COMPLEX - Cours 9
Résolution probabiliste de problèmes SAT

Damien Vergnaud

Sorbonne Université – CNRS

Table des matières

Problème 3-SAT

$x_1 \vee x_4 \vee \neg x_6 \qquad x_2 \vee \neg x_4 \vee x_3 \qquad x_1 \vee \neg x_2 \vee x_6$

$\underbrace{x_1 \vee x_4 \vee \neg x_6}_{\text{clause}} \qquad x_2 \vee \neg x_4 \vee x_3 \qquad x_1 \vee \neg x_2 \vee x_6$

littéraux

Problème 3-SAT

- Entrée : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec au plus 3 littéraux
- Sortie : VRAI si il existe une assignation des variables, qui rend la formule vraie (et FAUX sinon)

- problème de décision
- « Le » problème $\mathcal{NP}$-difficile

Problème MAX-3-SAT

Problème MAX-3-SAT

- Entrée : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec au plus 3 littéraux
- Sortie : une assignation des variables qui maximise le nombre de clauses satisfaites

Exemple.

$$x_1 \vee \neg x_2 \vee \neg x_3 \quad \neg x_1 \vee x_2 \vee x_3 \quad x_1 \vee \neg x_2 \vee \neg x_3 \quad x_1 \vee x_2 \vee x_3$$

$$x_1 \vee \neg x_2 \vee \neg x_4 \quad \neg x_1 \vee x_3 \quad x_1 \vee \neg x_3 \quad x_1 \vee x_4 \quad x_2 \vee \neg x_3 \quad \neg x_2 \quad \neg x_4$$

Problème MAX-3-SAT

Problème MAX-3-SAT

- Entrée : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec au plus 3 littéraux
- Sortie : une assignation des variables qui maximise le nombre de clauses satisfaites

Exemple.

$$x_1 \vee \neg x_2 \vee \neg x_3 \quad \neg x_1 \vee x_2 \vee x_3 \quad x_1 \vee \neg x_2 \vee \neg x_3 \quad x_1 \vee x_2 \vee x_3$$

$$x_1 \vee \neg x_2 \vee \neg x_4 \quad \neg x_1 \vee x_3 \quad x_1 \vee \neg x_3 \quad x_1 \vee x_4 \quad x_2 \vee \neg x_3 \quad \neg x_2 \quad \neg x_4$$

- $(0,0,0,0) \rightsquigarrow 9$ clauses satisfaites

Problème MAX-3-SAT

Problème MAX-3-SAT

- Entrée : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec au plus 3 littéraux
- Sortie : une assignation des variables qui maximise le nombre de clauses satisfaites

Exemple.

$$x_1 \vee \neg x_2 \vee \neg x_3 \quad \neg x_1 \vee x_2 \vee x_3 \quad x_1 \vee \neg x_2 \vee \neg x_3 \quad x_1 \vee x_2 \vee x_3$$

$$x_1 \vee \neg x_2 \vee \neg x_4 \quad \neg x_1 \vee x_3 \quad x_1 \vee \neg x_3 \quad x_1 \vee x_4 \quad x_2 \vee \neg x_3 \quad \neg x_2 \quad \neg x_4$$

- $(0, 0, 0, 0) \rightsquigarrow$ 9 clauses satisfaites
- $(1, 1, 1, 1) \rightsquigarrow$ 9 clauses satisfaites

Problème MAX-3-SAT

Problème MAX-3-SAT

- Entrée : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec au plus 3 littéraux
- Sortie : une assignation des variables qui maximise le nombre de clauses satisfaites

Exemple.

$$x_1 \vee \neg x_2 \vee \neg x_3 \quad \neg x_1 \vee x_2 \vee x_3 \quad x_1 \vee \neg x_2 \vee \neg x_3 \quad x_1 \vee x_2 \vee x_3$$

$$x_1 \vee \neg x_2 \vee \neg x_4 \quad \neg x_1 \vee x_3 \quad x_1 \vee \neg x_3 \quad x_1 \vee x_4 \quad x_2 \vee \neg x_3 \quad \neg x_2 \quad \neg x_4$$

- $(0,0,0,0) \rightsquigarrow$ 9 clauses satisfaites
- $(1,1,1,1) \rightsquigarrow$ 9 clauses satisfaites
- $(1,0,1,0) \rightsquigarrow$ 10 clauses satisfaites (et c'est le maximum !)

Problème MAX-E3-SAT

Problème MAX-E3-SAT

- Entrée : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec exactement 3 littéraux
- Sortie : une assignation des variables qui maximise le nombre de clauses satisfaites

Exemple.

$$
\begin{array}{cccc}
x_1 \vee \neg x_2 \vee \neg x_3 & \neg x_1 \vee x_2 \vee x_3 & x_1 \vee \neg x_2 \vee \neg x_3 & x_1 \vee x_2 \vee x_3 \\
x_1 \vee \neg x_2 \vee \neg x_4 & \neg x_1 \vee x_2 \vee x_4 & x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee \neg x_2 \vee x_4 \\
\neg x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee x_2 \vee x_4 & \neg x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee x_2 \vee x_4
\end{array}
$$

Problème MAX-E3-SAT

Problème MAX-E3-SAT

- Entrée : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec exactement 3 littéraux
- Sortie : une assignation des variables qui maximise le nombre de clauses satisfaites

Exemple.

$$\begin{array}{cccc}
x_1 \vee \neg x_2 \vee \neg x_3 & \neg x_1 \vee x_2 \vee x_3 & x_1 \vee \neg x_2 \vee \neg x_3 & x_1 \vee x_2 \vee x_3 \\
x_1 \vee \neg x_2 \vee \neg x_4 & \neg x_1 \vee x_2 \vee x_4 & x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee \neg x_2 \vee x_4 \\
\neg x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee x_2 \vee x_4 & \neg x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee x_2 \vee x_4
\end{array}$$

- $(0, 0, 0, 0) \rightsquigarrow 9$ clauses satisfaites

Problème MAX-E3-SAT

Problème MAX-E3-SAT

- Entrée : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec exactement 3 littéraux
- Sortie : une assignation des variables qui maximise le nombre de clauses satisfaites

Exemple.

$$
\begin{array}{cccc}
x_1 \vee \neg x_2 \vee \neg x_3 & \neg x_1 \vee x_2 \vee x_3 & x_1 \vee \neg x_2 \vee \neg x_3 & x_1 \vee x_2 \vee x_3 \\
x_1 \vee \neg x_2 \vee \neg x_4 & \neg x_1 \vee x_2 \vee x_4 & x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee \neg x_2 \vee x_4 \\
\neg x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee x_2 \vee x_4 & \neg x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee x_2 \vee x_4
\end{array}
$$

- $(0,0,0,0) \rightsquigarrow 9$ clauses satisfaites
- $(1,1,1,1) \rightsquigarrow 10$ clauses satisfaites

Problème MAX-E3-SAT

Problème MAX-E3-SAT

- Entrée : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec exactement 3 littéraux
- Sortie : une assignation des variables qui maximise le nombre de clauses satisfaites

Exemple.

$$
\begin{array}{cccc}
x_1 \vee \neg x_2 \vee \neg x_3 & \neg x_1 \vee x_2 \vee x_3 & x_1 \vee \neg x_2 \vee \neg x_3 & x_1 \vee x_2 \vee x_3 \\
x_1 \vee \neg x_2 \vee \neg x_4 & \neg x_1 \vee x_2 \vee x_4 & x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee \neg x_2 \vee x_4 \\
\neg x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee x_2 \vee x_4 & \neg x_1 \vee \neg x_2 \vee \neg x_4 & x_1 \vee x_2 \vee x_4
\end{array}
$$

- $(0,0,0,0) \rightsquigarrow$ 9 clauses satisfaites
- $(1,1,1,1) \rightsquigarrow$ 10 clauses satisfaites
- $(1,1,1,0) \rightsquigarrow$ 12 clauses satisfaites (et c'est le maximum !)

Problème MAX-E3-SAT

Problème MAX-E3-SAT

- Entrée : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec exactement 3 littéraux
- Sortie : une assignation des variables qui maximise le nombre de clauses satisfaites

- 1974 – D. Johnson ; 1992 – M. Yannakakis
  $\rightsquigarrow$ Il existe un algorithme probabiliste polynomial de $7/8$-approximation pour MAX-E3-SAT

- **Idée :** $x_i \longleftarrow \{0, 1\}$!

Assignation aléatoire pour une clause

Considérons une clause $C = (x \vee y \vee z)$. Nous avons

$$\Pr_{(x,y,z)\in\{0,1\}^3}(C \text{ satisfaite}) = \frac{7}{8}$$

Démonstration.

x	y	z	C	x	y	z	C
0	0	0	0	1	0	0	1
0	0	1	1	1	0	1	1
0	1	0	1	1	1	0	1
0	1	1	1	1	1	1	1

$\square$

Assignation aléatoire pour une clause

Considérons une clause $C = (x \vee y \vee z)$. Nous avons

$$\Pr_{(x,y,z)\in\{0,1\}^3}(C \text{ satisfaite}) = \frac{7}{8}$$

Démonstration.

x	y	z	C	x	y	z	C
0	0	0	0	1	0	0	1
0	0	1	1	1	0	1	1
0	1	0	1	1	1	0	1
0	1	1	1	1	1	1	1

□

Assignation aléatoire pour m clauses

Soient Φ une formule E3-SAT et X la variable aléatoire du nombres de clauses de Φ satisfaites par une assignation aléatoire des variables. Nous avons $\mathbb{E}(X) = 7m/8$

Démonstration.

- Notons pour $i \in \{1, \ldots, m\}$, la variable aléatoire

$$X_i = \begin{cases} 1 & \text{si la clause } C_i \text{ est satisfaite} \\ 0 & \text{sinon} \end{cases}$$

- Nous avons

$$X = \sum_{i=1}^{m} X_i$$

Assignation aléatoire pour m clauses

Démonstration (fin).

- Par linéarité de l'espérance

$$\begin{aligned}
\mathbb{E}(X) = \sum_{i=1}^{m} \mathbb{E}(X_i) &= \sum_{i=1}^{m} [1 \cdot \Pr(X_i = 1) + 0 \cdot \Pr(X_i = 0)] \\
&= \sum_{i=1}^{m} \Pr(X_i = 1) \\
&= \sum_{i=1}^{m} \frac{7}{8} = \frac{7m}{8}
\end{aligned}$$

$\square$

Applications – Méthode probabiliste

Soient Φ une formule E3-SAT. Il existe une assignation des variables de Φ pour laquelle au moins $7m/8$ clauses sont satisfaites

Démonstration. Une variable aléatoire est supérieure ou égale à son espérance au moins une fois . . . $\square$

Une formule E3-SAT avec moins de 7 clauses est satisfiable

Démonstration.

$$\left\lceil \frac{7 \times 7}{8} \right\rceil = 7 \ldots$$

$\square$

Applications – Méthode probabiliste

Soient Φ une formule E3-SAT. Il existe une assignation des variables de Φ pour laquelle au moins $7m/8$ clauses sont satisfaites

Démonstration. Une variable aléatoire est supérieure ou égale à son espérance au moins une fois ... □

Une formule E3-SAT avec moins de 7 clauses est satisfiable

Démonstration.

$$\left\lceil \frac{7 \times 7}{8} \right\rceil = 7 \dots$$

□

Algorithme probabiliste pour MAX-E3-SAT

Entrée: Φ formule booléenne MAX-E3-SAT en n variables et m clauses
Sortie: $y \in \{0,1\}$ tel que l'assignation y satisfait $\lceil 7m/8 \rceil$ clauses de Φ

tant que vrai **faire**
  $y = (y_1, \ldots, y_n) \xleftarrow{} \{0,1\}^n$
  si au moins $\lceil 7m/8 \rceil$ clauses de Φ sont satisfaites par y **alors**
    retourner y
  fin si
fin tant que

Loi géométrique – Rappel

- **Loi de Bernoulli** X

$$\mathbb{P}(X = x) = \begin{cases} p & \text{si } x = 1, \\ 1 - p & \text{si } x = 0, \\ 0 & \text{sinon.} \end{cases}$$

- **Loi géométrique** Y = loi du nombre d'épreuves de Bernoulli indépendantes nécessaire pour obtenir le premier succès

$$\mathbb{P}(Y = k) = (1 - p)^{k-1} p$$

$$\mathbb{E}[Y] = \frac{1}{p}$$

Probabilité de succès

Soient Φ une formule E3-SAT et S l'événement « une assignation des variables de Φ satisfait au moins $7m/8$ des clauses de Φ.
Nous avons $p = \Pr(S) \geq 1/8m$

Démonstration.

- p_j = probabilité de satisfaire exactement j clauses de Φ
- Nous avons

$$
\begin{aligned}
\frac{7m}{8} = \mathbb{E}(X) = \sum_{j=0}^{m} j \cdot p_j &= \sum_{0 \leq j < 7m/8} j \cdot p_j + \sum_{7m/8 \leq j < m} j \cdot p_j \\
&\leq \left(\frac{7m}{8} - \frac{1}{8}\right) \sum_{0 \leq j < 7m/8} p_j + m \sum_{7m/8 \leq j < m} p_j \\
&\leq \left(\frac{7m}{8} - \frac{1}{8}\right) + m \cdot p
\end{aligned}
$$

Probabilité de succès

Soient Φ une formule E3-SAT et S l'événement « une assignation des variables de Φ satisfait au moins $7m/8$ des clauses de Φ.
Nous avons $p = \Pr(S) \geq 1/8m$

Démonstration.

- p_j = probabilité de satisfaire exactement j clauses de Φ
- Nous avons

$$\frac{7m}{8} = \mathbb{E}(X) = \sum_{j=0}^{m} j \cdot p_j = \sum_{0 \leq j < 7m/8} j \cdot p_j + \sum_{7m/8 \leq j < m} j \cdot p_j$$

$$\leq \left(\frac{7m}{8} - \frac{1}{8}\right) \sum_{0 \leq j < 7m/8} p_j + m \sum_{7m/8 \leq j < m} p_j$$

$$\leq \left(\frac{7m}{8} - \frac{1}{8}\right) + m \cdot p$$

Probabilité de succès

Soient Φ une formule E3-SAT et S l'événement « une assignation des variables de Φ satisfait au moins $7m/8$ des clauses de Φ.
Nous avons $p = \Pr(S) \geq 1/8m$

Démonstration.

- $p_j =$ probabilité de satisfaire exactement j clauses de Φ
- Nous avons

$$\frac{7m}{8} = \mathbb{E}(X) = \sum_{j=0}^{m} j \cdot p_j = \sum_{0 \leq j < 7m/8} j \cdot p_j + \sum_{7m/8 \leq j < m} j \cdot p_j$$

$$\leq \left(\frac{7m}{8} - \frac{1}{8}\right) \sum_{0 \leq j < 7m/8} p_j + m \sum_{7m/8 \leq j < m} p_j$$

$$\leq \left(\frac{7m}{8} - \frac{1}{8}\right) + m \cdot p$$

Probabilité de succès

Soient Φ une formule E3-SAT et S l'événement « une assignation des variables de Φ satisfait au moins $7m/8$ des clauses de Φ.
Nous avons $p = \Pr(S) \geq 1/8m$

Démonstration.

- p_j = probabilité de satisfaire exactement j clauses de Φ
- Nous avons

$$\frac{7m}{8} = \mathbb{E}(X) = \sum_{j=0}^{m} j \cdot p_j = \sum_{0 \leq j < 7m/8} j \cdot p_j + \sum_{7m/8 \leq j < m} j \cdot p_j$$

$$\leq \left(\frac{7m}{8} - \frac{1}{8}\right) \sum_{0 \leq j < 7m/8} p_j + m \sum_{7m/8 \leq j < m} p_j$$

$$\leq \left(\frac{7m}{8} - \frac{1}{8}\right) + m \cdot p$$

Probabilité de succès

Soient Φ une formule E3-SAT et S l'événement « une assignation des variables de Φ satisfait au moins $7m/8$ des clauses de Φ.
Nous avons $p = \Pr(S) \geq 1/8m$

Démonstration.

- $p_j =$ probabilité de satisfaire exactement j clauses de Φ
- Nous avons

$$\frac{7m}{8} = \mathbb{E}(X) = \sum_{j=0}^{m} j \cdot p_j = \sum_{0 \leq j < 7m/8} j \cdot p_j + \sum_{7m/8 \leq j < m} j \cdot p_j$$

$$\leq \left(\frac{7m}{8} - \frac{1}{8}\right) \sum_{0 \leq j < 7m/8} p_j + m \sum_{7m/8 \leq j < m} p_j$$

$$\leq \left(\frac{7m}{8} - \frac{1}{8}\right) + m \cdot p$$

Probabilité de succès

Soient Φ une formule E3-SAT et S l'événement « une assignation des variables de Φ satisfait au moins $7m/8$ des clauses de Φ.
Nous avons $p = \Pr(S) \geq 1/8m$

Démonstration.

- $p_j =$ probabilité de satisfaire exactement j clauses de Φ
- Nous avons

$$\begin{aligned}
\frac{7m}{8} = \mathbb{E}(X) = \sum_{j=0}^{m} j \cdot p_j &= \sum_{0 \leq j < 7m/8} j \cdot p_j + \sum_{7m/8 \leq j < m} j \cdot p_j \\
&\leq \left(\frac{7m}{8} - \frac{1}{8}\right) \sum_{0 \leq j < 7m/8} p_j + m \sum_{7m/8 \leq j < m} p_j \\
&\leq \left(\frac{7m}{8} - \frac{1}{8}\right) + m \cdot p
\end{aligned}$$

Probabilité de succès

Soient Φ une formule E3-SAT et S l'événement « une assignation des variables de Φ satisfait au moins $7m/8$ des clauses de Φ.
Nous avons $p = \Pr(S) \geq 1/8m$

Démonstration.

- p_j = probabilité de satisfaire exactement j clauses de Φ
- Nous avons

$$\frac{7m}{8} = \mathbb{E}(X) = \sum_{j=0}^{m} j \cdot p_j = \sum_{0 \leq j < 7m/8} j \cdot p_j + \sum_{7m/8 \leq j < m} j \cdot p_j$$

$$\leq \left(\frac{7m}{8} - \frac{1}{8}\right) \sum_{0 \leq j < 7m/8} p_j + m \sum_{7m/8 \leq j < m} p_j$$

$$\leq \left(\frac{7m}{8} - \frac{1}{8}\right) + m \cdot p$$

Algorithme probabiliste pour MAX-E3-SAT

Entrée: Φ formule booléenne MAX-E3-SAT en n variables et m clauses
Sortie: $y \in \{0,1\}$ tel que l'assignation y satisfait $\lceil 7m/8 \rceil$ clauses de Φ

tant que vrai **faire**
  $y = (y_1, \ldots, y_n) \xleftarrow{\quad} \{0,1\}^n$
  si au moins $\lceil 7m/8 \rceil$ clauses de Φ sont satisfaites par y **alors**
    retourner y
  fin si
fin tant que

Complexité en moyenne : $O(m^2)$ $\triangleright$ $8m$ répétitions en moyenne

Probabilité d'erreur : 0

Algorithme de type Las Vegas

Algorithme probabiliste pour MAX-E3-SAT

Entrée: Φ formule booléenne MAX-E3-SAT en n variables et m clauses
Sortie: $y \in \{0,1\}$ tel que l'assignation y satisfait $\lceil 7m/8 \rceil$ clauses de Φ

tant que vrai **faire**
  $y = (y_1, \ldots, y_n) \xleftarrow{\text{⚂⚃}} \{0,1\}^n$
  si au moins $\lceil 7m/8 \rceil$ clauses de Φ sont satisfaites par y **alors**
    retourner y
  fin si
fin tant que

Complexité en moyenne : $O(m^2)$ $\triangleright$ $8m$ répétitions en moyenne

Probabilité d'erreur : 0

Algorithme de type Las Vegas

Algorithme probabiliste pour MAX-E3-SAT

Entrée: Φ formule booléenne MAX-E3-SAT en n variables et m clauses
Sortie: $y \in \{0,1\}$ tel que l'assignation y satisfait $\lceil 7m/8 \rceil$ clauses de Φ

tant que vrai **faire**
  $y = (y_1, \ldots, y_n) \xleftarrow{\text{⚂⚃}} \{0,1\}^n$
  si au moins $\lceil 7m/8 \rceil$ clauses de Φ sont satisfaites par y **alors**
    retourner y
  fin si
fin tant que

Complexité en moyenne : $O(m^2)$ $\quad \triangleright$ $8m$ répétitions en moyenne

Probabilité d'erreur : 0

Algorithme de type Las Vegas

Pour aller plus loin . . .

- Extensions possibles
  - Algorithme probabiliste d'approximation pour MAX-3-SAT (et non pas MAX-E3-SAT) – (*cf.* TD)
  - Algorithme probabiliste avec des poids sur les clauses

- 1997, H. J. Karloff, U. Zwick
  $\rightsquigarrow$ Il existe un algorithme probabiliste polynomial de 7/8-approximation pour MAX-3-SAT.

- 1997, J. Håstad
  $\rightsquigarrow$ Si $\mathcal{P} \neq \mathcal{NP}$, il n'existe pas d'algorithme probabiliste polynomial de c-approximation avec $c > 7/8$ pour MAX-3-SAT.

Pour aller plus loin . . .

- Extensions possibles
  - Algorithme probabiliste d'approximation pour MAX-3-SAT (et non pas MAX-E3-SAT) – (*cf.* TD)
  - Algorithme probabiliste avec des poids sur les clauses

- 1997, H. J. Karloff, U. Zwick
  $\rightsquigarrow$ Il existe un algorithme probabiliste polynomial de $7/8$-approximation pour MAX-3-SAT.

- 1997, J. Håstad
  $\rightsquigarrow$ Si $\mathcal{P} \neq \mathcal{NP}$, il n'existe pas d'algorithme probabiliste polynomial de c-approximation avec $c > 7/8$ pour MAX-3-SAT.

Pour aller plus loin ...

- Extensions possibles
  - Algorithme probabiliste d'approximation pour MAX-3-SAT (et non pas MAX-E3-SAT) – (*cf.* TD)
  - Algorithme probabiliste avec des poids sur les clauses

- 1997, H. J. Karloff, U. Zwick
  $\rightsquigarrow$ Il existe un algorithme probabiliste polynomial de $7/8$-approximation pour MAX-3-SAT.

- 1997, J. Håstad
  $\rightsquigarrow$ Si $\mathcal{P} \neq \mathcal{NP}$, il n'existe pas d'algorithme probabiliste polynomial de c-approximation avec $c > 7/8$ pour MAX-3-SAT.

Dérandomisation

- L'aléatoire permet d'obtenir des algorithmes parfois plus efficaces, souvent plus simples mais l'aléa n'est **pas gratuit**
- Éliminer (ou limiter) l'aléatoire dans un algorithme probabibiliste est un sujet de recherche important appelé dérandomisation
- L'algorithme de Johnson de 1974 est une version dérandomisée de l'algorithme que nous venons de voir
  - repose sur la même idée que la **méthode probabiliste**
  - c'est la méthode des espérances conditionnelles

Dérandomisation

- L'aléatoire permet d'obtenir des algorithmes parfois plus efficaces, souvent plus simples mais l'aléa n'est **pas gratuit**
- Éliminer (ou limiter) l'aléatoire dans un algorithme probabibiliste est un sujet de recherche important appelé dérandomisation
- L'algorithme de Johnson de 1974 est une version dérandomisée de l'algorithme que nous venons de voir
  - repose sur la même idée que la **méthode probabiliste**
  - c'est la méthode des espérances conditionnelles

Dérandomisation

- L'aléatoire permet d'obtenir des algorithmes parfois plus efficaces, souvent plus simples mais l'aléa n'est **pas gratuit**
- Éliminer (ou limiter) l'aléatoire dans un algorithme probabibiliste est un sujet de recherche important appelé dérandomisation
- L'algorithme de Johnson de 1974 est une version dérandomisée de l'algorithme que nous venons de voir
  - repose sur la même idée que la **méthode probabiliste**
  - c'est la méthode des espérances conditionnelles

Méthode des espérances conditionnelles (1/3)

- Soit Φ une formule E3-SAT en n variables $x_1, \ldots, x_n$
  Soit X la variable aléatoire du nombres de clauses de Φ satisfaites par une assignation aléatoire des variables.

- Par la formule des probabilités conditionnelles ($k \in \{0, \ldots, m\}$)

$$\begin{aligned}
&\Pr(X = k) \\
=&\Pr(x_1 = 0) \cdot \Pr(X = k | x_1 = 0) + \Pr(x_1 = 1) \cdot \Pr(X = k | x_1 = 1) \\
=&\frac{1}{2} \cdot \Pr(X = k | x_1 = 0) + \frac{1}{2} \cdot \Pr(X = k | x_1 = 1)
\end{aligned}$$

- Nous obtenons pour l'espérance

$$\mathbb{E}(X) = \frac{1}{2}\mathbb{E}(X | x_1 = 0) + \frac{1}{2}\mathbb{E}(X | x_1 = 1)$$

Méthode des espérances conditionnelles (2/3)

$$\mathbb{E}(X) = \frac{1}{2}\mathbb{E}(X|x_1 = 0) + \frac{1}{2}\mathbb{E}(X|x_1 = 1)$$

- Nous avons donc

$$\mathbb{E}(X|x_1 = 0) \geq \mathbb{E}(X) \text{ ou } \mathbb{E}(X|x_1 = 1) \geq \mathbb{E}(X)$$

  (les deux sont possibles)

- **Algorithme déterministe**
  - choisir $x_1 \in \{0, 1\}$ pour lequel l'espérance conditionnelle est maximale $\rightsquigarrow x_1 = b$
  - appliquer récursivement cette idée à la formule $\Phi' = \Phi_{|x_1=b}$

Méthode des espérances conditionnelles (3/3)

- $\ell_i \vee \ell_j \vee \ell_k$ (avec $1 \notin \{i,j,k\}$)
  $\rightsquigarrow$ la clause est satisfaite avec probabilité $7/8$
- $x_1 \vee \ell_j \vee \ell_k$ (avec $1 \notin \{j,k\}$)
  - $x_1 = 1 \rightsquigarrow$ la clause est satisfaite avec probabilité 1
  - $x_1 = 0 \rightsquigarrow \ell_j \vee \ell_k \rightsquigarrow$ la clause est satisfaite avec probabilité $3/4$
- $\neg x_1 \vee \ell_j \vee \ell_k$ (avec $1 \notin \{j,k\}$)
  - $x_1 = 1 \rightsquigarrow \ell_j \vee \ell_k \rightsquigarrow$ la clause est satisfaite avec probabilité $3/4$
  - $x_1 = 0 \rightsquigarrow$ la clause est satisfaite avec probabilité 1

Le calcul de $\mathbb{E}(X|x_1 = 0)$ et $\mathbb{E}(X|x_1 = 1)$ peut donc se faire en $O(m)$ opérations élémentaires

Complexité : $O(m \cdot n)$

Méthode des espérances conditionnelles (3/3)

- $\ell_i \vee \ell_j \vee \ell_k$ (avec $1 \notin \{i,j,k\}$)
  $\rightsquigarrow$ la clause est satisfaite avec probabilité $7/8$
- $x_1 \vee \ell_j \vee \ell_k$ (avec $1 \notin \{j,k\}$)
  - $x_1 = 1 \rightsquigarrow$ la clause est satisfaite avec probabilité 1
  - $x_1 = 0 \rightsquigarrow \ell_j \vee \ell_k \rightsquigarrow$ la clause est satisfaite avec probabilité $3/4$
- $\neg x_1 \vee \ell_j \vee \ell_k$ (avec $1 \notin \{j,k\}$)
  - $x_1 = 1 \rightsquigarrow \ell_j \vee \ell_k \rightsquigarrow$ la clause est satisfaite avec probabilité $3/4$
  - $x_1 = 0 \rightsquigarrow$ la clause est satisfaite avec probabilité 1

Le calcul de $\mathbb{E}(X|x_1 = 0)$ et $\mathbb{E}(X|x_1 = 1)$ peut donc se faire en $O(m)$ opérations élémentaires

Complexité : $O(m \cdot n)$

Méthode des espérances conditionnelles (3/3)

- $\ell_i \vee \ell_j \vee \ell_k$ (avec $1 \notin \{i, j, k\}$)
  $\rightsquigarrow$ la clause est satisfaite avec probabilité $7/8$
- $x_1 \vee \ell_j \vee \ell_k$ (avec $1 \notin \{j, k\}$)
  - $x_1 = 1 \rightsquigarrow$ la clause est satisfaite avec probabilité 1
  - $x_1 = 0 \rightsquigarrow \ell_j \vee \ell_k \rightsquigarrow$ la clause est satisfaite avec probabilité $3/4$
- $\neg x_1 \vee \ell_j \vee \ell_k$ (avec $1 \notin \{j, k\}$)
  - $x_1 = 1 \rightsquigarrow \ell_j \vee \ell_k \rightsquigarrow$ la clause est satisfaite avec probabilité $3/4$
  - $x_1 = 0 \rightsquigarrow$ la clause est satisfaite avec probabilité 1

Le calcul de $\mathbb{E}(X|x_1 = 0)$ et $\mathbb{E}(X|x_1 = 1)$ peut donc se faire en $O(m)$ opérations élémentaires

Complexité : $O(m \cdot n)$

Méthode des espérances conditionnelles (3/3)

- $\ell_i \vee \ell_j \vee \ell_k$ (avec $1 \notin \{i, j, k\}$)
  $\leadsto$ la clause est satisfaite avec probabilité $7/8$
- $x_1 \vee \ell_j \vee \ell_k$ (avec $1 \notin \{j, k\}$)
  - $x_1 = 1 \leadsto$ la clause est satisfaite avec probabilité 1
  - $x_1 = 0 \leadsto \ell_j \vee \ell_k \leadsto$ la clause est satisfaite avec probabilité $3/4$
- $\neg x_1 \vee \ell_j \vee \ell_k$ (avec $1 \notin \{j, k\}$)
  - $x_1 = 1 \leadsto \ell_j \vee \ell_k \leadsto$ la clause est satisfaite avec probabilité $3/4$
  - $x_1 = 0 \leadsto$ la clause est satisfaite avec probabilité 1

Le calcul de $\mathbb{E}(X|x_1 = 0)$ et $\mathbb{E}(X|x_1 = 1)$ peut donc se faire en $O(m)$ opérations élémentaires

Complexité : $O(m \cdot n)$

Méthode des espérances conditionnelles (3/3)

- $\ell_i \vee \ell_j \vee \ell_k$ (avec $1 \notin \{i, j, k\}$)
  $\rightsquigarrow$ la clause est satisfaite avec probabilité $7/8$
- $x_1 \vee \ell_j \vee \ell_k$ (avec $1 \notin \{j, k\}$)
  - $x_1 = 1 \rightsquigarrow$ la clause est satisfaite avec probabilité 1
  - $x_1 = 0 \rightsquigarrow \ell_j \vee \ell_k \rightsquigarrow$ la clause est satisfaite avec probabilité $3/4$
- $\neg x_1 \vee \ell_j \vee \ell_k$ (avec $1 \notin \{j, k\}$)
  - $x_1 = 1 \rightsquigarrow \ell_j \vee \ell_k \rightsquigarrow$ la clause est satisfaite avec probabilité $3/4$
  - $x_1 = 0 \rightsquigarrow$ la clause est satisfaite avec probabilité 1

Le calcul de $\mathbb{E}(X|x_1 = 0)$ et $\mathbb{E}(X|x_1 = 1)$ peut donc se faire en $O(m)$ opérations élémentaires

Complexité : $O(m \cdot n)$

Table des matières

Problème 2-SAT

Problème 2-SAT

- ENTRÉE : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec au plus 2 littéraux
- SORTIE : VRAI si il existe une assignation des variables, qui rend la formule vraie (et FAUX sinon)

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- problème de décision
- problème dans $\mathcal{P}$ (même linéaire !)
- 1991 – C. H. Papadimitriou
  $\rightsquigarrow$ algorithme probabiliste cubique par marche aléatoire

Algorithme de Papadimitriou

Entrée: Formule Φ de m clauses $C_1, \ldots, C_m$ en n variables $x_1, \ldots, x_n$: $T \in \mathbb{N}$

Sortie: $\textsc{Satisfiable}$ ou $\textsc{Non satisfiable}$

$y = (y_1, \ldots, y_n) \longleftarrow \{0,1\}^n$

pour i de 1 à T **faire**

  si $\Phi(y) = 0$ **alors**

    considérer une clause C_t (arbitraire) non satisfaite de Φ

    $j \longleftarrow \textsc{Variables}(C_t)$ ▷ tirage d'une variable dans C_t

    $y_j \leftarrow 1 - y_j$ ▷ changement de la valeur de y_j

  sinon

    retourner $\textsc{Satisfiable}$

  fin si

fin pour

retourner $\textsc{Non satisfiable}$

Algorithme de Papadimitriou

- L'algorithme termine toujours après T répétition de la boucle **pour . . . faire**

Complexité : $O(Tm)$

- Si l'algorithme retourne SATISFIABLE, alors Φ est satisfiable
- Si l'algorithme retourne NON SATISFIABLE, alors Φ est peut-être satisfiable

Probabilité d'erreur ?? (dépend notamment de T)

Algorithme de type Monte-Carlo

Algorithme de Papadimitriou

- L'algorithme termine toujours après T répétition de la boucle **pour ... faire**

Complexité : $O(Tm)$

- Si l'algorithme retourne $\textsc{Satisfiable}$, alors Φ est satisfiable
- Si l'algorithme retourne $\textsc{Non satisfiable}$, alors Φ est peut-être satisfiable

Probabilité d'erreur ?? (dépend notamment de T)

Algorithme de type Monte-Carlo

Algorithme de Papadimitriou

- L'algorithme termine toujours après T répétition de la boucle **pour ... faire**

Complexité : $O(Tm)$

- Si l'algorithme retourne Satisfiable, alors Φ est satisfiable
- Si l'algorithme retourne Non satisfiable, alors Φ est peut-être satisfiable

Probabilité d'erreur ?? (dépend notamment de T)

Algorithme de type Monte-Carlo

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge$$
$$(x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge$$
$$(x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0)$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$
- $(1, 1, 0, 0, 0, 1, 0)$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge$$
$$(x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$
- $(1, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_2 \vee \neg x_5)$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge$$
$$(x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$
- $(1, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_2 \vee \neg x_5)$ $\rightsquigarrow x_5$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$
- $(1, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_2 \vee \neg x_5)$ $\rightsquigarrow x_5$
- $(1, 1, 0, 0, 0, 0, 0)$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$
- $(1, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_2 \vee \neg x_5)$ $\rightsquigarrow x_5$
- $(1, 1, 0, 0, 0, 0, 0) \rightsquigarrow (x_3 \vee x_6)$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$
- $(1, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_2 \vee \neg x_5)$ $\rightsquigarrow x_5$
- $(1, 1, 0, 0, 0, 0, 0) \rightsquigarrow (x_3 \vee x_6)$ $\rightsquigarrow x_3$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$
- $(1, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_2 \vee \neg x_5)$ $\rightsquigarrow x_5$
- $(1, 1, 0, 0, 0, 0, 0) \rightsquigarrow (x_3 \vee x_6)$ $\rightsquigarrow x_3$
- $(1, 1, 0, 1, 0, 0, 0)$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$
- $(1, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_2 \vee \neg x_5)$ $\rightsquigarrow x_5$
- $(1, 1, 0, 0, 0, 0, 0) \rightsquigarrow (x_3 \vee x_6)$ $\rightsquigarrow x_3$
- $(1, 1, 0, 1, 0, 0, 0) \rightsquigarrow (x_4 \vee x_6)$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge$$
$$(x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$
- $(1, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_2 \vee \neg x_5)$ $\rightsquigarrow x_5$
- $(1, 1, 0, 0, 0, 0, 0) \rightsquigarrow (x_3 \vee x_6)$ $\rightsquigarrow x_3$
- $(1, 1, 0, 1, 0, 0, 0) \rightsquigarrow (x_4 \vee x_6)$ $\rightsquigarrow x_6$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0, 1, 0, 0, 0, 1, 0)$
- $(0, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$
- $(1, 1, 0, 0, 0, 1, 0) \rightsquigarrow (x_2 \vee \neg x_5)$ $\rightsquigarrow x_5$
- $(1, 1, 0, 0, 0, 0, 0) \rightsquigarrow (x_3 \vee x_6)$ $\rightsquigarrow x_3$
- $(1, 1, 0, 1, 0, 0, 0) \rightsquigarrow (x_4 \vee x_6)$ $\rightsquigarrow x_6$
- $(1, 1, 0, 1, 0, 0, 1)$

Exemple d'exécution

$$(x_0 \vee x_2) \wedge (x_0 \vee \neg x_3) \wedge (x_1 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_0 \vee \neg x_5) \wedge (x_1 \vee \neg x_5) \wedge (x_2 \vee \neg x_5) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6).$$

- Tirage initial : $(0,1,0,0,0,1,0)$
- $(0,1,0,0,0,1,0) \rightsquigarrow (x_0 \vee x_2)$ $\rightsquigarrow x_0$
- $(1,1,0,0,0,1,0) \rightsquigarrow (x_2 \vee \neg x_5)$ $\rightsquigarrow x_5$
- $(1,1,0,0,0,0,0) \rightsquigarrow (x_3 \vee x_6)$ $\rightsquigarrow x_3$
- $(1,1,0,1,0,0,0) \rightsquigarrow (x_4 \vee x_6)$ $\rightsquigarrow x_6$
- $(1,1,0,1,0,0,1)$ $\rightsquigarrow$ SATISFIABLE

Marche aléatoire (1/2)

Supposons que Φ est satisfiable ; soit $y^\star$ tel que $\Phi(y^\star) = 1$
Notons $M(y, y^\star)$ le nombre de coordonnées de y et $y^\star$ égales

- Si $M(y, y^\star) = n$, $\Phi(y) = \Phi(y^\star) = 1$ et l'algorithme s'arrête !
- Si $\Phi(y) = 0$, alors $M(y, y^\star) < n$
  - au moins une clause n'est pas satisfaite $\rightsquigarrow C_t = \ell_i \vee \ell_j$
  - avec y, $\ell_i = 0$ et $\ell_j = 0$
  - avec $y^\star$, $\ell_i = 1$ ou $\ell_j = 1$
  - l'algorithme modifie y en changeant aléatoirement y_i ou y_j
    $M(y, y^\star)$ augmente de 1 avec probabilité $\geq 1/2$
    $M(y, y^\star)$ diminue de 1 avec probabilité $\leq 1/2$

Marche aléatoire (1/2)

Supposons que Φ est satisfiable ; soit $y^\star$ tel que $\Phi(y^\star) = 1$
Notons $M(y, y^\star)$ le nombre de coordonnées de y et $y^\star$ égales

- Si $M(y, y^\star) = n$, $\Phi(y) = \Phi(y^\star) = 1$ et l'algorithme s'arrête !
- Si $\Phi(y) = 0$, alors $M(y, y^\star) < n$
  - au moins une clause n'est pas satisfaite $\rightsquigarrow C_t = \ell_i \vee \ell_j$
  - avec y, $\ell_i = 0$ et $\ell_j = 0$
  - avec $y^\star$, $\ell_i = 1$ ou $\ell_j = 1$
  - l'algorithme modifie y en changeant aléatoirement y_i ou y_j
    $M(y, y^\star)$ augmente de 1 avec probabilité $\geq 1/2$
    $M(y, y^\star)$ diminue de 1 avec probabilité $\leq 1/2$

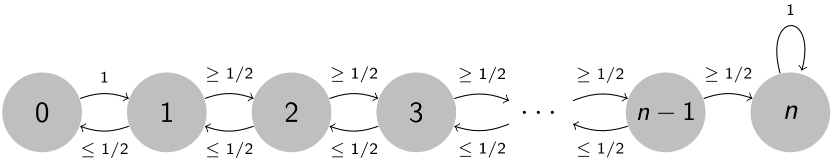
Marche aléatoire (1/2)

Supposons que Φ est satisfiable ; soit $y^\star$ tel que $\Phi(y^\star) = 1$
Notons $M(y, y^\star)$ le nombre de coordonnées de y et $y^\star$ égales

- Si $M(y, y^\star) = n$, $\Phi(y) = \Phi(y^\star) = 1$ et l'algorithme s'arrête !
- Si $\Phi(y) = 0$, alors $M(y, y^\star) < n$
  - au moins une clause n'est pas satisfaite $\rightsquigarrow C_t = \ell_i \vee \ell_j$
  - avec y, $\ell_i = 0$ et $\ell_j = 0$
  - avec $y^\star$, $\ell_i = 1$ ou $\ell_j = 1$
  - l'algorithme modifie y en changeant aléatoirement y_i ou y_j
    $M(y, y^\star)$ augmente de 1 avec probabilité $\geq 1/2$
    $M(y, y^\star)$ diminue de 1 avec probabilité $\leq 1/2$

Marche aléatoire (2/2)

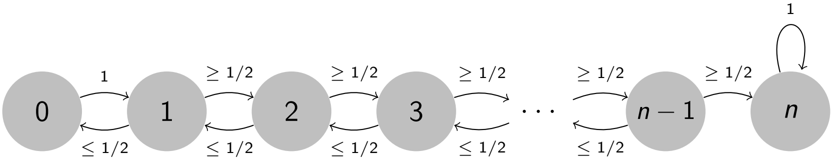
- Nous interprétons le comportement de l'algorithme comme une marche aléatoire dans un graphe



- Au départ $M(y, y^\star)$ prend n'importe quelle valeur de $\{0, \ldots, n\}$
- À chaque étape $M(y, y^\star)$ varie de ± 1
- **Combien de temps pour atteindre le sommet n ?**

Marche aléatoire (2/2)

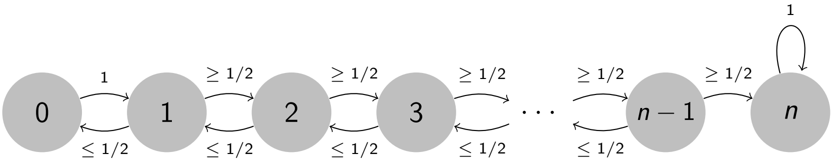
- Nous interprétons le comportement de l'algorithme comme une marche aléatoire dans un graphe



- Au départ $M(y, y^{\star})$ prend n'importe quelle valeur de $\{0, \ldots, n\}$
- À chaque étape $M(y, y^{\star})$ varie de ± 1
- **Combien de temps pour atteindre le sommet n ?**

Marche aléatoire (2/2)

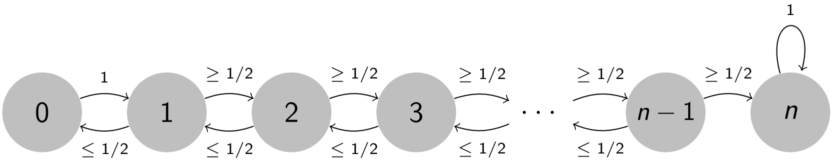
- Nous interprétons le comportement de l'algorithme comme une marche aléatoire dans un graphe



- Au départ $M(y, y^\star)$ prend n'importe quelle valeur de $\{0, \ldots, n\}$
- À chaque étape $M(y, y^\star)$ varie de ± 1
- Combien de temps pour atteindre le sommet n ?

Marche aléatoire (2/2)

- Nous interprétons le comportement de l'algorithme comme une marche aléatoire dans un graphe



- Au départ $M(y, y^\star)$ prend n'importe quelle valeur de $\{0, \ldots, n\}$
- À chaque étape $M(y, y^\star)$ varie de ± 1
- **Combien de temps pour atteindre le sommet n ?**

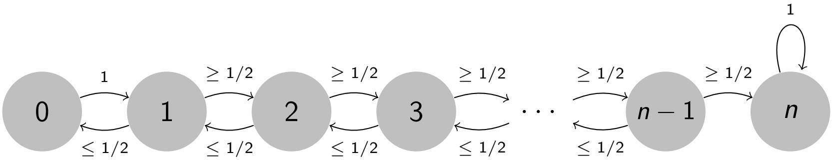
Algorithme de Papadimitriou modifié

Modification formelle de l'algorithme

$y = (y_1, \dots, y_n) \xleftarrow{} \{0,1\}^n$
tant que vrai **faire**
  si $y \neq y^\star$ **alors**
    considérer une clause C_t (arbitraire) non satisfaite de Φ
    $j \xleftarrow{} \textsc{Variables}(C_t)$ $\qquad \triangleright$ tirage d'une variable dans C_t
    $y_j \leftarrow 1 - y_j$ $\qquad \triangleright$ changement de la valeur de y_j
  fin si
fin tant que

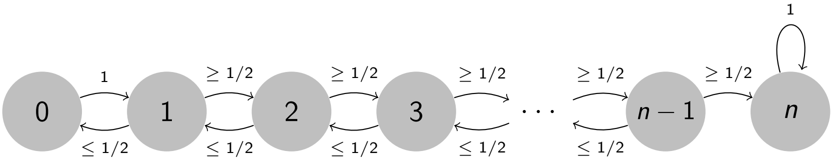
- Si cet « algorithme » s'arrête en moins de T étapes, l'algorithme de Papadimitriou retourne SATISFIABLE

Analyse (1/6)



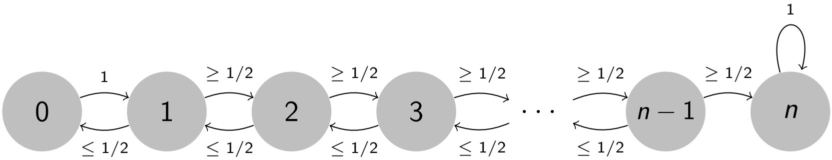
- Z_j = v.a. du nombre d'étapes pour atteindre n à partir de j
- $Z_n = 0$; $Z_0 = 1 + Z_1$
- $Z_i \geq Z_j$ pour $i \leq j$
- **Intuition :** « $Z_j = 1 + Z_{j+1}$ avec probabilité $\geq 1/2$ »
  « $Z_j = 1 + Z_{j-1}$ avec probabilité $\leq 1/2$ »

Analyse (1/6)



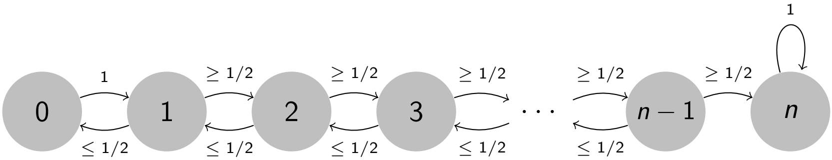
- Z_j = v.a. du nombre d'étapes pour atteindre n à partir de j
- $Z_n = 0$; $Z_0 = 1 + Z_1$
- $Z_i \geq Z_j$ pour $i \leq j$
- **Intuition :** « $Z_j = 1 + Z_{j+1}$ avec probabilité $\geq 1/2$ »
  « $Z_j = 1 + Z_{j-1}$ avec probabilité $\leq 1/2$ »

Analyse (1/6)



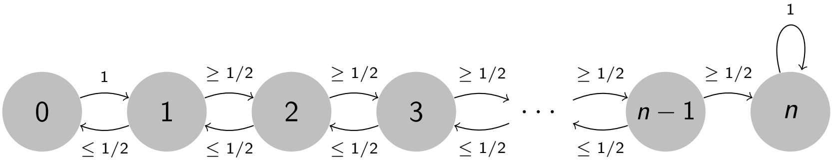
- Z_j = v.a. du nombre d'étapes pour atteindre n à partir de j
- $Z_n = 0$; $Z_0 = 1 + Z_1$
- $Z_i \geq Z_j$ pour $i \leq j$
- **Intuition :** « $Z_j = 1 + Z_{j+1}$ avec probabilité $\geq 1/2$ »
  « $Z_j = 1 + Z_{j-1}$ avec probabilité $\leq 1/2$ »

Analyse (1/6)



- Z_j = v.a. du nombre d'étapes pour atteindre n à partir de j
- $Z_n = 0$; $Z_0 = 1 + Z_1$
- $Z_i \geq Z_j$ pour $i \leq j$
- **Intuition :** « $Z_j = 1 + Z_{j+1}$ avec probabilité $\geq 1/2$ »
  « $Z_j = 1 + Z_{j-1}$ avec probabilité $\leq 1/2$ »

Analyse (2/6)



- Posons $h_j = \mathbb{E}(Z_j)$. Nous allons montrer $h_j \leq n^2$
- $h_n = \mathbb{E}(Z_n) = 0$
- $h_0 = \mathbb{E}(Z_0) = \mathbb{E}(1 + Z_1) = 1 + h_1$
- Notons p la probabilité de passer de j à $j+1$ (avec $p \geq 1/2$) :

$$h_j = \mathbb{E}(Z_j) = p \cdot \mathbb{E}(1 + Z_{j+1}) + (1-p) \cdot \mathbb{E}(1 + Z_{j-1})$$

Analyse (3/6)

$$h_j = \mathbb{E}(Z_j) = p \cdot (1 + h_{j+1}) + (1-p) \cdot (1 + h_{j-1}) \text{ avec } p \geq 1/2$$

- $Z_i \geq Z_j$ pour $i \leq j \rightsquigarrow Z_{j+1} \leq Z_{j-1} \rightsquigarrow h_{j+1} \leq h_{j-1}$
- L'expression est minimale pour $p = 1/2$ et

$$h_j \leq \frac{1}{2}(1 + h_{j+1}) + \frac{1}{2} \cdot (1 + h_{j-1}) = 1 + \frac{1}{2}h_{j+1} + \frac{1}{2}h_{j-1}$$

- Par récurrence, pour $j \in \{0, \dots, n-1\}$

$$h_j \leq h_{j+1} + 2j + 1$$

Analyse (3/6)

$$h_j = \mathbb{E}(Z_j) = p \cdot (1 + h_{j+1}) + (1 - p) \cdot (1 + h_{j-1}) \text{ avec } p \geq 1/2$$

- $Z_i \geq Z_j$ pour $i \leq j \rightsquigarrow Z_{j+1} \leq Z_{j-1} \rightsquigarrow h_{j+1} \leq h_{j-1}$
- L'expression est minimale pour $p = 1/2$ et

$$h_j \leq \frac{1}{2}(1 + h_{j+1}) + \frac{1}{2} \cdot (1 + h_{j-1}) = 1 + \frac{1}{2} h_{j+1} + \frac{1}{2} h_{j-1}$$

- Par récurrence, pour $j \in \{0, \ldots, n-1\}$

$$h_j \leq h_{j+1} + 2j + 1$$

Analyse (3/6)

$$h_j = \mathbb{E}(Z_j) = p \cdot (1 + h_{j+1}) + (1-p) \cdot (1 + h_{j-1}) \text{ avec } p \geq 1/2$$

- $Z_i \geq Z_j$ pour $i \leq j \rightsquigarrow Z_{j+1} \leq Z_{j-1} \rightsquigarrow h_{j+1} \leq h_{j-1}$
- L'expression est minimale pour $p = 1/2$ et

$$h_j \leq \frac{1}{2}(1 + h_{j+1}) + \frac{1}{2} \cdot (1 + h_{j-1}) = 1 + \frac{1}{2} h_{j+1} + \frac{1}{2} h_{j-1}$$

- Par récurrence, pour $j \in \{0, \ldots, n-1\}$

$$h_j \leq h_{j+1} + 2j + 1$$

Analyse (4/6)

$$\boxed{h_j \leq h_{j+1} + 2j + 1}$$

En effet,

- $h_0 = 1 + h_1 = h_1 + 2 \cdot 0 + 1$
- Par l'hypothèse de récurrence

$$\begin{aligned} h_j \leq 1 + \frac{1}{2}h_{j+1} + \frac{1}{2}h_{j-1} &\leq 1 + \frac{1}{2}h_{j+1} + \frac{1}{2}(h_j + 2(j-1) + 1) \\ &= 1 + \frac{1}{2}h_{j+1} + \frac{1}{2}h_j + (j-1) + \frac{1}{2} \end{aligned}$$

et

$$h_j - \frac{1}{2}h_j \leq \frac{1}{2}h_{j+1} + j + \frac{1}{2}$$

Analyse (5/6)

- Nous avons $h_n = 0$ et pour $j \in \{0, \dots, n-1\}$
$$h_j \leq h_{j+1} + 2j + 1$$
donc
$$\begin{aligned} h_j &\leq h_{j+1} + 2j + 1 \\ &\leq h_{j+2} + (2(j+1)+1) + (2j+1) \\ &\leq h_{j+3} + (2(j+2)+1) + (2(j+1)+1) + (2j+1) \\ &\leq \dots \\ &\leq h_n + (2(n-1)+1) + \cdots + (2j+1) \end{aligned}$$
et
$$h_j \leq \sum_{k=j}^{n-1} (2k+1) \leq \sum_{k=0}^{n-1} (2k+1) = n + n(n-1) = n^2$$

Analyse (6/6)

- Nous avons pour $j \in \{0, \ldots, n\}$

$$\mathbb{E}(Z_j) = h_j \leq n^2$$

- L'algorithme modifié termine donc en n^2 répétitions de la boucle **tant que . . . faire**

- **Comment fixer T dans l'algorithme de Papadimitriou pour assurer une bonne probabilité de succès ?**

  Inégalité de Markov !

Analyse (6/6)

- Nous avons pour $j \in \{0, \ldots, n\}$

$$\mathbb{E}(Z_j) = h_j \leq n^2$$

- L'algorithme modifié termine donc en n^2 répétitions de la boucle **tant que ... faire**

- **Comment fixer T dans l'algorithme de Papadimitriou pour assurer une bonne probabilité de succès ?**

Inégalité de Markov !

Analyse (6/6)

- Nous avons pour $j \in \{0, \ldots, n\}$

$$\mathbb{E}(Z_j) = h_j \leq n^2$$

- L'algorithme modifié termine donc en n^2 répétitions de la boucle **tant que … faire**

- **Comment fixer T dans l'algorithme de Papadimitriou pour assurer une bonne probabilité de succès ?**

  Inégalité de Markov !

Inégalité de Markov

Inégalité de Markov

Soit Z une variable aléatoire réelle positive.

$$\forall a > 0, \qquad \Pr(Z \geqslant a) \leqslant \frac{\mathbb{E}(Z)}{a}.$$

Exemples.

- Loi géométrique $\mathbb{E}(\mathbb{Y}) = 1/p \rightsquigarrow \Pr\left(Z \geqslant \frac{2}{p}\right) \leqslant \frac{1}{2}$
- Alg. de Papadimitriou $\mathbb{E}(Z_j) \leq n^2 \rightsquigarrow \Pr\left(Z_j \geqslant 2 \cdot n^2\right) \leqslant \frac{1}{2}$

Algorithme de Papadimitriou

- L'algorithme termine toujours après $T = 2n^2$ répétition de la boucle **pour . . . faire**

Complexité : $O(n^2 \cdot m)$

- Si l'algorithme retourne Satisfiable, alors Φ est satisfiable
- Si l'algorithme retourne Non satisfiable, alors Φ est peut-être satisfiable

Probabilité d'erreur $\leq 1/2$

Algorithme de type Monte Carlo

Algorithme de Papadimitriou – Amplification

Entrée: Formule Φ de m clauses $C_1, \dots, C_m$ en n variables $x_1, \dots, x_n$
Sortie: SATISFIABLE ou NON SATISFIABLE

  pour j de 1 à $T_0 = \lceil \log(m) \rceil$ **faire**
    $y = (y_1, \dots, y_n) \xleftarrow{\text{⚂⚃}} \{0,1\}^n$
    pour i de 1 à $T_1 = 2n^2$ **faire**
      si $\Phi(y) = 0$ **alors**
        considérer une clause C_t (arbitraire) non satisfaite de Φ
        $j \xleftarrow{\text{⚂⚃}} \textsc{Variables}(C_t)$ $\triangleright$ tirage d'une variable dans C_t
        $y_j \leftarrow 1 - y_j$ $\triangleright$ changement de la valeur de y_j
      sinon
        retourner SATISFIABLE
      fin si
    fin pour
  fin pour
  retourner NON SATISFIABLE

Algorithme de Papadimitriou – Amplification

- L'algorithme termine toujours après $T_0 \times T_1 = 2n^2 \log(m)$ répétition de la boucle **pour . . . faire**

Complexité : $O(n^2 \cdot m \cdot \log(m))$

- Si l'algorithme retourne SATISFIABLE, alors Φ est satisfiable
- Si l'algorithme retourne NON SATISFIABLE, alors Φ est peut-être satisfiable

Probabilité d'erreur $\leq 1/m$

Algorithme de type Monte Carlo

Table des matières

Problème 3-SAT

Problème 3-SAT

- ENTRÉE : une formule booléenne en n variables sous forme d'une conjonction de m clauses avec au plus 3 littéraux
- SORTIE : VRAI si il existe une assignation des variables, qui rend la formule vraie (et FAUX sinon)

- problème de décision
- « Le » problème $\mathcal{NP}$-difficile
- 1999 – U. Schöning
  $\rightsquigarrow$ algorithme probabiliste exponentiel par marche aléatoire

Algorithme de Schöning

Entrée: Formule Φ de m clauses $C_1, \ldots, C_m$ en n variables $x_1, \ldots, x_n$: $T_0, T_1 \in \mathbb{N}$

Sortie: $\textsc{Satisfiable}$ ou $\textsc{Non satisfiable}$

pour j de 1 à T_0 **faire**

  $y = (y_1, \ldots, y_n) \xleftarrow{⚂⚃} \{0,1\}^n$

  pour i de 1 à T_1 **faire**

    si $\Phi(y) = 0$ **alors**

      considérer une clause C_t (arbitraire) non satisfaite de Φ

      $j \xleftarrow{⚂⚃} \textsc{Variables}(C_t)$ ▷ tirage d'une variable dans C_t

      $y_j \leftarrow 1 - y_j$ ▷ changement de la valeur de y_j

    sinon

      retourner $\textsc{Satisfiable}$

    fin si

  fin pour

fin pour

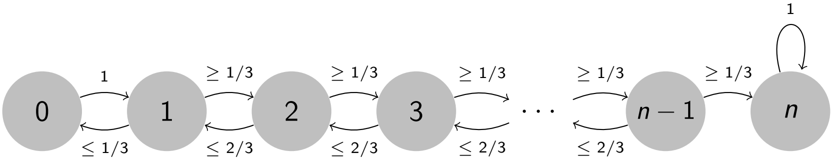
Marche aléatoire (1/2)

Supposons que Φ est satisfiable ; soit $y^\star$ tel que $\Phi(y^\star) = 1$
Notons $M(y, y^\star)$ le nombre de coordonnées de y et $y^\star$ égales

- Si $M(y, y^\star) = n$, $\Phi(y) = \Phi(y^\star) = 1$ et l'algorithme s'arrête !
- Si $\Phi(y) = 0$, alors $M(y, y^\star) < n$
  - au moins une clause n'est pas satisfaite $\rightsquigarrow$ $C_t = \ell_i \vee \ell_j \vee \ell_k$
  - avec y, $\ell_i = 0$ et $\ell_j = 0$ et $\ell_k = 0$
  - avec $y^\star$, $\ell_i = 1$ ou $\ell_j = 1$ ou $\ell_k = 1$
  - l'algorithme modifie y en changeant aléatoirement y_i ou y_j
    $M(y, y^\star)$ augmente de 1 avec probabilité $\geq 1/3$
    $M(y, y^\star)$ diminue de 1 avec probabilité $\leq 2/3$

Marche aléatoire (2/2)

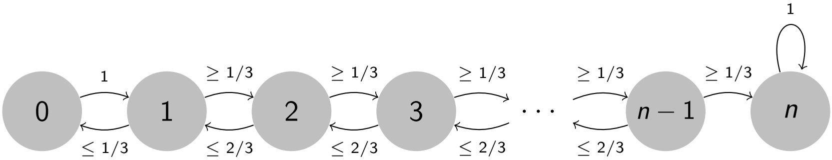
- Nous interprétons le comportement de l'algorithme comme une marche aléatoire dans un graphe



- Au départ $M(y, y^\star)$ prend n'importe quelle valeur de $\{0, \ldots, n\}$
- À chaque étape $M(y, y^\star)$ varie de ± 1
- **Combien de temps pour atteindre le sommet n ?**
- *A priori*, beaucoup plus que dans le cas de 2-SAT !

Marche aléatoire (2/2)

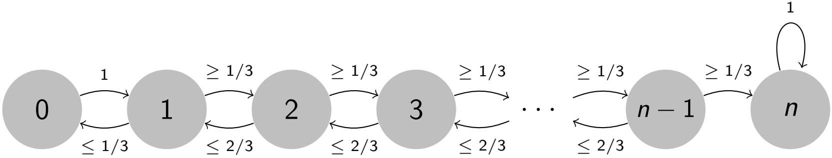
- Nous interprétons le comportement de l'algorithme comme une marche aléatoire dans un graphe



- Au départ $M(y, y^\star)$ prend n'importe quelle valeur de $\{0, \ldots, n\}$
- À chaque étape $M(y, y^\star)$ varie de ± 1
- **Combien de temps pour atteindre le sommet n ?**
- *A priori*, beaucoup plus que dans le cas de 2-SAT !

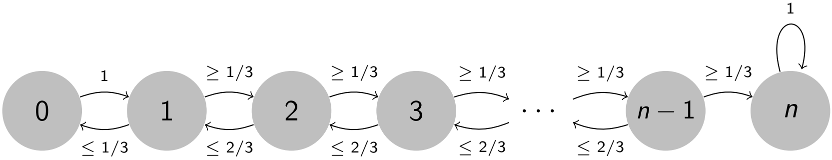
Marche aléatoire (2/2)

- Nous interprétons le comportement de l'algorithme comme une marche aléatoire dans un graphe



- Au départ $M(y, y^\star)$ prend n'importe quelle valeur de $\{0, \ldots, n\}$
- À chaque étape $M(y, y^\star)$ varie de ± 1
- **Combien de temps pour atteindre le sommet n ?**
- *A priori*, beaucoup plus que dans le cas de 2-SAT !

Algorithme de Schöning – Première approche (1/2)

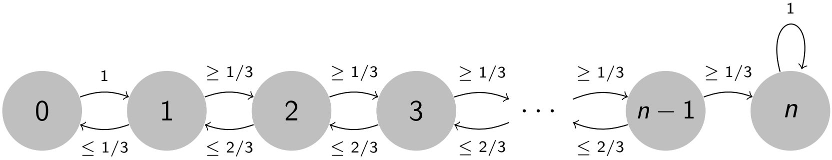


- On espère partir d'un état initial « proche » du sommet n

$$M(y, y^\star) \geq n/2 \quad \rightsquigarrow \quad \text{probabilité} \geq 1/2$$

- Si c'est le cas, on espère faire $\leq n/2$ étapes vers la droite $\rightsquigarrow$ probabilité $\geq (1/3)^{n/2}$
- Avec $T_1 = \lceil n/2 \rceil$, pour Φ satisfiable, on obtient un succès dans la boucle **pour** ... **faire** avec probabilité $p \geq (1/2) \cdot (1/3)^{n/2}$

Algorithme de Schöning – Première approche (1/2)

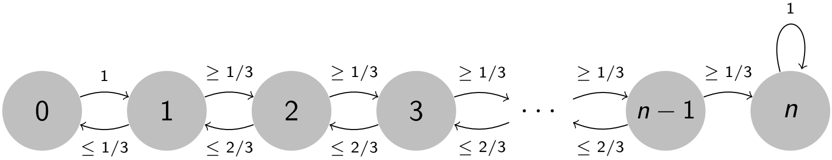


- On espère partir d'un état initial « proche » du sommet n

$$M(y, y^\star) \geq n/2 \quad \rightsquigarrow \quad \text{probabilité} \geq 1/2$$

- Si c'est le cas, on espère faire $\leq n/2$ étapes vers la droite $\rightsquigarrow$ probabilité $\geq (1/3)^{n/2}$
- Avec $T_1 = \lceil n/2 \rceil$, pour Φ satisfiable, on obtient un succès dans la boucle **pour . . . faire** avec probabilité $p \geq (1/2) \cdot (1/3)^{n/2}$

Algorithme de Schöning – Première approche (1/2)



- On espère partir d'un état initial « proche » du sommet n

$$M(y, y^\star) \geq n/2 \quad \rightsquigarrow \quad \text{probabilité} \geq 1/2$$

- Si c'est le cas, on espère faire $\leq n/2$ étapes vers la droite
  $\rightsquigarrow$ probabilité $\geq (1/3)^{n/2}$
- Avec $T_1 = \lceil n/2 \rceil$, pour Φ satisfiable, on obtient un succès dans la boucle **pour . . . faire** avec probabilité $p \geq (1/2) \cdot (1/3)^{n/2}$

Algorithme de Schöning – Première approche (2/2)

- En répétant $T_0 = \ln(m)/p$ fois cette boucle, l'algorithme échoue pour Φ satisfiable avec probabilité

$$\leq (1-p)^{\ln(m)/p} \leq \frac{1}{m}$$

Complexité :

$$O(T_0 \cdot T_1 \cdot m) = O(n \cdot m \cdot \log m \cdot 3^{n/2}) = \tilde{O}(3^{n/2}) = O(1.733^n)$$

Probabilité d'erreur $\leq 1/m$

Algorithme de Schöning – Première approche (2/2)

- En répétant $T_0 = \ln(m)/p$ fois cette boucle, l'algorithme échoue pour Φ satisfiable avec probabilité

$$\leq (1-p)^{\ln(m)/p} \leq \frac{1}{m}$$

Complexité :

$$O(T_0 \cdot T_1 \cdot m) = O(n \cdot m \cdot \log m \cdot 3^{n/2}) = \tilde{O}(3^{n/2}) = O(1.733^n)$$

Probabilité d'erreur $\leq 1/m$

Algorithme de Schöning – Première approche (2/2)

- En répétant $T_0 = \ln(m)/p$ fois cette boucle, l'algorithme échoue pour Φ satisfiable avec probabilité

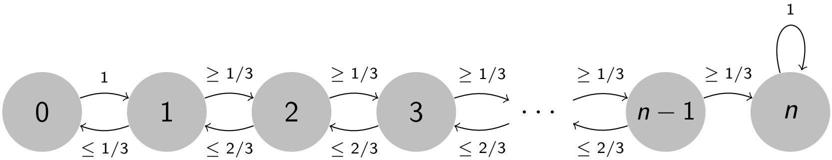
$$\leq (1-p)^{\ln(m)/p} \leq \frac{1}{m}$$

Complexité :

$$O(T_0 \cdot T_1 \cdot m) = O(n \cdot m \cdot \log m \cdot 3^{n/2}) = \tilde{O}(3^{n/2}) = O(1.733^n)$$

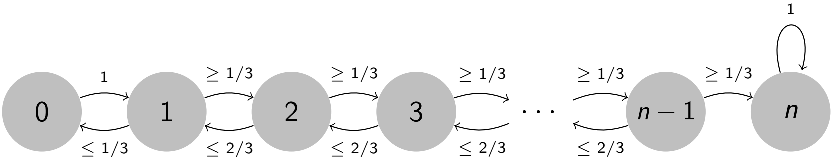
Probabilité d'erreur $\leq 1/m$

Algorithme de Schöning – Analyse (1/5)



- **Idée 1 :** Exécuter la boucle **pour . . . faire** plus longtemps pour permettre des étapes vers la gauche $\rightsquigarrow T_1 = 3n$
- **Idée 2 :** Faire une analyse plus fine en prenant en compte tous les états initiaux possibles $j \in \{0, \ldots, n\}$

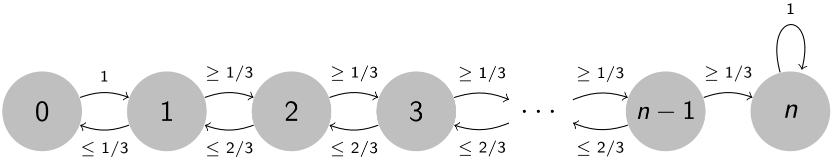
Algorithme de Schöning – Analyse (1/5)



- **Idée 1 :** Exécuter la boucle **pour . . . faire** plus longtemps pour permettre des étapes vers la gauche $\rightsquigarrow T_1 = 3n$

- **Idée 2 :** Faire une analyse plus fine en prenant en compte tous les états initiaux possibles $j \in \{0, \ldots, n\}$

Algorithme de Schöning – Analyse (2/5)



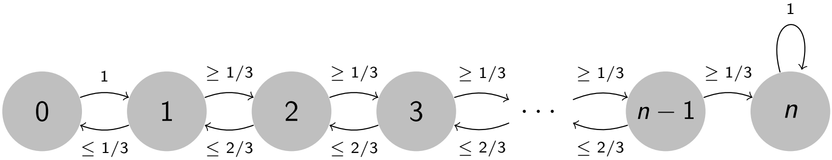
- D_j : « pour y aléatoire, nous avons $M(y, y^\star) = n - j$ »

$$\Pr(D_j) = \binom{n}{j}\left(\frac{1}{2}\right)^j\left(\frac{1}{2}\right)^{n-j} = \binom{n}{j}\left(\frac{1}{2}\right)^n$$

- E_j : « en partant de $n-j$, nous arrivons à n en $\leq 3n$ étapes »
- F_j : « en partant de $n-j$, nous arrivons à n en $2j$ étapes à droite et j étapes à gauche »

$$\Pr(E_i) \geq \Pr(F_j)$$

Algorithme de Schöning – Analyse (2/5)



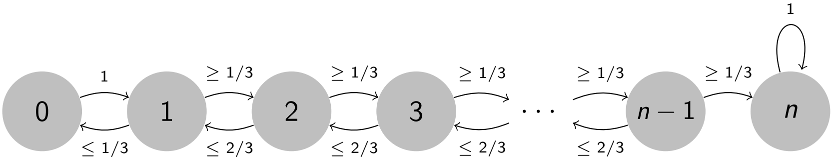
- D_j : « pour y aléatoire, nous avons $M(y, y^\star) = n - j$ »

$$\Pr(D_j) = \binom{n}{j} \left(\frac{1}{2}\right)^j \left(\frac{1}{2}\right)^{n-j} = \binom{n}{j} \left(\frac{1}{2}\right)^n$$

- E_j : « en partant de $n - j$, nous arrivons à n en $\leq 3n$ étapes »
- F_j : « en partant de $n - j$, nous arrivons à n en $2j$ étapes à droite et j étapes à gauche »

$$\Pr(E_i) \geq \Pr(F_j)$$

Algorithme de Schöning – Analyse (2/5)



- D_j : « pour y aléatoire, nous avons $M(y, y^\star) = n - j$ »

$$\Pr(D_j) = \binom{n}{j}\left(\frac{1}{2}\right)^j\left(\frac{1}{2}\right)^{n-j} = \binom{n}{j}\left(\frac{1}{2}\right)^n$$

- E_j : « en partant de $n - j$, nous arrivons à n en $\leq 3n$ étapes »
- F_j : « en partant de $n - j$, nous arrivons à n en $2j$ étapes à droite et j étapes à gauche »

$$\Pr(E_i) \geq \Pr(F_j)$$

Algorithme de Schöning – Analyse (3/5)

- F_j : « en partant de $n-j$, nous arrivons à n en $2j$ étapes à droite et j étapes à gauche »

$$\Pr(F_j) \geq \binom{3j}{j}\left(\frac{2}{3}\right)^j\left(\frac{1}{3}\right)^{2j} \text{ avec } \binom{3j}{j} = \frac{3j!}{2j!\cdot j!}$$

Formule de Stirling

$$\sqrt{2\pi t}\left(\frac{t}{e}\right)^t \leq t! \leq 2\cdot\sqrt{2\pi t}\left(\frac{t}{e}\right)^t$$

$$\binom{3j}{j} \geq \frac{\sqrt{2\pi 3j}\left(\frac{3j}{e}\right)^{3j}}{2\cdot\sqrt{2\pi j}\left(\frac{j}{e}\right)^j 2\cdot\sqrt{2\pi 2j}\left(\frac{2j}{e}\right)^{2j}} = \frac{\sqrt{3}}{8\sqrt{\pi j}}\left(\frac{27}{4}\right)^j$$

et

$$\Pr(F_j) \geq \frac{\sqrt{3}}{8\sqrt{\pi j}}\left(\frac{27}{4}\right)^j\left(\frac{2}{3}\right)^j\left(\frac{1}{3}\right)^{2j} = \frac{\sqrt{3}}{8\sqrt{\pi j}}\left(\frac{1}{2}\right)^j$$

Algorithme de Schöning – Analyse (4/5)

- Nous avons pour tout $j \in \{1, \ldots, n\}$

$$\Pr(E_j) \geq \Pr(F_j) \geq \frac{\sqrt{3}}{8\sqrt{\pi j}} \left(\frac{1}{2}\right)^j = \frac{c}{\sqrt{j}} \left(\frac{1}{2}\right)^j \geq \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^j$$

- Nous avons une probabilité de succès p' telle que

$$p' \geq \sum_{j=0}^{n} \Pr(D_j) \Pr(E_j) \geq \sum_{j=0}^{n} \binom{n}{j} \left(\frac{1}{2}\right)^n \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^j$$

$$\geq \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^n \sum_{j=0}^{n} \binom{n}{j} \left(\frac{1}{2}\right)^j$$

Algorithme de Schöning – Analyse (4/5)

- Nous avons pour tout $j \in \{1, \ldots, n\}$

$$\Pr(E_j) \geq \Pr(F_j) \geq \frac{\sqrt{3}}{8\sqrt{\pi j}} \left(\frac{1}{2}\right)^j = \frac{c}{\sqrt{j}} \left(\frac{1}{2}\right)^j \geq \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^j$$

- Nous avons une probabilité de succès p' telle que

$$p' \geq \sum_{j=0}^{n} \Pr(D_j) \Pr(E_j) \geq \sum_{j=0}^{n} \binom{n}{j} \left(\frac{1}{2}\right)^n \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^j$$

$$\geq \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^n \sum_{j=0}^{n} \binom{n}{j} \left(\frac{1}{2}\right)^j$$

Algorithme de Schöning – Analyse (4/5)

- Nous avons pour tout $j \in \{1, \dots, n\}$

$$\Pr(E_j) \geq \Pr(F_j) \geq \frac{\sqrt{3}}{8\sqrt{\pi j}} \left(\frac{1}{2}\right)^j = \frac{c}{\sqrt{j}} \left(\frac{1}{2}\right)^j \geq \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^j$$

- Nous avons une probabilité de succès p' telle que

$$\begin{aligned} p' \geq \sum_{j=0}^{n} \Pr(D_j) \Pr(E_j) &\geq \sum_{j=0}^{n} \binom{n}{j} \left(\frac{1}{2}\right)^n \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^j \\ &\geq \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^n \sum_{j=0}^{n} \binom{n}{j} \left(\frac{1}{2}\right)^j \end{aligned}$$

Algorithme de Schöning – Analyse (4/5)

- Nous avons pour tout $j \in \{1, \ldots, n\}$

$$\Pr(E_j) \geq \Pr(F_j) \geq \frac{\sqrt{3}}{8\sqrt{\pi j}} \left(\frac{1}{2}\right)^j = \frac{c}{\sqrt{j}} \left(\frac{1}{2}\right)^j \geq \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^j$$

- Nous avons une probabilité de succès p' telle que

$$\begin{aligned} p' \geq \sum_{j=0}^{n} \Pr(D_j) \Pr(E_j) &\geq \sum_{j=0}^{n} \binom{n}{j} \left(\frac{1}{2}\right)^n \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^j \\ &\geq \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^n \sum_{j=0}^{n} \binom{n}{j} \left(\frac{1}{2}\right)^j 1^{n-j} \end{aligned}$$

Algorithme de Schöning – Analyse (4/5)

- Nous avons pour tout $j \in \{1, \ldots, n\}$

$$\Pr(E_j) \geq \Pr(F_j) \geq \frac{\sqrt{3}}{8\sqrt{\pi j}} \left(\frac{1}{2}\right)^j = \frac{c}{\sqrt{j}} \left(\frac{1}{2}\right)^j \geq \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^j$$

- Nous avons une probabilité de succès p' telle que

$$\begin{aligned} p' \geq \sum_{j=0}^{n} \Pr(D_j) \Pr(E_j) &\geq \sum_{j=0}^{n} \binom{n}{j} \left(\frac{1}{2}\right)^n \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^j \\ &\geq \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^n \sum_{j=0}^{n} \binom{n}{j} \left(\frac{1}{2}\right)^j 1^{n-j} \\ &\geq \frac{c}{\sqrt{n}} \left(\frac{1}{2}\right)^n \left(\frac{1}{2}+1\right)^n = \frac{c}{\sqrt{n}} \left(\frac{3}{4}\right)^n \end{aligned}$$

Algorithme de Schöning – Analyse (5/5)

- Avec $T_1 = 3n$, pour Φ satisfiable, on obtient un succès dans la boucle **pour . . . faire** avec probabilité $p' \geq \frac{c}{\sqrt{n}} \left(\frac{3}{4}\right)^n$
- En répétant $T_0 = \ln(m)/p'$ fois cette boucle, l'algorithme échoue pour Φ satisfiable avec probabilité

$$\leq (1 - p')^{\ln(m)/p'} \leq \frac{1}{m}$$

Complexité :

$$O(T_0 T_1 m) = O(m \cdot n \cdot \sqrt{n} \cdot (4/3)^n) = \tilde{O}((4/3)^n) = O(1.334^n)$$

Probabilité d'erreur $\leq 1/m$

Algorithme de type Monte Carlo

Algorithme de Schöning – Analyse (5/5)

- Avec $T_1 = 3n$, pour Φ satisfiable, on obtient un succès dans la boucle **pour . . . faire** avec probabilité $p' \geq \frac{c}{\sqrt{n}} \left(\frac{3}{4}\right)^n$
- En répétant $T_0 = \ln(m)/p'$ fois cette boucle, l'algorithme échoue pour Φ satisfiable avec probabilité

$$\leq (1 - p')^{\ln(m)/p'} \leq \frac{1}{m}$$

Complexité :

$$O(T_0 T_1 m) = O(m \cdot n \cdot \sqrt{n} \cdot (4/3)^n) = \tilde{O}((4/3)^n) = O(1.334^n)$$

Probabilité d'erreur $\leq 1/m$

Algorithme de type Monte Carlo

Algorithme de Schöning – Analyse (5/5)

- Avec $T_1 = 3n$, pour Φ satisfiable, on obtient un succès dans la boucle **pour ... faire** avec probabilité $p' \geq \frac{c}{\sqrt{n}} \left(\frac{3}{4}\right)^n$
- En répétant $T_0 = \ln(m)/p'$ fois cette boucle, l'algorithme échoue pour Φ satisfiable avec probabilité

$$\leq (1 - p')^{\ln(m)/p'} \leq \frac{1}{m}$$

Complexité :

$$O(T_0 T_1 m) = O(m \cdot n \cdot \sqrt{n} \cdot (4/3)^n) = \tilde{O}((4/3)^n) = O(1.334^n)$$

Probabilité d'erreur $\leq 1/m$

Algorithme de type Monte Carlo

Algorithme de Schöning – Analyse (5/5)

- Avec $T_1 = 3n$, pour Φ satisfiable, on obtient un succès dans la boucle **pour ... faire** avec probabilité $p' \geq \frac{c}{\sqrt{n}} \left(\frac{3}{4}\right)^n$
- En répétant $T_0 = \ln(m)/p'$ fois cette boucle, l'algorithme échoue pour Φ satisfiable avec probabilité

$$\leq (1 - p')^{\ln(m)/p'} \leq \frac{1}{m}$$

Complexité :

$$O(T_0 T_1 m) = O(m \cdot n \cdot \sqrt{n} \cdot (4/3)^n) = \tilde{O}((4/3)^n) = O(1.334^n)$$

Probabilité d'erreur $\leq 1/m$

Algorithme de type Monte Carlo