polynomial, donc $O(|X|^k)$
 $k = \text{constante}$

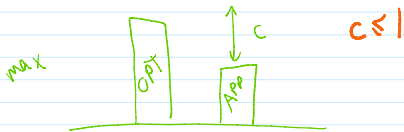
② $APP(A, X) \leq c \cdot OPT(X)$



Si P est un pb de maximisation, A est une c -approx. si \forall instance X de P

① A s'exécute en temps poly.

② $APP(A, X) \geq c \cdot OPT(X)$



Note: si A et X sont claires, on écrit OPT et APP

Note: c peut être une constante
• une fonction de n ($n = |X|$)

ex: SET-COVER admet une $\log n$ -approx.

MAX-SAT (satisfaisabilité booléenne)

Clause = variables bool. liées par des \vee (ou)

ex: $C = x_1 \vee \bar{x}_2 \vee x_3$

$C_3 = x_1 \vee (x_2 \wedge x_3) \equiv (x_1 \vee x_2) \wedge (x_1 \vee x_3)$

Une variable x_i peut être positive x_i
negative \bar{x}_i

Une assignation attribue une valeur T ou F à chaque x_i

Elle satisfait une clause C si cette dernière inclut à true.

ex: $x_1 \vee \bar{x}_2 \vee x_3$ $x_1 = F$ $x_2 = F$ $x_3 = T$

$F \vee \bar{F} \vee T \equiv F \vee T \vee T \equiv T$

La seule façon de ne pas satisfaire C est
 $x_1 = F$ $x_2 = T$ $x_3 = F$

$C_1 = x_1$

$C_2 = \bar{x}_2$

$C_1 \wedge C_2 \wedge C_3$

$(x_1 \wedge \bar{x}_2) \wedge (x_2 \wedge x_3) \wedge (x_1 \vee \bar{x}_2) \wedge (x_1 \vee x_3)$

$(\bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_3)$

MAX-SAT:

Entrée: ensemble de clauses C_1, C_2, \dots, C_m sur variables x_1, x_2, \dots, x_n

Cost: ...

Entrée: ensemble de clauses C_1, C_2, \dots, C_m sur variables x_1, x_2, \dots, x_n

Sortie: une assignation des x_i qui maximise le # de clauses satisfaites

$\text{max sat}(C_1, C_2, \dots, C_m, x_1, x_2, \dots, x_n)$

Soit A l'assign. t.q. $x_1=T, x_2=T, \dots, x_n=T$
si A satisfait au moins $m/2$ clause
return A

soit \bar{A} l'assign. t.q. $x_1=F, x_2=F, \dots, x_n=F$
return \bar{A}

Théorème: max sat est une $\frac{1}{2}$ -approx.

Sketch: On a $\text{OPT} \leq m$ car il y a m clauses.

Pour APP , si l'algo retourne A , alors $\text{APP} \geq m/2$ (c'est vérifié explicitement).

Si l'algo retourne \bar{A} , c'est parce que A satisfait $< m/2$ clauses.
On remarque que tout ce que A ne sat. pas est sat. par \bar{A}

$$C_i = \bar{x}_a \vee \bar{x}_b \vee \bar{x}_c \vee \bar{x}_d$$

Donc, \bar{A} satisfait $> m/2$ clauses.

Dans les 2 cas, on a $\text{APP} \geq m/2$

$$\text{OPT} \leq m$$

$$\text{APP} \geq m/2$$

$$\Rightarrow \text{APP} \geq \frac{m}{2} \geq \frac{\text{OPT}}{2} \geq \frac{1}{2} \text{OPT}$$

Technique fondamentale d'analyse d'approx.

- ① Trouver une borne sur OPT
- ② Montrer que APP est "proche" de cette borne

• Minimisation

$$\text{OPT} \geq k$$

$$\text{APP} \leq c \cdot k \quad c \geq 1$$

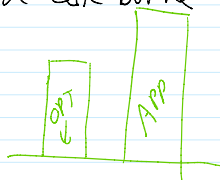
$$\Rightarrow \text{APP} \leq c \cdot k \leq c \cdot \text{OPT}$$

• Maximisation

$$\text{OPT} \leq k$$

$$\text{APP} \geq c \cdot k \quad c \leq 1$$

$$\Rightarrow \text{APP} \geq c \cdot k \geq c \cdot \text{OPT}$$



VERTEX-COVER

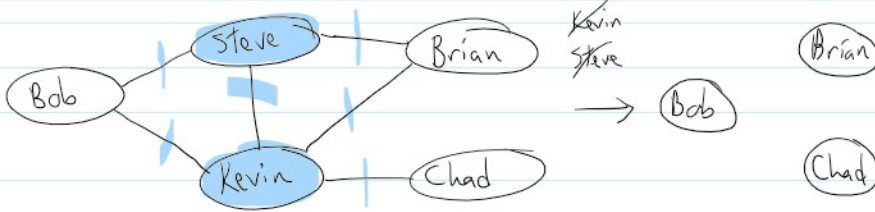
Soit $G=(V, E)$ un graphe. Un ensemble $X \subseteq V$ est couvrant si: $\forall uv \in E, u \in X$ ou $v \in X$
(ou les deux)

Entrée: un graphe $G=(V,E)$

Sortie: un ens. couvrant $X \subseteq V$ de taille min.

ex: vous avez un bar, où certaines paires de clients qui se battent si présent en même.

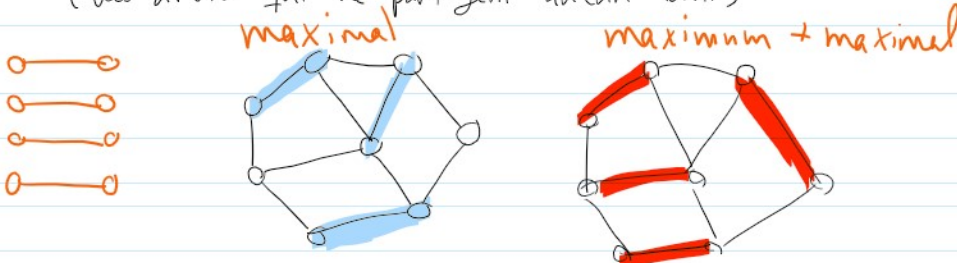
Vous voulez bloquer l'entrée à un # min pour 0 bataille.



① Trouver une borne sur OPT ($OPT \geq k$)

La taille d'un matching maximal est une borne sur OPT.

Def.: sur un graphe $G=(V,E)$, un matching est un ensemble d'arêtes $M \subseteq E$ tel que $\forall uv, xy \in M, u \neq x, u \neq y, v \neq x, v \neq y$ (des arêtes qui ne partagent aucun bout)



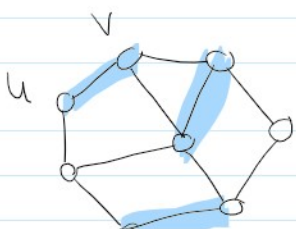
Un matching est maximal si on ne peut plus lui ajouter d'arêtes.

M est maximal si $\forall uv \in E \setminus M, M \cup \{uv\}$ n'est pas un matching.

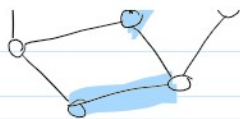
Lemme: soit M un matching maximal, et soit X un ens. couvrant minimum.

Alors $|X| \geq |M|$. ($OPT \geq |M|$)

Sketch: X doit couvrir chaque arête de M ,



Puisque chaque arête de M ne partage pas de sommet, X doit contenir au moins un sommet de chaque arête de M .



Voici une 2-approx.

vc-matching ($G=(V,E)$)

$X = \{\}$

tant que $|E| > 0$

soit $uv \in E$

Ajouter u à X , ajouter v à X

Enlever u et v de G (avec leurs arêtes)

return X

À prouver: le X retourné est faisable (ens. couvrant)

$$APP \leq 2 \cdot OPT$$

Théorème: l'algo vc-matching est une 2-approx. au pb VERTEX-COVER

Preuve: Soit X l'ens. retourné par l'algo.

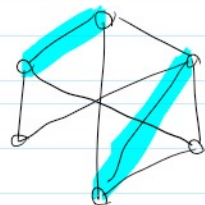
On a que X contient les sommets d'un matching maximal M .

On sait que $OPT \geq |M|$.

D'un autre côté, $APP = |X| = 2|M|$.

$$\Rightarrow APP = 2|M| \leq 2 \cdot OPT.$$

Reste à prouver: X est un ens. couvrant. \square



Analyse "serrée" (tight analysis)

Est-ce notre analyse sur un algo peut être améliorée?

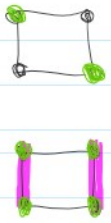
ex: est-ce que vc-matching est une 1.5-approx?

Rep: non, vc-matching n'est pas mieux qu'une 2-approx



$$OPT = 2$$

$\Rightarrow \exists$ instances telles que



$$\text{OPT} = 2$$
$$\text{APP} = 4$$

$\Rightarrow \exists$ instances telles que
 $\text{APP} = 2 \cdot \text{OPT}$

Revenons à 3-SET-COVER

algo(U, S)

pour chaque $u_i \in U$

si u_i n'est pas couvert

ajouter un $S \in S$ qui couvre u_i

$$\text{OPT} \geq \frac{n}{3} \Rightarrow \text{APP} < 3 \text{OPT}$$

$$\text{APP} \leq n - 2$$

Q: exemple(s) montrant que $\text{APP} \leq c \cdot \text{OPT}$, $c < 3$ est impossible