

Vorlesung

Effiziente Algorithmen und Komplexitätstheorie

Sommersemester 2008

Ingo Wegener

Was bisher geschah...

SAT **Eingabe** m Klauseln c_1, c_2, \dots, c_m
 über n Variablen x_1, x_2, \dots, x_n
 (z. B. $c_j = x_3 \vee \overline{x_5} \vee \overline{x_7} \vee x_9$)

zulässige Lösungen

Belegungen $b \in \{0, 1\}^n$

Bewertung Anzahl durch b erfüllter Klauseln

klar

- $\text{SAT} \in \mathcal{NP}$
- zugehöriges Entscheidungsproblem NP-vollständig

Theorem 10.9

Algorithmus 10.9 berechnet zu einer MAXSAT-Instanz, in der höchstens OPT Klauseln gleichzeitig erfüllt werden können, in Polynomialzeit eine Belegung, in der im Erwartungswert mindestens $(3/4) \cdot \text{OPT}$ Klauseln gleichzeitig erfüllt sind.

Und unser Thema heute...

Wir haben randomisiert in Polynomialzeit
im Erwartungswert Belegung mit Güte $\leq 4/3$

Was tun, wenn man eine bessere Lösung braucht?

Fakt MAXSAT hat **kein PTAS**, falls $P \neq NP$

klar für nicht zu große Eingaben
exakte Lösung bestimmbar in akzeptabler Zeit

Was kann man tun, um „zu groß“ zu verschieben?

Ziel möglichst schneller 3-SAT-Algorithmus

Üben am „kleinen Bruder“

Erinnerung Fakt 2-SAT $\in P$

vorab **Warnhinweis**

Es folgen „merkwürdige“ Algorithmen.

Eigenschaften

- randomisiert
- für erfüllbare 2-SAT-Instanzen erfüllende Belegung in erwarteter Polynomialzeit
- für unerfüllbare 2-SAT-Instanzen zunächst **keine** hilfreiche Information
- für unerfüllbare 2-SAT-Instanzen **herleitbar** (machen wir aber kaum) Ausgabe „vermutlich nicht erfüllbar“ und Prob (doch erfüllbar) **sehr klein**

später für 3-SAT

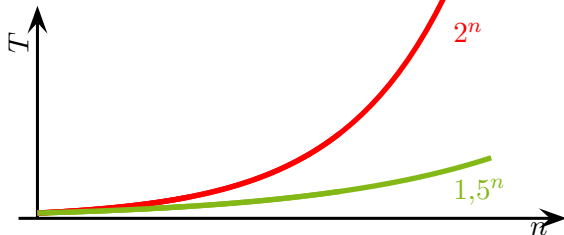
- **analog**, aber in Exponentialzeit

Motivation

Was sollen wir mit Exponentialzeitalgorithmen?

Erinnerung wenn $ZPP \neq NP$
Algorithmen mit erwarteter Polynomialzeit
nicht möglich

Erinnerung Ziel „zu groß“ verschieben



Ein Zufallsprozess

Definition $X_0, X_1, X_2, X_3, \dots \in \mathbb{Z}$
 $X_0 := a \in \{0, 1, 2, \dots, n\}$ fest
 $X_t \in \{X_{t-1} - 1, X_{t-1} + 1\}$ mit
 $\text{Prob}(X_t = X_{t-1} - 1) = \text{Prob}(X_t = X_{t-1} + 1) = 1/2$

Beobachtung $(X_i)_{i \geq 0}$ ist Markow-Kette

Definition $T := \min\{t \mid |X_t| = n\}$

Was ist $E(T)$?

Definition 10.11

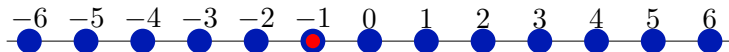
ZV $T: \Omega \rightarrow \mathbb{N} \cup \{\infty\}$ heißt **Stopzeit** von $(X_i)_{i \geq 0}$, wenn für alle $n \in \mathbb{N}_0$ das Ereignis $T = n$ basierend auf X_0, X_1, \dots, X_n ausgedrückt werden kann.

Beobachtung T ist Stopzeit

Ein fairer Random Walk

Definition $X_0, X_1, X_2, X_3, \dots \in \mathbb{Z}$
 $X_0 := a \in \{0, 1, 2, \dots, n\}$ fest
 $X_t \in \{X_{t-1} - 1, X_{t-1} + 1\}$ mit
 $\text{Prob}(X_t = X_{t-1} - 1) = \text{Prob}(X_t = X_{t-1} + 1) = 1/2$

Beispiel für $n = 5, a = 3$



Wie bestimmen wir $E(T)$?

Anmerkung Ad-hoc-Analyse möglich (gar nicht so schwierig)

hier Anwendung einer **allgemeinen Methode**

Über Martingale

Definition 10.12

Ein Zufallsprozess $(Y_i)_{i \geq 0}$ mit $Y_i \in \mathbb{R}$ für alle $i \in \mathbb{N}_0$ heißt **Martingal** in Bezug auf einen Zufallsprozess $(X_i)_{i \geq 0}$, wenn für alle $n \in \mathbb{N}_0$ die folgenden drei Aussagen gelten.

- 1 Y_n ist eine Funktion von X_0, X_1, \dots, X_n .
- 2 $E(|Y_n|) < \infty$ oder $Y_n \geq 0$
- 3 $E(Y_{n+1} \mid X_0, X_1, \dots, X_n) = Y_n$

Theorem 10.13 (Optional-Stopping-Theorem)

Sei $(Y_i)_{i \geq 0}$ Martingal in Bezug auf $(X_i)_{i \geq 0}$, T Stoppzeit von $(X_i)_{i \geq 0}$.

$(\exists k \in \mathbb{N}_0: T \leq k \text{ fast sicher})$

$\vee (T < \infty \wedge (\exists k \in \mathbb{N}_0: \forall t < T: |Y_t| \leq k \text{ fast sicher})) :$

$$E(Y_T) = E(Y_0)$$

Über unseren fairen Random Walk

Lemma

$(Y_i)_{i \geq 0}$ mit $Y_i := X_i^2 - i$ ist ein Martingal in Bezug auf den fairen Random Walk $(X_i)_{i \geq 0}$.

Beweis.

Beobachtung Y_i ist Funktion von X_0, X_1, \dots, X_n ✓

Beobachtung $|Y_i| = |X_i^2 - i| \leq X_i^2 + i \leq (a + i)^2 + i < \infty$ ✓

$$\begin{aligned}
 & \mathbb{E}(Y_{i+1} \mid X_0, X_1, \dots, X_i) \\
 = & \mathbb{E}(X_{i+1}^2 - (i+1) \mid X_0, X_1, \dots, X_i) \\
 = & \mathbb{E}(X_{i+1}^2 \mid X_i) - i - 1 \\
 = & \frac{1}{2} \cdot (X_i + 1)^2 + \frac{1}{2} \cdot (X_i - 1)^2 - i - 1 \\
 = & X_i^2 + 1 - i - 1 = X_i^2 - i = Y_i \quad \checkmark
 \end{aligned}$$

Erwartete Stoppzeit des fairen Random Walk

Theorem 10.14

Der faire Random Walk mit Start in a hat eine erwartete Stoppzeit $E(T) = (n - a) \cdot (n + a)$.

Beweis.

Betrachte Martingal $(Y_i)_{i \geq 0}$ mit $Y_i := X_i^2 - i$

Wunsch Anwendung des Optional-Stopping-Theorems

zuerst Voraussetzungen prüfen

Beobachtung $T < \infty$ fast sicher
 denn immer nach $\leq n$ gleichen Schritten am Ende
 also $\forall t: \text{Prob}(T > t) < (1 - 2^{-n})^{\lceil t/n \rceil}$ ✓

Erinnerung $|Y_i| \leq (a + i)^2 + i$
 also $|Y_i| \leq k := (a + i)^2 + i$ ✓

also Optional-Stopping-Theorem **anwendbar**

Anwendung des Optional-Stopping-Theorems

Betrachte Martingal $(Y_i)_{i \geq 0}$ mit $Y_i := X_i^2 - i$

Optional-Stopping-Theorem
$$\begin{aligned} \mathbb{E}(Y_T) &= \mathbb{E}(Y_0) \\ &= \mathbb{E}(X_0^2 - 0) = \mathbb{E}(a^2) = a^2 \end{aligned}$$

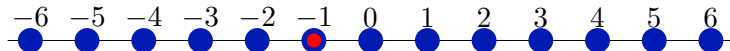
aber auch
$$\begin{aligned} \mathbb{E}(Y_T) &= \mathbb{E}(X_T^2 - T) \\ &= \mathbb{E}(X_T^2) - \mathbb{E}(T) = \mathbb{E}(n^2) - \mathbb{E}(T) = n^2 - \mathbb{E}(T) \end{aligned}$$

also zusammen
$$\begin{aligned} a^2 &= n^2 - \mathbb{E}(T) \\ \Leftrightarrow \mathbb{E}(T) &= n^2 - a^2 = (n - a) \cdot (n + a) \end{aligned}$$

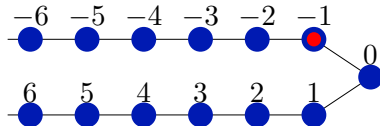


Eine winzige Modifikation der Sichtweise

Wir kennen schon unseren fairen Random Walk

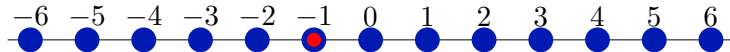


ohne Unterschied eine andere Darstellung

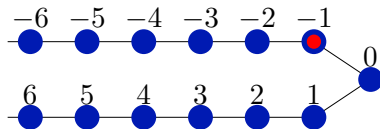


Ein reflektierender fairer Random Walk

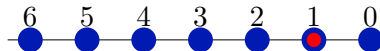
Wir kennen schon unseren fairen Random Walk



ohne Unterschied eine andere Darstellung



jetzt „neu“ ein reflektierender Random Walk



Der reflektierende faire Random Walk gespiegelt

Definition $(X_i)_{i \geq 0}$ mit
 $X_0 := b \in \{0, 1, \dots, n\}$ fest
 $X_t \in \{X_{t-1} - 1, X_{t-1} + 1\}$ mit

$$\text{Prob}(X_t = X_{t-1} - 1) = \begin{cases} \frac{1}{2} & \text{für } 0 < X_{t-1} < n \\ 1 & \text{für } X_{t-1} = n \\ 0 & \text{sonst} \end{cases}$$

$$\text{Prob}(X_t = X_{t-1} + 1) = \begin{cases} \frac{1}{2} & \text{für } 0 < X_{t-1} < n \\ 1 & \text{für } X_{t-1} = 0 \\ 0 & \text{sonst} \end{cases}$$

Betrachte $T := \min\{t \mid X_t = 0\}$

Beobachtung ist wie fairer Random Walk
aber „geknickt“ und „verschmolzen“

Beobachtung Startpunkt $b \rightsquigarrow a = n - b$

also $E(T) = (n - a) \cdot (n + a) = b \cdot (2n - b)$

Ein Algorithmus für 2-SAT

Parameter $T \in \mathbb{N} \cup \{\infty\}$

Algorithmus 10.15

1. Für alle $i \in \{1, 2, \dots, n\}$ setze $b[i] := 0$.
2. Für alle $t \in \{1, 2, \dots, T\}$
 3. Falls b alle Klauseln erfüllt, gib b aus. STOP.
 4. Wähle eine Klausel c_j , die unter b nicht erfüllt ist.
 5. Wähle uniform zufällig ein Literal aus c_j .
 6. Invertiere die zu diesem Literal gehörige Stelle in b .
7. Ausgabe „Es gibt vermutlich keine erfüllende Belegung.“

Theorem 10.16

Für eine erfüllbare 2-SAT-Instanz über n Variablen liefert Algorithmus 10.15 mit $T = \infty$ im Durchschnitt nach höchstens n^2 Schleifendurchläufen eine erfüllende Belegung.

Analyse von Algorithmus 10.15

Voraussetzung 2-SAT-Instanz erfüllbar

Sei b^* eine erfüllende Belegung

Definition $d(b, b^*) := \sum_{i=1}^n |b[i] - b^*[i]|$

Hammingabstand von b und b^*

einfache Beobachtungen

- $0 \leq d(b, b^*) \leq n$
- $d(b, b^*) = 0 \Rightarrow b$ erfüllend

Betrachte Klausel c_j mit Variablen x_{j_1}, x_{j_2}

Beobachtung $d(b[j_1]b[j_2], b^*[j_1]b^*[j_2]) > 0$

Beweis von Theorem 10.16

Wir haben erfüllende Belegung b^*
 $0 \leq d(b, b^*) \leq n$ mit $d(b, b^*) = 0 \Rightarrow b$ erfüllend
 bei Klausel c_j mit Variablen x_{j_1}, x_{j_2}
 $d(b[j_1]b[j_2], b^*[j_1]b^*[j_2]) > 0$

also Prob ($d(b, b^*)$ wächst) $\in \{0, 1/2\} \leq 1/2$
 Prob ($d(b, b^*)$ fällt) $\in \{1/2, 1\} \geq 1/2$

also stochastisch dominiert vom fairen Random Walk
 auf $\{0, 1, \dots, n\}$ (reflektierend in n)

also Anzahl Runden $\leq i \cdot (2n - i) \leq n^2$



Ein sinnvoller Algorithmus

Ist Algorithmus 10.15 auch allgemein sinnvoll?

Was wünschen wir uns?

Wünsche

- für erfüllbare Instanzen mit großer Wahrscheinlichkeit erfüllende Belegung
- möglichst schnelle Antwort

Korollar 10.17

Für jedes (nicht notwendig konstante) ε mit $0 < \varepsilon < 1$ kann man durch unabhängige Wiederholung von Algorithmus 10.15 einen Algorithmus erhalten, der mit $O(-\log(\varepsilon) \cdot n^2)$ Runden von Algorithmus 10.15 auskommt und für eine erfüllbare 2-SAT-Instanz über n Variablen nur mit Wahrscheinlichkeit höchstens ε die Ausgabe „Es gibt vermutlich keine erfüllende Belegung.“ erzeugt.

Beweis von Korollar 10.17

Beweis.

Wir haben $E(\# \text{Runden bis erfüllende Belegung}) \leq n^2$

also für erfüllbare 2-SAT-Instanz
 Prob (nach $2n^2$ Runden ohne Erfolg) $\leq 1/2$
 (Markow-Ungleichung)

also nach w Wiederholungen mit $T := 2n^2$
 Misserfolgsw'keit $\leq (1/2)^w$

nachrechnen $(1/2)^{\lceil -\log \varepsilon \rceil} \leq (1/2)^{-\log \varepsilon} \leq \varepsilon$



Eine gute Idee. . .

im Rückblick Algorithmus 10.15 ist

- bestechend einfach
- erstaunlich effizient

verwegene Idee Benutze Algorithmus 10.15 für 3-SAT.

klar Wir erwarten nicht Polynomialzeit.

Wir hoffen $E(T) = O(\text{poly}(n) \cdot b^n)$ mit $b > 1$ klein

Was leistet Algorithmus 10.15 für 3-SAT überhaupt?

Theorem 10.18

Für eine 3-SAT-Instanz mit erfüllender Belegung b^* findet Algorithmus 10.15 mit $T = 3n + 1$ mit Wahrscheinlichkeit mindestens $\frac{1}{3n+1} \cdot \left(\frac{1}{2}\right)^{d(0^n, b^*)}$ eine erfüllende Belegung.

Zum Beweis von Theorem 10.18

Wie gehabt 3-SAT-Instanz als erfüllbar vorausgesetzt
 b^* sei erfüllende Belegung
 $d(b, b^*)$ Hammingabstand, initial $d(0^n, b^*)$
 bei nicht erfüllter Klausel c_j mit Variablen $x_{j_1}, x_{j_2}, x_{j_3}$
 $d(b[j_1]b[j_2]b[j_3], b^*[j_1]b^*[j_2]b^*[j_3]) > 0$

also Prob ($d(b, b^*)$ wächst) $\in \{0, 1/3, 2/3\} \leq 2/3$
 Prob ($d(b, b^*)$ fällt) $\in \{1/3, 2/3, 1\} \geq 1/3$

Beobachtung auch ein Random Walk
 aber ein **unfairer** zu unseren **Ungunsten**

Die ersten t Schritte

Betrachte die ersten t Schritte, Initialabstand $d^* := d(0^n, b^*)$

Definition $S^+(t) = \#\text{Schritte davon, die } d(b, b^*) \text{ vergrößern}$
 $S^-(t) = \#\text{Schritte davon, die } d(b, b^*) \text{ verkleinern}$

$$\begin{aligned}
 & \text{Prob (finde erfüllende Belegung in } 3n + 1 \text{ Schritten)} \\
 & \geq \text{Prob (finde } b^* \text{ in } 3n + 1 \text{ Schritten)} \\
 & \geq \text{Prob (finde } b^* \text{ in } 3d(0^n, b^*) \text{ Schritten)} \\
 & \geq \text{Prob} \left((S^-(3d^*) \geq 2d^*) \wedge (S^+(3d^*) \leq d^*) \right) \\
 & = \text{Prob} (S^-(3d^*) \geq 2d^*) \geq \text{Prob} (S^-(3d^*) = 2d^*) \\
 & = \binom{3d^*}{2d^*} \cdot \left(\frac{1}{3}\right)^{2d^*} \cdot \left(\frac{2}{3}\right)^{d^*}
 \end{aligned}$$

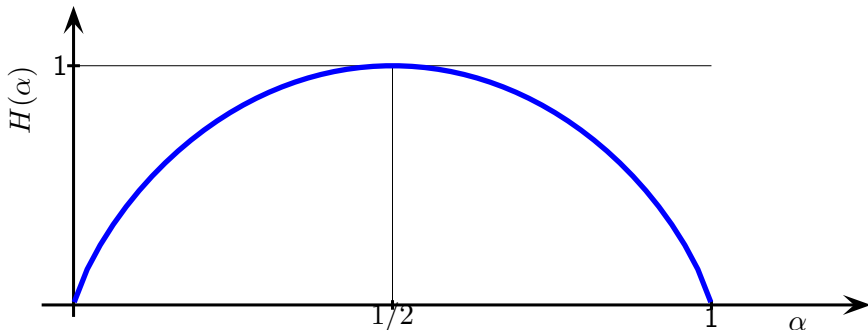
Zur Abschätzung von Binomialkoeffizienten

Definition 10.19

Für $\alpha \in [0; 1]$ heißt

$$H(\alpha) := -\alpha \log(\alpha) - (1 - \alpha) \log(1 - \alpha)$$

die **Entropie** von α . Es ist $H(0) := H(1) := 0$.



Abschätzung von Binomialkoeffizienten

Lemma 10.20

$\forall n \in \mathbb{N}: \forall k \in \{1, 2, \dots, n\}:$

$$\frac{1}{n+1} \cdot 2^{H(k/n) \cdot n} \leq \binom{n}{k} \leq 2^{H(k/n) \cdot n}$$

Beweis.

Betrachte n unabhängige Münzwürfe
jeweils Prob (Kopf) = p

Sei K = Anzahl „Kopf“

klar K binomialverteilt mit Parametern n, p

also $\text{Prob}(K = k) = \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k} \leq 1$

Beweis von Lemma 10.20

Wir haben $\text{Prob}(K = k) = \binom{n}{k} \cdot p^k \cdot (1 - p)^{n-k} \leq 1$
für beliebiges $p \in [0; 1]$

Wähle $p := k/n$

$$\begin{aligned} \text{Prob}(K = k) &= \binom{n}{k} \cdot \left(\frac{k}{n}\right)^k \cdot \left(1 - \left(\frac{k}{n}\right)\right)^{n-k} && \leq 1 \\ \Leftrightarrow \binom{n}{k}^{1/n} \cdot \left(\frac{k}{n}\right)^{k/n} \cdot \left(1 - \left(\frac{k}{n}\right)\right)^{(n-k)/n} && \leq 1 \\ \Leftrightarrow \left(\frac{k}{n}\right)^{k/n} \cdot \left(1 - \left(\frac{k}{n}\right)\right)^{(n-k)/n} && \leq \binom{n}{k}^{-1/n} \end{aligned}$$

Erinnerung $H(\alpha) = -\alpha \log(-\alpha) - (1 - \alpha) \log(1 - \alpha)$

also $2^{-H(\alpha)} = \alpha^\alpha \cdot (1 - \alpha)^{1-\alpha}$

Tapfer rechnen...

Wir haben $\left(\frac{k}{n}\right)^{k/n} \cdot \left(1 - \left(\frac{k}{n}\right)\right)^{(n-k)/n} \leq \binom{n}{k}^{-1/n}$
 und $2^{-H(\alpha)} = \alpha^\alpha \cdot (1 - \alpha)^{1-\alpha}$

$$\begin{aligned} \left(\frac{k}{n}\right)^{k/n} \cdot \left(1 - \left(\frac{k}{n}\right)\right)^{(n-k)/n} &\leq \binom{n}{k}^{-1/n} \\ \Leftrightarrow 2^{-H(k/n)} &\leq \binom{n}{k}^{-1/n} \\ \Leftrightarrow \binom{n}{k}^{1/n} &\leq 2^{H(k/n)} \\ \Leftrightarrow \binom{n}{k} &\leq 2^{H(k/n) \cdot n} \checkmark \end{aligned}$$

für zweite Ungleichung etwas ausholen...

Über den Zentralterm der Binomialverteilung

Wollen zeigen mit $p = k/n$

$$\text{Prob}(K = k) = \max \{ \text{Prob}(K = i) \mid i \in \{0, 1, \dots, n\} \}$$

Erinnerung $E(K) = n \cdot p = n \cdot k/n = k$

$$\begin{aligned} \frac{\text{Prob}(K = i)}{\text{Prob}(K = i - 1)} &= \frac{\binom{n}{i} \cdot \left(\frac{k}{n}\right)^i \cdot \left(1 - \frac{k}{n}\right)^{n-i}}{\binom{n}{i-1} \cdot \left(\frac{k}{n}\right)^{i-1} \cdot \left(1 - \frac{k}{n}\right)^{n-(i-1)}} \\ &= \frac{n!}{i! \cdot (n-i)!} \cdot \frac{(i-1)! \cdot (n-i+1)!}{n!} \cdot \frac{k/n}{1 - k/n} \\ &= \frac{n-i+1}{i} \cdot \frac{k}{n-k} = \frac{nk - ik + k}{in - ik} = 1 + \frac{k(n+1) - in}{in - ik} \end{aligned}$$

also $i > k \cdot \frac{n+1}{n} \Rightarrow \frac{\text{Prob}(K=i)}{\text{Prob}(K=i-1)} < 1$

$i < k \cdot \frac{n+1}{n} \Rightarrow \frac{\text{Prob}(K=i)}{\text{Prob}(K=i-1)} > 1$

klar $k < k \cdot \frac{n+1}{n} < k + 1$ ✓

Beweis der zweiten Ungleichung

Wir haben $\text{Prob}(K = k)$
 $= \max \{ \text{Prob}(k = i) \mid i \in \{0, 1, \dots, n\} \}$

Beobachtung $\sum_{i=0}^n \binom{n}{i} \cdot \left(\frac{k}{n}\right)^i \cdot \left(1 - \frac{k}{n}\right)^{n-i} = 1$
 weil Binomialverteilung **Verteilung** ist

also $\binom{n}{k} \cdot \left(\frac{k}{n}\right)^k \cdot \left(1 - \frac{k}{n}\right)^{n-k} \geq \frac{1}{n+1}$

wie vorhin $\frac{1}{n+1} \cdot 2^{H(n/k) \cdot n} \geq \binom{n}{k}$



Wozu sollte das gleich gut sein?

Erinnerung Beweis von Theorem 10.18

Zurück zum Beweis von Theorem 10.18

Theorem 10.18

Für eine 3-SAT-Instanz mit erfüllender Belegung b^* findet Algorithmus 10.15 mit $T = 3n + 1$ mit Wahrscheinlichkeit mindestens $\frac{1}{3n+1} \cdot \left(\frac{1}{2}\right)^{d(0^n, b^*)}$ eine erfüllende Belegung.

Beweis.

Definition $S^+(t) = \#\text{Schritte in } t, \text{ die } d(b, b^*) \text{ vergrößern}$
 $S^-(t) = \#\text{Schritte in } t, \text{ die } d(b, b^*) \text{ verkleinern}$
 Initialabstand $d^* := d(0^n, b^*)$

Wir haben Prob (finde erfüllende Belegung in $3n + 1$ Schritten)
 $\geq \text{Prob}(S^-(3d^*) = 2d^*) = \binom{3d^*}{2d^*} \cdot \left(\frac{1}{3}\right)^{2d^*} \cdot \left(\frac{2}{3}\right)^{d^*}$

Zum Ende rechnen...

Prob (finde erfüllende Belegung in $3n + 1$ Schritten)

$$\begin{aligned}
 &\geq \binom{3d^*}{2d^*} \cdot \left(\frac{1}{3}\right)^{2d^*} \cdot \left(\frac{2}{3}\right)^{d^*} \\
 &\geq \frac{1}{3d^* + 1} \cdot 2^{H(2/3) \cdot 3d^*} \cdot \left(\frac{1}{3}\right)^{2d^*} \cdot \left(\frac{2}{3}\right)^{d^*} \\
 &= \frac{1}{3d^* + 1} \cdot 2^{H(2/3) \cdot 3d^*} \cdot \left(\left(\frac{1}{3}\right)^{1/3} \cdot \left(\frac{2}{3}\right)^{2/3} \right)^{3d^*} \cdot \left(\frac{1}{2}\right)^{d^*} \\
 &= \frac{1}{3d^* + 1} \cdot 2^{H(2/3) \cdot 3d^*} \cdot \left(2^{-H(2/3)}\right)^{3d^*} \cdot \left(\frac{1}{2}\right)^{d^*} \\
 &= \frac{1}{3d^* + 1} \cdot \left(\frac{1}{2}\right)^{d^*} \geq \frac{1}{3n + 1} \cdot \left(\frac{1}{2}\right)^{d^*}
 \end{aligned}$$



Ein Zwischenfazit

Wir haben Erfolgsw'keit $\geq \frac{1}{3n+1} \cdot 2^{-d(0^n, b^*)}$

Wie können wir $d(0^n, b^*)$ abschätzen?

Einsicht im Worst Case $d(0^n, b^*) = n$

also \rightsquigarrow erwartete Laufzeit $O(\text{poly}(n) \cdot 2^n)$
Mist!

Einsicht Wir müssen **und können** arbeiten
am initialen Abstand.

Ein erster 3-SAT-Algorithmus

Erinnerung Wir wollen am initialen Abstand arbeiten.

Definition Funktion $\text{RandomWalk}(T, b)$
Algorithmus 10.15 mit initialer Belegung b

Algorithmus 10.21

1. Für alle $i \in \{1, 2, \dots, T\}$
2. $\text{RandomWalk}(3n + 1, 0^n)$
3. $\text{RandomWalk}(3n + 1, 1^n)$
4. If keine erfüllende Belegung gefunden
 Then Ausgabe „Es gibt vermutlich keine erfüllende Belegung.“

Theorem 10.22

Für eine erfüllbare 3-SAT-Instanz über n Variablen liefert Algorithmus 10.21 mit $T = \infty$ im Erwartungswert in Zeit $O(\text{poly}(n) \cdot 1,42^n)$ eine erfüllende Belegung.

Beweis von Theorem 10.22

Beobachtung im mindestens einem RandomWalk-Aufruf
initialer Abstand $d^* \leq n/2$

also in diesem RandomWalk-Aufruf
Prob (finde erfüllende Belegung in $3n + 1$ Schritten)

$$\begin{aligned} &\geq \frac{1}{3n+1} \cdot \left(\frac{1}{2}\right)^{n/2} = \frac{1}{3n+1} \cdot \left(\frac{1}{\sqrt{2}}\right)^n \\ &\geq \frac{1}{3n+1} \cdot 1,42^{-n} \end{aligned}$$

also erwartete Anzahl benötigter Aufruf
 $\leq (6n + 2) \cdot 1,42^n$

klar je Aufruf polynomielle Laufzeit



Ein besserer 3-SAT-Algorithmus

Erinnerung Wir betrachten randomisierte Algorithmen.

Erinnerung Randomisierung hilft, den Worst Case zu vermeiden.

Algorithmus 10.23

1. Für alle $i \in \{1, 2, \dots, T\}$
2. Wähle $b \in \{0, 1\}^n$ uniform zufällig.
3. RandomWalk($3n + 1, b$)
4. If keine erfüllende Belegung gefunden
 Then Ausgabe „Es gibt vermutlich keine erfüllende Belegung.“

Theorem 10.24

Für eine erfüllbare 3-SAT-Instanz über n Variablen liefert Algorithmus 10.23 mit $T = \infty$ im Erwartungswert in Zeit $O(\text{poly}(n) \cdot (4/3)^n)$ eine erfüllende Belegung.

Zum Beweis von Theorem 10.24

Definition Ereignis A
 = RandomWalk-Aufruf findet erf. Belegung

$$\begin{aligned}
 & \text{Prob}(A) \\
 = & \sum_{b \in \{0,1\}^n} \text{Prob}(b \text{ gewählt}) \\
 & \cdot \text{Prob}(\text{RandomWalk}(3n+1, b) \text{ findet erf. Belegung}) \\
 \geq & \sum_{b \in \{0,1\}^n} \text{Prob}(b \text{ gewählt}) \cdot \frac{1}{3n+1} \cdot \left(\frac{1}{2}\right)^{d(b,b^*)} \\
 = & \frac{1}{3n+1} \cdot \sum_{b \in \{0,1\}^n} \text{Prob}(b \text{ gewählt}) \cdot \left(\frac{1}{2}\right)^{d(b,b^*)} \\
 = & \frac{1}{3n+1} \cdot \mathbb{E} \left(\left(\frac{1}{2}\right)^{d(b,b^*)} \right)
 \end{aligned}$$

Erwartungswert ausrechnen

Wir haben $\text{Prob}(A) \geq \frac{1}{3n+1} \cdot \mathbb{E} \left(\left(\frac{1}{2} \right)^{d(b,b^*)} \right)$

Definition Indikatorvariable

$$X_i := \begin{cases} 1 & \text{falls } b[i] \neq b^*[i] \\ 0 & \text{sonst} \end{cases}$$

$$\begin{aligned} \text{Prob}(A) &\geq \frac{1}{3n+1} \cdot \mathbb{E} \left(\left(\frac{1}{2} \right)^{d(b,b^*)} \right) = \frac{1}{3n+1} \cdot \mathbb{E} \left(\left(\frac{1}{2} \right)^{\sum_{i=1}^n X_i} \right) \\ &= \frac{1}{3n+1} \cdot \mathbb{E} \left(\prod_{i=1}^n \left(\frac{1}{2} \right)^{X_i} \right) = \frac{1}{3n+1} \cdot \prod_{i=1}^n \mathbb{E} \left(\left(\frac{1}{2} \right)^{X_i} \right) \\ &= \frac{1}{3n+1} \cdot \prod_{i=1}^n \left(\text{Prob}(X_i = 0) + \text{Prob}(X_i = 1) \cdot \frac{1}{2} \right) \\ &= \frac{1}{3n+1} \cdot \prod_{i=1}^n \left(\frac{1}{2} + \frac{1}{4} \right) = \frac{1}{3n+1} \cdot \left(\frac{3}{4} \right)^n \end{aligned}$$



Noch einmal intensiv nachdenken. . .

Erinnerung Wahl der Anfangsbelegung entscheidend

Erinnerung

- ein fester Startpunkt (Algorithmus 10.15) $\rightsquigarrow O(\text{poly}(n) \cdot 2^n)$
- zwei feste Startpunkte mit max. Hammingabstand (Algorithmus 10.21) $\rightsquigarrow O(\text{poly}(n) \cdot 1,42^n)$
- zufällige Startpunkte (Algorithmus 10.23) $\rightsquigarrow O(\text{poly}(n) \cdot (4/3)^n)$

Beobachtung Wahl der Anfangsbelegung
ohne Betrachten der Eingabe

Hoffnung durch Berücksichtigung der Eingabe
nachweislich bessere Anfangsbelegung findbar

Ein Blick auf die 3-SAT-Instanz

Annahme wieder nur positive Literale
sonst von $\overline{x_i}$ zu $\hat{x}_i = \overline{x_i}$ übergehen

Betrachte Klausel $c_j = x_1 \vee x_2 \vee x_3$

klar uniform zufällige Belegung
 $\Rightarrow \text{Prob}(c_j \text{ nicht erfüllt}) = \text{Prob}(x_1 = x_2 = x_3 = 0) = 1/8$

besser Belegung $x_1 = x_2 = x_3 = 0$ vermeiden

klar für eine Klausel einfach und fast sinnlos

Wunsch für möglichst viele Klauseln durchführen

Problem Klauseln sind nicht unabhängig

Unabhängige Klauseln

Definition

Zwei Klauseln heißen **unabhängig**, wenn sie keine Variable gemeinsam haben.

Eine Menge von Klauseln heißt **unabhängig**, wenn alle Klauseln der Menge paarweise unabhängig sind.

Eine Menge von Klauseln M heißt **inklusionsmaximal unabhängig** für eine 3-SAT-Instanz c_1, c_2, \dots, c_m , wenn M unabhängig ist und für jede Klausel $c \in \{c_1, c_2, \dots, c_m\} \setminus M$ gilt, dass $M \cup \{c\}$ nicht unabhängig ist.

klar inklusionsmaximale unabhängige Menge M greedy **berechenbar** in Polynomialzeit

wie gehabt **Annahme** alle Literale positiv in M
sonst Übergang zu $\hat{x}_i = \overline{x_i}$

Unabhängige Klauseln belegen

Parameter inklusionsmaximale unabhängige Menge C'

$$p_1, p_2, p_3 \in [0; 1] \text{ mit } 3p_1 + 3p_2 + p_3 = 1$$

AssignIndependentClauses(C', p_1, p_2, p_3)

1. Für jede Klausel $c_j = \hat{x}_{j_1} \vee \hat{x}_{j_2} \vee \hat{x}_{j_3} \in C'$
2. Führe ein Zufallsexperiment mit drei Ausgängen durch:
3. Mit Wahrscheinlichkeit $3p_1$
4. Wähle $(b[j_1], b[j_2], b[j_3])$ uniform zufällig aus $\{(0, 0, 1), (0, 1, 0), (1, 0, 0)\}$.
5. Mit Wahrscheinlichkeit $3p_2$
6. Wähle $(b[j_1], b[j_2], b[j_3])$ uniform zufällig aus $\{(0, 1, 1), (1, 0, 1), (1, 1, 0)\}$.
7. Mit Wahrscheinlichkeit p_3
8. Wähle $(b[j_1], b[j_2], b[j_3]) = (1, 1, 1)$.

Berücksichtigung unabhängiger Klauseln

Algorithmus 10.25

1. Für alle $i \in \{1, 2, \dots, T\}$
2. Berechne inklusionsmaximale unabhängige Klauselmenge C' .
3. Berechne eine initiale Belegung $b \in \{0, 1\}^n$ durch
 AssignIndependentClauses(C', p_1, p_2, p_3)
 Uniform zufällige Wahl für die restlichen Variablen.
4. RandomWalk($3n + 1, b$)
5. If keine erfüllende Belegung gefunden
 Then Ausgabe „Es gibt vermutlich keine erfüllende Belegung.“

Bringt das asymptotisch eine Laufzeitverbesserung?

Lässt sich das überhaupt noch geschätzt abschätzen?

Analyse von Algorithmus 10.25

Definition Ereignis B
 = RandomWalk-Aufruf findet erf. Belegung

wie bisher $\text{Prob}(B) \geq \frac{1}{3n+1} \cdot \mathbb{E} \left(\left(\frac{1}{2} \right)^{d(b,b^*)} \right)$

noch offen Abschätzung von $\mathbb{E} \left(\left(\frac{1}{2} \right)^{d(b,b^*)} \right)$

Beobachtung $b[i]$ nicht alle unabhängig
 darum **schwieriger**

Was tun?

klar Abhängigkeiten genauer betrachten

Betrachtung der Abhängigkeiten

Annahme C' enthält genau $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{3|C'|}$
(nur Notation)

Beobachtungen

- alle \hat{x}_i mit $i > 3|C'|$ vollständig unabhängig
- Variablen in C' in Dreiergruppen einteilbar: innerhalb der Gruppen abhängig, die Gruppen jeweils vollständig unabhängig

Annahme abhängige Dreiergruppen sind $\hat{x}_{3i+1}, \hat{x}_{3i+2}, \hat{x}_{3i+3}$
mit $i \in \{0, 1, \dots, |C'| - 1\}$ (nur Notation)

passend dazu „Indikatorvariable“ $X_{3i+1,3i+2,3i+3}$ mit
 $X_{3i+1,3i+2,3i+3} = d(b[3i+1]b[3i+2]b[3i+3],$
 $b^*[3i+1]b^*[3i+2]b^*[3i+3])$

klar $d(b, b^*) = \sum_{i=0}^{|C'|-1} X_{3i+1,3i+2,3i+3} + \sum_{i=3|C'|+1}^n X_i$

Erfolgswahrscheinlichkeit abschätzen

Wir haben
$$d(b, b^*) = \sum_{i=0}^{|C'|-1} X_{3i+1, 3i+2, 3i+3} + \sum_{i=3|C'|+1}^n X_i$$

Prob(B)

$$\begin{aligned} &\geq \frac{1}{3n+1} \cdot \mathbb{E} \left(\left(\frac{1}{2} \right)^{d(b, b^*)} \right) \\ &= \frac{1}{3n+1} \cdot \left(\prod_{i=0}^{|C'|-1} \mathbb{E} \left(\left(\frac{1}{2} \right)^{X_{3i+1, 3i+2, 3i+3}} \right) \right) \cdot \left(\prod_{i=3|C'|+1}^n \mathbb{E} \left(\left(\frac{1}{2} \right)^{X_i} \right) \right) \\ &\geq \frac{1}{3n+1} \cdot \left(\prod_{i=0}^{|C'|-1} \mathbb{E} \left(\left(\frac{1}{2} \right)^{X_{3i+1, 3i+2, 3i+3}} \right) \right) \cdot \left(\frac{3}{4} \right)^{n-3|C'|} \end{aligned}$$

Erfolgsw'keit in der inklusionsmax. unabhängigen Menge

Ziel $E\left(\left(\frac{1}{2}\right)^{X_{3i+1,3i+2,3i+3}}\right)$ abschätzen

1. Fall $b^*[3i+1] + b^*[3i+2] + b^*[3i+3] = 3$

$$\begin{aligned}
 & E\left(\left(\frac{1}{2}\right)^{X_{3i+1,3i+2,3i+3}}\right) \\
 = & p_3 \cdot \left(\frac{1}{2}\right)^0 + 3p_2 \cdot \left(\frac{1}{2}\right)^1 + 3p_1 \cdot \left(\frac{1}{2}\right)^2 \\
 = & p_3 + \frac{3}{2}p_2 + \frac{3}{4}p_1
 \end{aligned}$$

Erfolgsw'keit in der inklusionsmax. unabhängigen Menge

Ziel $E\left(\left(\frac{1}{2}\right)^{X_{3i+1,3i+2,3i+3}}\right)$ abschätzen

2. Fall $b^*[3i+1] + b^*[3i+2] + b^*[3i+3] = 1$

$$\begin{aligned}
 & E\left(\left(\frac{1}{2}\right)^{X_{3i+1,3i+2,3i+3}}\right) \\
 = & p_1 \cdot \left(\frac{1}{2}\right)^0 + 2p_2 \cdot \left(\frac{1}{2}\right)^1 + (2p_1 + p_3) \cdot \left(\frac{1}{2}\right)^2 + p_2 \cdot \left(\frac{1}{2}\right)^3 \\
 = & \frac{1}{4}p_3 + \frac{9}{8}p_2 + \frac{3}{2}p_1
 \end{aligned}$$

Erfolgsw'keit in der inklusionsmax. unabhängigen Menge

Ziel $E\left(\left(\frac{1}{2}\right)^{X_{3i+1,3i+2,3i+3}}\right)$ abschätzen

3. Fall $b^*[3i+1] + b^*[3i+2] + b^*[3i+3] = 2$

$$\begin{aligned} & E\left(\left(\frac{1}{2}\right)^{X_{3i+1,3i+2,3i+3}}\right) \\ &= p_2 \cdot \left(\frac{1}{2}\right)^0 + (p_3 + 2p_1) \cdot \left(\frac{1}{2}\right)^1 + 2p_2 \cdot \left(\frac{1}{2}\right)^2 + p_1 \cdot \left(\frac{1}{2}\right)^3 \\ &= \frac{1}{2}p_3 + \frac{3}{2}p_2 + \frac{9}{8}p_1 \end{aligned}$$

4. Fall $b^*[3i+1] + b^*[3i+2] + b^*[3i+3] = 0$

klar kommt nicht vor ✓

Erfolgsw'keit in der inklusionsmax. unabhängigen Menge

Wir haben $E \left(\left(\frac{1}{2} \right)^{X_{3i+1, 3i+2, 3i+3}} \right)$

$$= \begin{cases} p_3 + \frac{3}{2}p_2 + \frac{3}{4}p_1 & \text{falls } \sum_{j=3i+1}^{3j+3} b^*[j] = 3 \\ \frac{1}{4}p_3 + \frac{9}{8}p_2 + \frac{3}{2}p_1 & \text{falls } \sum_{j=3i+1}^{3j+3} b^*[j] = 1 \\ \frac{1}{2}p_3 + \frac{3}{2}p_2 + \frac{9}{8}p_1 & \text{falls } \sum_{j=3i+1}^{3j+3} b^*[j] = 2 \end{cases}$$

und brauchen $3p_1 + 3p_2 + p_3 = 1$

Beobachtung $p_1 := 4/21, p_2 := 2/21, p_3 := 3/21$
 $\Rightarrow E \left(\left(\frac{1}{2} \right)^{X_{3i+1, 3i+2, 3i+3}} \right) = \frac{3}{7}$

Wieso denn $\frac{3}{7}$? Wir hatten vorher doch schon $\frac{3}{4}$!

Erinnerung $\frac{3}{7}$ für drei Variable

und $\frac{3}{7} > 0,42857 > 0,421875 = \frac{27}{64} = \left(\frac{3}{4} \right)^3$

Ergebnis zusammensetzen

Erinnerung Ereignis B

= RandomWalk-Aufruf findet erf. Belegung

$$\text{Prob}(B) \geq \frac{1}{3n+1} \cdot \left(\prod_{i=0}^{|C'|-1} \mathbb{E} \left(\left(\frac{1}{2} \right)^{X_{3i+1,3i+2,3i+3}} \right) \right) \cdot \left(\frac{3}{4} \right)^{n-3|C'|}$$

$$\geq \frac{1}{3n+1} \cdot \left(\frac{3}{4} \right)^{n-3|C'|} \cdot \left(\frac{3}{7} \right)^{|C'|}$$

$$= \frac{1}{3n+1} \cdot \left(\frac{3}{4} \right)^n \cdot \left(\frac{64}{63} \right)^{|C'|}$$

mit der Wahl $p_1 = 4/21, p_2 = 2/21, p_3 = 3/21$

Beobachtung für große C' asymptotisch besser
aber auch nur dann

Problem C' kann für jede Wahl **klein** sein

Kleine inklusionsmax. unabhängige Mengen

Beobachtung C' inklusionsmaximal unabhängig
 \Rightarrow jedes $c_j \notin C'$ enthält Variable $\in C'$

Beobachtung nach Belegung der Variablen aus C'
 diese Variablen auch außerhalb von C' konstant
 also außerhalb von C' nur noch 2-SAT
 nach Vereinfachung

klar Löse 2-SAT-Restinstanz deterministisch optimal
 in Polynomialzeit.

Ein Algorithmus für kleine C'

Algorithmus 10.26

1. AssignIndependentClauses(q_1, q_2, q_3)
2. Führe die Konstantsetzungen in allen Klauseln durch.
3. Löse die entstandene 2-SAT-Instanz det. in pol. Zeit.

Beobachtung Erfolgsw'keit von Algorithmus 10.26
 $\geq \text{Prob}(\text{Belegung aus AssIndCl}(q_1, q_2, q_3) = b^*)$
 $\geq \min\{p_1, p_2, p_3\}^{|C'|}$

Erinnerung brauchen $3p_1 + 3p_2 + p_3 = 1$

also $p_1 = p_2 = p_3 = 1/7$ gute Wahl
damit Erfolgsw'keit $\geq (1/7)^{|C'|}$

klar gut für kleine C'

naheliegend guter allgemeiner Algorithmus durch **Kombination**

Verbesserter 3-SAT-Algorithmus

Algorithmus 10.27

1. Algorithmus 10.25 mit $T = 1$, $p_1 = 4/21$, $p_2 = 2/21$, $p_3 = 3/21$.
2. Algorithmus 10.26 mit $q_1 = q_2 = q_3 = 1/7$.

Theorem 10.28

Algorithmus 10.27 berechnet zur einer erfüllbaren 3-SAT-Instanz über n Variablen eine erfüllende Belegung mit Wahrscheinlichkeit $> 0,75177^n$ eine erfüllende Belegung. Er lässt sich durch Wiederholung zu einem Algorithmus mit erwarteter Laufzeit $O(\text{poly}(n) \cdot 1,3302^n)$ ausbauen.

Beweis von Theorem 10.28

Wir wissen Erfolgsw'keit von Algorithmus 10.27

$$\geq \frac{1}{3n+1} \cdot \max \left\{ \left(\frac{1}{7}\right)^{|C'|}, \left(\frac{3}{4}\right)^n \cdot \left(\frac{64}{63}\right)^{|C'|} \right\}$$

Beobachtung $\left(\frac{1}{7}\right)^{|C'|}$ fällt streng monoton
 $\left(\frac{3}{4}\right)^n \cdot \left(\frac{64}{63}\right)^{|C'|}$ wächst streng monoton

also zur Abschätzung bestimme $|C'|$ mit

$$\begin{aligned} \left(\frac{1}{7}\right)^{|C'|} &= \left(\frac{3}{4}\right)^n \cdot \left(\frac{64}{63}\right)^{|C'|} \\ \Leftrightarrow \left(\frac{63}{7 \cdot 64}\right)^{|C'|} &= \left(\frac{3}{4}\right)^n \\ \Leftrightarrow |C'| &= n \cdot \frac{\log(3/4)}{\log(63/448)} \approx 0,1446625247n \end{aligned}$$

also Erfolgsw'keit $\geq \left(\frac{1}{7}\right)^{0,144662524n} \approx 0,75177^n \checkmark$

also erwartete Laufzeit
 $= O(\text{poly}(n) \cdot 0,75177^{-n}) = O(\text{poly}(n) \cdot 1,3302^n) \quad \square$

Fazit 3-SAT-Algorithmen

Anmerkung analog zu 2-SAT

Ausbau zu „endliche Laufzeit, kleine Fehlerw'keit“ möglich

Haben wir überhaupt Wesentliches erreicht?

Rechenzeit bei 1 000 000 Belegungen pro Sekunde (nur exp. Anteil)

n	bei 2^n	bei $1,42^n$	bei $(4/3)^n$	bei $1,3302^n$
10	0,001 Sek.	0,000 Sek.	0,000 Sek.	0,000 Sek.
20	1,048 Sek.	0,001 Sek.	0,000 Sek.	0,000 Sek.
30	17,895 Min.	0,033 Sek.	0,006 Sek.	0,005 Sek.
40	12,726 Tage	1,049 Sek.	0,099 Sek.	0,091 Sek.
50	35,702 Jahre	33,554 Sek.	1,766 Sek.	1,570 Sek.
60	36 558,9 Jahre	17,896 Min.	31,356 Sek.	27,228 Sek.
70	> 37 436 300 Jahre	9,544 Std.	9,280 Min.	7,871 Min.
100	> $4 \cdot 10^{16}$ Jahre	35,702 Jahre	36,089 Tage	28,522 Tage