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) $O(\log n)$
- Hier: $W = n^{O(1)}$, 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 ([Chechik/Mukhtar '20, Cao/Fineman '23, Rozhoň et al. 23])

Das SSSP-Problem kann in $O((\sqrt{n}D^{1/4} + D) \cdot \log^{O(1)} n)$ bzw. in $O(\sqrt{n} + D + n^{2/5+o(1)} D^{2/5})$ 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 ([Chechik/Mukhtar '20, Cao/Fineman '23, Rozhoň et al. 23])

Das SSSP-Problem kann in $O((\sqrt{n}D^{1/4} + D) \cdot \log^{O(1)} n)$ bzw. in $O(\sqrt{n} + D + n^{2/5+o(1)} D^{2/5})$ 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:

$$\mathrm{d}(s, v) \leq \delta(s, v) \leq (1 + \epsilon)\,\mathrm{d}(s, v), \text{für } 0 < \epsilon < 1.$$

Approximationsalgorithmen

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

$$\mathrm{d}(s, v) \leq \delta(s, v) \leq (1+\epsilon)\,\mathrm{d}(s, v), \text{ für } 0 < \epsilon < 1.$$

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:

$$\mathrm{d}(s, v) \leq \delta(s, v) \leq (1 + \epsilon)\,\mathrm{d}(s, v), \text{ für } 0 < \epsilon < 1.$$

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:

$$\mathrm{d}(s,v) \leq \delta(s,v) \leq (1+\epsilon)\,\mathrm{d}(s,v), \text{ für } 0 < \epsilon < 1.$$

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 weniger als h Kanten hat oder innerhalb der ersten h vom Startknoten verschiedenen 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 weniger als h Kanten hat oder innerhalb der ersten h vom Startknoten verschiedenen 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:

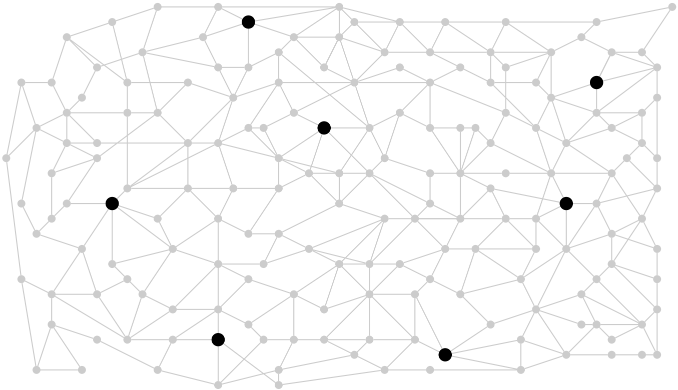
$$\mathrm{d}(x, v) \leq \tilde{\mathrm{d}}(x, v) \leq (1+\epsilon)\,\mathrm{d}^h(x, v)\,.$$

Idee: Reduktion auf Overlay Netzwerk

Bilde Graph mit Zentren als Knoten:

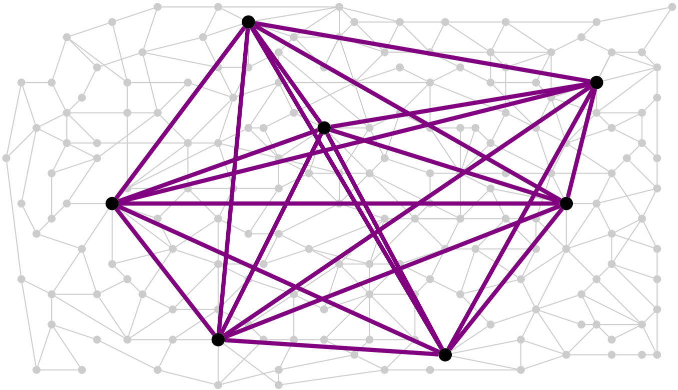
Idee: Reduktion auf Overlay Netzwerk

Bilde Graph mit Zentren als Knoten:



Idee: Reduktion auf Overlay Netzwerk

Bilde Graph mit Zentren als Knoten:



$H = (Z, Z \times Z)$ mit $w_H(x, y) = \tilde{\mathrm{d}}(x, y)$ für alle $x, y \in Z$

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

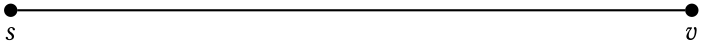
Korrektheit

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 $\leq h$ 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 $\leq h$ Kanten voneinander entfernt
- Somit $\mathrm{d}_G^h(x_i, x_{i+1}) = \mathrm{d}_G(x_i, x_{i+1})$ für alle $0 \leq i \leq k$

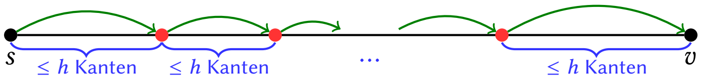
Korrektheit

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 $x_1, x_2, \dots, x_k$ 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 $\leq h$ 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 $x_0, \dots, x_k \in Z$, 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 $x_0, \ldots, x_k \in Z$, 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

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
- $w_{H_v}(x_i, x_{i+1}) = \tilde{\mathrm{d}}(x_i, x_{i+1}) \leq (1+\epsilon)\,\mathrm{d}_G^h(x_i, x_{i+1}) = (1+\epsilon)\,\mathrm{d}_G(x_i, x_{i+1})$

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
- $w_{H_v}(x_i, x_{i+1}) = \tilde{\mathrm{d}}(x_i, x_{i+1}) \leq (1+\epsilon)\,\mathrm{d}^h_G(x_i, x_{i+1}) = (1+\epsilon)\,\mathrm{d}_G(x_i, x_{i+1})$
- 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 weniger als h Kanten hat oder innerhalb der ersten h vom Startknoten verschiedenen Knoten ein Zentrum enthält.

Beweis des Lemmas

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

Beweis des Lemmas

- **Technisches Detail:** Fixiere im Beweis 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 mit mehr als h Knoten

Beweis des Lemmas

- **Technisches Detail:** Fixiere im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für Position des ersten Zentrums auf π_i

Beweis des Lemmas

- **Technisches Detail:** Fixiere im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für Position des ersten Zentrums auf π_i
- X_i is geometrisch verteilt (mit Einzelerfolgswahrscheinlichkeit p):

Beweis des Lemmas

- **Technisches Detail:** Fixiere im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für 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 im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für 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 im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für 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 im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für 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 im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für 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 im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für 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 $x \geq 1$

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

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

Beweis des Lemmas

- **Technisches Detail:** Fixiere im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für 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 $x \geq 1$

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

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

Beweis des Lemmas

- **Technisches Detail:** Fixiere im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für 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 $x \geq 1$

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

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

Beweis des Lemmas

- **Technisches Detail:** Fixiere im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für 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 $x \geq 1$

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

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

Beweis des Lemmas

- **Technisches Detail:** Fixiere im Beweis 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 mit mehr als h Knoten
- Sei $X_i \geq 1$ Zufallsvariable für 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 $x \geq 1$

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

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

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 $\log W$)

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:

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

Modifikation der Breitensuche

Lemma

Seien $M > 0$, $T \geq 0$ und sei G ein Graph mit Kantengewichten $\geq M$. Dann können von einem Startknoten s aus alle Knoten v mit $\mathrm{d}_G(s, v) \leq T$ sowie deren Distanz $\mathrm{d}_G(s, v)$ in $O(T/M)$ Runden berechnet werden, wobei jeder Knoten höchstens in einer Runde Nachrichten sendet.

Modifikation der Breitensuche

Lemma

Seien $M > 0$, $T \geq 0$ und sei G ein Graph mit Kantengewichten $\geq M$. Dann können von einem Startknoten s aus alle Knoten v mit $\mathrm{d}_G(s, v) \leq T$ sowie deren Distanz $\mathrm{d}_G(s, v)$ in $O(T/M)$ Runden berechnet werden, wobei jeder Knoten höchstens in einer Runde Nachrichten sendet.

Idee: Unterteile Distanzen in Intervalle der Form $[i \cdot M, (i + 1) \cdot M)$

Aufrunden der Gewichte

Für jedes $i \geq 0$, betrachte Graph $G_i^{\uparrow}$: Graph mit aufgerundeten Gewichten

$$w_{G_i^{\uparrow}}(u, v) := \begin{cases} M_i & \text{falls } w_G(u, v) < M_i \\ w(u, v) & \text{andernfalls} \end{cases}$$

$$M_i := \frac{\epsilon 2^i}{h}$$

$$T_i := (1 + \epsilon) \cdot 2^{i+1}$$

Aufrunden der Gewichte

Für jedes $i \geq 0$, betrachte Graph $G_i^{\uparrow}$: Graph mit aufgerundeten Gewichten

$$w_{G_i^{\uparrow}}(u,v) := \begin{cases} M_i & \text{falls } w_G(u,v) < M_i \\ w(u,v) & \text{andernfalls} \end{cases}$$

$$M_i := \frac{\epsilon 2^i}{h}$$

$$T_i := (1+\epsilon) \cdot 2^{i+1}$$

Eigenschaften: Für alle Paare von Knoten $x, y \in V$ gilt:

1. $\mathrm{d}_{G_i^{\uparrow}}(x,y) \geq \mathrm{d}_G(x,y)$
2. Falls $\mathrm{d}_G^h(x,y) \geq 2^i$: $\mathrm{d}_{G_i^{\uparrow}}(x,y) \leq (1+\epsilon)\,\mathrm{d}_G^h(x,y)$
3. Falls $2^i \leq \mathrm{d}_G^h(x,y) \leq 2^{i+1}$: $\mathrm{d}_{G_i^{\uparrow}}(x,y) \leq T_i$

Aufrunden der Gewichte

Für jedes $i \geq 0$, betrachte Graph $G_i^{\uparrow}$: Graph mit aufgerundeten Gewichten

$$w_{G_i^{\uparrow}}(u,v) := \begin{cases} M_i & \text{falls } w_G(u,v) < M_i \\ w(u,v) & \text{andernfalls} \end{cases}$$

$$M_i := \frac{\epsilon 2^i}{h}$$

$$T_i := (1+\epsilon) \cdot 2^{i+1}$$

Eigenschaften: Für alle Paare von Knoten $x, y \in V$ gilt:

1. $\mathrm{d}_{G_i^{\uparrow}}(x,y) \geq \mathrm{d}_G(x,y)$
2. Falls $\mathrm{d}_G^h(x,y) \geq 2^i$: $\mathrm{d}_{G_i^{\uparrow}}(x,y) \leq (1+\epsilon)\,\mathrm{d}_G^h(x,y)$
3. Falls $2^i \leq \mathrm{d}_G^h(x,y) \leq 2^{i+1}$: $\mathrm{d}_{G_i^{\uparrow}}(x,y) \leq T_i$

(1) ist offensichtlich, (3) folgt aus (2)

Beweis von (2)

Zu zeigen

Für alle Paare von Knoten $x, y \in V$ gilt: Falls $\mathrm{d}_G^h(x, y) \geq 2^i$, dann ist $\mathrm{d}_{G_i^{\uparrow}}(x, y) \leq (1 + \epsilon)\, \mathrm{d}_G^h(x, y)$

Beweis von (2)

Zu zeigen

Für alle Paare von Knoten $x, y \in V$ gilt: Falls $\mathrm{d}_G^h(x, y) \geq 2^i$, dann ist $\mathrm{d}_{G_i^{\uparrow}}(x, y) \leq (1 + \epsilon)\,\mathrm{d}_G^h(x, y)$

Sei π kürzester Weg von x nach y mit höchstens h Kanten in G

Betrachte Gewicht von π in $G_i^{\uparrow}$ im Vergleich zu G

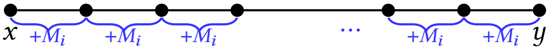
Beweis von (2)

Zu zeigen

Für alle Paare von Knoten $x, y \in V$ gilt: Falls $\mathrm{d}_G^h(x, y) \geq 2^i$, dann ist $\mathrm{d}_{G_i^\uparrow}(x, y) \leq (1 + \epsilon)\, \mathrm{d}_G^h(x, y)$

Sei π kürzester Weg von x nach y mit höchstens h Kanten in G

Betrachte Gewicht von π in $G_i^\uparrow$ im Vergleich zu G



Beobachtung: Jedes Kantengewicht ist in $G_i^\uparrow$ höchstens um M_i höher

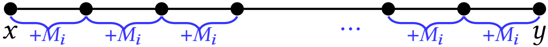
Beweis von (2)

Zu zeigen

Für alle Paare von Knoten $x, y \in V$ gilt: Falls $\mathrm{d}_G^h(x, y) \geq 2^i$, dann ist $\mathrm{d}_{G_i^\uparrow}(x, y) \leq (1 + \epsilon)\, \mathrm{d}_G^h(x, y)$

Sei π kürzester Weg von x nach y mit höchstens h Kanten in G

Betrachte Gewicht von π in $G_i^\uparrow$ im Vergleich zu G



Beobachtung: Jedes Kantengewicht ist in $G_i^\uparrow$ höchstens um M_i höher

Somit:

$$\mathrm{d}_{G_i^\uparrow}(x, y) \leq w_{G_i^\uparrow}(\pi)$$

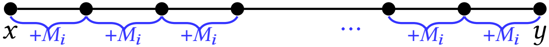
Beweis von (2)

Zu zeigen

Für alle Paare von Knoten $x, y \in V$ gilt: Falls $\mathrm{d}_G^h(x,y) \geq 2^i$, dann ist $\mathrm{d}_{G_i^\uparrow}(x,y) \leq (1+\epsilon)\,\mathrm{d}_G^h(x,y)$

Sei π kürzester Weg von x nach y mit höchstens h Kanten in G

Betrachte Gewicht von π in $G_i^\uparrow$ im Vergleich zu G



Beobachtung: Jedes Kantengewicht ist in $G_i^\uparrow$ höchstens um M_i höher

Somit:

$$\mathrm{d}_{G_i^\uparrow}(x,y) \leq w_{G_i^\uparrow}(\pi) \leq w_G(\pi) + h \cdot M_i$$

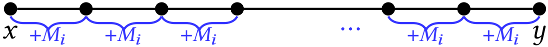
Beweis von (2)

Zu zeigen

Für alle Paare von Knoten $x, y \in V$ gilt: Falls $\mathrm{d}_G^h(x, y) \geq 2^i$, dann ist $\mathrm{d}_{G_i^\uparrow}(x, y) \leq (1 + \epsilon)\, \mathrm{d}_G^h(x, y)$

Sei π kürzester Weg von x nach y mit höchstens h Kanten in G

Betrachte Gewicht von π in $G_i^\uparrow$ im Vergleich zu G



Beobachtung: Jedes Kantengewicht ist in $G_i^\uparrow$ höchstens um M_i höher

Somit:

$$\begin{aligned}\mathrm{d}_{G_i^\uparrow}(x, y) \leq w_{G_i^\uparrow}(\pi) &\leq w_G(\pi) + h \cdot M_i \\ &= w_G(\pi) + \epsilon 2^i\end{aligned}$$

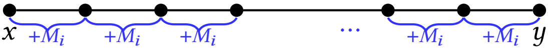
Beweis von (2)

Zu zeigen

Für alle Paare von Knoten $x, y \in V$ gilt: Falls $\mathrm{d}^h_G(x, y) \geq 2^i$, dann ist $\mathrm{d}_{G_i^\uparrow}(x, y) \leq (1 + \epsilon)\,\mathrm{d}^h_G(x, y)$

Sei π kürzester Weg von x nach y mit höchstens h Kanten in G

Betrachte Gewicht von π in $G_i^\uparrow$ im Vergleich zu G



Beobachtung: Jedes Kantengewicht ist in $G_i^\uparrow$ höchstens um M_i höher

Somit:

$$\begin{aligned} \mathrm{d}_{G_i^\uparrow}(x, y) \leq w_{G_i^\uparrow}(\pi) &\leq w_G(\pi) + h \cdot M_i \\ &= w_G(\pi) + \epsilon 2^i \\ &= \mathrm{d}^h_G(x, y) + \epsilon 2^i \end{aligned}$$

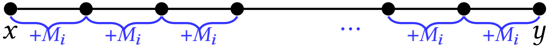
Beweis von (2)

Zu zeigen

Für alle Paare von Knoten $x, y \in V$ gilt: Falls $\mathrm{d}^h_G(x,y) \geq 2^i$, dann ist $\mathrm{d}_{G_i^\uparrow}(x,y) \leq (1+\epsilon)\,\mathrm{d}^h_G(x,y)$

Sei π kürzester Weg von x nach y mit höchstens h Kanten in G

Betrachte Gewicht von π in $G_i^\uparrow$ im Vergleich zu G



Beobachtung: Jedes Kantengewicht ist in $G_i^\uparrow$ höchstens um M_i höher

Somit:

$$\begin{aligned}
\mathrm{d}_{G_i^\uparrow}(x,y) \leq w_{G_i^\uparrow}(\pi) &\leq w_G(\pi) + h \cdot M_i \\
&= w_G(\pi) + \epsilon 2^i \\
&= \mathrm{d}^h_G(x,y) + \epsilon 2^i \\
&\leq \mathrm{d}^h_G(x,y) + \epsilon\,\mathrm{d}^h_G(x,y)
\end{aligned}$$

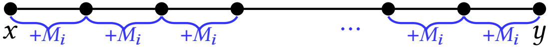
Beweis von (2)

Zu zeigen

Für alle Paare von Knoten $x, y \in V$ gilt: Falls $\mathrm{d}_G^h(x,y) \geq 2^i$, dann ist $\mathrm{d}_{G_i^\uparrow}(x,y) \leq (1+\epsilon)\,\mathrm{d}_G^h(x,y)$

Sei π kürzester Weg von x nach y mit höchstens h Kanten in G

Betrachte Gewicht von π in $G_i^\uparrow$ im Vergleich zu G



Beobachtung: Jedes Kantengewicht ist in $G_i^\uparrow$ höchstens um M_i höher

Somit:

$$
\begin{aligned}
\mathrm{d}_{G_i^\uparrow}(x,y) \leq w_{G_i^\uparrow}(\pi) &\leq w_G(\pi) + h \cdot M_i \\
&= w_G(\pi) + \epsilon 2^i \\
&= \mathrm{d}_G^h(x,y) + \epsilon 2^i \\
&\leq \mathrm{d}_G^h(x,y) + \epsilon\,\mathrm{d}_G^h(x,y) = (1+\epsilon)\,\mathrm{d}_G^h(x,y)
\end{aligned}
$$

Aufrunden der Gewichte II

Eigenschaften: Für alle Paare von Knoten $x, y \in V$ gilt:

1. $\mathrm{d}_{G_i^{\uparrow}}(x, y) \geq \mathrm{d}_G(x, y)$
2. Falls $\mathrm{d}_G^h(x, y) \geq 2^i$: $\mathrm{d}_{G_i^{\uparrow}}(x, y) \leq (1 + \epsilon)\, \mathrm{d}_G^h(x, y)$
3. Falls $2^i \leq \mathrm{d}_G^h(x, y) \leq 2^{i+1}$: $\mathrm{d}_{G_i^{\uparrow}}(x, y) \leq T_i$

Aufrunden der Gewichte II

Eigenschaften: Für alle Paare von Knoten $x, y \in V$ gilt:

1. $\mathrm{d}_{G_i^\uparrow}(x, y) \geq \mathrm{d}_G(x, y)$
2. Falls $\mathrm{d}_G^h(x, y) \geq 2^i$: $\mathrm{d}_{G_i^\uparrow}(x, y) \leq (1 + \epsilon)\, \mathrm{d}_G^h(x, y)$
3. Falls $2^i \leq \mathrm{d}_G^h(x, y) \leq 2^{i+1}$: $\mathrm{d}_{G_i^\uparrow}(x, y) \leq T_i$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i < \lfloor \log(hW) \rfloor$: Berechne $\tilde{\mathrm{d}}_i(s, v) := \mathrm{d}_{G_i^\uparrow}(s, v)$ für alle v mit $\mathrm{d}_{G_i^\uparrow}(s, v) \leq T_i$ (und $\tilde{\mathrm{d}}_i(s, v) := \infty$ sonst)

Aufrunden der Gewichte II

Eigenschaften: Für alle Paare von Knoten $x, y \in V$ gilt:

1. $\mathrm{d}_{G_i^{\uparrow}}(x,y) \geq \mathrm{d}_G(x,y)$
2. Falls $\mathrm{d}_G^h(x,y) \geq 2^i$: $\mathrm{d}_{G_i^{\uparrow}}(x,y) \leq (1+\epsilon)\,\mathrm{d}_G^h(x,y)$
3. Falls $2^i \leq \mathrm{d}_G^h(x,y) \leq 2^{i+1}$: $\mathrm{d}_{G_i^{\uparrow}}(x,y) \leq T_i$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i < \lfloor \log(hW) \rfloor$: Berechne $\tilde{\mathrm{d}}_i(s,v) := \mathrm{d}_{G_i^{\uparrow}}(s,v)$ für alle v mit $\mathrm{d}_{G_i^{\uparrow}}(s,v) \leq T_i$ (und $\tilde{\mathrm{d}}_i(s,v) := \infty$ sonst) Laufzeit $O(\log(hW) \cdot h/\epsilon)$

Aufrunden der Gewichte II

Eigenschaften: Für alle Paare von Knoten $x, y \in V$ gilt:

1. $\mathrm{d}_{G_i^\uparrow}(x, y) \geq \mathrm{d}_G(x, y)$
2. Falls $\mathrm{d}_G^h(x, y) \geq 2^i$: $\mathrm{d}_{G_i^\uparrow}(x, y) \leq (1 + \epsilon)\, \mathrm{d}_G^h(x, y)$
3. Falls $2^i \leq \mathrm{d}_G^h(x, y) \leq 2^{i+1}$: $\mathrm{d}_{G_i^\uparrow}(x, y) \leq T_i$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i < \lfloor \log (hW) \rfloor$: Berechne $\tilde{\mathrm{d}}_i(s, v) := \mathrm{d}_{G_i^\uparrow}(s, v)$ für alle v mit $\mathrm{d}_{G_i^\uparrow}(s, v) \leq T_i$ (und $\tilde{\mathrm{d}}_i(s, v) := \infty$ sonst) Laufzeit $O(\log (hW) \cdot h/\epsilon)$
2. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}(s, v) := \min_{0 \leq i < \lfloor \log (hW) \rfloor} \tilde{\mathrm{d}}_i(s, v)$

Aufrunden der Gewichte II

Eigenschaften: Für alle Paare von Knoten $x, y \in V$ gilt:

1. $\mathrm{d}_{G_i^{\uparrow}}(x, y) \geq \mathrm{d}_G(x, y)$
2. Falls $\mathrm{d}_G^h(x, y) \geq 2^i$: $\mathrm{d}_{G_i^{\uparrow}}(x, y) \leq (1 + \epsilon)\, \mathrm{d}_G^h(x, y)$
3. Falls $2^i \leq \mathrm{d}_G^h(x, y) \leq 2^{i+1}$: $\mathrm{d}_{G_i^{\uparrow}}(x, y) \leq T_i$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i < \lfloor \log (hW) \rfloor$: Berechne $\tilde{\mathrm{d}}_i(s, v) := \mathrm{d}_{G_i^{\uparrow}}(s, v)$ für alle v mit $\mathrm{d}_{G_i^{\uparrow}}(s, v) \leq T_i$ (und $\tilde{\mathrm{d}}_i(s, v) := \infty$ sonst) Laufzeit $O(\log (hW) \cdot h/\epsilon)$
2. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}(s, v) := \min_{0 \leq i < \lfloor \log (hW) \rfloor} \tilde{\mathrm{d}}_i(s, v)$

Korrektheit:

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

Aufrunden der Gewichte II

Eigenschaften: Für alle Paare von Knoten $x, y \in V$ gilt:

1. $\mathrm{d}_{G_i^\uparrow}(x, y) \geq \mathrm{d}_G(x, y)$
2. Falls $\mathrm{d}_G^h(x, y) \geq 2^i$: $\mathrm{d}_{G_i^\uparrow}(x, y) \leq (1 + \epsilon)\, \mathrm{d}_G^h(x, y)$
3. Falls $2^i \leq \mathrm{d}_G^h(x, y) \leq 2^{i+1}$: $\mathrm{d}_{G_i^\uparrow}(x, y) \leq T_i$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i < \lfloor \log (hW) \rfloor$: Berechne $\tilde{\mathrm{d}}_i(s, v) := \mathrm{d}_{G_i^\uparrow}(s, v)$ für alle v mit $\mathrm{d}_{G_i^\uparrow}(s, v) \leq T_i$ (und $\tilde{\mathrm{d}}_i(s, v) := \infty$ sonst) Laufzeit $O(\log (hW) \cdot h/\epsilon)$
2. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}(s, v) := \min_{0 \leq i < \lfloor \log (hW) \rfloor} \tilde{\mathrm{d}}_i(s, v)$

Korrektheit:

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

Aufrunden der Gewichte II

Eigenschaften: Für alle Paare von Knoten $x, y \in V$ gilt:

1. $\mathrm{d}_{G_i^{\uparrow}}(x,y) \geq \mathrm{d}_G(x,y)$
2. Falls $\mathrm{d}_G^h(x,y) \geq 2^i$: $\mathrm{d}_{G_i^{\uparrow}}(x,y) \leq (1+\epsilon)\,\mathrm{d}_G^h(x,y)$
3. Falls $2^i \leq \mathrm{d}_G^h(x,y) \leq 2^{i+1}$: $\mathrm{d}_{G_i^{\uparrow}}(x,y) \leq T_i$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i < \lfloor \log (hW) \rfloor$: Berechne $\tilde{\mathrm{d}}_i(s,v) := \mathrm{d}_{G_i^{\uparrow}}(s,v)$ für alle v mit $\mathrm{d}_{G_i^{\uparrow}}(s,v) \leq T_i$ (und $\tilde{\mathrm{d}}_i(s,v) := \infty$ sonst) Laufzeit $O(\log (hW) \cdot h/\epsilon)$
2. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}(s,v) := \min_{0 \leq i < \lfloor \log (hW) \rfloor} \tilde{\mathrm{d}}_i(s,v)$

Korrektheit:

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

Aufrunden der Gewichte II

Eigenschaften: Für alle Paare von Knoten $x, y \in V$ gilt:

1. $\mathrm{d}_{G_i^{\uparrow}}(x, y) \geq \mathrm{d}_G(x, y)$
2. Falls $\mathrm{d}_G^h(x, y) \geq 2^i$: $\mathrm{d}_{G_i^{\uparrow}}(x, y) \leq (1 + \epsilon)\, \mathrm{d}_G^h(x, y)$
3. Falls $2^i \leq \mathrm{d}_G^h(x, y) \leq 2^{i+1}$: $\mathrm{d}_{G_i^{\uparrow}}(x, y) \leq T_i$

Algorithmus (für fixes $s \in Z$):

1. Für jedes $0 \leq i < \lfloor \log(hW) \rfloor$: Berechne $\tilde{\mathrm{d}}_i(s, v) := \mathrm{d}_{G_i^{\uparrow}}(s, v)$ für alle v mit $\mathrm{d}_{G_i^{\uparrow}}(s, v) \leq T_i$ (und $\tilde{\mathrm{d}}_i(s, v) := \infty$ sonst) Laufzeit $O(\log(hW) \cdot h/\epsilon)$
2. Intern, für jeden Knoten v: Berechne $\tilde{\mathrm{d}}(s, v) := \min_{0 \leq i < \lfloor \log(hW) \rfloor} \tilde{\mathrm{d}}_i(s, v)$

Korrektheit:

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

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, Modifikation der 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, Modifikation der 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, Modifikation der 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
  $\longrightarrow$ 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 $\longrightarrow$ 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
  $\longrightarrow$ 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 $\longrightarrow$ 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“. *SIAM Journal on Computing* 50(3): 815–856 (2021)
- Shiri Chechik, Doron Mukhtar. „Single-Source Shortest Paths in the CONGEST Model with Improved Bound. In: *Proc. of the Symposium on Principles of Distributed Computing (PODC).* 2020, S. 464–473
- 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)