

Randomisierte Algorithmen

4. Routing in Hyperwürfeln

Thomas Worsch

Fakultät für Informatik
Karlsruher Institut für Technologie

Wintersemester 2016/2017

Überblick

Das Problem und ein deterministischer Algorithmus

Markov- und Chebyshev-Ungleichung

Chernoff-Schranken

Erster randomisierter Algorithmus

Die probabilistische Methode

Zweiter „randomisierter Algorithmus“

Überblick

Das Problem und ein deterministischer Algorithmus

Markov- und Chebyshev-Ungleichung

Chernoff-Schranken

Erster randomisierter Algorithmus

Die probabilistische Methode

Zweiter „randomisierter Algorithmus“

4.1 Definition Hyperwürfel

- ▶ $d \geq 1$: *d-dimensionaler Hyperwürfel* $H_d = (V_d, E_d)$ mit
- ▶ Knotenmenge $V_d = \{0, 1\}^d$ und
- ▶ Kantenmenge E_d : Knoten x und y verbunden, wenn Hamming-Distanz 1

4.2 Beobachtungen

- ▶ $|V_d| = N = 2^d$
- ▶ $|E_d| = d \cdot 2^{d-1} \in \Theta(N \log N)$
- ▶ H_d hat Durchmesser

4.2 Beobachtungen

- ▶ $|V_d| = N = 2^d$
- ▶ $|E_d| = d \cdot 2^{d-1} \in \Theta(N \log N)$
- ▶ H_d hat Durchmesser $d = \log N$, denn
für $x = (x_1 x_2 \cdots x_d)$ und $y = (y_1 y_2 \cdots y_d)$

4.2 Beobachtungen

- ▶ $|V_d| = N = 2^d$
- ▶ $|E_d| = d \cdot 2^{d-1} \in \Theta(N \log N)$
- ▶ H_d hat Durchmesser $d = \log N$, denn für $x = (x_1 x_2 \cdots x_d)$ und $y = (y_1 y_2 \cdots y_d)$ ist

$$(x_1 x_2 x_3 \cdots x_{d-1} x_d)$$

$$(y_1 x_2 x_3 \cdots x_{d-1} x_d)$$

$$(y_1 y_2 x_3 \cdots x_{d-1} x_d)$$

$$\vdots$$

$$(y_1 y_2 y_3 \cdots y_{d-1} x_d)$$

$$(y_1 y_2 y_3 \cdots y_{d-1} y_d)$$

ein Weg von x nach y (nach Entfernung aller Doppelten).

4.3 Problemstellung

- ▶ Knoten von H_d seien Prozessoren.
- ▶ *Routing*: Jeder Knoten x habe „Nachricht“ (oder „Paket“) vorliegen, die auf einem Pfad in H_d zu Zielknoten $f(x)$ transportiert werden muss.
- ▶ *Permutationsrouting*: $f: V \rightarrow V$ ist eine Bijektion, beschreibt also eine Permutation der Knoten.

4.3 Problemstellung

- ▶ Knoten von H_d seien Prozessoren.
- ▶ **Routing**: Jeder Knoten x habe „Nachricht“ (oder „Paket“) vorliegen, die auf einem Pfad in H_d zu Zielknoten $f(x)$ transportiert werden muss.
- ▶ **Permutationsrouting**: $f: V \rightarrow V$ ist eine Bijektion, beschreibt also eine Permutation der Knoten.
- ▶ Einschränkung: in jedem Schritt kann über jede Kante maximal ein Paket transportiert werden.
- ▶ für Staus: FIFO-Warteschlangen
- ▶ **gesucht**: für jedes Paar $(x, f(x))$ ein „Reiseplan“ (Kanten, Zeitpunkte) von x nach $f(x)$, so dass möglichst schnell alle Pakete am Ziel

4.4 Einschränkung

Zur Vermeidung einer zentralen Instanz, die die Transporte organisieren muss:

- ▶ hier nur Algorithmen, die *oblivious* bzw. *datenunabhängig* sind
- ▶ Das heißt: die Route für Paket x hängt nicht von den Routen anderer Pakete ab.
- ▶ Algorithmus also vollständig charakterisiert durch die N^2 denkbaren Pfade $P_{x,y}$

4.5 Bit-Fixing-Algorithmus

Von $x = (x_1x_2 \cdots x_d)$ nach $y = (y_1y_2 \cdots y_d)$ benutze

- ▶ Pfad, der sich aus

$$(x_1x_2x_3 \cdots x_{d-1}x_d)$$

$$(y_1x_2x_3 \cdots x_{d-1}x_d)$$

$$(y_1y_2x_3 \cdots x_{d-1}x_d)$$

$$\vdots$$

$$(y_1y_2y_3 \cdots y_{d-1}x_d)$$

$$(y_1y_2y_3 \cdots y_{d-1}y_d)$$

durch Streichen aller Doppelten ergibt.

- ▶ datenunabhängig

4.6 „Matrix-Transposition“ mit dem Bit-Fixing-Algorithmus

- ▶ $f(x_1 \cdots x_{d/2} x_{d/2+1} \cdots x_d) = (x_{d/2+1} \cdots x_d x_1 \cdots x_{d/2})$
- ▶ Problem: fixiere beliebiges Bitmuster $z_{d/2+1} \cdots z_d$.
- ▶ Transport der $2^{d/2} = \sqrt{N}$ Pakete
 von allen Knoten $x_1 \cdots x_{d/2} z_{d/2+1} \cdots z_d$
 über gleichen Knoten $z_{d/2+1} \cdots z_d z_{d/2+1} \cdots z_d$
- ▶ pro Schritt maximal $d = \log N$ Pakete transportierbar
- ▶ also untere Schranke von $\sqrt{N}/\log N$ Schritten
 - ▶ viel größer als der Durchmesser $\log N$

4.7 Satz

Zu jedem deterministischen datenunabhängigen Algorithmus für Permutationsrouting in einem Graphen mit N Knoten, die alle Ausgangsgrad d haben, gibt es eine Permutation, für die der Algorithmus $\Omega(\sqrt{N}/d)$ Schritte benötigt.

4.8 Beweis

- ▶ A : deterministischer datenunabhängiger Algorithmus für Permutationsrouting
- ▶ $P_{u,v}$: von A für ein Paket von u nach v gewählter Pfad
- ▶ Beweisidee:
 - ▶ finde eine Kante e
 - ▶ für die „große“ Mengen von Quellknoten $U' = \{u_1, \dots, u_k\}$ und zugehörigen Zielknoten $V' = \{v_1, \dots, v_k\}$ existieren
 - ▶ so dass alle Pfade P_{u_i, v_i} über e führen.
- ▶ Da in jedem Schritt über e nur je ein Paket in jede Richtung transportiert kann, folgt untere Schranke von $k/2$.
- ▶ Wir werden sehen, dass man $k = \sqrt{N}/d$ solche Pfade finden kann.

4.8 Beweis (2)

- ▶ Betrachte beliebigen Knoten v und alle $N - 1$ Pfade $P_{u,v}$ von anderen u zu ihm.
- ▶ Für $k \geq 1$ sei $S_k(v)$ die Menge aller Kanten, durch die mindestens k dieser Pfade führen.
- ▶ $S_k^*(v)$ sei die Menge aller Endknoten der Kanten in $S_k(v)$.
- ▶ Offensichtlich ist $|S_k^*(v)| \leq 2|S_k(v)|$.

4.8 Beweis (2)

- ▶ Betrachte beliebigen Knoten v und alle $N - 1$ Pfade $P_{u,v}$ von anderen u zu ihm.
- ▶ Für $k \geq 1$ sei $S_k(v)$ die Menge aller Kanten, durch die mindestens k dieser Pfade führen.
- ▶ $S_k^*(v)$ sei die Menge aller Endknoten der Kanten in $S_k(v)$.
- ▶ Offensichtlich ist $|S_k^*(v)| \leq 2|S_k(v)|$.
- ▶ Da $N - 1$ Pfade zu v hinführen, aber nur d Kanten, müssen über mindestens eine dieser Kanten mindestens $\frac{N-1}{d}$ Pfade führen.
- ▶ Also ist für $k \leq \frac{N-1}{d}$ auch $v \in S_k^*(v)$.
- ▶ Von nun an stets $k \leq \frac{N-1}{d}$ und daher $v \in S_k^*(v)$.

4.8 Beweis (3)

- ▶ Zeige:

$$|V \setminus S_k^*(v)| \leq (d-1)(k-1)|S_k^*(v)| \quad (1)$$

- ▶ Wegen der eben gemachten Annahme führt jeder Pfad $P_{u,v}$ von einem Knoten $u \in V \setminus S_k^*(v)$ „nach $S_k^*(v)$ hinein“.
- ▶ Für jeweils erstes „Hineinführen“ über eine Kante $(w, w') \in V \setminus S_k^*(v) \times S_k^*(v)$ gilt:
 - ▶ Es gibt $|S_k^*(v)|$ mögliche w' .
 - ▶ zu jedem w' maximal $d-1$ Kanten „von außerhalb“
 - ▶ über solche Kante (w, w') führen höchstens $k-1$ Pfade
- ▶ folglich „außerhalb“ von $S_k^*(v)$, in $V \setminus S_k^*(v)$, nur in (1) behauptete Anzahl von Knoten

4.8 Beweis (4)

- ▶ Folglich gilt für jedes $k \leq (N - 1)/d$:

$$\begin{aligned} N &= |V \setminus S_k^*(v)| + |S_k^*(v)| \\ &\leq (d - 1)(k - 1)|S_k^*(v)| + |S_k^*(v)| \\ &\leq ((d - 1)(k - 1) + 1) \cdot |S_k^*(v)| \\ &\leq 2kd|S_k(v)| \end{aligned}$$

und daher $|S_k(v)| \geq \frac{N}{2kd}$

- ▶ Summation über alle Knoten ergibt

$$\sum_{v \in V} |S_k(v)| \geq \frac{N^2}{2kd} .$$

4.8 Beweis (5)

- ▶ Da es aber maximal $Nd/2$ Kanten im Graphen gibt, muss mindestens eine Kante in mindestens

$$\frac{N^2/2kd}{Nd/2} = \frac{N}{kd^2}$$

Mengen $S_k(v)$ vorkommen.

- ▶ Wir wählen nun k so, dass diese Anzahl gerade wieder k ist, also $k = \sqrt{N}/d$.
- ▶ Dieses k ist kleiner gleich $(N - 1)/d$.
- ▶ Es sei nun e eine Kante, die in $k = \sqrt{N}/d$ Mengen $S_k(v_1), \dots, S_k(v_k)$ liegt.

4.8 Beweis (6)

- ▶ Es sei nun e eine Kante, die in $k = \sqrt{N}/d$ Mengen $S_k(v_1), \dots, S_k(v_k)$ liegt.
- ▶ Es sei u_1 einer der k Knoten, für die P_{u_1, v_1} über e führt.
- ▶ Nach Wahl von k gibt es zu jedem v_i mindestens k Knoten, für die P_{u_i, v_i} über e führt.
- ▶ Daher können wir induktiv u_i festlegen, indem wir verlangen, dass u_i einer der mindestens $k - (i - 1)$ Knoten ungleich u_1, \dots, u_{i-1} sei, für die P_{u_i, v_i} über e führt.
- ▶ Also gibt es mindestens $k = \sqrt{N}/d$ Pfade $P_{u_1, v_1}, \dots, P_{u_k, v_k}$, die alle über die gleiche Kante e führen.

4.9 Bemerkung

- ▶ *Frage:* Gibt es zumindest einen deterministischen datenunabhängigen Algorithmus, für den nur sehr wenige Permutationen tatsächlich „sehr schlimm“ sind?
- ▶ *Antwort:* Für jeden deterministischen datenunabhängigen Algorithmus gibt es sogar $(\sqrt{N}/d)!$ Permutationen, die mindestens $\sqrt{N}/2d$ Routingschritte nötig machen.

Überblick

Das Problem und ein deterministischer Algorithmus

Markov- und Chebyshev-Ungleichung

Chernoff-Schranken

Erster randomisierter Algorithmus

Die probabilistische Methode

Zweiter „randomisierter Algorithmus“

4.10 Satz (Markov-Ungleichung)

Es sei Y eine Zufallsvariable mit Erwartungswert $\mathbf{E}[Y]$, die nur nichtnegative Werte annahme.

Dann gilt für alle $t, k \in \mathbb{R}_+$:

$$\mathbf{P}(Y \geq t) \leq \frac{\mathbf{E}[Y]}{t} \quad \text{bzw.} \quad \mathbf{P}(Y \geq k\mathbf{E}[Y]) \leq \frac{1}{k} .$$

4.11 Beweis

- ▶ betrachte Zufallsvariable

$$X = \begin{cases} 0 & \text{falls } Y < t \\ t & \text{falls } Y \geq t \end{cases}$$

- ▶ dann $X \leq Y$ und $\mathbf{E}[X] \leq \mathbf{E}[Y]$
- ▶ also

$$t \cdot \mathbf{P}(Y \geq t) \leq \mathbf{E}[Y]$$

4.12 Satz (Chebyshev-Ungleichung)

Es sei X eine Zufallsvariable mit Erwartungswert μ_X und Standardabweichung σ_X . Dann gilt für alle $t \in \mathbb{R}_+$:

$$\mathbf{P}(|X - \mu_X| \geq t\sigma_X) \leq \frac{1}{t^2}$$

bzw.

$$\mathbf{P}(|X - \mu_X| \geq t) \leq \frac{\sigma_X^2}{t^2}.$$

4.13 Beweis

- ▶ Zufallsvariable $Y = (X - \mu_X)^2$ hat Erwartungswert $\mu_Y = \sigma_X^2$.
- ▶ Nach der Markov-Ungleichung:

$$\mathbf{P}(Y \geq t^2 \mu_Y) \leq \frac{1}{t^2} .$$

- ▶ Die linke Seite ist aber

$$\begin{aligned} \mathbf{P}(Y \geq t^2 \mu_Y) &= \mathbf{P}((X - \mu_X)^2 \geq t^2 \sigma_X^2) \\ &= \mathbf{P}(|X - \mu_X| \geq t \sigma_X) . \end{aligned}$$

Überblick

Das Problem und ein deterministischer Algorithmus

Markov- und Chebyshev-Ungleichung

Chernoff-Schranken

Erster randomisierter Algorithmus

Die probabilistische Methode

Zweiter „randomisierter Algorithmus“

4.14 Problemstellung

- ▶ Im folgenden stets: X_1, \dots, X_n *unabhängige* 0-1-Zufallsvariablen mit $\mathbf{P}(X_i = 1) = p_i$ für $1 \leq i \leq n$.
- ▶ Solche Zufallsvariablen heißen auch *Poisson-Versuche*.
- ▶ Außerdem sei $X = X_1 + \dots + X_n$ und $\mu = \mathbf{E}[X] = \sum_{i=1}^n p_i$.
- ▶ Falls alle $p_i = p$ sind, spricht man auch von *Bernoulli-Versuchen*, X ist dann binomialverteilt.
- ▶ *Gesucht*: Abschätzungen für Abweichungen von X vom Erwartungswert.

4.15 Satz

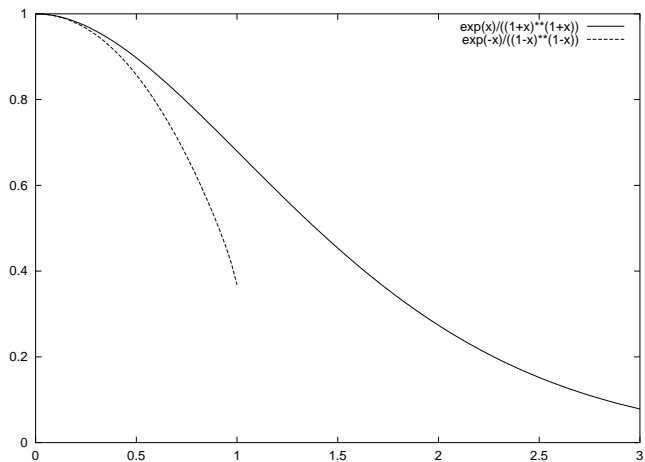
Mit den Bezeichnungen wie in Punkt 4.14 gilt:

- ▶ für $0 \leq \delta$:

$$\mathbf{P}(X \geq (1 + \delta)\mu) \leq \left(\frac{e^\delta}{(1 + \delta)^{(1+\delta)}} \right)^\mu$$

- ▶ für $1 > \delta \geq 0$ also $0 < 1 - \delta \leq 1$:

$$\mathbf{P}(X \leq (1 - \delta)\mu) \leq \left(\frac{e^{-\delta}}{(1 - \delta)^{(1-\delta)}} \right)^\mu$$



4.16 Beobachtung

- ▶ Für $x \geq 0$ ist $1 + x \leq e^x$.

4.17 Beweis (von Satz 4.15)

- ▶ sei t positiv; Markov-Ungleichung liefert:

$$\mathbf{P}(X \geq (1 + \delta)\mu) = \mathbf{P}\left(e^{tX} \geq e^{t(1+\delta)\mu}\right) \leq \frac{\mathbf{E}\left[e^{tX}\right]}{e^{t(1+\delta)\mu}}$$

- ▶ mit den X_i sind auch die e^{tX_i} unabhängig:

$$\mathbf{E}\left[e^{tX}\right] = \mathbf{E}\left[e^{t\sum X_i}\right] = \mathbf{E}\left[\prod e^{tX_i}\right] = \prod \mathbf{E}\left[e^{tX_i}\right]$$

- ▶ $\mathbf{E}\left[e^{tX_i}\right] = p_i \cdot e^t + (1 - p_i) \cdot 1 = 1 + p_i(e^t - 1) \leq e^{p_i(e^t - 1)}$

- ▶
$$\begin{aligned} \mathbf{P}(X \geq (1 + \delta)\mu) &\leq \frac{\prod e^{p_i(e^t - 1)}}{e^{t(1+\delta)\mu}} = \frac{e^{\sum p_i(e^t - 1)}}{e^{t(1+\delta)\mu}} \\ &= \frac{e^{\mu(e^t - 1)}}{e^{t(1+\delta)\mu}} = \left(\frac{e^{(e^t - 1)}}{e^{t(1+\delta)}}\right)^\mu \end{aligned}$$

- ▶ wähle $t = \ln(1 + \delta)$ (positiv!)

4.17 Beweis (2)

Fall $\mathbf{P}(X \leq (1 - \delta)\mu) \leq \dots$ für $1 > \delta \geq 0$

- ▶ analoge Rechnung
- ▶ wähle $t = -\ln(1 - \delta)$ (positiv!)

4.18 Bemerkung

- ▶ In Satz 4.15 von Interesse:

$$F(\mu, \delta) = \left(\frac{e^\delta}{(1 + \delta)^{(1+\delta)}} \right)^\mu \quad \text{für } \delta > -1$$

- ▶ Betrachte statt dessen:

$$\ln(F(\mu, \delta)^{1/\mu}) = \delta - (1 + \delta) \ln(1 + \delta) = f(\delta)$$

4.19 Lemma

1. Für $-1 < x \leq 0$ gilt: $f(x) \leq -x^2/2$.
2. Für $0 < x$ gilt: $-x^2/2 \leq f(x)$.
3. Die Funktion $g(x) = f(x)/x^2$ ist monoton wachsend.
4. Für $0 < \delta < x$ gilt: $f(\delta) \leq g(x)\delta^2$.
5. Für $0 < \delta < 2e - 1$ gilt: $f(\delta) \leq -\delta^2/5$.
6. Für $0 < \delta < 1$ gilt: $f(\delta) \leq -\delta^2/3$.

Beweis: rechnen

4.21 Korollar

- ▶ Für $0 \leq \delta \leq 2e - 1$ gilt:

$$\mathbf{P}(X \geq (1 + \delta)\mu) \leq \left(\frac{e^\delta}{(1 + \delta)^{(1+\delta)}} \right)^\mu \leq e^{-\delta^2 \mu / 5}$$

- ▶ Für $0 \leq \delta \leq 1$ gilt:

$$\mathbf{P}(X \geq (1 + \delta)\mu) \leq \left(\frac{e^\delta}{(1 + \delta)^{(1+\delta)}} \right)^\mu \leq e^{-\delta^2 \mu / 3}$$

- ▶ Für $1 > \delta \geq 0$ gilt:

$$\mathbf{P}(X \leq (1 - \delta)\mu) \leq \left(\frac{e^{-\delta}}{(1 - \delta)^{(1-\delta)}} \right)^\mu \leq e^{-\delta^2 \mu / 2}$$

4.23 Korollar

- ▶ Für $0 \leq \delta$ gilt:

$$\mathbf{P}(X \geq (1 + \delta)\mu) \leq \left(\frac{e}{1 + \delta}\right)^{(1+\delta)\mu}$$

- ▶ Für $2e - 1 \leq \delta$ gilt:

$$\mathbf{P}(X \geq (1 + \delta)\mu) \leq 2^{-(1+\delta)\mu}$$

Beweis: rechnen

Überblick

Das Problem und ein deterministischer Algorithmus

Markov- und Chebyshev-Ungleichung

Chernoff-Schranken

Erster randomisierter Algorithmus

Die probabilistische Methode

Zweiter „randomisierter Algorithmus“

4.25 Algorithmus

1. Für jedes Paket b_x in Startknoten x wird unabhängig und gleichverteilt zufällig ein Zwischenknoten z_x gewählt.
2. Unter Verwendung des Bit-Fixing-Algorithmus wird jedes b_x von x nach z_x transportiert.
3. Unter Verwendung des Bit-Fixing-Algorithmus wird jedes b_x von z_x zu Zielknoten $f(x)$ transportiert.

4.26 Satz

1. Die Wahrscheinlichkeit, dass jedes Paket seinen Zwischenknoten nach spätestens $7d$ Schritten erreicht hat, ist mindestens $1 - 2^{-5d}$.
2. Die Wahrscheinlichkeit, dass jedes Paket sein Ziel nach spätestens $14d$ Schritten erreicht hat, ist mindestens $1 - 2/N^5$.
3. Für $d \geq 3$ ist der Erwartungswert für die Laufzeit von Alg. 4.25 kleiner oder gleich $14d + 1$.

4.27 Lemma

- ▶ x Startknoten, beliebig aber fest,
 z_x zufällig gewählt
- ▶ $\rho_x = (e_1, e_2, \dots, e_k)$: „Bit-Fixing-Pfad“ von x nach z_x
- ▶ „ $\rho_y \cap \rho_x$ “: Menge der gemeinsamen Kanten zweier Pfade
- ▶ $S_x = \{b_y \mid y \neq x \wedge \rho_y \cap \rho_x \neq \emptyset\}$
- ▶ t : tatsächlicher Ankunftszeitpunkt von b_x in z_x ,
also aufgelaufene „Verspätung“ $\ell_x = t - k$
- ▶ **Behauptung:** $\ell_x \leq |S_x|$.

4.28 Beweis

- ▶ Zwei Pfade ρ_x und ρ_y mögen Kante gemeinsam haben.
- ▶ *Behauptung 1:* Sobald sich Pfade getrennt haben, führen sie nicht wieder zusammen.
- ▶ betrachte Knoten u und v auf Bit-Fixing-Pfad von $x \rightsquigarrow z_x$
- ▶ der Weg von u nach v ist eindeutig festgelegt (bit fixing!)
- ▶ unabhängig davon
 - ▶ von wo man zu u gekommen ist und
 - ▶ wohin es von v aus weiter geht

4.28 Beweis (2)

Sprechweisen:

- ▶ Paket $b_y \in S_x$ „*verlässt*“ ρ_x , wenn es zum letzten Mal eine Kante von ρ_x benutzt.
- ▶ Behauptung 1: dieser Zeitpunkt für jedes b_y eindeutig
- ▶ Ein Paket habe beim Transport über e_i von ρ_x „*Verspätung*“ ℓ , falls Transport erst in Schritt $t = i + \ell$
- ▶ Für b_x ist das wirklich die Zeitdifferenz zwischen frühest möglicher Ankunft $t = i$ und tatsächlicher Ankunft $t = i + \ell$ am Endpunkt von e_i
- ▶ Für andere $b_y \in S_x$ ist das nur „irgendeine“ Zahl.

4.28 Beweis (3)

- ▶ **Behauptung 2:** Jedes Mal, wenn sich die Verspätung von b_x von ℓ auf $\ell + 1$ erhöht, verlässt ein Paket $b_y \in S_x$ den Pfad ρ_x mit Verspätung ℓ .
- ▶ Wegen Behauptung 1 kann dies für jedes Paket in S_x nur ein einziges Mal passieren.
- ▶ Aus Behauptung 2 folgt *also*: $\ell_x \leq |S_x|$.

4.28 Beweis (4)

- ▶ Betrachte e_i , die von Paket b_x zu Zeitpunkt t benutzt werden möchte aber nicht kann, wegen anderem b_y
- ▶ Verspätung von b_x erhöht sich von $\ell = t - i$ auf $\ell + 1$, und das kann auch nur so passieren.
- ▶ „Verspätung“ von b_y bei Benutzung von e_i ist $t - i = \ell$
- ▶ Sei t' letzter Zeitpunkt, zu dem ein Paket b aus S_x Verspätung ℓ hat.
- ▶ sei $e_{j'}$ die Kante, die b benutzen „will“; also ist $t' - j' = \ell$.
- ▶ Dann verlässt auch ein Paket in S_x zu t' Pfad ρ_x :
 - ▶ Da b Kante $e_{j'}$ benutzen „will“,
 - ▶ wird ein Paket b' sicher Kante $e_{j'}$ benutzen.
 - ▶ Es hat offensichtlich Verzögerung $t' - j' = \ell$.

4.28 Beweis (5)

- ▶ Würde b' Pfad ρ_x *nicht* verlassen, dann gäbe es b'' , das Kante $e_{j'+1}$ zum Zeitpunkt $t' + 1$ mit Verzögerung $t' + 1 - (j' + 1) = \ell$ benutzen würde.
- ▶ Widerspruch zur Wahl von t' :
letzter Zeitpunkt, zu dem ein Paket Verspätung ℓ hat.
- ▶ Also verlässt b' Pfad ρ_x zum Zeitpunkt t' .
- ▶ Wir schreiben daher nun b' zu, bei Paket b_x die Erhöhung der Verspätung von ℓ auf $\ell + 1$ verursacht zu haben.
- ▶ Da b' den Pfad ρ_x verlässt und nie wieder betritt, wird so keinem Paket doppelt Verspätungserhöhung angerechnet.
- ▶ Also ist $\ell_x \leq |S_x|$.

4.29 Lemma

- ▶ Betrachte Zufallsvariable $H_{xy} = \begin{cases} 1 & \text{falls } \rho_x \cap \rho_y \neq \emptyset \\ 0 & \text{sonst} \end{cases}$.
- ▶ Dann gilt:
 1. Die Gesamtverspätung von b_x beim Eintreffen in z_x ist $\ell_x \leq \sum_{y \neq x} H_{xy}$.
 2. $\mathbf{E}[\sum_{y \neq x} H_{xy}] \leq d/2$.
 3. $\mathbf{P}(\ell_x \geq 6d) \leq 2^{-6d}$.

4.30 Beweis

1. Lemma von eben: $\ell_x \leq |S_x| = \sum_{y \neq x} H_{xy}$.
2. Betrachte irgendein $\rho_x = (e_1, \dots, e_k)$ mit $k \leq d$.
 - ▶ Die Zufallsvariable $T(e)$ gebe die Anzahl Pfade ρ_y mit $y \neq x$ an, die über eine Kante e führen.
 - ▶ Dann ist

$$\sum_{y \neq x} H_{xy} \leq \sum_{i=1}^k T(e_i)$$

und

$$\mathbf{E} \left[\sum_{y \neq x} H_{xy} \right] \leq \sum_{i=1}^k \mathbf{E}[T(e_i)]$$

- ▶ Zeige:

$$\mathbf{E}[T(e_i)] \leq 1/2$$

(wegen $k \leq d$ folgt die Behauptung)

4.30 Beweis (2)

- ▶ O. B. d. A. führe e_i von $(x_1 \cdots x_r 0 x_{r+2} \cdots x_d)$ zu $(x_1 \cdots x_r 1 x_{r+2} \cdots x_d)$.
- ▶ Bit-Fixing-Algorithmus: Ein Pfad von y nach z_y führt genau dann über e_i , wenn gilt:
 - ▶ $y = u_1 \cdots u_r 0 x_{r+2} \cdots x_d$ und
 - ▶ z_y beginnt mit $x_1 \cdots x_r 1$
- ▶ Solche $y \neq x$ gibt es $2^r - 1$.
- ▶ Die Zwischenknoten werden zufällig gleichverteilt und unabhängig gewählt,
- ▶ also ist für jedes y die Wahrscheinlichkeit für das Präfix $x_1 \cdots x_r 1$ in z_y stets $2^{-(r+1)}$.

4.30 Beweis (3)

Also ist

$$\begin{aligned} \mathbf{E}[T(e_i)] &= \sum_{y \neq x} \mathbf{P}(\rho_y \text{ benutzt } e_i) \\ &= \sum_{u_1 \cdots u_r \neq x_1 \cdots x_r} \mathbf{P}(\rho_{u_0 x_{r+2} \cdots x_d} \text{ benutzt } e_i) \\ &= (2^r - 1) \cdot 2^{-(r+1)} \leq 1/2 \end{aligned}$$

4.30 Beweis (4)

- ▶ wegen der ersten beiden Punkte: $\mathbf{E}[\ell_x] \leq d/2$
- ▶ man vergesse die $T(e_i)$ und denke wieder an die H_{xy}
 - ▶ also Chernoff-Schranken anwendbar
- ▶ da $2e - 1 < 11$, liefert Korollar 4.23

$$\begin{aligned}\mathbf{P}(\ell_x \geq 6d) &\leq \mathbf{P}\left(\sum H_{xy} \geq 6d\right) \\ &= \mathbf{P}\left(\sum H_{xy} \geq 12 \cdot d/2\right) \\ &\leq \mathbf{P}\left(\sum H_{xy} \geq (1 + 11)\mathbf{E}\left[\sum H_{xy}\right]\right) \\ &\leq 2^{-12d/2} = 2^{-6d}\end{aligned}$$

4.26 Satz (zur Erinnerung)

1. Die Wahrscheinlichkeit, dass jedes Paket seinen Zwischenknoten nach spätestens $7d$ Schritten erreicht hat, ist mindestens $1 - 2^{-5d}$.
2. Die Wahrscheinlichkeit, dass jedes Paket sein Ziel nach spätestens $14d$ Schritten erreicht hat, ist mindestens $1 - 2/N^5$.
3. Für $d \geq 3$ ist der Erwartungswert für die Laufzeit von Alg. 4.25 kleiner oder gleich $14d + 1$.

4.31 Beweis von Satz 4.26

1. Mit Wahrscheinlichkeit $1 - 2^{-5d}$ alle Pakete nach $\leq 7d$ Schritten am Zwischenknoten:
 - ▶ Wahrscheinlichkeit, dass Paket um $\geq 6d$ Schritte verzögert wird, ist $\leq 2^{-6d}$.
 - ▶ $N = 2^d$ Pakete unabhängig voneinander transportiert: Wahrscheinlichkeit, dass wenigstens eines um $\geq 6d$ Schritte verzögert wird, ist $\leq 2^d \cdot 2^{-6d} = 2^{-5d}$.
 - ▶ Zusätzlich jedes Paket über maximal d Kanten transportiert: Wahrscheinlichkeit, dass wenigstens ein Paket erst nach $\geq 7d$ Schritten am Ziel ist, $\leq 2^{-5d}$.
 - ▶ Also: mit Wahrscheinlichkeit $\geq 1 - 2^{-5d}$ alle Pakete nach $\leq 7d$ Schritten am Zwischenknoten.

4.31 Beweis von Satz 4.26 (2)

2. Mit Wahrscheinlichkeit mindestens $1 - 2/N^5$ alle Pakete nach $\leq 14d$ Schritten am Ziel:
- ▶ Zweite Phase: Umkehrung der ersten Phase.
 - ▶ Deshalb gilt getrennt hierfür die gleiche Analyse.
 - ▶ Damit bei Nacheinanderausführung beider Phasen durch ihre Überlappung keine zusätzlichen nachteiligen Effekte auftreten, Erweiterung der erste Phase:
 - ▶ Jedes Paket verharrt im Zwischenknoten, bis insgesamt seit Beginn des Routing $7d$ Schritte vergangen sind.
 - ▶ Wahrscheinlichkeit, dass alles nach $\leq 14d$ Schritten erledigt, mindestens
- $$(1 - 2^{-5d})(1 - 2^{-5d}) = 1 - 2/N^5 + 1/N^{10} \geq 1 - 2/N^5$$

4.31 Beweis von Satz 4.26 (3)

3. Für $d \geq 3$ ist die erwartete Laufzeit $\leq 14d + 1$.

- ▶ Die größte Laufzeit beschränkt durch Zeit für „sequentielles Routing“. Zeitbedarf dafür $\leq 2dN$.
- ▶ Für $d \geq 3$ ist $N \geq 8$, also $14d \leq 2dN$,
- ▶ also ist dann der Erwartungswert für die Laufzeit

$$\begin{aligned} &\leq (1 - 2/N^5)14d + (2/N^5) \cdot 2dN \\ &= 14d - 28d/N^5 + 4d/N^4 \\ &\leq 14d - 0 + 1/N^2 \\ &\leq 14d + 1 \end{aligned}$$

Überblick

Das Problem und ein deterministischer Algorithmus

Markov- und Chebyshev-Ungleichung

Chernoff-Schranken

Erster randomisierter Algorithmus

Die probabilistische Methode

Zweiter „randomisierter Algorithmus“

Die probabilistische Methode (1)

- ▶ Tragweite zuerst von Erdős erkannt
- ▶ *Nachweis der Existenz* gewisser Objekte

- ▶ Variante 1: Zufallsvariable X
 - ▶ hat reelle numerische Werte
 - ▶ $E[X]$ existiert

Die probabilistische Methode (1)

- ▶ Tragweite zuerst von Erdős erkannt
- ▶ *Nachweis der Existenz* gewisser Objekte

- ▶ Variante 1: Zufallsvariable X
 - ▶ hat reelle numerische Werte
 - ▶ $E[X]$ existiert

dann nimmt X

- ▶ mindestens einen Wert an, der nicht kleiner als $E[X]$ ist, und ebenso einen Wert, der nicht größer als $E[X]$ ist.

Die probabilistische Methode (1)

- ▶ Tragweite zuerst von Erdős erkannt
- ▶ *Nachweis der Existenz* gewisser Objekte

- ▶ Variante 1: Zufallsvariable X
 - ▶ hat reelle numerische Werte
 - ▶ $E[X]$ existiert

dann nimmt X

- ▶ mindestens einen Wert an, der nicht kleiner als $E[X]$ ist, und ebenso einen Wert, der nicht größer als $E[X]$ ist.
- ▶ Also *existieren* Ereignisse, für die X diese Werte annimmt.

Die probabilistische Methode (2)

- ▶ Variante 2: endliches nichtleeres Universum U
 - ▶ mit Objekten O
 - ▶ Eigenschaft P , die Objekte haben können

Die probabilistische Methode (2)

- ▶ Variante 2: endliches nichtleeres Universum U
 - ▶ mit Objekten O
 - ▶ Eigenschaft P , die Objekte haben können
- ▶ Wähle zufällig ein Objekt O aus U aus.
Wahrscheinlichkeit p , dass O Eigenschaft P hat?

Die probabilistische Methode (2)

- ▶ Variante 2: endliches nichtleeres Universum U
 - ▶ mit Objekten O
 - ▶ Eigenschaft P , die Objekte haben können
- ▶ Wähle zufällig ein Objekt O aus U aus.
Wahrscheinlichkeit p , dass O Eigenschaft P hat?
 - ▶ Wenn $p > 0$ ist,

Die probabilistische Methode (2)

- ▶ Variante 2: endliches nichtleeres Universum U
 - ▶ mit Objekten O
 - ▶ Eigenschaft P , die Objekte haben können
- ▶ Wähle zufällig ein Objekt O aus U aus.
Wahrscheinlichkeit p , dass O Eigenschaft P hat?
 - ▶ Wenn $p > 0$ ist,
 - ▶ dann muss in U ein O mit $P(O)$ *existieren*.

Überblick

Das Problem und ein deterministischer Algorithmus

Markov- und Chebyshev-Ungleichung

Chernoff-Schranken

Erster randomisierter Algorithmus

Die probabilistische Methode

Zweiter „randomisierter Algorithmus“

Plan

- ▶ *Sprechweisen:*
 - ▶ *RPH-Algorithmus:* datenunabhängiger randomisierter Algorithmus für Permutationsrouting in Hyperwürfeln
 - ▶ RPH-Algorithmus *schnell:* erwartete Laufzeit in $O(d)$.
- ▶ *Bisher:*
 - ▶ Algorithmus 4.25 nutzt $\Theta(Nd)$ Zufallsbits und ist schnell
 - ▶ Datenunabhängige deterministische Algorithmen nutzen 0 Zufallsbits und sind nie schnell. (Satz 4.7)
- ▶ *Frage:* Gibt es RPH-Algorithmen, die weniger als $\Theta(Nd)$ Zufallsbits benutzen und trotzdem schnell sind?

Plan

- ▶ *Sprechweisen:*
 - ▶ *RPH-Algorithmus:* datenunabhängiger randomisierter Algorithmus für Permutationsrouting in Hyperwürfeln
 - ▶ RPH-Algorithmus *schnell:* erwartete Laufzeit in $O(d)$.
- ▶ *Bisher:*
 - ▶ Algorithmus 4.25 nutzt $\Theta(Nd)$ Zufallsbits und ist schnell
 - ▶ Datenunabhängige deterministische Algorithmen nutzen 0 Zufallsbits und sind nie schnell. (Satz 4.7)
- ▶ *Frage:* Gibt es RPH-Algorithmen, die weniger als $\Theta(Nd)$ Zufallsbits benutzen und trotzdem schnell sind?
- ▶ *Ziel:* $\Theta(d)$ Zufallsbits
 - ▶ notwendig und
 - ▶ *in gewissem Sinne* „hinreichend“, um sein schnell zu können

4.32 Satz

Wenn ein RPH-Algorithmus in Würfeln mit $N = 2^d$ Knoten nur k Zufallsbits benutzt, dann ist seine erwartete Laufzeit in $\Omega(2^{-k} \sqrt{N}/d)$.

4.33 Beweis

- ▶ Fasse RPH-Algorithmus R auf als Wahrscheinlichkeitsverteilung über 2^k deterministischen Algorithmen.
- ▶ Dann wird mindestens ein det. Alg A mit Wahrscheinlichkeit $\geq 2^{-k}$ ausgewählt.
- ▶ Es sei x eine Eingabe, für die A Laufzeit $\Omega(\sqrt{N}/d)$ hat.
- ▶ Betrachte die Bearbeitung von x durch R :
 - ▶ Mit Wahrscheinlichkeit $\geq 2^{-k}$ wird R wie A arbeiten.
 - ▶ Also ist der Erwartungswert für die Laufzeit von R mindestens $\Omega(2^{-k} \sqrt{N}/d)$.

4.34 Korollar

Jeder schnelle RPH-Algorithmus muss $\Omega(d)$ Zufallsbits verwenden.

4.35 Beweis

- ▶ Es sei c irgendeine positive Konstante.
- ▶ Damit $2^{-k} \sqrt{N}/d \leq cd$ ist, muss gelten:
 - ▶ $2^k \geq \sqrt{N}/(cd^2)$, also
 - ▶ $k \geq \log \sqrt{N} - O(\log d)$, also
 - ▶ $k \in \Omega(d)$.

4.36 Satz

Für jedes d gibt es einen schnellen RPH-Algorithmus, der $3d$ Zufallsbits benötigt und erwartete Laufzeit $22d$ hat.

4.36 Satz

Für jedes d gibt es einen schnellen RPH-Algorithmus, der $3d$ Zufallsbits benötigt und erwartete Laufzeit $22d$ hat.

Achtung! Hier wird **nicht** die Existenz *eines* RPH-Algorithmus für Hyperwürfel *aller* Größen zugesichert wird.

4.37 Beweis

- ▶ $\mathcal{A} = (A_1, \dots, A_t)$: Liste det. PH-Algorithmen
- ▶ $\mathcal{A} \rightsquigarrow$ rand. $R_{\mathcal{A}}$ (zufällig gleichverteilt ein $A_i \in \mathcal{A}$)
- ▶ \mathcal{A} *effizientes N -Schema*, falls für jede N -Permutation $E[\text{Laufzeit}] \leq 22d$

4.37 Beweis

- ▶ $\mathcal{A} = (A_1, \dots, A_t)$: Liste det. PH-Algorithmen
- ▶ $\mathcal{A} \rightsquigarrow$ rand. $R_{\mathcal{A}}$ (zufällig gleichverteilt ein $A_i \in \mathcal{A}$)
- ▶ \mathcal{A} *effizientes N -Schema*, falls für jede N -Permutation $E[\text{Laufzeit}] \leq 22d$
- ▶ *Zeige*: es ex. effizientes N -Schema mit $t = N^3$ Algs.
- ▶ $R_{\mathcal{A}}$ braucht nur $\log t \in O(\log N) = O(d)$ Zufallsbits

- ▶ Nachweis der Existenz eines so kleinen effizienten N -Schemas mittels der probabilistischen Methode

4.37 Beweis (2)

- ▶ Fasse Algorithmus 4.25 als Menge $\mathcal{B} = \{B_1, \dots, B_{N^N}\}$ von N^N deterministischen PH-Algorithmen auf.
- ▶ Betrachte das folgende Zufallsexperiment: Aus \mathcal{B} wird zufällig (mit Zurücklegen) eine Liste $\mathcal{A} = (A_1, \dots, A_{N^3})$ von N^3 Algorithmen $A_i = B_{j_i}$ ausgewählt.
- ▶ *Zeige:* \mathcal{A} ist effizientes N -Schema mit Wahrscheinlichkeit echt größer Null.
- ▶ Also existiert ein effizientes N -Schema.

4.37 Beweis (3)

- ▶ π_i : die $N!$ Permutationen
- ▶ Det. PH-Algorithmus A heie *gut* fr π_i , wenn er π_i in hchstens $14d$ Schritten routet, sonst *schlecht*.
- ▶ Satz 4.26.2. sagt: Fr jedes π_i ist ein Bruchteil von $\leq 1/N$ aller B_j schlecht fr π_i .
- ▶ zunchst ein beliebiges π_i fixiert.
- ▶ Erwartungswert fr die Anzahl der fr π_i schlechten Algorithmen in \mathcal{A} ist $\leq N^3/N = N^2$
- ▶ X_j sei die 0-1-Zufallsvariable mit $X_j = 1$, gdw. A_j schlecht fr π_i .
- ▶ Also: $\mu = \mathbf{E}\left[\sum_{j=1}^{N^3} X_j\right] \leq N^2$.

4.37 Beweis (4)

- ▶ $\mu = \mathbf{E}\left[\sum_{j=1}^{N^3} X_j\right] \leq N^2$
- ▶ sei $c = N^2/\mu \geq 1$
- ▶ X_j unabhängige Zufallsvariablen
- ▶ Obere Schranke für $\mathbf{P}\left(\sum_{j=1}^{N^3} X_j > 4N^2\right)$ mittels Chernoff-Schranke (Kor. 4.23):

$$\mathbf{P}\left(\sum_{j=1}^{N^3} X_j > (1+3)N^2\right) < \left(\frac{1}{e}\right)^{N^2} = e^{-N^2}.$$

(siehe nächste Folie)

- ▶ Ereignis E_i : $> 4N^2$ Algorithmen in \mathcal{A} schlecht für π_i .
- ▶ Also $\mathbf{P}(E_i) < e^{-N^2}$.

4.37 Beweis (5)

$$\begin{aligned} \mathbf{P}\left(\sum_{j=1}^{N^3} X_j > (1+3)N^2\right) &= \mathbf{P}\left(\sum_{j=1}^{N^3} X_j > (1+\delta)c\mu\right) \\ &\leq \mathbf{P}\left(\sum_{j=1}^{N^3} X_j > (1+c\delta)\mu\right) \\ &\leq \left(\frac{e^{c\delta}}{(1+c\delta)^{1+c\delta}}\right)^\mu \leq \left(\frac{e^{c\delta}}{(1+c\delta)^{c\delta}}\right)^\mu \\ &= \left(\frac{e}{1+c\delta}\right)^{c\delta\mu} \leq \left(\frac{e}{1+3}\right)^{c\delta\mu} \\ &= \left(\frac{e}{4}\right)^{3N^2} \leq \left(\frac{1}{e}\right)^{N^2} \end{aligned}$$

4.37 Beweis (6)

- ▶ Wahrscheinlichkeit, dass \mathcal{A} für mindestens ein π_i schlecht:

$$\mathbf{P}\left(\bigcup_{i=1}^{N!} E_i\right) \leq \sum_{i=1}^{N!} \mathbf{P}(E_i) \leq N! \cdot e^{-N^2} < 1 .$$

- ▶ Also: Wahrscheinlichkeit, dass von den Algorithmen in \mathcal{A} für jede Permutation höchstens $4N^2$ schlecht sind, ist echt größer 0. Also existiert ein solches \mathcal{A} .

4.37 Beweis (7)

- ▶ *Zeige:* dieses \mathcal{A} sogar ein effizientes N -Schema.
- ▶ Es sei π_i beliebig.
- ▶ Mit Wahrscheinlichkeit $\geq 1 - (4N^2/N^3) = 1 - 4/N$ wird $R_{\mathcal{A}}$ diese Permutation in $\leq 14d$ Schritten durchführen. Andernfalls werden höchstens $2dN$ Schritte benötigt.
- ▶ Also Erwartungswert für die Laufzeit höchstens

$$\left(1 - \frac{4}{N}\right)14d + \frac{4}{N}2dN \leq 22d .$$

4.38 Bemerkung

- ▶ Zeige: Für hinreichend große N gilt: $N! \cdot e^{-N^2} < 1$.
- ▶ Stirlingsche Formel:

$$N! = \sqrt{2\pi N} \frac{N^N}{e^N} (1 + h(N)) \text{ mit}$$

$$h(N) = \frac{1}{12N} + \frac{1}{288N^2} - \frac{139}{5140N^3} \pm \dots \in O\left(\frac{1}{N}\right)$$

- ▶ Also:

$$\lim_{N \rightarrow \infty} \frac{N!}{e^{N^2}} = \lim_{N \rightarrow \infty} \frac{\sqrt{2\pi N} \cdot N^N}{e^N \cdot e^{N^2}} = \lim_{N \rightarrow \infty} \frac{\sqrt{2\pi N} \cdot e^{N \log N}}{e^N \cdot e^{N^2}} = 0$$

- ▶ Genaueres Nachrechnen ergibt, dass schon für $N \geq 4$ gilt:
 $N! \cdot e^{-N^2} < 1$.

4.39 Bemerkung

- ▶ Satz 4.36 behauptet nur die *Existenz* eines – noch dazu *nichtuniformen* – schnellen RPH-Algorithmus, der nur $O(d)$ Zufallsbits braucht.
- ▶ Schwieriger: man gebe explizit RPH-Algorithmen an, die möglichst wenige Zufallsbits benötigen.
- ▶ Bislang beste Lösung: uniform $\Theta(d^2)$ Zufallsbits.
- ▶ Aufgabe: Man senke diesen Wert.

Zusammenfassung

- ▶ Chernoff-Schranken
- ▶ die probabilistische Methode
- ▶ Routing in Hyperwürfeln:
 - ▶ deterministisch ist der schlimmste Fall immer schlimm
 - ▶ randomisiert im Erwartungswert immer harmlos
 - ▶ aber eben nur der Erwartungswert