

- **Simulated Annealing** führt eine Irrfahrt, bzw. einen Random Walk auf den Lösungen eines Optimierungsproblems durch, um eine optimale Lösung zu bestimmen.
- Google berechnet den **Page-Rank** einer Webseite über einen Random Walk auf dem World-Wide-Web Graphen.
- Ein „sorgfältig kontrollierter“ Random Walk bestimmt **erfüllende Belegungen von 3-Sat Formeln** schneller als bisher bekannte deterministische Algorithmen.
- Ein Random Walk auf einem ungerichteten Graphen bestimmt **Zusammenhangskomponenten** auf logarithmischem Platz.
- In der **Volumenbestimmung konvexer Mengen** legt man ein Gitter über die konvexe Menge und exploriert das Gitter mit Hilfe eines Random Walks.

Löse das Optimierungsproblem

minimiere  $f(x)$ , so dass Prädikat  $L(x)$  wahr ist.

- Das Prädikat  $L$  ist genau dann wahr, falls  $x$  eine zulässige Lösung ist.
- Annahme: Zu jeder Lösung  $x$  gibt es eine Umgebung  $U(x)$  benachbarter Lösungen mit:
  - ▶ Für jedes  $x$  gilt  $x \in U(x)$  ( $x$  liegt in seiner eigenen Umgebung).
  - ▶ Für alle  $x, y$  gilt:  $x \in U(y) \Leftrightarrow y \in U(x)$ .
  - ▶ Für  $x \in U(y)$  gilt  $|U(x)| = |U(y)|$ .

$U$  heißt **symmetrisch**.

## Erhitze einen Stoff über seinen Schmelzpunkt hinaus.

- Kühle behutsam ab, um eine eine reine Kristallstruktur zu erhalten.

Man erhält ein globales Energie-Minimum.

- Kühlt man zu schnell ab, treten kleine Fehler in der Kristallgitterstruktur auf.

Die fehlerhafte Struktur entspricht einem lokalen Energie-Minimum.

- Wir führen eine **lokale Minimierung** durch:
  - ▶ Wechsel von einer zulässigen Lösung  $x$ , wenn möglich, zu einer besseren benachbarten Lösung  $y$ .
- Und wenn es nur schlechtere benachbarte Lösungen gibt?
  - ▶ Wir stecken in einem lokalen Minimum fest.
  - ▶ Mache „**widerstrebend**“ mit einer schlechteren benachbarten Lösung  $y$  weiter.
  - ▶ Wie widerstrebend? Entsprechend einem „Abkühl-Fahrplan“.

# Simulated Annealing: Der Algorithmus

- (1) Die Zielfunktion  $f(x)$  ist für Lösungen  $x$  zu minimieren.
- (2) Setze  $x = x_0$  für eine Anfangslösung  $x_0$  und wähle eine Anfangstemperatur  $T$ .

Der Temperaturparameter  $T$  bestimmt unsere Bereitschaft, eine Uphill-Bewegung in Kauf zu nehmen.

- (3) Wiederhole „hinreichend oft“:
  - (3a) Wähle zufällig eine Nachbar-Lösung  $y \in U(x)$ .
  - (3b) IF  $f(y) \leq f(x)$  THEN  $x = y$  ELSE

Setze  $x = y$  mit Wahrscheinlichkeit  $e^{-\frac{f(y)-f(x)}{T}}$ . (Warum?)

- (3c) Wähle eine neue Temperatur  $T$ .

Wenn die Temperatur nicht verändert wird, dann spricht man vom **Metropolis Algorithmus**.

# Beispiel: Das Vertex Cover Problem I

Warum kühlt man ab? **Der leere Graph  $G = (V, \emptyset)$ :**

- Beginne mit Anfangslösung  $V$ .
- Anfänglich wird die Knotenmenge nur reduziert und Metropolis verhält sich wie die lokale Suche.
- Umso kleiner die gegenwärtige Lösung ist, umso größer ist die Anzahl der schlechteren Nachbarn.
  - ▶ Die schlechteren Nachbarn sind also höher wahrscheinlich als die guten.
  - ▶ Die Akzeptanzwahrscheinlichkeit einer schlechteren Lösung, bei nicht zu kleiner Temperatur, ist aber nur unwesentlich kleiner.
  - ▶ Schlechtere Lösungen werden kurz über lang gewählt!

Der Metropolis-Algorithmus bekommt Angst vor der eigenen Courage, wenn gute, aber längst nicht optimale Lösungen erreicht werden.

## Der Sterngraph:

- Der Metropolis-Algorithmus zeigt seine Stärke:
  - ▶ Beginne wieder mit der Menge aller Knoten als Anfangslösung.
  - ▶ Selbst wenn das Zentrum anfänglich entfernt wird, so ist die Wahrscheinlichkeit hoch, dass es nach nicht zu langer Zeit wieder betrachtet wird.
  - ▶ Nach Aufnahme, wenn mindestens ein Satellit **nicht** in der jeweiligen Lösung liegt, bleibt das Zentrum erhalten.  
Warum? Sonst liegt keine Überdeckung vor!
- Allerdings hat Metropolis auch für den Sterngraphen Angst vor der eigenen Courage, wenn die Überdeckung genügend klein ist:
  - ▶ Satelliten werden aufgenommen, obwohl sie nicht benötigt werden!

Simulated Annealing erschwert Aufwärtsbewegungen „mit der Zeit“.

Wenn die Temperatur hoch ist, dann ist

$$e^{-\frac{f(y)-f(x)}{T}} \approx 1 - \frac{f(y) - f(x)}{T}.$$

Warum?  $e^{-x} \approx 1 - x$ , wenn  $x$  klein.

- Bei hoher Temperatur sind selbst große Verschlechterungen durchaus wahrscheinlich.

Simulated Annealing durchläuft den Lösungsraum in fast zufälliger Manier.

- Wenn  $T$  gegen Null strebt, dann sind Verschlechterungen entsprechend niedrig-wahrscheinlich.
- **Geometrisches Abkühlen**, ein populärer Abkühl-Fahrplan:
  - ▶ Halte die Temperatur  $T$  für längere Zeit konstant und
  - ▶ erniedrige  $T$  dann auf  $(1 - \epsilon) \cdot T$ .

## Graph-Partitionierung

Für einen ungerichteten Graphen  $G = (V, E)$ : Bestimme  $W \subseteq V$  mit  $|W| = \frac{1}{2} \cdot |V|$ , so dass die Anzahl

$$|\{e \in E : |e \cap W| = 1\}|$$

kreuzender Kanten minimal ist.

- Wichtiges Problem im VLSI-Layout:
  - ▶ Zerschneide einen Graphen  $G$  in zwei Teilgraphen  $G_1$  und  $G_2$ .
  - ▶ Entwerfe Layouts für  $G_1$  und  $G_2$ .
  - ▶ Füge die Layouts zusammen und schaffe Platz für die einzufügenden kreuzenden Kanten.
- Graph-Partitionierung ist komplex:
  - ▶ Die Sprachenversion ist  $\mathcal{NP}$ -vollständig.



# Simulated Annealing für Graph-Partitionierung

Ein Vorschlag für die Parameterwahl:

- Alle Teilmengen  $W \subseteq V$  sind zulässige Lösungen.
- Die Zielfunktion ist

$$f(W) := |\{e \in E : |e \cap W| = 1\}| + \underbrace{\alpha \cdot (|W| - |V \setminus W|)^2}_{\text{Strafterm}}.$$

Belohne eine perfekte Aufteilung  $|W| = |V|/2$ .

- Die Umgebung von  $W$ :

$$U(W) := \{W' \subseteq V \mid |W \oplus W'| \leq 1\}.$$

- Die Temperatur-Steuerung:
  - ▶ Die Anfangstemperatur wird so gewählt, dass 40% aller Nachbarn akzeptiert werden.
  - ▶ Die Temperatur wird für die Dauer von  $16 \cdot |V|$  Schritten konstant gehalten
  - ▶ und dann auf  $0.95 \cdot T$  abgekühlt.

- Für **zufällig gewählte** Graphen:
  - ▶ Simulated Annealing schneidet sogar besser als maßgeschneiderte Algorithmen ab
  - ▶ und dies gilt selbst dann, wenn den beteiligten Algorithmen die gleiche Laufzeit zugestanden wird.
- Für **strukturierte** Graphen:
  - ▶ Wähle 500 (bzw. 1000) Punkte zufällig aus dem Quadrat  $[0, 1]^2$  und verbinde zwei Punkte in hinreichender Nähe.
  - ▶ Simulated Annealing-Methode bricht ein und die maßgeschneiderten Algorithmen sind deutlich überlegen.
- Eine Erklärung dieses Phänomens?

Simulated Annealing ist definitiv kein Allheilmittel, aber ein universell einsetzbares Schema für eine erste Approximation.

# Simulated Annealing: Die Analyse

Für Temperatur  $T$  und aktuelle Lösung  $x$

- $G_T[x, y]$  ist die Wahrscheinlichkeit, dass Lösung  $y$  gewählt wird.
- $A_T(x, y)$  ist die Wahrscheinlichkeit, dass  $y$  akzeptiert wird.
- $P_T[x, y]$  ist die Wahrscheinlichkeit, dass Simulated Annealing zur Lösung  $y$  wechselt.

$$G_T[x, y] = \begin{cases} 0 & y \notin U(x) \\ \frac{1}{|U(x)|} & \text{sonst,} \end{cases} \quad A_T[x, y] = \begin{cases} 1 & f(y) \leq f(x) \\ e^{-\frac{f(x)-f(y)}{T}} & \text{sonst.} \end{cases}$$

Für  $x \neq y$  ist  $P_T[x, y] = G_T[x, y] \cdot A_T[x, y]$

und  $P_T[x, x] = 1 - \sum_{y, y \in U(x), y \neq x} P_T[x, y]$ .

Eine Markoff-Kette  $(G, P)$  besteht aus

- einem gerichteten Graphen  $G = (V, E)$  und
- einer (stochastischen) Übergangsmatrix  $P$ :
  - ▶ Für jeden Knoten  $v \in V$  gilt

$$\sum_{w \in V} P[v, w] = 1.$$

- ▶ Zusätzlich ist  $P[v, w] = 0$  für  $(v, w) \notin E$  und  $P[v, w] \geq 0$  gilt stets.
- Die Knoten werden auch **Zustände** genannt.
- Der Graph von Simulated Annealing:
  - ▶ Die Zustände entsprechen zulässigen Lösungen
  - ▶ und  $(x, y)$  ist genau dann eine Kante, wenn  $y \in U(x)$ .
- $P_T$  ist die Übergangsmatrix von Simulated Annealing bei fixierter Temperatur  $T$ .

# Markoff-Ketten und Random Walks

$(G, P)$  sei eine Markoff-Kette und  $q = (q_i \mid i \in V)$  eine Anfangsverteilung auf der Menge der Zustände.

Die Markoff-Kette definiert einen Random Walk auf  $G$ .

- Der Weg beginnt in einem zufällig gemäß  $q$  gewählten Zustand  $x$ .
- Ein Nachbar  $y$  wird mit Wahrscheinlichkeit  $P[x, y]$  zufällig gewählt.  
 $P[x, y]$  ist also die Wahrscheinlichkeit, in einem Schritt vom Zustand  $x$  zum Zustand  $y$  zu wandern.
- Dieser Prozess wird sodann „hinreichend“ oft wiederholt.

## Die zentralen Fragen

Sei  $q^t$  die Verteilung auf den Zuständen, wenn wir Random Walks der Länge  $t$  durchführen.

- **Wie bestimmt man die Grenzverteilung  $\lim_{t \rightarrow \infty} q^t$**
- **und wie schnell ist die Konvergenz?**

# Der Zustand nach $k$ Schritten

$$p_{x,y}^{(k)} = \left[ \begin{array}{l} \text{Wahrscheinlichkeit, dass Lösung } y \text{ nach } k \\ \text{Schritten von Lösung } x \text{ aus erreicht wird.} \end{array} \right].$$

**Behauptung:**  $p_{x,y}^{(k)} = P_T^k[x, y]$ .

- Induktion nach  $k$ : Die Verankerung  $k = 1$  ist offensichtlich.
- Es ist  $p_{x,y}^{(k+1)} = \sum_r P_T^k[x, r] \cdot P_T[r, y]$ . Warum?
  - ▶ Nach Induktionsannahme ist  $P_T^k[x, r]$  die Wahrscheinlichkeit in  $k$  Schritten von  $x$  nach  $r$  zu wandern und
  - ▶  $P_T[r, y]$  ist die Wahrscheinlichkeit, dass  $y$  in einem Schritt von  $r$  aus erreicht wird.
- Die Behauptung folgt.

Die Markoff-Kette  $(G, P)$  heißt genau dann **ergodisch**, wenn für alle Zustände  $x, y, z$

- die Grenzwerte  $\lim_{k \rightarrow \infty} P^k[x, z]$  und  $\lim_{k \rightarrow \infty} P^k[y, z]$  existieren
- und  $\lim_{k \rightarrow \infty} P^k[x, z] = \lim_{k \rightarrow \infty} P^k[y, z] > 0$  gilt.

$\pi^{(\infty)} = (\lim_{k \rightarrow \infty} P^k[1, j] \mid j \in V)$  heißt **Grenzverteilung** von  $(G, P)$ .

- Ergodische Ketten sind „gute“ Markoff-Ketten:
  - ▶ Die Wahrscheinlichkeit, einen Zustand  $y$  zu erreichen, hängt nicht vom Anfangszustand ab
  - ▶ und jeder Zustand  $z$  kann von jedem anderen Zustand erreicht werden.
- Grenzverteilungen ergodischer Ketten lassen sich einfach charakterisieren!

$$P^\infty = \lim_{k \rightarrow \infty} P^k = \left( \lim_{k \rightarrow \infty} P^k[x, y] \right)_{x, y \in V}$$

- Beachte  $P^\infty[x, z] = P^\infty[y, z]$  für alle Zustände  $x, y, z$ .
- Für eine beliebige Verteilung  $\pi$

$$\begin{aligned} \pi^T \cdot P^\infty &= \left( \sum_{x=1}^N \pi_x \cdot P^\infty[x, 1], \dots, \sum_{x=1}^N \pi_x \cdot P^\infty[x, N] \right) \\ &= \left( P^\infty[1, 1] \cdot \sum_{x=1}^N \pi_x, \dots, P^\infty[1, N] \cdot \sum_{x=1}^N \pi_x \right) \\ &= ( P^\infty[1, y] \mid y = 1, \dots, N ) =: \pi^{(\infty)} \end{aligned}$$

- $\pi^{(\infty)}$  ist die Grenzverteilung!
- Es gilt also stets  $\pi^T \cdot P^\infty = \pi^{(\infty)}$  und deshalb  $\pi^{(\infty)} \cdot P^\infty = \pi^{(\infty)}$ .



# Stationäre Verteilungen II

Sei  $(G, P)$  eine Markoff-Kette mit  $G = (V, E)$ .

- Eine Verteilung  $\pi$  heißt **stationär**, falls  $\pi^T \cdot P = \pi$  gilt.
- Jede ergodische Markoff-Kette  $(G, P)$  besitzt eine eindeutig bestimmte stationäre Verteilung  $\pi$  und es gilt  $\pi = \pi^{(\infty)}$ .

**Stationäre Verteilung und Grenzverteilung stimmen für ergodische Ketten überein!**

- $\pi^{(\infty)}$  ist stationär:

$$\begin{aligned}\pi^{(\infty)} &= \pi^{(\infty)} \cdot P^\infty = \pi^{(\infty)} \cdot \lim_{k \rightarrow \infty} P^k \\ &= \pi^{(\infty)} \cdot \lim_{k \rightarrow \infty} P^{k+1} = \pi^{(\infty)} \cdot P^\infty \cdot P = \pi^{(\infty)} \cdot P.\end{aligned}$$

- Sei  $\pi$  eine beliebige **stationäre** Verteilung.
  - ▶ Wir erhalten  $\pi = \pi^T \cdot P$  und damit natürlich auch  $\pi = \pi^T \cdot P^\infty$ .
  - ▶ Andererseits ist  $\pi^T \cdot P^\infty = \pi^{(\infty)}$  und deshalb  $\pi = \pi^{(\infty)}$ .

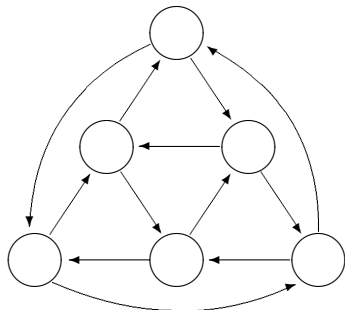
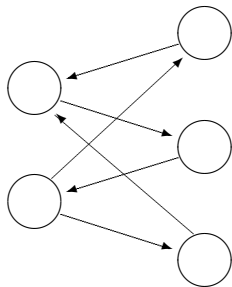
# Ergodisch = aperiodisch und irreduzibel

Sei  $(G, P)$  eine Markoff-Kette mit  $G = (V, E)$ .

- (a)  $(G, P)$  heißt **irreduzibel**, wenn es für alle Zustände  $x$  und  $y$  einen Weg von  $x$  nach  $y$  mit Kanten positiver Wahrscheinlichkeit gibt.
- (b)  $(G, P)$  hat **Periode  $p$** , wenn für alle Zustände  $x$  alle Wege von  $x$  nach  $x$  eine durch  $p$  teilbare Länge besitzen.
- (c)  $(G, P)$  ist **aperiodisch** genau dann, wenn  $(G, P)$  keine Periode  $p > 1$  besitzt.

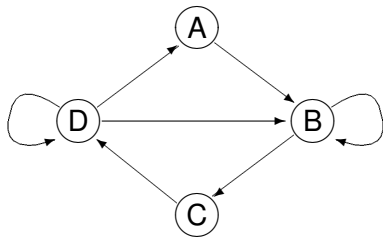
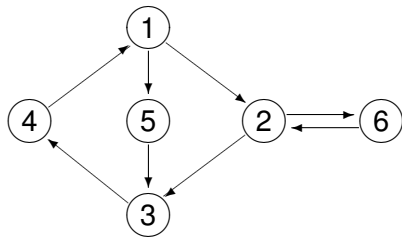
Eine Kette ist genau dann ergodisch, wenn sie aperiodisch und irreduzibel ist.

Die beiden folgenden Ketten sind nicht ergodisch.



Warum?

Und diese beiden?



# Was können wir über Simulated Annealing sagen?

Die Markoff-Kette für Simulated-Annealing ist irreduzibel und aperiodisch und damit ergodisch.

- Wenn die Kette reduzibel ist, liegt ein Definitionsfehler der Nachbarschaft vor:
  - Ein globales Minimum ist möglicherweise nicht von einer Anfangslösung erreichbar.
- Die Kette ist stets aperiodisch, denn jede Lösung ist auch ihr eigener Nachbar.

- + Jede vernünftige Implementierung von Simulated-Annealing führt auf eine ergodische Markoff-Kette.
- + Die Kette besitzt eine Grenzverteilung, die mit der eindeutig bestimmten stationären Verteilung zusammenfällt.

# Wie sieht die Grenzverteilung aus?

## Vermutung

$$q_T(x) = \frac{1}{N_T} \cdot e^{-\frac{f(x)}{T}}$$

mit der Normalisierung

$$N_T := \sum_{x \text{ Lösung}} e^{-\frac{f(x)}{T}}$$

ist die Grenzverteilung.

Wenn die Vermutung richtig ist,

dann nimmt die Wahrscheinlichkeit einer Lösung  $x$  zu, je kleiner  $f(x)$  ist.

Und genau das wollen wir!

# Wie zeigt man die Vermutung?

Eine ergodische Markoff-Kette  $(G, P)$  heißt **umkehrbar**, wenn es eine Verteilung  $q$  gibt, so dass

$$q_x \cdot P[x, y] = q_y \cdot P[y, x].$$

- Wir summieren über alle Zustände  $y$

$$\sum_y q_x \cdot P[x, y] = \sum_y q_y \cdot P[y, x].$$

- Und die linke Seite? Es ist

$$\sum_y q_x \cdot P[x, y] = q_x \sum_y P[x, y] = q_x.$$

und  $q_x = \sum_y q_y \cdot P[y, x]$  folgt.

$q$  ist die stationäre Verteilung und damit die Grenzverteilung.

# Warum sprechen wir von umkehrbaren Ketten?

$(G, P)$  ist umkehrbar falls  $q_x \cdot P[x, y] = q_y \cdot P[y, x]$ .

- $q$  ist die Grenzverteilung und  $q_x$  ist deshalb die Wahrscheinlichkeit, den Zustand  $x$  zu erreichen.
- Wie groß ist die Wahrscheinlichkeit, im vorigen Schritt im Zustand  $y$  gewesen zu sein (**Ereignis vorher<sub>y</sub>**), gegeben, dass wir jetzt im Zustand  $x$  sind (**Ereignis jetzt<sub>x</sub>**)?

$$\text{prob[ vorher}_y \mid \text{jetzt}_x ] = \frac{\text{prob[ vorher}_y \wedge \text{jetzt}_x ]}{\text{prob[ jetzt}_x ]} = \frac{q_y \cdot P[y, x]}{q_x}.$$

- Umkehrbarkeit bedeutet  $P[x, y] = \text{prob[ vorher}_y \mid \text{jetzt}_x ]$ .

Die Wahrscheinlichkeit  $P[x, y]$  von  $x$  nach  $y$  zu wandern, stimmt überein mit der Wahrscheinlichkeit in  $y$  gewesen zu sein, wenn  $x$  erreicht wird!



$q_T(x) = \frac{1}{N_T} \cdot e^{-\frac{f(x)}{T}}$  ist die Grenzverteilung!

- Wir zeigen  $q_T(x) \cdot P_T[x, y] = q_T(y) \cdot P_T[y, x]$ : Die Kette von Simulated annealing ist umkehrbar.
- Es ist

$$P_T[x, y] = G_T[x, y] \cdot A_T[x, y] \text{ und}$$
$$G_T[x, y] = \frac{1}{|U(x)|} = \frac{1}{|U(y)|} = G_T[y, x].$$

- Also zeige  $q_T(x) \cdot A_T[x, y] = q_T(y) \cdot A_T[y, x]$ .
- Für  $[z]^+ := \max\{z, 0\}$

$$\begin{aligned} q_T(x) \cdot A_T[x, y] &= \frac{1}{N_T} \cdot e^{-\frac{f(x)}{T}} \cdot e^{-\frac{[f(y)-f(x)]^+}{T}} \\ &= \frac{1}{N_T} \cdot e^{-\frac{f(y)}{T}} \cdot e^{-\frac{f(x)-f(y)+[f(y)-f(x)]^+}{T}} \\ &= \frac{1}{N_T} \cdot e^{-\frac{f(y)}{T}} \cdot e^{-\frac{[f(x)-f(y)]^+}{T}} = q_T(y) \cdot A_T[y, x]. \end{aligned}$$

# Und die Konsequenzen?

Wir haben die Akzeptanzwahrscheinlichkeit

$$e^{-\frac{f(y)-f(x)}{T}}$$

motiviert: Bei fixierter Temperatur  $T$  ist die Wahrscheinlichkeit, eine Lösung  $x$  zu erreichen, proportional zu  $e^{-\frac{f(x)}{T}}$ .

# Die guten und die schlechten Nachrichten

- Die guten Nachrichten:

- ▶ Wenn  $\text{opt}$  der minimale Wert von  $f$  ist, dann folgt

$$\lim_{T \rightarrow 0} q_T(x) = \lim_{T \rightarrow 0} \frac{e^{-\frac{f(x)}{T}}}{\sum_{y \text{ Lösung}} e^{-\frac{f(y)}{T}}} = \lim_{T \rightarrow 0} \frac{e^{-\frac{f(x) - \text{opt}}{T}}}{\sum_{y \text{ Lösung}} e^{-\frac{f(y) - \text{opt}}{T}}}.$$

- ▶ Wenn  $f(x) \neq \text{opt}$ , dann strebt der Zähler gegen Null. Der Nenner strebt hingegen gegen die Anzahl  $G$  der globalen Minima.
- ▶  $\lim_{T \rightarrow 0} q_T$  ist die uniforme Verteilung auf den **globalen Minima!**

- Die schlechten Nachrichten:

- ▶ Die Konvergenzgeschwindigkeit bei fixierter Temperatur ist im Allgemeinen miserabel. Warum?
- ▶ Es dauert lange, aus vielen tiefen lokalen Minima zu entkommen!

# Einsatzgebiet von Simulated Annealing

- + Simulated Annealing kann auf praktisch jedes Optimierungsproblem angewandt werden, solange ein natürlicher Umgebungsbegriff vorhanden ist.
  - + Wenn “genügend lange“ Rechenzeit vorhanden ist, dann wird Simulated Annealing Lösungen mit kleinem  $f$ -Wert bevorzugen.
    - Gutes Verhalten bei sehr flachen lokalen Optima.
  - Miserable Konvergenzgeschwindigkeit bei tiefen lokalen Minima.
- 
- Simulated Annealing gehört als eines von vielen Instrumenten in den Werkzeugkasten eines Optimierers.
  - Nur approximative Lösungen dürfen selbst bei extrem langer Rechenzeit erwartet werden.

# Random Walks (Irrfahrten) in Graphen

Sei  $G = (V, E)$  ein Graph.

- Wir starten in einem Knoten  $v \in V$ .
- Wenn Knoten  $w$  erreicht ist, dann wähle einen Nachbarn (oder direkten Nachfolger) zufällig nach der uniformen Verteilung.
- Wie groß ist die **Überdeckungszeit** (cover time), also die erwartete Schrittzahl, nach der alle Knoten besucht sind?
  - ▶ Selbst für stark zusammenhängende, **gerichtete** Graphen kann es extrem lange dauern, bis alle Knoten besucht werden.  
Die Überdeckungszeit kann exponentiell in der Knotenzahl sein.
  - ▶ Wir zeigen für zusammenhängende **ungerichtete** Graphen, dass die Überdeckungszeit durch  $2 \cdot (|V| - 1) \cdot |E|$  beschränkt ist.

# Die Markoff-Kette eines Random Walks

- $G = (V, E)$  sei **zusammenhängend** und besitze die Knotenmenge  $\{1, \dots, n\}$ .
- Jeder Knoten besitze eine Eigenschleife: Ein Knoten ist also sein eigener Nachbar.

- Die zugehörige Markoff-Kette:

- ▶ Zustände sind die Knoten von  $G$ .
- ▶ Definiere die Übergangsmatrix  $P$  durch

$$P[i, j] = \begin{cases} \frac{1}{d_i} & \{i, j\} \in E \\ 0 & \text{sonst,} \end{cases}$$

( $d(i)$  ist die Anzahl der Nachbarn von  $i$ , wobei  $i$  mitgezählt wird.)

- Die Kette ist ergodisch.

- ▶ Da  $G$  zusammenhängend ist, ist die Kette **irreduzibel**.
- ▶ Jeder Knoten besitzt eine Eigenschleife: Die Kette ist **aperiodisch**.

# Die Grenzverteilung für einen Random Walk

Wir stellen wieder eine Vermutung auf.

- Die Wahrscheinlichkeit, den Knoten  $i$  zu besuchen, sollte von der Zahl  $d(i)$  seiner Nachbar abhängen.

Vermutung: Die Verteilung  $q$  mit  $q_i = \frac{d_i}{2 \cdot |E|}$  ist stationär und damit die Grenzverteilung.

- Wir zeigen, dass die Kette sogar umkehrbar ist, denn für jede Kante  $\{i, j\}$

$$q_i \cdot P[i, j] = \frac{d_i}{2 \cdot |E|} \cdot \frac{1}{d_i} = \frac{1}{2 \cdot |E|} = \frac{d_j}{2 \cdot |E|} \cdot \frac{1}{d_j} = q_j \cdot P[j, i].$$

Es gilt  $\lim_{k \rightarrow \infty} P^k[i, j] = \frac{d_j}{2 \cdot |E|}$  für jeden Knoten  $i$ .

# Häufigkeit des Besuchs einer Kante

Wie häufig wird eine Kante  $e = \{i, j\}$  durchlaufen?

- $i$  wird mit relativer Häufigkeit  $q_i = \frac{d_i}{2 \cdot |E|}$  besucht.
- Die Wahrscheinlichkeit eines Besuchs von  $e$  beginnend in  $i$  ist deshalb

$$q_i \cdot \frac{1}{d_j} = \frac{d_i}{2 \cdot |E|} \cdot \frac{1}{d_j} = \frac{1}{2 \cdot |E|}.$$

Jede Kante des Graphen wird in jeder Richtung mit Wahrscheinlichkeit genau  $\frac{1}{2 \cdot |E|}$  durchlaufen.



Für eine Kante  $e = \{i, j\}$  in  $G$  sei

- die Hitting Time  $H_{i,j}$  die **erwartete** Anzahl von Schritten, um  $j$  von  $i$  aus zu erreichen.
- die Commute Time  $C_{i,j}$  die **erwartete** Anzahl von Schritten, um von  $i$  nach  $j$  und zurück zu  $i$  zu wandern.

Es ist  $C_{i,j} = H_{i,j} + H_{j,i} = C_{j,i}$ .

- $C_{i,j}$  ist höchstens so groß wie die erwartete Zeit zwischen zwei Durchläufen der Kante  $\{i, j\}$ .
- Die Kante  $e$  wird aber mit Wahrscheinlichkeit genau  $\frac{1}{2 \cdot |E|}$  durchlaufen.

Für jede Kante  $\{i, j\}$  ist

$$C_{i,j} \leq 2 \cdot |E|.$$

# Die Überdeckungszeit

- $G$  besitzt einen Spannbaum  $T$  (mit  $n - 1$  Kanten).
- Also gibt es einen Weg in  $G$ , der jede Kante von  $T$  zweimal, einmal in jeder Richtung, besucht.
- Wenn  $e_1, \dots, e_{n-1}$  die Kanten von  $T$  sind, dann ist die Überdeckungszeit durch

$$\sum_{i=1}^{n-1} 2 \cdot |E|$$

beschränkt, denn

die Commute Time  $C_{i,j}$  ist höchstens so groß wie die erwartete Zeit  $2 \cdot |E|$  zwischen zwei Durchläufen der Kanten  $(i, j)$  und  $(j, i)$ .

Die Überdeckungszeit für  $G$  ist höchstens  $2 \cdot (n - 1) \cdot |E|$ .

# Gibt es einen Weg von Knoten 1 nach Knoten 2?

Wir können diese Frage mit Tiefen- oder Breitensuche beantworten, arbeiten dann aber mit einem linear großen Speicherplatz.

- Was passiert, wenn wir einen Random Walk in  $G$  durchführen?
  - ▶ Wir brauchen Speicherplatz höchstens  $O(\log_2 n)$ .
  - ▶ Wie lange sollte unserer Random Walk sein?
- Wenn wir nach  $2 \cdot \text{Überdeckungszeit}$  vielen Schritten abbrechen, dann machen wir einen Fehler von höchstens  $1/2$ .

Nach  $k$  unabhängigen Random Walks der Länge  $4n \cdot |E|$ :

Ein Weg von Knoten 1 nach Knoten 2 wird mit Wahrscheinlichkeit höchstens  $2^{-k}$  nicht gefunden.

# Das $k$ -Sat Problem

Für eine Formel  $\alpha$  in konjunktiver Normalform mit höchstens  $k$  Literalen pro Klausel:

Entscheide, ob  $\alpha$  erfüllbar ist.

- $k$ -Sat ist  $\mathcal{NP}$ -vollständig für  $k \geq 3$ .
- 2-Sat ist einfach:
  - ▶ Setze  $x_1 = 0$  und bestimme alle **erzwungenen** Setzungen.
  - ▶ Wenn wir keinen Widerspruch erhalten, dann bleibt die Erfüllung der Restformel  $\alpha'$  als Aufgabe.
    - ★  $\alpha'$  ist eine Teilformel von  $\alpha$ .
    - ★ Wenn  $\alpha$  erfüllbar ist, dann auch  $\alpha'$ .
    - ★ Die ursprüngliche Setzung  $x_1 = 0$  ist OK.
  - ▶ Wenn wir einen Widerspruch erhalten, dann bleibt nur die Setzung  $x_1 = 1$ .

Für Formeln mit  $n$  Variablen: Können wir 3-Sat schneller lösen als alle  $2^n$  Belegungen auszuprobieren?

# Ein deterministischer Algorithmus für 3-Sat

- (1) Die Eingabeformel  $\alpha \equiv k_1 \wedge \dots \wedge k_N$  bestehe aus den 3-Klauseln  $k_1, \dots, k_N$ .
- (2) Setze  $\mathcal{K} = \{k_1\}$ . Solange es eine Klausel  $k_i$  gibt, die **keine** Variable mit einer Klausel in  $\mathcal{K}$  gemeinsam hat, füge  $k_i$  zu  $\mathcal{K}$  hinzu.
  - ▶ Je zwei Klauseln in  $\mathcal{K}$  sind variablen-disjunkt.
  - ▶ Jede Klausel  $k_i$  hat mindestens eine Variable mit einer Klausel in  $\mathcal{K}$  gemeinsam.
- (3)  $\mathcal{B}$ : Die Menge aller Belegungen, die alle Klauseln in  $\mathcal{K}$  erfüllen.
  - ▶ Jede Klausel wird von sieben der acht Belegungen erfüllt.
  - ▶ Wenn  $\mathcal{K}$  aus  $m$  Klauseln besteht, dann ist  $m \leq n/3$ .
  - ▶ Also besteht  $\mathcal{B}$  aus höchstens  $7^m \leq 7^{n/3}$  Belegungen.  
Zähle nur Belegungen für in  $\mathcal{K}$  vorkommende Variablen.
- (4) Für alle Belegungen  $b \in \mathcal{B}$ : Setze  $b$  in  $\alpha$  ein.
  - ▶ Jede Klausel besitzt mindestens eine durch  $b$  gesetzte Variable.
  - ▶ Das resultierende 2-KNF Erfüllbarkeitsproblem ist zu lösen.

- Wenn  $\mathcal{K}$  aus  $m$  Klauseln besteht, dann lösen wir das 3-Sat Problem in Zeit

$$\text{poly}(n + N) \cdot 7^m$$

für Formeln mit  $n$  Variablen und  $N$  Klauseln.

- Die worst-case Laufzeit ist durch

$$\text{poly}(n + N) \cdot 7^{n/3} \ll 8^{n/3} = 2^n$$

beschränkt.

- Kann Randomisierung helfen?
  - ▶ Würfel eine zufällige Belegung  $b$  aus.
  - ▶ Verbessere  $b$ .
  - ▶ Wieviele Verbesserungsschritte?
  - ▶ Wenn erfolglos, dann wiederhole mit neuer zufälliger Belegung.

# Ein randomisierter Algorithmus

- (1) Die Eingabe  $\alpha \equiv k_1 \wedge \dots \wedge k_N$  sei eine  $k$ -KNF Formel mit  $n$  Variablen und  $N$  Klauseln.
- (2) Wähle eine zufällige Belegung  $b$ .
- (3) Wiederhole  $3 \cdot n$ -mal:
  - (3a) Akzeptiere  $\alpha$ , wenn  $\alpha$  durch  $b$  erfüllt wird.
  - (3b) Ansonsten bestimme eine Klausel  $k_i$ , die nicht durch  $b$  erfüllt wird.
  - (3c) Wähle zufällig eine in  $k_i$  vorkommende Variable und flippe ihren Wahrheitswert.

- Nach der zufälligen Wahl einer Belegung arbeiten wir mit nur  $3 \cdot n$  Verbesserungsschritten.

Ein „Verbesserungsschritt“ kann tatsächlich zu einem Verlust bereits erfüllter Klauseln führen.

- Ist die Sogwirkung erfüllender Belegungen stark genug?

Sei  $a$  eine erfüllende Belegung von  $\alpha$ .

- $d(a, b)$  bezeichne den **Hamming-Abstand** zwischen  $a$  und der zufällig ausgewürfelten Belegung  $b$ .
- Unsere Kette hat die Zustände  $0, 1, \dots, n$ .
- Wir beginnen im Zustand  $d(a, b)$ .
- Wenn wir im Zustand  $i$  sind, dann
  - ▶ wechseln wir in den Zustand  $i - 1$  mit Wahrscheinlichkeit  $\geq 1/k$ :  
Mit Wahrscheinlichkeit mindestens  $1/k$  flippen wir eine Variable auf ihren Wert in  $a$ .
  - ▶ und in den Zustand  $i + 1$  mit der Gegenwahrscheinlichkeit von **höchstens**  $(k - 1)/k$ .



Wir bestimmen die Wahrscheinlichkeit  $q(b)$ , dass der Random Walk den Zustand 0 nach höchstens  $3 \cdot d(a, b)$  Schritten aufsucht.

- Wir sagen, dass der Random Walk sich falsch entscheidet, wenn ein mit  $a$  übereinstimmender Wahrheitswert geflippt wird.
- Wenn der Random Walk nach höchstens  $3 \cdot d(a, b)$  Schritten im Zustand 0 endet, hat er  $i$  falsche Entscheidungen getroffen.
  - ▶ Der Random Walk ist insgesamt  $d(a, b) + 2 \cdot i$  Schritte lang:  
Die  $i$  falschen müssen durch  $d(a, b) + i$  richtige Entscheidungen wettgemacht werden.
  - ▶ Also ist insbesondere  $i \leq d(a, b)$ .

**Wege( $i$ )** ist die Anzahl der Irrfahrten, die zum ersten Mal nach genau  $d(a, b) + 2 \cdot i$  Schritten im Zustand 0 enden.

- $q(b)$  ist die Wahrscheinlichkeit nach höchstens  $3 \cdot d(a, b)$  Schritten im Zustand 0 zu enden.

$$q(b) \geq \sum_{i=0}^{d(a,b)} \text{Wege}(i) \cdot \left(\frac{1}{k}\right)^{d(a,b)+i} \cdot \left(\frac{k-1}{k}\right)^i.$$

- Das Zwischenziel:

$$q(b) \geq \sum_{i=0}^{d(a,b)} \text{Wege}(i) \cdot \left(\frac{1}{k}\right)^{d(a,b)+i} \cdot \left(\frac{k-1}{k}\right)^i \stackrel{!}{\geq} \rho \cdot \left(\frac{1}{k-1}\right)^{d(a,b)}$$

oder äquivalent

$$\sum_{i=0}^{d(a,b)} \text{Wege}(i) \cdot \left(\frac{1}{k}\right)^i \cdot \left(\frac{k-1}{k}\right)^{d(a,b)+i} \stackrel{!}{\geq} \rho.$$

Wir müssen für  $k \geq 3$  zeigen:

$$\sum_{i=0}^{d(a,b)} \text{Wege}(i) \cdot \left(\frac{1}{k}\right)^i \cdot \left(\frac{k-1}{k}\right)^{d(a,b)+i} \stackrel{!}{\geq} \rho.$$

- Wir interpretieren die linke Seite als die Wahrscheinlichkeit, dass ein Random Walk,
  - ▶ der im Zustand  $d(a, b)$  beginnt und
  - ▶ mit Wahrscheinlichkeit  $\frac{k-1}{k}$  nach links (bzw. mit Wahrscheinlichkeit  $\frac{1}{k}$  nach rechts) wandert,
  - ▶ den Zustand 0 nach höchstens  $3 \cdot d(a, b)$  Schritten erreicht.
- Mit Wahrscheinlichkeit  $\geq 2/3$  geht es in die richtige Richtung!

Wir erreichen Zustand 0 mit Wahrscheinlichkeit  $\geq \rho$  !?

Die Wahrscheinlichkeit  $q(b)$ , den Zustand 0 nach höchstens  $3 \cdot d(a, b)$  Schritten zu erreichen, ist mindestens

$$\rho \cdot \left(\frac{1}{k-1}\right)^{d(a,b)}$$

- Für  $k = 3$  ist somit  $q(b) \geq 2^{-d(a,b)-1}$ .
  - ▶ Wir sollten  $d(a, b) \approx n/2$  erwarten und deshalb  $q(b) \approx 2^{-n/2}$ .
  - ▶ Also ist die erwartete Laufzeit höchstens  $q(b)^{-1} \approx 2^{n/2}$ .
- Aber es kommt noch besser:  
Bei vielen Versuchen werden wir recht häufig eine geringere Hammingdistanz  $d(a, b)$  erreichen und  $q(b)$  steigt entsprechend.

Wir zeigen: Eine erfüllende Belegung wird nach höchstens

$$\frac{1}{\rho} \cdot L \cdot \left( \frac{2 \cdot (k-1)}{k} \right)^n = \frac{1}{\rho} \cdot L \cdot \left( \frac{k-1}{k} \right)^n \cdot 2^n$$

Iterationen mit Wahrscheinlichkeit mindestens  $1 - e^{-L}$  gefunden.

- Für  $k = 3$  und konstante Erfolgswahrscheinlichkeit reichen  $(4/3)^n$  Iterationen.
- Signifikante Laufzeitverbesserungen für kleine Werte von  $k$ ,
- so gut wie keine Verbesserungen, wenn  $k$  linear in der Anzahl der Variablen ist.

Sei  $p$  die Wahrscheinlichkeit, dass eine Iteration erfolgreich ist.

$$\begin{aligned} p &\geq \sum_{d=0}^n \text{prob}[b \text{ mit } d(a, b) = d \text{ wird gew\u00e4hlt}] \cdot \rho \cdot \left(\frac{1}{k-1}\right)^d \\ &\geq 2^{-n} \cdot \rho \cdot \sum_{d=0}^n \binom{n}{d} \cdot \left(\frac{1}{k-1}\right)^d \\ &= 2^{-n} \cdot \rho \cdot \left(1 + \frac{1}{k-1}\right)^n \\ &= \rho \cdot \left(\frac{k}{2 \cdot (k-1)}\right)^n. \end{aligned}$$

- Nach  $t$  Iterationen haben wir Pech mit Wahrscheinlichkeit  $\leq (1-p)^t \leq e^{-p \cdot t}$ .
- F\u00fcr  $t = \frac{1}{p} \cdot L$  haben wir Pech mit Wahrscheinlichkeit  $\leq e^{-L}$ .

Eine Anfrage liegt als eine Folge von Stichworten vor.

- Google bestimmt zuerst alle Webseiten, die die Stichworte der Anfrage enthalten.
- Die wesentlichen Kriterien der **lokalen (abfrage-abhängigen) Bewertung** der Webseite:
  - Schriftgröße und Nähe der Stichworte zueinander,
  - Häufigkeit des Vorkommens der Stichworte und
  - Vorkommen der Stichworte in der Beschriftung von Hyperlinks, die auf die Seite zeigen.
- Das entscheidende Kriterium in der **globalen (abfrage-unabhängigen) Bewertung** ist der Pagerank:  
Der Pagerank wird in einem Peer Review ermittelt.

## Peer Review

Der Pagerank  $\text{pr}(w)$  ist hoch, wenn viele Seiten  $u$  mit hohem Pagerank  $\text{pr}(u)$  auf die Seite  $w$  zeigen.

- Umsetzung: ( $d(u)$  ist die Anzahl der Seiten, auf die  $u$  zeigt).

$$\text{pr}(w) = \sum_{u \text{ zeigt auf } w} \frac{\text{pr}(u)}{d_u}.$$

Der Pagerank von  $u$  wird zu gleichen Teilen auf alle Webseiten vererbt, auf die  $u$  zeigt.

- Ein zweiter Ansatz: Interpretiere das WWW als eine Markoff-Kette.
  - ▶ Zustände entsprechen den Webseiten, Kanten den Hyperlinks.
  - ▶ Dann sollte doch die stationäre Verteilung der WWW-Kette auch eine interessante Bewertung sein!



# Die stationäre Verteilung der WWW-Kette

Die Übergangsmatrix  $P$  der WWW-Kette ist

$$P[u, w] = \begin{cases} 1/d_u & (u, w) \text{ ist ein Hyperlink,} \\ 0 & \text{sonst.} \end{cases}$$

- Für die stationäre Verteilung  $q$  der Kette gilt  $q = q^T \cdot P$ , bzw.

$$q_w = \sum_{u \text{ zeigt auf } w} \frac{q_u}{d_u},$$

und **das ist genau die Definition der Peer Review Bewertung!**

- Wir erhalten den Pagerank von zwei verwandten Sichtweisen:
  1. der des Peer-Review und
  2. der des „Random Surfers“ (oder einer stationären Verteilung).
- Also, bewerte eine Webseite  $w$  mit der relativen Häufigkeit mit der ein Random Surfer die Seite  $w$  besucht.

Die WWW-Kette ist definitiv aperiodisch, aber **nicht irreduzibel!**

- Google nimmt deshalb einen Random Surfer an, der
  - ▶ mit Wahrscheinlichkeit  $(1 - \alpha)$  eine benachbarte Seite aufsucht und
  - ▶ mit Wahrscheinlichkeit  $\alpha$  zu einer zufälligen Webseite springt.
- Die Übergangsmatrix bei insgesamt  $n$  Seiten

$$P[u, w] = \begin{cases} \frac{\alpha}{n} + \frac{1-\alpha}{d_u} & (u, w) \text{ ist ein Hyperlink,} \\ \frac{\alpha}{n} & \text{sonst.} \end{cases}$$

- Die neue Kette ist jetzt auch irreduzibel und damit ergodisch:  
Eine stationäre Verteilung existiert also!

# Und noch ein Problemchen

## Wie berechnet man die stationäre Verteilung?

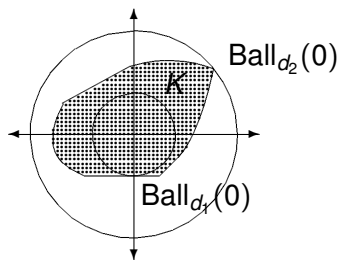
- Eine direkte Lösung des Gleichungssystems  $\text{pr}^T \cdot P = \text{pr}$  steht außer Frage:
  - ▶  $P$  hat mehrere Milliarden Zeilen und Spalten.
  - ▶ Gaußsche Eliminierung hat kubische Rechenzeit und würde größenordnungsmäßig  $10^{27}$  Zyklen verschlucken.
- Aber: Der Web-Graph ist „hochgradig vernetzt“:  
Geringe Distanz zwischen den meisten Paaren von Webseiten.

## Approximiere den Pagerank mit einer Reihe von Matrix-Vektor Operationen:

- Beginne mit einer Anfangsverteilung  $\pi_0$ ,
- und aktualisiere gemäß  $\pi_{i+1} = \pi_i^T \cdot P$ .

# Volumenbestimmung: Das Modell

- Wir haben Zugriff auf eine unbekannte konvexe Menge  $K \subseteq \mathbb{R}^n$  über ein **Orakel**, das Fragen der Form  $x \stackrel{?}{\in} K$  beantwortet.
- Wir wissen, dass  $\text{Ball}_{d_1}(0) \subseteq K \subseteq \text{Ball}_{d_2}(0)$  für Bälle mit Radien  $d_1, d_2$  um den Nullpunkt gilt.



Wieviele Fragen müssen wir stellen, bis wir eine approximative Volumenbestimmung abgeben können?

# Deterministische Algorithmen

Wir müssen ein Objekt  $K \subseteq \mathbb{R}^n$  erkunden. Was leisten deterministische Algorithmen?

- $A$  sei ein deterministischer Algorithmus, der für jede konvexe Menge  $K \subseteq \mathbb{R}^n$  mit  $\text{Ball}_{d_1}(0) \subseteq K \subseteq \text{Ball}_{d_2}(0)$  untere und obere Schranken  $\text{vol}_{\text{unten}}(K)$ ,  $\text{vol}_{\text{oben}}(K)$  mit

$$\text{vol}_{\text{unten}}(K) \leq \text{volumen}(K) \leq \text{vol}_{\text{oben}}(K)$$

in Zeit  $\text{poly}(n, \frac{1}{d_1}, d_2)$  bestimmt.

- Dann gibt eine konvexe Menge  $K_0$  und eine Konstante  $c > 0$  mit

$$\frac{\text{vol}_{\text{oben}}(K_0)}{\text{vol}_{\text{unten}}(K_0)} \geq \left( c \cdot \frac{n}{\log_2 n} \right)^n.$$

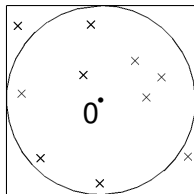
# Ein grundsätzliches Problem

Wie kann man die Fläche des Einheitskreises approximativ bestimmen?

- Wir ziehen zufällig eine genügend große Anzahl  $L$  von Punkten  $x_1, \dots, x_L$  aus dem Quadrat  $[-1, +1]^2$ .
- Wenn  $L^*$  Punkte dem Einheitskreis angehören, dann gilt

$$\frac{L^*}{L} \approx \frac{\text{volumen}(\text{Ball}_1(0))}{\text{volumen}([-1, +1]^2)}$$

- $4 \cdot \frac{L^*}{L}$  ist eine gute Approximation der Kreisfläche  $\pi$ .



# Ein grundsätzliches Problem II

- Das Volumen des  $n$ -dimensionalen Würfels  $[-1, +1]^n$  ist  $2^n$ .
- Das Volumen der Kugel mit Radius 1 ist durch

$$\text{volumen}(\text{Ball}_1(0)) = \frac{\pi^{\frac{n}{2}}}{\Gamma(1 + \frac{1}{2}n)} = \frac{2 \cdot \pi^{\frac{n}{2}}}{n \cdot \Gamma(\frac{1}{2}n)}$$

gegeben.

$\Gamma$  ist die Gamma-Funktion mit  $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ ,  $\Gamma(1) = 1$  und  $\Gamma(x+1) = x \cdot \Gamma(x)$  für  $x \in \mathbb{R}^+$ .

- Für eine geeignete Konstante  $c > 1$  gilt

$$\text{volumen}(\text{Ball}_1(0)) \leq \frac{c^n}{\sqrt{n!}}.$$

Wenn wir polynomiell viele Vektoren zufällig aus dem Einheitswürfel  $[-1, +1]^n$  ziehen, befindet sich darunter mit hoher Wahrscheinlichkeit **kein einziger Punkt** der Kugel!

- Wir definieren eine Folge  $d_1 = r_0 < r_1 < \dots < r_{N-1} < r_N \geq d_2$  langsam wachsender Radien, so dass sich die Volumina von  $K_{i-1} = \text{Ball}_{r_{i-1}}(0) \cap K$  und  $K_i = \text{Ball}_{r_i}(0) \cap K$  nur um höchstens einen kleinen konstanten Faktor unterscheiden.
- Wir ziehen  $N$  zufällige Punkte  $x \in K_i$  und fragen, ob  $x \in K_{i-1}$  gilt: Wenn  $e_i$  die Anzahl der Punkte ist, die in  $K_{i-1}$  liegen, dann ist

$$\frac{\text{volumen}(K_i)}{\text{volumen}(K_{i-1})} \approx \frac{N}{e_i}.$$

- **Wie zieht man einen zufälligen Punkt aus  $K_i$ ?**
  - ▶ Wir legen ein Gitter über die Menge  $K_i$  und führen einen Random Walk auf dem Gitter aus.
  - ▶ Der Endpunkt des Random Walk ist der zu ziehende Punkt.



# Wie sollten die Parameter gewählt werden?

- Wie sollte die Gitterbreite  $\varepsilon$  gewählt werden?

Wenn wir einen zufälligen Weg polynomieller Länge auf dem Gitter durchlaufen, dann sollte der Endpunkt ein zufälliger Punkt aus  $K_i$  sein.

- Um wieviel sollte der Radius  $r_i$  gegenüber  $r_{i-1}$  anwachsen?

- ▶ Bei einem zu geringen Anstieg ist die Laufzeit zu groß,
- ▶ bei einem zu starken Anstieg ist das Volumen von  $K_i$  im Vergleich zum Volumen von  $K_{i-1}$  zu groß:

Ein zufälliger Punkt aus  $K_i$  liegt hochwahrscheinlich nicht in  $K_{i-1}$ .

# Wahl der Radien

Wir wissen  $\text{Ball}_{d_1}(0) \subseteq K \subseteq \text{Ball}_{d_2}(0)$ .

- Es gilt

$$\text{volumen}(\text{Ball}_r(0)) = r^n \cdot \text{volumen}(\text{Ball}_1(0)).$$

- Also folgt

$$\frac{\text{volumen}(\text{Ball}_{r_{i+1}}(0))}{\text{volumen}(\text{Ball}_{r_i}(0))} = \left(\frac{r_{i+1}}{r_i}\right)^n.$$

- Setze  $r_{i+1} = \left(1 + \frac{1}{n}\right) \cdot r_i$ .

Die Volumina steigen um höchstens  $(1 + 1/n)^n \leq e$  an.

- Wieviele Iterationen benötigen wir? Setze  $N = n \cdot \ln\left(\frac{d_2}{d_1}\right)$ .

Dann  $r_N = d_1 \cdot \left(1 + \frac{1}{n}\right)^N \approx d_1 \cdot e^{\frac{N}{n}} = d_1 \cdot e^{\ln \frac{d_2}{d_1}} = d_2$ .

$N = n \cdot \ln\left(\frac{d_2}{d_1}\right)$  Iterationen genügen.

# Der Random Walk auf $\text{Ball}_{r_i}(0) \cap K$

- (0) Wir wählen  $\epsilon = \left(\text{poly}(n, \frac{1}{d_1})\right)^{-1}$  als Gitterbreite und betrachten das Gitter

$$\Gamma = \{\epsilon \cdot x \mid x \in \mathbb{Z}^n\}.$$

- (1) Wir beginnen den Random-Walk in  $z_0 = 0 \in \Gamma$ .
- (2) FOR  $j = 1$  TO  $\text{poly}(n, \frac{1}{d_1}, d_2)$  DO
- (2a) Wähle einen mit  $z_{j-1}$  benachbarten Gitterpunkt  $z$  zufällig. (Beachte, dass  $z$  auch Nachbar von sich selbst ist.)
  - (2b) IF  $z \in \text{Ball}_{r_i}(0) \cap K$  THEN  $z_j = z$  ELSE  $z_j = z_{j-1}$ .

# Liefert der Random Walk einen zufälligen Punkt?

- Die Markoff-Kette:
  - ▶  $K_i \cap \Gamma$  ist die Menge der Zustände.
  - ▶ Wenn  $z \in K_i \cap \Gamma$  insgesamt  $d$  Gitterpunkte als Nachbarn hat (inklusive sich selbst), erhält jeder Übergang die Wahrscheinlichkeit  $\frac{1}{d}$ .
- Die Kette ist irreduzibel + aperiodisch (warum?) und damit ergodisch.
- Wie sieht die eindeutig bestimmte stationäre Verteilung aus?
  - ▶ Die Verteilung sollte „beinahe“ die Gleichverteilung sein, denn die relativ wenigen Randpunkte haben geringe Wahrscheinlichkeit.
  - ▶ Ohne Beweis: Die Kette „**mischt schnell**“, d.h. der Random Walk approximiert die stationäre Verteilung schnell.

- Für jedes  $i$  führe  $\text{poly}(n)$  Random Walks in  $K_i \cap \Gamma$  aus.  
Dann kann das relative Volumen von  $K_{i-1}$  bezüglich  $K_i$  scharf approximiert werden.
- Jeder Random Walk hat die Länge  $\text{poly}(n, \frac{1}{d_1}, d_2)$ .
- Insgesamt  $N = n \cdot \ln(\frac{d_2}{d_1})$  verschiedene Mengen  $K_i = \text{Ball}_{r_i} \cap K$  müssen behandelt werden.

Die Laufzeit ist polynomiell in  $\frac{1}{d_1}$ ,  $d_2$  und  $n$ .