Kürzeste Wege II

Algorithmen für verteilte Systeme

Sebastian Forster

Universität Salzburg

Problemstellung

Gegeben: Gewichteter, ungerichteter Graph G mit Startknoten s
Ziel: Jeder Knoten v kennt Distanz $\mathrm{d}_G(s, v)$ von s zu v

Problemstellung

Gegeben: Gewichteter, ungerichteter Graph G mit Startknoten s
Ziel: Jeder Knoten v kennt Distanz $\mathrm{d}_G(s, v)$ von s zu v

SSSP: „Single-Source Shortest Paths“

Problemstellung

Gegeben: Gewichteter, ungerichteter Graph G mit Startknoten s
Ziel: Jeder Knoten v kennt Distanz $\mathrm{d}_G(s, v)$ von s zu v

SSSP: „Single-Source Shortest Paths“

Annahmen:

- Positive, ganzzahlige Kantengewichte von 1 bis W
- Jeder Knoten weiß initial, ob er Startknoten ist

Problemstellung

Gegeben: Gewichteter, ungerichteter Graph G mit Startknoten s
Ziel: Jeder Knoten v kennt Distanz $\mathrm{d}_G(s, v)$ von s zu v

SSSP: „Single-Source Shortest Paths“

Annahmen:

- Positive, ganzzahlige Kantengewichte von 1 bis W
- Jeder Knoten weiß initial, ob er Startknoten ist

CONGEST Modell:

- Kommunikation mit Nachbarn in synchronen Runden
- Bandbreite (= maximale Nachrichtengröße) $O(\log n)$
- Heute: $W = n^{O(1)}$, also $\log W = O(\log n)$

Obere und untere Schranken

Theorem ([Peleg/Rubinovich '99])

Im Allgemeinen werden $\Omega(\sqrt{n}/\log n + D)$ *Runden benötigt, um das SSSP-Problem zu lösen.*

Obere und untere Schranken

Theorem ([Peleg/Rubinovich '99])

Im Allgemeinen werden $\Omega(\sqrt{n}/\log n + D)$ *Runden benötigt, um das SSSP-Problem zu lösen.*

Theorem (Bellman-Ford)

Das SSSP-Problem kann in $O(n)$ *Runden gelöst werden.*

Obere und untere Schranken

Theorem ([Peleg/Rubinovich '99])

Im Allgemeinen werden $\Omega(\sqrt{n}/\log n + D)$ *Runden benötigt, um das SSSP-Problem zu lösen.*

Theorem (Bellman-Ford)

Das SSSP-Problem kann in $O(n)$ *Runden gelöst werden.*

Theorem ([Forster/Nanongkai '18])

Das SSSP-Problem kann in $O((\sqrt{n}D^{1/4} + n^{3/5} + D) \cdot \log^{O(1)} n)$ *Runden gelöst werden (mit hoher Wahrscheinlichkeit).*

Obere und untere Schranken

Theorem ([Peleg/Rubinovich '99])

Im Allgemeinen werden $\Omega(\sqrt{n}/\log n + D)$ Runden benötigt, um das SSSP-Problem zu lösen.

Theorem (Bellman-Ford)

Das SSSP-Problem kann in $O(n)$ Runden gelöst werden.

Theorem ([Forster/Nanongkai '18])

Das SSSP-Problem kann in $O((\sqrt{n}D^{1/4} + n^{3/5} + D) \cdot \log^{O(1)} n)$ Runden gelöst werden (mit hoher Wahrscheinlichkeit).

Enge obere/untere Schranke ist großes offenes Problem!

Approximationsalgorithmen

Ziel: Berechne für jeden Knoten v eine Distanzschätzung $\delta(s, v)$, für die gilt:

$$\mathrm{d}(s, v) \leq \delta(s, v) \leq (1 + \epsilon)\,\mathrm{d}(s, v)$$

Approximationsalgorithmen

Ziel: Berechne für jeden Knoten v eine Distanzschätzung $\delta(s,v)$, für die gilt:

$$\mathrm{d}(s,v) \leq \delta(s,v) \leq (1+\epsilon)\,\mathrm{d}(s,v)$$

Theorem ([Elkin '04])

Im Allgemeinen werden $\Omega(\sqrt{n/(\alpha \log n)} + D)$ Runden benötigt, um eine α-Approximation für das SSSP Problem zu berechnen.

Approximationsalgorithmen

Ziel: Berechne für jeden Knoten v eine Distanzschätzung $\delta(s,v)$, für die gilt:

$$\mathrm{d}(s,v) \leq \delta(s,v) \leq (1+\epsilon)\,\mathrm{d}(s,v)$$

Theorem ([Elkin '04])

Im Allgemeinen werden $\Omega(\sqrt{n/(\alpha \log n)} + D)$ Runden benötigt, um eine α-Approximation für das SSSP Problem zu berechnen.

Theorem ([Becker et al. '17])

Eine $(1+\epsilon)$-Approximation für das SSSP Problem kann in $O((\sqrt{n} + D) \cdot \log^{O(1)}(n)/\epsilon^{O(1)})$ Runden berechnet werden (mit hoher Wahrscheinlichkeit).

Approximationsalgorithmen

Ziel: Berechne für jeden Knoten v eine Distanzschätzung $\delta(s, v)$, für die gilt:

$$\mathrm{d}(s, v) \leq \delta(s, v) \leq (1 + \epsilon)\, \mathrm{d}(s, v)$$

Theorem ([Elkin '04])

Im Allgemeinen werden $\Omega(\sqrt{n/(\alpha \log n)} + D)$ Runden benötigt, um eine α-Approximation für das SSSP Problem zu berechnen.

Theorem ([Becker et al. '17])

Eine $(1 + \epsilon)$-Approximation für das SSSP Problem kann in $O((\sqrt{n} + D) \cdot \log^{O(1)}(n)/\epsilon^{O(1)})$ Runden berechnet werden (mit hoher Wahrscheinlichkeit).

Heute: $O(n^{2/3} \log^2(n)/\epsilon + D)$ Runden

Tools

Lemma

Sei h ein Parameter und sei Z (Zentren) eine Menge zu der jeder Knoten unabhängig mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ hinzugefügt wurde. Dann gilt mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$: Für jedes Knotenpaar u und v gibt es einen kürzesten Weg von u nach v, der innerhalb der ersten h Knoten ein Zentrum enthält.

Tools

Lemma

Sei h ein Parameter und sei Z (Zentren) eine Menge zu der jeder Knoten unabhängig mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ hinzugefügt wurde. Dann gilt mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$: Für jedes Knotenpaar u und v gibt es einen kürzesten Weg von u nach v, der innerhalb der ersten h Knoten ein Zentrum enthält.

Lemma

Sei Z eine Teilmenge von Knoten und h ein Parameter. In $O((|Z| + h) \log(nW) \log(n)/\epsilon)$ Runden kann für jedes $x \in Z$ und jedes $v \in V$ eine approximative Distanz $\tilde{\mathrm{d}}(x,v)$ berechnet werden (die v am Ende kennt), für die mit hoher Wahrscheinlichkeit gilt:

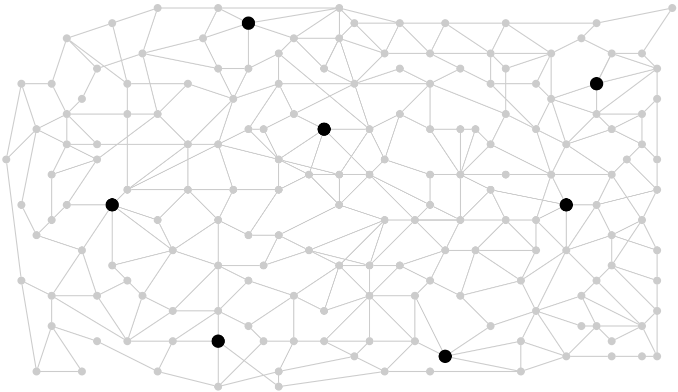
$$\mathrm{d}(x,v) \leq \tilde{\mathrm{d}}(x,v) \leq (1+\epsilon)\,\mathrm{d}^h(x,v)\,.$$

Idee: Reduktion auf Overlay Netzwerk

Bilde Graph mit Zentren als Knoten:

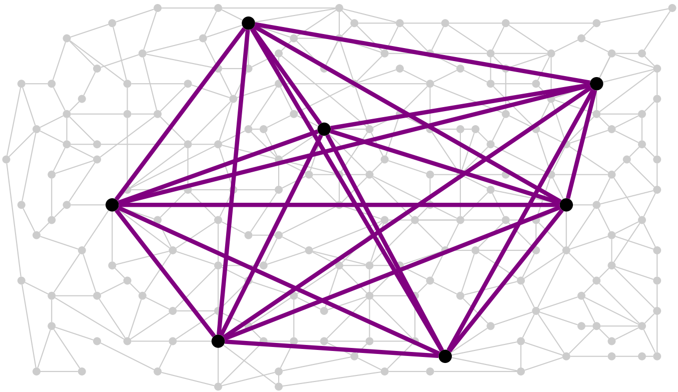
Idee: Reduktion auf Overlay Netzwerk

Bilde Graph mit Zentren als Knoten:



Idee: Reduktion auf Overlay Netzwerk

Bilde Graph mit Zentren als Knoten:



Algorithmus

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c + 2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu

Algorithmus

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu
2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x, v)$, für die gilt: $\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1+\epsilon)\, \mathrm{d}^h(x, v)$

Algorithmus

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu

2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x, v)$, für die gilt: $\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1+\epsilon)\,\mathrm{d}^h(x, v)$

3. Mache $\tilde{\mathrm{d}}(x, y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcasts bekannt

Algorithmus

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu

2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x,v)$, für die gilt: $\mathrm{d}(x,v) \leq \tilde{\mathrm{d}}(x,v) \leq (1+\epsilon)\,\mathrm{d}^h(x,v)$

3. Mache $\tilde{\mathrm{d}}(x,y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcasts bekannt

4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $w_{H_v}(x,y) = \tilde{\mathrm{d}}(x,y)$ für jede Kante (x,y)

Algorithmus

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu

2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x, v)$, für die gilt: $\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1 + \epsilon)\,\mathrm{d}^h(x, v)$

3. Mache $\tilde{\mathrm{d}}(x, y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcasts bekannt

4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $w_{H_v}(x, y) = \tilde{\mathrm{d}}(x, y)$ für jede Kante (x, y)
5. Intern für jeden Knoten v: Berechne $\delta(s, v) = \mathrm{d}_{H_v}(s, v)$ als Ergebnis

Algorithmus

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu
   $|Z| = O((n/h)\log n)$ mit hoher Wahrscheinlichkeit
2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x,v)$, für die gilt: $\mathrm{d}(x,v) \leq \tilde{\mathrm{d}}(x,v) \leq (1+\epsilon)\,\mathrm{d}^h(x,v)$
3. Mache $\tilde{\mathrm{d}}(x,y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcasts bekannt
4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $w_{H_v}(x,y) = \tilde{\mathrm{d}}(x,y)$ für jede Kante (x,y)
5. Intern für jeden Knoten v: Berechne $\delta(s,v) = \mathrm{d}_{H_v}(s,v)$ als Ergebnis

Algorithmus

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu
   $|Z| = O((n/h)\log n)$ mit hoher Wahrscheinlichkeit
2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x,v)$, für die gilt: $\mathrm{d}(x,v) \leq \tilde{\mathrm{d}}(x,v) \leq (1+\epsilon)\,\mathrm{d}^h(x,v)$
   #Runden: $O((|Z|+h) \cdot \log(nW)\log(n)/\epsilon)$
3. Mache $\tilde{\mathrm{d}}(x,y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcasts bekannt
4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $w_{H_v}(x,y) = \tilde{\mathrm{d}}(x,y)$ für jede Kante (x,y)
5. Intern für jeden Knoten v: Berechne $\delta(s,v) = \mathrm{d}_{H_v}(s,v)$ als Ergebnis

Algorithmus

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu
   $|Z| = O((n/h)\log n)$ mit hoher Wahrscheinlichkeit
2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x,v)$, für die gilt: $\mathrm{d}(x,v) \leq \tilde{\mathrm{d}}(x,v) \leq (1+\epsilon)\,\mathrm{d}^h(x,v)$
   #Runden: $O((|Z|+h)\cdot\log(nW)\log(n)/\epsilon)$
3. Mache $\tilde{\mathrm{d}}(x,y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcasts bekannt
   #Runden: $O(|Z|^2 + D)$
4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $w_{H_v}(x,y) = \tilde{\mathrm{d}}(x,y)$ für jede Kante (x,y)
5. Intern für jeden Knoten v: Berechne $\delta(s,v) = \mathrm{d}_{H_v}(s,v)$ als Ergebnis

Algorithmus

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu
   $|Z| = O((n/h)\log n)$ mit hoher Wahrscheinlichkeit
2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x,v)$, für die gilt: $\mathrm{d}(x,v) \leq \tilde{\mathrm{d}}(x,v) \leq (1+\epsilon)\,\mathrm{d}^h(x,v)$
   #Runden: $O((|Z|+h)\cdot\log(nW)\log(n)/\epsilon)$
3. Mache $\tilde{\mathrm{d}}(x,y)$ für alle Paare $x,y \in Z$ im gesamten Netzwerk durch Up- und Downcasts bekannt
   #Runden: $O(|Z|^2 + D)$
4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $w_{H_v}(x,y) = \tilde{\mathrm{d}}(x,y)$ für jede Kante (x,y)
5. Intern für jeden Knoten v: Berechne $\delta(s,v) = \mathrm{d}_{H_v}(s,v)$ als Ergebnis

#Runden: $O((|Z|+h)\cdot\log(nW)\log(n)/\epsilon + |Z|^2 + D)$ mit $|Z| = O(n\log n/h)$

Algorithmus

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu
   $|Z| = O((n/h)\log n)$ mit hoher Wahrscheinlichkeit
2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x,v)$, für die gilt: $\mathrm{d}(x,v) \leq \tilde{\mathrm{d}}(x,v) \leq (1+\epsilon)\,\mathrm{d}^h(x,v)$
   #Runden: $O((|Z|+h) \cdot \log(nW)\log(n)/\epsilon)$
3. Mache $\tilde{\mathrm{d}}(x,y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcasts bekannt
   #Runden: $O(|Z|^2 + D)$
4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $w_{H_v}(x,y) = \tilde{\mathrm{d}}(x,y)$ für jede Kante (x,y)
5. Intern für jeden Knoten v: Berechne $\delta(s,v) = \mathrm{d}_{H_v}(s,v)$ als Ergebnis

#Runden: $O((|Z|+h) \cdot \log(nW)\log(n)/\epsilon + |Z|^2 + D)$ mit $|Z| = O(n\log n/h)$
Setze $h = n^{2/3}$:

Algorithmus

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu
   $|Z| = O((n/h)\log n)$ mit hoher Wahrscheinlichkeit
2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x,v)$, für die gilt: $\mathrm{d}(x,v) \leq \tilde{\mathrm{d}}(x,v) \leq (1+\epsilon)\,\mathrm{d}^h(x,v)$
   #Runden: $O((|Z|+h)\cdot\log(nW)\log(n)/\epsilon)$
3. Mache $\tilde{\mathrm{d}}(x,y)$ für alle Paare $x,y \in Z$ im gesamten Netzwerk durch Up- und Downcasts bekannt
   #Runden: $O(|Z|^2 + D)$
4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $w_{H_v}(x,y) = \tilde{\mathrm{d}}(x,y)$ für jede Kante (x,y)
5. Intern für jeden Knoten v: Berechne $\delta(s,v) = \mathrm{d}_{H_v}(s,v)$ als Ergebnis

#Runden: $O((|Z|+h)\cdot\log(nW)\log(n)/\epsilon + |Z|^2 + D)$ mit $|Z| = O(n\log n/h)$
Setze $h = n^{2/3}$: $O((n^{1/3}\log n + n^{2/3})\log(nW)\log(n)/\epsilon + n^{2/3}\log^2 n + D) = O(n^{2/3}\log(nW)\log(n)/\epsilon + D)$

Korrektheit

Lemma

Mit hoher Wahrscheinlichkeit gilt: $\mathrm{d}_G(s, v) \leq \mathrm{d}_{H_v}(s, v) \leq (1 + \epsilon)\, \mathrm{d}_G(s, v)$

Korrektheit

Lemma

Mit hoher Wahrscheinlichkeit gilt: $\mathrm{d}_G(s, v) \leq \mathrm{d}_{H_v}(s, v) \leq (1 + \epsilon)\, \mathrm{d}_G(s, v)$

Beweis:

- Erste Ungleichung $\mathrm{d}_G(s, v) \leq \mathrm{d}_{H_v}(s, v)$ gilt, weil Kantengewichte in H_v echte Distanz nicht unterschätzen

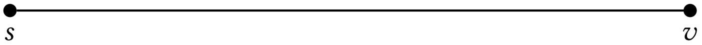
Korrektheit

Lemma

Mit hoher Wahrscheinlichkeit gilt: $\mathrm{d}_G(s, v) \leq \mathrm{d}_{H_v}(s, v) \leq (1 + \epsilon)\,\mathrm{d}_G(s, v)$

Beweis:

- Erste Ungleichung $\mathrm{d}_G(s, v) \leq \mathrm{d}_{H_v}(s, v)$ gilt, weil Kantengewichte in H_v echte Distanz nicht unterschätzen
- Sei π kürzester Weg von s nach v in G



Korrektheit

Lemma

Mit hoher Wahrscheinlichkeit gilt: $\mathrm{d}_G(s,v) \leq \mathrm{d}_{H_v}(s,v) \leq (1+\epsilon)\,\mathrm{d}_G(s,v)$

Beweis:

- Erste Ungleichung $\mathrm{d}_G(s,v) \leq \mathrm{d}_{H_v}(s,v)$ gilt, weil Kantengewichte in H_v echte Distanz nicht unterschätzen
- Sei π kürzester Weg von s nach v in G
- Mit wiederholter Anwendung des Lemmas: Kann π so wählen, dass nach höchstens h Kanten immer ein Zentrum getroffen wird

Korrektheit

Lemma

Mit hoher Wahrscheinlichkeit gilt: $\mathrm{d}_G(s, v) \leq \mathrm{d}_{H_v}(s, v) \leq (1 + \epsilon)\, \mathrm{d}_G(s, v)$

Beweis:

- Erste Ungleichung $\mathrm{d}_G(s, v) \leq \mathrm{d}_{H_v}(s, v)$ gilt, weil Kantengewichte in H_v echte Distanz nicht unterschätzen
- Sei π kürzester Weg von s nach v in G
- Mit wiederholter Anwendung des Lemmas: Kann π so wählen, dass nach höchstens h Kanten immer ein Zentrum getroffen wird
- Sei $x_1, x_2, \ldots, x_k$ die Sequenz der Zentren auf inneren Knoten von π und setze $x_0 = s$ und $x_{k+1} = v$

Korrektheit

Lemma

Mit hoher Wahrscheinlichkeit gilt: $\mathrm{d}_G(s, v) \leq \mathrm{d}_{H_v}(s, v) \leq (1 + \epsilon)\, \mathrm{d}_G(s, v)$

Beweis:

- Erste Ungleichung $\mathrm{d}_G(s, v) \leq \mathrm{d}_{H_v}(s, v)$ gilt, weil Kantengewichte in H_v echte Distanz nicht unterschätzen
- Sei π kürzester Weg von s nach v in G
- Mit wiederholter Anwendung des Lemmas: Kann π so wählen, dass nach höchstens h Kanten immer ein Zentrum getroffen wird
- Sei $x_1, x_2, \ldots, x_k$ die Sequenz der Zentren auf inneren Knoten von π und setze $x_0 = s$ und $x_{k+1} = v$
- Auf π sind x_i und x_{i+1} nur h Kanten voneinander entfernt

Korrektheit

Lemma

Mit hoher Wahrscheinlichkeit gilt: $\mathrm{d}_G(s, v) \leq \mathrm{d}_{H_v}(s, v) \leq (1+\epsilon)\,\mathrm{d}_G(s, v)$

Beweis:

- Erste Ungleichung $\mathrm{d}_G(s, v) \leq \mathrm{d}_{H_v}(s, v)$ gilt, weil Kantengewichte in H_v echte Distanz nicht unterschätzen
- Sei π kürzester Weg von s nach v in G
- Mit wiederholter Anwendung des Lemmas: Kann π so wählen, dass nach höchstens h Kanten immer ein Zentrum getroffen wird
- Sei $x_1, x_2, \ldots, x_k$ die Sequenz der Zentren auf inneren Knoten von π und setze $x_0 = s$ und $x_{k+1} = v$
- Auf π sind x_i und x_{i+1} nur h Kanten voneinander entfernt
- Somit $\mathrm{d}^h_G(x_i, x_{i+1}) = \mathrm{d}_G(x_i, x_{i+1})$ für alle $0 \leq i \leq k$

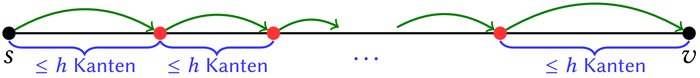
Korrektheit

Lemma

Mit hoher Wahrscheinlichkeit gilt: $\mathrm{d}_G(s,v) \leq \mathrm{d}_{H_v}(s,v) \leq (1+\epsilon)\,\mathrm{d}_G(s,v)$

Beweis:

- Erste Ungleichung $\mathrm{d}_G(s,v) \leq \mathrm{d}_{H_v}(s,v)$ gilt, weil Kantengewichte in H_v echte Distanz nicht unterschätzen
- Sei π kürzester Weg von s nach v in G
- Mit wiederholter Anwendung des Lemmas: Kann π so wählen, dass nach höchstens h Kanten immer ein Zentrum getroffen wird
- Sei $x_1, x_2, \ldots, x_k$ die Sequenz der Zentren auf inneren Knoten von π und setze $x_0 = s$ und $x_{k+1} = v$
- Auf π sind x_i und x_{i+1} nur h Kanten voneinander entfernt
- Somit $\mathrm{d}_G^h(x_i, x_{i+1}) = \mathrm{d}_G(x_i, x_{i+1})$ für alle $0 \leq i \leq k$



Fortsetzung des Beweises

- Da $x_0, \ldots, x_k \in Z$, enthält H_v Kante (x_i, x_{i+1}) für alle $0 \leq i \leq k-1$

Fortsetzung des Beweises

- Da $x_0, \ldots, x_k \in Z$, enthält H_v Kante (x_i, x_{i+1}) für alle $0 \leq i \leq k-1$
- Ebenso enthält H_v Kante (x_k, x_{k+1})

Fortsetzung des Beweises

- Da $x_0, \dots, x_k \in Z$, enthält H_v Kante (x_i, x_{i+1}) für alle $0 \leq i \leq k-1$
- Ebenso enthält H_v Kante (x_k, x_{k+1})
- Betrachte Pfad $\pi' = (x_0, x_1, \dots, x_k, x_{k+1})$ in H_v

Fortsetzung des Beweises

- Da $x_0, \ldots, x_k \in Z$, enthält H_v Kante (x_i, x_{i+1}) für alle $0 \leq i \leq k-1$
- Ebenso enthält H_v Kante (x_k, x_{k+1})
- Betrachte Pfad $\pi' = (x_0, x_1, \ldots, x_k, x_{k+1})$ in H_v
- $w_{H_v}(x_i, x_{i+1}) = \tilde{\mathrm{d}}(x_i, x_{i+1}) \leq (1+\epsilon)\, \mathrm{d}_G^h(x_i, x_{i+1}) = (1+\epsilon)\, \mathrm{d}_G(x_i, x_{i+1})$

Fortsetzung des Beweises

- Da $x_0, \dots, x_k \in Z$, enthält H_v Kante (x_i, x_{i+1}) für alle $0 \leq i \leq k-1$
- Ebenso enthält H_v Kante (x_k, x_{k+1})
- Betrachte Pfad $\pi' = (x_0, x_1, \dots, x_k, x_{k+1})$ in H_v
- $w_{H_v}(x_i, x_{i+1}) = \tilde{\mathrm{d}}(x_i, x_{i+1}) \leq (1+\epsilon)\, \mathrm{d}_G^h(x_i, x_{i+1}) = (1+\epsilon)\, \mathrm{d}_G(x_i, x_{i+1})$
- Somit:

$$
\begin{aligned}
\mathrm{d}_{H_v}(s, v) &\leq w_{H_v}(\pi') \\
&= \sum_{i=0}^{k} w_{H_v}(x_i, x_{i+1}) \\
&\leq (1+\epsilon) \sum_{i=0}^{k} \mathrm{d}_G(x_i, x_{i+1}) \\
&= (1+\epsilon)\, \mathrm{d}_G(s, v)
\end{aligned}
$$

Sampling

Lemma

Sei h ein Parameter und sei Z (Zentren) eine Menge zu der jeder Knoten unabhängig mit Wahrscheinlichkeit $p = ((c + 2) \ln n)/h$ hinzugefügt wurde. Dann gilt mit Wahrscheinlichkeit mindestens $1 - \frac{1}{n^c}$: Für jedes Knotenpaar u und v gibt es einen kürzesten Weg von u nach v, der innerhalb der ersten h Knoten ein Zentrum enthält.

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \ldots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \ldots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \ldots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \dots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

$$\Pr[X_i > h] = (1-p)^h$$

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \dots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

$$\Pr[X_i > h] = (1-p)^h = (1-p)^{((c+2)\ln n)/p}$$

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \dots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

$$\Pr[X_i > h] = (1-p)^h = (1-p)^{((c+2)\ln n)/p} = \left((1-p)^{\frac{1}{p}}\right)^{\ln n^{c+2}}$$

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \dots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

$$\Pr[X_i > h] = (1-p)^h = (1-p)^{((c+2)\ln n)/p} = \left((1-p)^{\frac{1}{p}}\right)^{\ln n^{c+2}}$$

Wichtige Ungleichung

$(1-x)^{\frac{1}{x}} \leq \frac{1}{e}$ für $x \geq 1$

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \dots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

$$\Pr[X_i > h] = (1-p)^h = (1-p)^{((c+2)\ln n)/p} = \left((1-p)^{\frac{1}{p}}\right)^{\ln n^{c+2}}$$

Wichtige Ungleichung

$(1-x)^{\frac{1}{x}} \leq \frac{1}{e}$ für $x \geq 1$

$$\Pr[X_i > h] \leq \left(\frac{1}{e}\right)^{\ln n^{c+2}} = \frac{1}{n^{c+2}}$$

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \dots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

$$\Pr[X_i > h] = (1-p)^h = (1-p)^{((c+2)\ln n)/p} = \left((1-p)^{\frac{1}{p}}\right)^{\ln n^{c+2}}$$

Wichtige Ungleichung

$(1-x)^{\frac{1}{x}} \leq \frac{1}{e}$ für $x \geq 1$

$$\Pr[X_i > h] \leq \left(\frac{1}{e}\right)^{\ln n^{c+2}} = \frac{1}{n^{c+2}}$$

$$\Pr\left[\bigwedge_{i=1}^{\ell} X_i \leq h\right]$$

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \dots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

$$\Pr[X_i > h] = (1-p)^h = (1-p)^{((c+2)\ln n)/p} = \left((1-p)^{\frac{1}{p}}\right)^{\ln n^{c+2}}$$

Wichtige Ungleichung

$(1-x)^{\frac{1}{x}} \leq \frac{1}{e}$ für $x \geq 1$

$$\Pr[X_i > h] \leq \left(\frac{1}{e}\right)^{\ln n^{c+2}} = \frac{1}{n^{c+2}}$$

$$\Pr\left[\bigwedge_{i=1}^{\ell} X_i \leq h\right] = 1 - \Pr\left[\bigvee_{i=1}^{\ell} X_i > h\right]$$

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \dots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

$$\Pr[X_i > h] = (1-p)^h = (1-p)^{((c+2)\ln n)/p} = \left((1-p)^{\frac{1}{p}}\right)^{\ln n^{c+2}}$$

Wichtige Ungleichung

$(1-x)^{\frac{1}{x}} \leq \frac{1}{e}$ für $x \geq 1$

$$\Pr[X_i > h] \leq \left(\frac{1}{e}\right)^{\ln n^{c+2}} = \frac{1}{n^{c+2}}$$

$$\Pr\left[\bigwedge_{i=1}^{\ell} X_i \leq h\right] = 1 - \Pr\left[\bigvee_{i=1}^{\ell} X_i > h\right] \geq 1 - \sum_{i=1}^{\ell} \Pr\left[X_i > h\right]$$

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \dots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

$$\Pr[X_i > h] = (1-p)^h = (1-p)^{((c+2)\ln n)/p} = \left((1-p)^{\frac{1}{p}}\right)^{\ln n^{c+2}}$$

Wichtige Ungleichung

$(1-x)^{\frac{1}{x}} \leq \frac{1}{e}$ für $x \geq 1$

$$\Pr[X_i > h] \leq \left(\frac{1}{e}\right)^{\ln n^{c+2}} = \frac{1}{n^{c+2}}$$

$$\Pr\left[\bigwedge_{i=1}^{\ell} X_i \leq h\right] = 1 - \Pr\left[\bigvee_{i=1}^{\ell} X_i > h\right] \geq 1 - \sum_{i=1}^{\ell} \Pr\left[X_i > h\right] \geq 1 - \ell \cdot \frac{1}{n^{c+2}}$$

Beweis des Lemmas

- **Technisches Detail:** Fixiere in Analyse für jedes Paar von Knoten u, v *einen* der kürzesten Wege von u nach v
- Sei Z die Menge der Zentren und seien $\pi_1, \pi_2, \dots \pi_\ell$ (mit $\ell \leq n^2$) die paarweisen kürzesten Wege
- Sei $X_i \geq 1$ die Zufallsvariable für die Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

$$\Pr[X_i > h] = (1-p)^h = (1-p)^{((c+2)\ln n)/p} = \left((1-p)^{\frac{1}{p}}\right)^{\ln n^{c+2}}$$

Wichtige Ungleichung

$(1-x)^{\frac{1}{x}} \leq \frac{1}{e}$ für $x \geq 1$

$$\Pr[X_i > h] \leq \left(\frac{1}{e}\right)^{\ln n^{c+2}} = \frac{1}{n^{c+2}}$$

$$\Pr\left[\bigwedge_{i=1}^{\ell} X_i \leq h\right] = 1 - \Pr\left[\bigvee_{i=1}^{\ell} X_i > h\right] \geq 1 - \sum_{i=1}^{\ell} \Pr\left[X_i > h\right] \geq 1 - \ell \cdot \frac{1}{n^{c+2}} \geq 1 - \frac{1}{n^c}$$

Approximation der h-Distanzen für Zentren

Lemma

Sei Z eine Teilmenge von Knoten und h ein Parameter. In $O((|Z| + h) \log(nW) \log(n)/\epsilon)$ Runden kann für jedes $x \in Z$ und jedes $v \in V$ eine approximative Distanz $\tilde{\mathrm{d}}(x, v)$ berechnet werden (die v am Ende kennt), für die mit hoher Wahrscheinlichkeit gilt:

$$\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1 + \epsilon)\, \mathrm{d}^h(x, v)\,.$$

Approximation der h-Distanzen für Zentren

Lemma

Sei Z eine Teilmenge von Knoten und h ein Parameter. In $O((|Z| + h) \log(nW) \log(n)/\epsilon)$ Runden kann für jedes $x \in Z$ und jedes $v \in V$ eine approximative Distanz $\tilde{\mathrm{d}}(x, v)$ berechnet werden (die v am Ende kennt), für die mit hoher Wahrscheinlichkeit gilt:

$$\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1 + \epsilon)\, \mathrm{d}^h(x, v)\,.$$

Anmerkungen:

- Für jeden Knoten können h-Distanzen mit Bellman-Ford Algorithmus berechnet werden

Approximation der h-Distanzen für Zentren

Lemma

Sei Z eine Teilmenge von Knoten und h ein Parameter. In $O((|Z| + h) \log(nW) \log(n)/\epsilon)$ Runden kann für jedes $x \in Z$ und jedes $v \in V$ eine approximative Distanz $\tilde{\mathrm{d}}(x, v)$ berechnet werden (die v am Ende kennt), für die mit hoher Wahrscheinlichkeit gilt:

$$\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1 + \epsilon)\, \mathrm{d}^h(x, v)\,.$$

Anmerkungen:

- Für jeden Knoten können h-Distanzen mit Bellman-Ford Algorithmus berechnet werden
- Parallelisierung mit Random-Delay Technik mit Bellman-Ford nicht sinvoll, da jeder Knoten in jeder Runde Nachrichten sendet

Approximation der h-Distanzen für Zentren

Lemma

Sei Z eine Teilmenge von Knoten und h ein Parameter. In $O((|Z| + h) \log(nW) \log(n)/\epsilon)$ Runden kann für jedes $x \in Z$ und jedes $v \in V$ eine approximative Distanz $\tilde{\mathrm{d}}(x, v)$ berechnet werden (die v am Ende kennt), für die mit hoher Wahrscheinlichkeit gilt:

$$\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1 + \epsilon)\, \mathrm{d}^h(x, v)\,.$$

Anmerkungen:

- Für jeden Knoten können h-Distanzen mit Bellman-Ford Algorithmus berechnet werden
- Parallelisierung mit Random-Delay Technik mit Bellman-Ford nicht sinvoll, da jeder Knoten in jeder Runde Nachrichten sendet
- Alternative: Bei gewichteter Breitensuche sendet jeder Knoten insgesamt nur in $O(1)$ vielen Runden, aber Laufzeit hat Faktor W

Approximation der h-Distanzen für Zentren

Lemma

Sei Z eine Teilmenge von Knoten und h ein Parameter. In $O((|Z| + h)\log(nW)\log(n)/\epsilon)$ Runden kann für jedes $x \in Z$ und jedes $v \in V$ eine approximative Distanz $\tilde{\mathrm{d}}(x, v)$ berechnet werden (die v am Ende kennt), für die mit hoher Wahrscheinlichkeit gilt:

$$\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1 + \epsilon)\,\mathrm{d}^h(x, v)\,.$$

Anmerkungen:

- Für jeden Knoten können h-Distanzen mit Bellman-Ford Algorithmus berechnet werden
- Parallelisierung mit Random-Delay Technik mit Bellman-Ford nicht sinvoll, da jeder Knoten in jeder Runde Nachrichten sendet
- Alternative: Bei gewichteter Breitensuche sendet jeder Knoten insgesamt nur in $O(1)$ vielen Runden, aber Laufzeit hat Faktor W
- Daher: Parallele Ausführung eines Approximationsalgorithmus für SSSP (mit Abhängigkeit $\log W$)

Runden der Gewichte I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Runden der Gewichte I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v) / \rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Notation:

- $\tilde{\mathrm{d}}_i(\cdot, \cdot)$: Distanz mit Gewichten $\tilde{w}_i(\cdot, \cdot)$
- $\mathrm{d}_i^{\downarrow}(\cdot, \cdot)$: Distanz mit Gewichten $w_i^{\downarrow}(\cdot, \cdot)$

Runden der Gewichte I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v)/\rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s, v) \geq \mathrm{d}(s, v)$

Notation:

- $\tilde{\mathrm{d}}_i(\cdot, \cdot)$: Distanz mit Gewichten $\tilde{w}_i(\cdot, \cdot)$
- $\mathrm{d}_i^{\downarrow}(\cdot, \cdot)$: Distanz mit Gewichten $w_i^{\downarrow}(\cdot, \cdot)$

Runden der Gewichte I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$
2. Falls $\mathrm{d}^h(s,v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$

Notation:

- $\tilde{\mathrm{d}}_i(\cdot,\cdot)$: Distanz mit Gewichten $\tilde{w}_i(\cdot,\cdot)$
- $\mathrm{d}_i^{\downarrow}(\cdot,\cdot)$: Distanz mit Gewichten $w_i^{\downarrow}(\cdot,\cdot)$

Runden der Gewichte I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$
2. Falls $\mathrm{d}^h(s,v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$
3. Falls $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$

Notation:

- $\tilde{\mathrm{d}}_i(\cdot,\cdot)$: Distanz mit Gewichten $\tilde{w}_i(\cdot,\cdot)$
- $\mathrm{d}_i^{\downarrow}(\cdot,\cdot)$: Distanz mit Gewichten $w_i^{\downarrow}(\cdot,\cdot)$

Runden der Gewichte I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v) / \rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s, v) \geq \mathrm{d}(s, v)$
2. Falls $\mathrm{d}^h(s, v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s, v) \leq (1 + \epsilon)\, \mathrm{d}^h(s, v)$
3. Falls $2^i \leq \mathrm{d}^h(s, v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s, v) \leq \frac{4}{\epsilon} h$

Notation:

- $\tilde{\mathrm{d}}_i(\cdot, \cdot)$: Distanz mit Gewichten $\tilde{w}_i(\cdot, \cdot)$
- $\mathrm{d}_i^{\downarrow}(\cdot, \cdot)$: Distanz mit Gewichten $w_i^{\downarrow}(\cdot, \cdot)$

Idee:

- $\tilde{w}_i(u, v)$ rundet Gewicht $w(u, v)$ auf das nächste Vielfache von ρ_i

Runden der Gewichte I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$
2. Falls $\mathrm{d}^h(s,v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$
3. Falls $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$

Notation:

- $\tilde{\mathrm{d}}_i(\cdot,\cdot)$: Distanz mit Gewichten $\tilde{w}_i(\cdot,\cdot)$
- $\mathrm{d}_i^{\downarrow}(\cdot,\cdot)$: Distanz mit Gewichten $w_i^{\downarrow}(\cdot,\cdot)$

Idee:

- $\tilde{w}_i(u,v)$ rundet Gewicht $w(u,v)$ auf das nächste Vielfache von ρ_i
- $\tilde{\mathrm{d}}_i(s,v)$ approximiert h-Distanz $\mathrm{d}^h(s,v)$ sofern $\mathrm{d}^h(s,v) \geq 2^i$

Runden der Gewichte I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$
2. Falls $\mathrm{d}^h(s,v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$
3. Falls $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$

Notation:

- $\tilde{\mathrm{d}}_i(\cdot,\cdot)$: Distanz mit Gewichten $\tilde{w}_i(\cdot,\cdot)$
- $\mathrm{d}_i^{\downarrow}(\cdot,\cdot)$: Distanz mit Gewichten $w_i^{\downarrow}(\cdot,\cdot)$

Idee:

- $\tilde{w}_i(u,v)$ rundet Gewicht $w(u,v)$ auf das nächste Vielfache von ρ_i
- $\tilde{\mathrm{d}}_i(s,v)$ approximiert h-Distanz $\mathrm{d}^h(s,v)$ sofern $\mathrm{d}^h(s,v) \geq 2^i$
- Berechnung von $\tilde{\mathrm{d}}_i(s,v)$ durch Berechnung von $\mathrm{d}_i^{\downarrow}(s,v)$
  Es gilt: $\tilde{\mathrm{d}}_i(s,v) = \mathrm{d}_i^{\downarrow}(s,v) \cdot \rho_i$

Runden der Gewichte I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v) / \rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s, v) \geq \mathrm{d}(s, v)$
2. Falls $\mathrm{d}^h(s, v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s, v) \leq (1 + \epsilon)\, \mathrm{d}^h(s, v)$
3. Falls $2^i \leq \mathrm{d}^h(s, v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s, v) \leq \frac{4}{\epsilon} h$

Notation:

- $\tilde{\mathrm{d}}_i(\cdot, \cdot)$: Distanz mit Gewichten $\tilde{w}_i(\cdot, \cdot)$
- $\mathrm{d}_i^{\downarrow}(\cdot, \cdot)$: Distanz mit Gewichten $w_i^{\downarrow}(\cdot, \cdot)$

Idee:

- $\tilde{w}_i(u, v)$ rundet Gewicht $w(u, v)$ auf das nächste Vielfache von ρ_i
- $\tilde{\mathrm{d}}_i(s, v)$ approximiert h-Distanz $\mathrm{d}^h(s, v)$ sofern $\mathrm{d}^h(s, v) \geq 2^i$
- Berechnung von $\tilde{\mathrm{d}}_i(s, v)$ durch Berechnung von $\mathrm{d}_i^{\downarrow}(s, v)$
  Es gilt: $\tilde{\mathrm{d}}_i(s, v) = \mathrm{d}_i^{\downarrow}(s, v) \cdot \rho_i$
- $\Rightarrow$ Effiziente Berechnung von $\mathrm{d}_i^{\downarrow}(s, v)$ durch gewichtete Breitensuche sofern $2^i \leq \mathrm{d}^h(s, v) \leq 2^{i+1}$

Runden der Gewichte II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v) / \rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s, v) \geq \mathrm{d}(s, v)$
2. Falls $\mathrm{d}^h(s, v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s, v) \leq (1 + \epsilon)\, \mathrm{d}^h(s, v)$
3. Falls $2^i \leq \mathrm{d}^h(s, v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s, v) \leq \frac{4}{\epsilon} h$

Runden der Gewichte II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$
2. Falls $\mathrm{d}^h(s,v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$
3. Falls $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i \leq \log(hW)$: Berechne $\mathrm{d}_i^{\downarrow}(s,v)$ für jeden Knoten v mit $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$

Runden der Gewichte II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$
2. Falls $\mathrm{d}^h(s,v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$
3. Falls $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon}h$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i \leq \log(hW)$: Berechne $\mathrm{d}_i^{\downarrow}(s,v)$ für jeden Knoten v mit $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon}h$ Laufzeit $O(\log(hW) \cdot h/\epsilon)$

Runden der Gewichte II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v) / \rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s, v) \geq \mathrm{d}(s, v)$
2. Falls $\mathrm{d}^h(s, v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s, v) \leq (1 + \epsilon)\, \mathrm{d}^h(s, v)$
3. Falls $2^i \leq \mathrm{d}^h(s, v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s, v) \leq \frac{4}{\epsilon} h$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i \leq \log(hW)$: Berechne $\mathrm{d}_i^{\downarrow}(s, v)$ für jeden Knoten v mit $\mathrm{d}_i^{\downarrow}(s, v) \leq \frac{4}{\epsilon} h$ Laufzeit $O(\log(hW) \cdot h/\epsilon)$
2. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}_i(s, v) = \mathrm{d}_i^{\downarrow}(s, v) \cdot \rho_i$

Runden der Gewichte II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$
2. Falls $\mathrm{d}^h(s,v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$
3. Falls $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i \leq \log(hW)$: Berechne $\mathrm{d}_i^{\downarrow}(s,v)$ für jeden Knoten v mit $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$ Laufzeit $O(\log(hW) \cdot h/\epsilon)$
2. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}_i(s,v) = \mathrm{d}_i^{\downarrow}(s,v) \cdot \rho_i$
3. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}(s,v) := \min_{0 \leq i \leq \log(hW)} \tilde{\mathrm{d}}_i(s,v)$

Runden der Gewichte II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v) / \rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s, v) \geq \mathrm{d}(s, v)$
2. Falls $\mathrm{d}^h(s, v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s, v) \leq (1 + \epsilon)\, \mathrm{d}^h(s, v)$
3. Falls $2^i \leq \mathrm{d}^h(s, v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s, v) \leq \frac{4}{\epsilon} h$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i \leq \log (hW)$: Berechne $\mathrm{d}_i^{\downarrow}(s, v)$ für jeden Knoten v mit $\mathrm{d}_i^{\downarrow}(s, v) \leq \frac{4}{\epsilon} h$ Laufzeit $O(\log (hW) \cdot h / \epsilon)$
2. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}_i(s, v) = \mathrm{d}_i^{\downarrow}(s, v) \cdot \rho_i$
3. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}(s, v) := \min_{0 \leq i \leq \log (hW)} \tilde{\mathrm{d}}_i(s, v)$

Korrektheit:

- Wegen (1): $\tilde{\mathrm{d}}(s, v) \geq \mathrm{d}(s, v)$

Runden der Gewichte II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$
2. Falls $\mathrm{d}^h(s,v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$
3. Falls $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i \leq \log(hW)$: Berechne $\mathrm{d}_i^{\downarrow}(s,v)$ für jeden Knoten v mit $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$ Laufzeit $O(\log(hW) \cdot h/\epsilon)$
2. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}_i(s,v) = \mathrm{d}_i^{\downarrow}(s,v) \cdot \rho_i$
3. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}(s,v) := \min_{0 \leq i \leq \log(hW)} \tilde{\mathrm{d}}_i(s,v)$

Korrektheit:

- Wegen (1): $\tilde{\mathrm{d}}(s,v) \geq \mathrm{d}(s,v)$
- Jede h-Distanz $\mathrm{d}^h(s,v)$ fällt in einen Bereich $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$

Runden der Gewichte II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$
2. Falls $\mathrm{d}^h(s,v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$
3. Falls $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i \leq \log(hW)$: Berechne $\mathrm{d}_i^{\downarrow}(s,v)$ für jeden Knoten v mit $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$ Laufzeit $O(\log(hW) \cdot h/\epsilon)$
2. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}_i(s,v) = \mathrm{d}_i^{\downarrow}(s,v) \cdot \rho_i$
3. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}(s,v) := \min_{0 \leq i \leq \log(hW)} \tilde{\mathrm{d}}_i(s,v)$

Korrektheit:

- Wegen (1): $\tilde{\mathrm{d}}(s,v) \geq \mathrm{d}(s,v)$
- Jede h-Distanz $\mathrm{d}^h(s,v)$ fällt in einen Bereich $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$
- Wegen (3): $\tilde{\mathrm{d}}_i(s,v)$ korrekt berechnet

Runden der Gewichte II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Eigenschaften:

1. $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$
2. Falls $\mathrm{d}^h(s,v) \geq 2^i$: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$
3. Falls $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i \leq \log(hW)$: Berechne $\mathrm{d}_i^{\downarrow}(s,v)$ für jeden Knoten v mit $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$ Laufzeit $O(\log(hW) \cdot h/\epsilon)$
2. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}_i(s,v) = \mathrm{d}_i^{\downarrow}(s,v) \cdot \rho_i$
3. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}(s,v) := \min_{0 \leq i \leq \log(hW)} \tilde{\mathrm{d}}_i(s,v)$

Korrektheit:

- Wegen (1): $\tilde{\mathrm{d}}(s,v) \geq \mathrm{d}(s,v)$
- Jede h-Distanz $\mathrm{d}^h(s,v)$ fällt in einen Bereich $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$
- Wegen (3): $\tilde{\mathrm{d}}_i(s,v)$ korrekt berechnet
- Wegen (2): $\tilde{\mathrm{d}}(s,v) \leq \tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$

Parallelisierung

Approximation der h-Distanzen

Lemma

Eine Distanzapproximation $\tilde{\mathrm{d}}(s, \cdot)$*, für die*

$$\mathrm{d}(s, v) \leq \tilde{\mathrm{d}}(s, v) \leq (1 + \epsilon)\, \mathrm{d}^h(s, v)$$

für jeden Knoten v *gilt, kann in* $O(h \log(hW)/\epsilon)$ *vielen Runden berechnet werden. Dabei sendet jeder Knoten in* $O(\log(hW))$ *vielen Runden.*

Parallelisierung

Approximation der h-Distanzen

Lemma

Eine Distanzapproximation $\tilde{\mathrm{d}}(s, \cdot)$, *für die*

$$\mathrm{d}(s, v) \leq \tilde{\mathrm{d}}(s, v) \leq (1 + \epsilon)\, \mathrm{d}^h(s, v)$$

für jeden Knoten v *gilt, kann in* $O(h \log(hW)/\epsilon)$ *vielen Runden berechnet werden. Dabei sendet jeder Knoten in* $O(\log(hW))$ *vielen Runden.*

Parallelisierung mit Random Delay Technik:

Lemma

Sei Z *eine Teilmenge von Knoten und* h *ein Parameter. In* $O((|Z| + h) \log(nW) \log(n)/\epsilon)$ *Runden kann für jedes* $x \in Z$ *und jedes* $v \in V$ *eine approximative Distanz* $\tilde{\mathrm{d}}(x, v)$ *berechnet werden (die* v *am Ende kennt), für die mit hoher Wahrscheinlichkeit gilt:*

$$\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1 + \epsilon)\, \mathrm{d}^h(x, v)\,.$$

Beweis der Eigenschaften I

$$\rho_i = \epsilon 2^i / h$$

$$w_i^{\downarrow}(u, v) = \lceil w(u, v) / \rho_i \rceil$$

$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Beweis der Eigenschaften I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$

Beweis der Eigenschaften I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$

Sei $\tilde{\pi}$ kürzester Weg von s nach v für Gewichte $\tilde{w}_i(\cdot,\cdot)$

Beweis der Eigenschaften I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v) / \rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s, v) \geq \mathrm{d}(s, v)$

Sei $\tilde{\pi}$ kürzester Weg von s nach v für Gewichte $\tilde{w}_i(\cdot, \cdot)$

$$\tilde{\mathrm{d}}_i(s, v) = \sum_{(x,y) \in \tilde{\pi}} \tilde{w}_i(x, y)$$

Beweis der Eigenschaften I

$$\begin{aligned} \rho_i &= \epsilon 2^i / h \\ w_i^{\downarrow}(u,v) &= \lceil w(u,v)/\rho_i \rceil \\ \tilde{w}_i(u,v) &= w_i^{\downarrow}(u,v) \cdot \rho_i \end{aligned}$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$

Sei $\tilde{\pi}$ kürzester Weg von s nach v für Gewichte $\tilde{w}_i(\cdot,\cdot)$

$$\begin{aligned} \tilde{\mathrm{d}}_i(s,v) &= \sum_{(x,y)\in\tilde{\pi}} \tilde{w}_i(x,y) \\ &= \sum_{(x,y)\in\tilde{\pi}} \left\lceil \frac{w(x,y)}{\rho_i} \right\rceil \cdot \rho_i \end{aligned}$$

Beweis der Eigenschaften I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v)/\rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s, v) \geq \mathrm{d}(s, v)$

Sei $\tilde{\pi}$ kürzester Weg von s nach v für Gewichte $\tilde{w}_i(\cdot, \cdot)$

$$\begin{aligned} \tilde{\mathrm{d}}_i(s, v) &= \sum_{(x,y) \in \tilde{\pi}} \tilde{w}_i(x, y) \\ &= \sum_{(x,y) \in \tilde{\pi}} \left\lceil \frac{w(x, y)}{\rho_i} \right\rceil \cdot \rho_i \\ &\geq \sum_{(x,y) \in \tilde{\pi}} w(x, y) \end{aligned}$$

Beweis der Eigenschaften I

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s,v) \geq \mathrm{d}(s,v)$

Sei $\tilde{\pi}$ kürzester Weg von s nach v für Gewichte $\tilde{w}_i(\cdot,\cdot)$

$$\begin{aligned}
\tilde{\mathrm{d}}_i(s,v) &= \sum_{(x,y)\in\tilde{\pi}} \tilde{w}_i(x,y) \\
&= \sum_{(x,y)\in\tilde{\pi}} \left\lceil \frac{w(x,y)}{\rho_i} \right\rceil \cdot \rho_i \\
&\geq \sum_{(x,y)\in\tilde{\pi}} w(x,y) \\
&= w(\tilde{\pi}) \geq \mathrm{d}(s,v)
\end{aligned}$$

Beweis der Eigenschaften II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$, wenn $\mathrm{d}^h(s,v) \geq 2^i$

Beweis der Eigenschaften II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v)/\rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s, v) \leq (1 + \epsilon)\, \mathrm{d}^h(s, v)$, wenn $\mathrm{d}^h(s, v) \geq 2^i$

Sei π kürzester Weg s nach v mit höchstens h Kanten für Gewichte $w(\cdot, \cdot)$

Beweis der Eigenschaften II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v) / \rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s, v) \leq (1 + \epsilon) \, \mathrm{d}^h(s, v)$, wenn $\mathrm{d}^h(s, v) \geq 2^i$

Sei π kürzester Weg s nach v mit höchstens h Kanten für Gewichte $w(\cdot, \cdot)$

$$\tilde{\mathrm{d}}_i(s, v) \leq \tilde{w}_i(\pi)$$

Beweis der Eigenschaften II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$, wenn $\mathrm{d}^h(s,v) \geq 2^i$

Sei π kürzester Weg s nach v mit höchstens h Kanten für Gewichte $w(\cdot,\cdot)$

$$\tilde{\mathrm{d}}_i(s,v) \leq \tilde{w}_i(\pi) = \sum_{(x,y)\in\pi} \tilde{w}_i(x,y)$$

Beweis der Eigenschaften II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v) / \rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s, v) \leq (1 + \epsilon)\, \mathrm{d}^h(s, v)$, wenn $\mathrm{d}^h(s, v) \geq 2^i$

Sei π kürzester Weg s nach v mit höchstens h Kanten für Gewichte $w(\cdot, \cdot)$

$$\begin{aligned}\tilde{\mathrm{d}}_i(s, v) \leq \tilde{w}_i(\pi) &= \sum_{(x,y) \in \pi} \tilde{w}_i(x, y) \\ &= \sum_{(x,y) \in \pi} \left\lceil \frac{w(x, y)}{\rho_i} \right\rceil \cdot \rho_i\end{aligned}$$

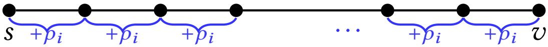
Beweis der Eigenschaften II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v)/\rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s, v) \leq (1 + \epsilon)\, \mathrm{d}^h(s, v)$, wenn $\mathrm{d}^h(s, v) \geq 2^i$

Sei π kürzester Weg s nach v mit höchstens h Kanten für Gewichte $w(\cdot, \cdot)$

$$\tilde{\mathrm{d}}_i(s, v) \leq \tilde{w}_i(\pi) = \sum_{(x,y) \in \pi} \tilde{w}_i(x, y)$$
$$= \sum_{(x,y) \in \pi} \left\lceil \frac{w(x, y)}{\rho_i} \right\rceil \cdot \rho_i \leq \sum_{(x,y) \in \pi} (w(x, y) + \rho_i)$$



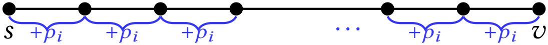
Beweis der Eigenschaften II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$, wenn $\mathrm{d}^h(s,v) \geq 2^i$

Sei π kürzester Weg s nach v mit höchstens h Kanten für Gewichte $w(\cdot,\cdot)$

$$\begin{aligned}
\tilde{\mathrm{d}}_i(s,v) \leq \tilde{w}_i(\pi) &= \sum_{(x,y)\in\pi} \tilde{w}_i(x,y) \\
&= \sum_{(x,y)\in\pi} \left\lceil \frac{w(x,y)}{\rho_i} \right\rceil \cdot \rho_i \leq \sum_{(x,y)\in\pi} (w(x,y) + \rho_i) \\
&= w(\pi) + |\pi| \cdot \rho_i
\end{aligned}$$



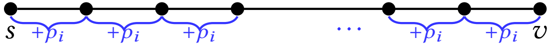
Beweis der Eigenschaften II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$, wenn $\mathrm{d}^h(s,v) \geq 2^i$

Sei π kürzester Weg s nach v mit höchstens h Kanten für Gewichte $w(\cdot,\cdot)$

$$\begin{aligned}\tilde{\mathrm{d}}_i(s,v) \leq \tilde{w}_i(\pi) &= \sum_{(x,y)\in\pi} \tilde{w}_i(x,y) \\ &= \sum_{(x,y)\in\pi} \left\lceil \frac{w(x,y)}{\rho_i} \right\rceil \cdot \rho_i \leq \sum_{(x,y)\in\pi} (w(x,y) + \rho_i) \\ &= w(\pi) + |\pi| \cdot \rho_i = \mathrm{d}^h(s,v) + |\pi| \cdot \rho_i\end{aligned}$$



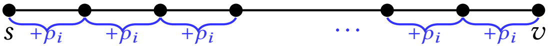
Beweis der Eigenschaften II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v)/\rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s, v) \leq (1 + \epsilon)\, \mathrm{d}^h(s, v)$, wenn $\mathrm{d}^h(s, v) \geq 2^i$

Sei π kürzester Weg s nach v mit höchstens h Kanten für Gewichte $w(\cdot, \cdot)$

$$\begin{aligned}
\tilde{\mathrm{d}}_i(s, v) \leq \tilde{w}_i(\pi) &= \sum_{(x,y) \in \pi} \tilde{w}_i(x, y) \\
&= \sum_{(x,y) \in \pi} \left\lceil \frac{w(x, y)}{\rho_i} \right\rceil \cdot \rho_i \leq \sum_{(x,y) \in \pi} (w(x, y) + \rho_i) \\
&= w(\pi) + |\pi| \cdot \rho_i = \mathrm{d}^h(s, v) + |\pi| \cdot \rho_i \\
&\leq \mathrm{d}^h(s, v) + h \cdot \rho_i
\end{aligned}$$



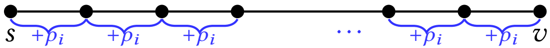
Beweis der Eigenschaften II

$$\begin{aligned} \rho_i &= \epsilon 2^i / h \\ w_i^{\downarrow}(u,v) &= \lceil w(u,v)/\rho_i \rceil \\ \tilde{w}_i(u,v) &= w_i^{\downarrow}(u,v) \cdot \rho_i \end{aligned}$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$, wenn $\mathrm{d}^h(s,v) \geq 2^i$

Sei π kürzester Weg s nach v mit höchstens h Kanten für Gewichte $w(\cdot,\cdot)$

$$\begin{aligned} \tilde{\mathrm{d}}_i(s,v) \leq \tilde{w}_i(\pi) = \sum_{(x,y)\in\pi} \tilde{w}_i(x,y) \\ &= \sum_{(x,y)\in\pi} \left\lceil \frac{w(x,y)}{\rho_i} \right\rceil \cdot \rho_i \leq \sum_{(x,y)\in\pi} (w(x,y) + \rho_i) \\ &= w(\pi) + |\pi| \cdot \rho_i = \mathrm{d}^h(s,v) + |\pi| \cdot \rho_i \\ &\leq \mathrm{d}^h(s,v) + h \cdot \rho_i = \mathrm{d}^h(s,v) + \epsilon 2^i \end{aligned}$$



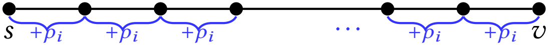
Beweis der Eigenschaften II

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$, wenn $\mathrm{d}^h(s,v) \geq 2^i$

Sei π kürzester Weg s nach v mit höchstens h Kanten für Gewichte $w(\cdot,\cdot)$

$$\begin{aligned}
\tilde{\mathrm{d}}_i(s,v) \leq \tilde{w}_i(\pi) &= \sum_{(x,y)\in\pi} \tilde{w}_i(x,y) \\
&= \sum_{(x,y)\in\pi} \left\lceil \frac{w(x,y)}{\rho_i} \right\rceil \cdot \rho_i \leq \sum_{(x,y)\in\pi} (w(x,y) + \rho_i) \\
&= w(\pi) + |\pi| \cdot \rho_i = \mathrm{d}^h(s,v) + |\pi| \cdot \rho_i \\
&\leq \mathrm{d}^h(s,v) + h \cdot \rho_i = \mathrm{d}^h(s,v) + \epsilon 2^i \\
&\leq \mathrm{d}^h(s,v) + \epsilon\,\mathrm{d}^h(s,v)
\end{aligned}$$



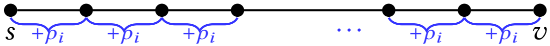
Beweis der Eigenschaften II

$$\begin{aligned} \rho_i &= \epsilon 2^i / h \\ w_i^{\downarrow}(u,v) &= \lceil w(u,v)/\rho_i \rceil \\ \tilde{w}_i(u,v) &= w_i^{\downarrow}(u,v) \cdot \rho_i \end{aligned}$$

Zu zeigen: $\tilde{\mathrm{d}}_i(s,v) \leq (1+\epsilon)\,\mathrm{d}^h(s,v)$, wenn $\mathrm{d}^h(s,v) \geq 2^i$

Sei π kürzester Weg s nach v mit höchstens h Kanten für Gewichte $w(\cdot,\cdot)$

$$\begin{aligned} \tilde{\mathrm{d}}_i(s,v) \leq \tilde{w}_i(\pi) &= \sum_{(x,y)\in\pi} \tilde{w}_i(x,y) \\ &= \sum_{(x,y)\in\pi} \left\lceil \frac{w(x,y)}{\rho_i} \right\rceil \cdot \rho_i \leq \sum_{(x,y)\in\pi} (w(x,y) + \rho_i) \\ &= w(\pi) + |\pi| \cdot \rho_i = \mathrm{d}^h(s,v) + |\pi| \cdot \rho_i \\ &\leq \mathrm{d}^h(s,v) + h \cdot \rho_i = \mathrm{d}^h(s,v) + \epsilon 2^i \\ &\leq \mathrm{d}^h(s,v) + \epsilon\,\mathrm{d}^h(s,v) = (1+\epsilon)\,\mathrm{d}^h(s,v) \end{aligned}$$



Beweis der Eigenschaften III

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$, wenn $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$

Beweis der Eigenschaften III

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$, wenn $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$

$$\mathrm{d}_i^{\downarrow}(s,v) = \frac{\tilde{\mathrm{d}}_i(s,v)}{\rho_i}$$

Beweis der Eigenschaften III

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v) / \rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Zu zeigen: $\mathrm{d}_i^{\downarrow}(s, v) \leq \frac{4}{\epsilon} h$, wenn $2^i \leq \mathrm{d}^h(s, v) \leq 2^{i+1}$

$$\begin{aligned} \mathrm{d}_i^{\downarrow}(s, v) &= \frac{\tilde{\mathrm{d}}_i(s, v)}{\rho_i} \\ &\leq \frac{(1 + \epsilon)\, \mathrm{d}^h(u, v)}{\rho_i} \end{aligned}$$

Beweis der Eigenschaften III

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$, wenn $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$

$$\begin{aligned}
\mathrm{d}_i^{\downarrow}(s,v) &= \frac{\tilde{\mathrm{d}}_i(s,v)}{\rho_i} \\
&\leq \frac{(1+\epsilon)\,\mathrm{d}^h(u,v)}{\rho_i} \\
&= \frac{(1+\epsilon) h\, \mathrm{d}^h(u,v)}{\epsilon 2^i}
\end{aligned}$$

Beweis der Eigenschaften III

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u,v) = \lceil w(u,v)/\rho_i \rceil$$
$$\tilde{w}_i(u,v) = w_i^{\downarrow}(u,v) \cdot \rho_i$$

Zu zeigen: $\mathrm{d}_i^{\downarrow}(s,v) \leq \frac{4}{\epsilon} h$, wenn $2^i \leq \mathrm{d}^h(s,v) \leq 2^{i+1}$

$$\begin{aligned}
\mathrm{d}_i^{\downarrow}(s,v) &= \frac{\tilde{\mathrm{d}}_i(s,v)}{\rho_i} \\
&\leq \frac{(1+\epsilon)\,\mathrm{d}^h(u,v)}{\rho_i} \\
&= \frac{(1+\epsilon)h\,\mathrm{d}^h(u,v)}{\epsilon 2^i} \\
&\leq \frac{(1+\epsilon)h 2^{i+1}}{\epsilon 2^i}
\end{aligned}$$

Beweis der Eigenschaften III

$$\rho_i = \epsilon 2^i / h$$
$$w_i^{\downarrow}(u, v) = \lceil w(u, v)/\rho_i \rceil$$
$$\tilde{w}_i(u, v) = w_i^{\downarrow}(u, v) \cdot \rho_i$$

Zu zeigen: $\mathrm{d}_i^{\downarrow}(s, v) \leq \frac{4}{\epsilon} h$, wenn $2^i \leq \mathrm{d}^h(s, v) \leq 2^{i+1}$

$$\begin{aligned}
\mathrm{d}_i^{\downarrow}(s, v) &= \frac{\tilde{\mathrm{d}}_i(s, v)}{\rho_i} \\
&\leq \frac{(1+\epsilon)\,\mathrm{d}^h(u, v)}{\rho_i} \\
&= \frac{(1+\epsilon) h\, \mathrm{d}^h(u, v)}{\epsilon 2^i} \\
&\leq \frac{(1+\epsilon) h 2^{i+1}}{\epsilon 2^i} \\
&\leq \frac{4h}{\epsilon}
\end{aligned}$$

Zusammenfassung

Algorithmus:

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu

2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x, v)$, für die gilt: $\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1 + \epsilon)\, \mathrm{d}^h(x, v)$

3. Mache $\tilde{\mathrm{d}}(x, y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcast bekannt

4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $\tilde{\mathrm{d}}(x, y)$ für jede Kante (x, y)

5. Intern für jeden Knoten v: Berechne $\delta(s, v) = \mathrm{d}_{H_v}(s, v)$ als Ergebnis

Zusammenfassung

Algorithmus:

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu
   Probabilistisches Argument zur Bestimmung von Zentren ohne Kommunikationsoverhead
2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x, v)$, für die gilt: $\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1+\epsilon)\,\mathrm{d}^h(x, v)$
3. Mache $\tilde{\mathrm{d}}(x, y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcast bekannt
4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $\tilde{\mathrm{d}}(x, y)$ für jede Kante (x, y)
5. Intern für jeden Knoten v: Berechne $\delta(s, v) = \mathrm{d}_{H_v}(s, v)$ als Ergebnis

Zusammenfassung

Algorithmus:

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu
   Probabilistisches Argument zur Bestimmung von Zentren ohne Kommunikationsoverhead
2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x, v)$, für die gilt: $\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1+\epsilon)\,\mathrm{d}^h(x, v)$
   Runden der Gewichte, gewichtete Breitensuche, parallele Ausführung mit geringer Bandbreite durch Random Delay
3. Mache $\tilde{\mathrm{d}}(x, y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcast bekannt
4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $\tilde{\mathrm{d}}(x, y)$ für jede Kante (x, y)
5. Intern für jeden Knoten v: Berechne $\delta(s, v) = \mathrm{d}_{H_v}(s, v)$ als Ergebnis

Zusammenfassung

Algorithmus:

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c + 2) \ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu
   Probabilistisches Argument zur Bestimmung von Zentren ohne Kommunikationsoverhead
2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x, v)$, für die gilt: $\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1 + \epsilon)\, \mathrm{d}^h(x, v)$
   Runden der Gewichte, gewichtete Breitensuche, parallele Ausführung mit geringer Bandbreite durch Random Delay
3. Mache $\tilde{\mathrm{d}}(x, y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcast bekannt
   Queuing und Pipelining durch globalen Breitensuchbaum
4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $\tilde{\mathrm{d}}(x, y)$ für jede Kante (x, y)
5. Intern für jeden Knoten v: Berechne $\delta(s, v) = \mathrm{d}_{H_v}(s, v)$ als Ergebnis

Zusammenfassung

Algorithmus:

1. Intern für jeden Knoten v: Füge v mit Wahrscheinlichkeit $p = ((c+2)\ln n)/h$ zu Z hinzu, füge s immer zu Z hinzu
   Probabilistisches Argument zur Bestimmung von Zentren ohne Kommunikationsoverhead
2. Berechne, für alle Paare $x \in Z$, $v \in V$, approximative Distanzen $\tilde{\mathrm{d}}(x, v)$, für die gilt: $\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1+\epsilon)\,\mathrm{d}^h(x, v)$
   Runden der Gewichte, gewichtete Breitensuche, parallele Ausführung mit geringer Bandbreite durch Random Delay
3. Mache $\tilde{\mathrm{d}}(x, y)$ für alle Paare $x, y \in Z$ im gesamten Netzwerk durch Up- und Downcast bekannt
   Queuing und Pipelining durch globalen Breitensuchbaum
4. Intern für jeden Knoten v: Konstruiere Graph $H_v = (Z \cup \{v\}, (Z \cup \{v\})^2)$ mit Gewicht $\tilde{\mathrm{d}}(x, y)$ für jede Kante (x, y)
   Zerstückeln und Zusammenfügen kürzester Wege
5. Intern für jeden Knoten v: Berechne $\delta(s, v) = \mathrm{d}_{H_v}(s, v)$ als Ergebnis

Weiterführende Themen

Schnellere Berechnung von $\mathrm{d}_H(s, v)$:

Weiterführende Themen

Schnellere Berechnung von $\mathrm{d}_H(s, v)$**:**

- Broadcast ist simpelste Lösung

Weiterführende Themen

Schnellere Berechnung von $\mathrm{d}_H(s, v)$**:**

- Broadcast ist simpelste Lösung
- Besser: Simuliere (approximativen) SSSP Algorithmus auf Overlay Netzwerk

Weiterführende Themen

Schnellere Berechnung von $\mathrm{d}_H(s, v)$**:**

- Broadcast ist simpelste Lösung
- Besser: Simuliere (approximativen) SSSP Algorithmus auf Overlay Netzwerk
- Senden von Nachrichten wird über globalen Spannbaum simuliert

Weiterführende Themen

Schnellere Berechnung von $\mathrm{d}_H(s, v)$**:**

- Broadcast ist simpelste Lösung
- Besser: Simuliere (approximativen) SSSP Algorithmus auf Overlay Netzwerk
- Senden von Nachrichten wird über globalen Spannbaum simuliert
- Entspricht Berechnung von SSSP auf einer gewichteten Clique

Weiterführende Themen

Schnellere Berechnung von $\mathrm{d}_H(s, v)$:

- Broadcast ist simpelste Lösung
- Besser: Simuliere (approximativen) SSSP Algorithmus auf Overlay Netzwerk
- Senden von Nachrichten wird über globalen Spannbaum simuliert
- Entspricht Berechnung von SSSP auf einer gewichteten Clique
- Schnellster Algorithmus für Clique: $O(\log^{O(1)}(n)/\epsilon^{O(1)})$ Runden
  $\rightarrow$ Gradientenabstiegsverfahren

Weiterführende Themen

Schnellere Berechnung von $\mathrm{d}_H(s, v)$:

- Broadcast ist simpelste Lösung
- Besser: Simuliere (approximativen) SSSP Algorithmus auf Overlay Netzwerk
- Senden von Nachrichten wird über globalen Spannbaum simuliert
- Entspricht Berechnung von SSSP auf einer gewichteten Clique
- Schnellster Algorithmus für Clique: $O(\log^{O(1)}(n)/\epsilon^{O(1)})$ Runden $\rightarrow$ Gradientenabstiegsverfahren

Exakte Berechnung der Distanz:

Weiterführende Themen

Schnellere Berechnung von $\mathrm{d}_H(s, v)$**:**

- Broadcast ist simpelste Lösung
- Besser: Simuliere (approximativen) SSSP Algorithmus auf Overlay Netzwerk
- Senden von Nachrichten wird über globalen Spannbaum simuliert
- Entspricht Berechnung von SSSP auf einer gewichteten Clique
- Schnellster Algorithmus für Clique: $O(\log^{O(1)}(n)/\epsilon^{O(1)})$ Runden $\rightarrow$ Gradientenabstiegsverfahren

Exakte Berechnung der Distanz:

- Reduziere auf $O(\log(nW))$ *approximative* SSSP-Berechnungen

Weiterführende Themen

Schnellere Berechnung von $\mathrm{d}_H(s, v)$:

- Broadcast ist simpelste Lösung
- Besser: Simuliere (approximativen) SSSP Algorithmus auf Overlay Netzwerk
- Senden von Nachrichten wird über globalen Spannbaum simuliert
- Entspricht Berechnung von SSSP auf einer gewichteten Clique
- Schnellster Algorithmus für Clique: $O(\log^{O(1)}(n)/\epsilon^{O(1)})$ Runden $\longrightarrow$ Gradientenabstiegsverfahren

Exakte Berechnung der Distanz:

- Reduziere auf $O(\log(nW))$ *approximative* SSSP-Berechnungen
- Aber: Reduktion nur möglich, wenn approximative Distanzen eine Metrik bilden

Weiterführende Themen

Schnellere Berechnung von $\mathrm{d}_H(s, v)$**:**

- Broadcast ist simpelste Lösung
- Besser: Simuliere (approximativen) SSSP Algorithmus auf Overlay Netzwerk
- Senden von Nachrichten wird über globalen Spannbaum simuliert
- Entspricht Berechnung von SSSP auf einer gewichteten Clique
- Schnellster Algorithmus für Clique: $O(\log^{O(1)}(n)/\epsilon^{O(1)})$ Runden $\rightarrow$ Gradientenabstiegsverfahren

Exakte Berechnung der Distanz:

- Reduziere auf $O(\log(nW))$ *approximative* SSSP-Berechnungen
- Aber: Reduktion nur möglich, wenn approximative Distanzen eine Metrik bilden
- Zusätzlicher Aufwand beim Design des Algorithmus

Quellen

Literatur:

- Ruben Becker, Andreas Karrenbauer, Sebastian Krinninger, Christoph Lenzen. „Near-Optimal Approximate Shortest Paths and Transshipment in Distributed and Streaming Models“. In: *Proc. of the International Symposium on Distributed Computing (DISC)*. 2017, S. 7:1–7:16
- Michael Elkin. „An Unconditional Lower Bound on the Time-Approximation Trade-off for the Distributed Minimum Spanning Tree Problem“. *SIAM Journal on Computing* 36(2): 433–456 (2006)
- Sebastian Forster, Danupon Nanongkai. „A Faster Distributed Single-Source Shortest Paths Algorithm“. In: *Proc. of the Symposium on Foundations of Computer Science (FOCS)*. 2018, S. 686–697
- David Peleg, Vitaly Rubinovich. „A Near-Tight Lower Bound on the Time Complexity of Distributed Minimum-Weight Spanning Tree Construction“. *SIAM Journal on Computing* 30(5): 1427–1442 (2006)