

**Universität Stuttgart  
Fakultät Informatik**



Institut für Informatik  
Breitwiesenstraße 20-22  
D-70565 Stuttgart

**Hierarchische Graphen  
zur kürzesten Wegesuche  
in planaren Graphen**

Friedhelm Buchholz und Bernhard Riedhofer

Report Nr. 1997/13

August 1997

## Zusammenfassung

Für ungerichtete Graphen werden hierarchische und regionenhierarchische Graphen definiert. In planaren hierarchischen Graphen lässt sich die kürzeste Entfernung zwischen zwei Knoten in der Laufzeit  $O(\sqrt{n})$  ermitteln ( $n$  sei die Anzahl der Knoten in dem gegebenen Graphen). Der kürzeste Weg mit  $l$  Knoten kann in  $O(\sqrt{n} + l)$  Operationen berechnet werden. Beide Typen von hierarchischen Graphen benötigen dabei  $O(n^{1.5})$  Platz für die Speicherung der Hilfsstabellen. Während für den Aufbau von allgemeinen hierarchischen Graphen  $O(n^2)$  Operationen benötigt werden, können regionenhierarchische Graphen in  $O(n^{1.5})$  Operationen aufgebaut werden. Zur Konstruktion der hierarchischen Graphen benutzen wir das Separatortheorem von Lipton und Tarjan [11].

## 1 Einführung

Die kürzeste Wegesuche ist ein ausführlich untersuchtes Problem in der Graphentheorie, das allerdings insbesondere in den letzten Jahren erneut bzgl. anderer Nebenbedingungen bearbeitet wird. Wenn man keine Preprocessing-Phase erlaubt, lässt sich die kürzeste Entfernung in einem planaren Graphen mit  $n$  Knoten in  $O(n)$  Zeit finden [10]. Wenn man in einer Preprocessing-Phase in  $O(n^2)$  Operationen alle kürzesten Entfernungen berechnet und in einer Tabelle zwischenspeichert, kann man durch Zugriff auf die Tabelle in  $O(1)$  Operationen die kürzeste Entfernung bestimmen. Es stellt sich die Frage nach hierarchischen Datenstrukturen, die wenig Platz benötigen und sich in einer Preprocessing-Phase effizient konstruieren lassen, so daß die Entfernungs berechnung ebenfalls effizient möglich ist. Diese Fragestellung ist insbesondere im Bereich von Verkehrs informationssystemen von großem Interesse. In [9],[1],[3],[4],[8],[13] finden sich diverse praktische Ansätze zur Hierarchiebildung, deren Effektivität zum Teil an realen Straßendaten nachgewiesen wurde. Leider wurden diese Verfahren entweder nicht bezüglich der Komplexität untersucht, oder sie berechnen lediglich Näherungen zum Bestweg.

Um Komplexitätsuntersuchungen zu vereinfachen, d.h. um insbesondere die Ergebnisse der Separation von planaren Graphen ausnutzen zu können [11], betrachten wir in dieser Arbeit planare Graphen. Für diese konstruieren wir in einer Preprocessing-Phase in  $O(n^{1.5})$  Zeit auf  $O(n^{1.5})$  Platz einen hierarchischen Graph, der die Berechnung der kürzesten Entfernung zwischen zwei Knoten in  $O(\sqrt{n})$  Zeit zusichert. Zur Berechnung kürzester Wege innerhalb unseres Verfahrens benutzen wir einen linearen Algorithmus, der von Klein u.a. entwickelt wurde [10]. Einzelne Ergebnisse dieser Arbeit sind auch in [12] nachzulesen.

Die Arbeit ist folgendermaßen aufgebaut: Abschnitt 2 enthält grundlegende Definitionen, in Abschnitt 3 werden hierarchische Graphen definiert, und in Abschnitt 4 wird ein Konstruktionsverfahren für hierarchische Graphen vorgestellt. In Abschnitt 5 werden einige vorbereitende Überlegungen über den Verlauf von Wegen gemacht, die in den Abschnitten 6 und 7 beim Aufbau von gewichteten und regionengewichteten hierarchischen Graphen benutzt werden. Hinweise zur Implementierung und Ergebnisse von Experimenten sind in den Abschnitten 8 und 9 zusammengefaßt.

## 2 Definitionen

In dieser Arbeit betrachten wir nur ungerichtete Graphen ohne Mehrfachkanten. Schlingen, d.h. Kanten von einem Knoten zu sich selbst, spielen bei unseren Untersuchungen keine Rolle, so daß wir darauf nicht weiter eingehen. Die Übertragung der Ergebnisse auf gerichtete Graphen bereitet keine Probleme. Ein ungerichteter Graph  $G = (V, E)$  besitzt die Knotenmenge  $V$  und die Kantenmenge  $E \subseteq \{\{u, v\} \mid u \in V, v \in V\}$ . Der Graph heißt **vollständig**, wenn für die Kantenmenge gilt:  $E = \{\{u, v\} \mid u \in V, v \in V\}$ . Für beliebige Knotenmengen  $A, B \subseteq V$  sei:

$$[A, B] := \{\{a, b\} \mid a \in A, b \in B\}$$

In dem Graphen  $G = (V, E)$  **induziert** die Knotenmenge  $A \subseteq V$  den Graph

$$G_A = (A, \{\{u, v\} \in E \mid u, v \in A\})$$

Ein Graph  $G' = (V', E')$  ist ein **Teilgraph** vom Graphen  $G = (V, E)$ , wenn gilt:  $V' \subseteq V$  und  $E' \subseteq E$  (Schreibweise  $G' \subseteq G$ ). Ein Graph  $G = (V, E)$  heißt **planar**, wenn sich der Graph  $G$  so in die Ebene zeichnen läßt, daß sich keine Kanten überkreuzen.

Eine Menge  $\{A_1, A_2, \dots, A_k\}$  heißt eine **Partition** einer Menge  $M$ , wenn gilt:

$$\begin{aligned} A_i &\subseteq M \quad \forall 1 \leq i \leq k \\ M &= \bigcup_{i=1}^k A_i \\ A_i \cap A_j &= \emptyset \quad \forall 1 \leq i < j \leq k \end{aligned}$$

Die Mächtigkeit einer Menge  $M$  bezeichnen wir mit  $|M|$ . Zu einem Graphen  $G = (V, E)$  heißt die Partition  $\{A, B, C\}$  der Knotenmenge  $V$  eine **Separation** mit **trennender Knotenmenge**  $C$ , wenn gilt:  $E \cap [A, B] = \emptyset$ . Es gibt also keine Kante in  $E$  zwischen den Knotenmengen  $A, B$ . Ein Graph  $G = (V, E, \delta)$  ist ein Graph  $G = (V, E)$  mit Kantengewichtsfunktion  $\delta : E \rightarrow \mathbb{R}^+$ . Ein Weg  $w = [u_1, u_2, \dots, u_k]$  im Graphen  $G$  ist eine Knotenfolge  $u_i \in V$  für  $1 \leq i \leq k$ , wobei die Kanten  $\{u_i, u_{i+1}\}$  für  $1 \leq i < k$  in der Kantenmenge  $E$  liegen.  $s(w)$  sei der Startknoten  $u_1$  des Weges  $w$  und  $e(w)$  sei der Endknoten  $u_k$ . Der Weg  $w$  **verläuft vollständig** in einem induzierten Graphen  $G'$  von  $G$ , falls  $w$  ein Weg in  $G$  ist und alle Knoten auf  $w$  in der Knotenmenge von  $G'$  liegen. Die Kantengewichtsfunktion  $\delta$  sei wie folgt auf Wege  $w = [u_1, u_2, \dots, u_k]$  für  $k \geq 2$  (für  $k = 1$  sei  $\delta(w) = 0$ ) und Knotenpaare  $u, v \in V$  fortgesetzt:

$$\begin{aligned} \delta(w) &:= \sum_{i=1}^{k-1} \delta(\{u_i, u_{i+1}\}) \\ \delta(u, v) &:= \min\{\delta(w) \mid s(w) = u, e(w) = v, w \text{ ist ein Weg in } G\} \end{aligned}$$

Falls kein Weg von  $u$  nach  $v$  in  $G$  existiert, setzen wir  $\delta(u, v) := \infty$ .

Die endliche Menge  $\Sigma$  nennen wir auch ein endliches Alphabet. Die Menge aller Wörter über dem Alphabet  $\Sigma$  sei  $\Sigma^* := \bigcup_{i=0}^{\infty} \Sigma^i$ . Dabei enthält  $\Sigma^i$  alle Wörter  $\lambda$  der Länge  $i$  (Schreibweise  $|\lambda| = i$  und  $\Sigma^0 = \{\epsilon\}$ ).  $\Sigma^{i,j} := \{\lambda \in \Sigma^* \mid i \leq |\lambda| \leq j\}$  sind alle Wörter mit einer Länge zwischen  $i$  und  $j$ , wobei  $i \leq j$  gelte.  $(\lambda)i$  sei das **Präfix** der Länge  $0 \leq i \leq |\lambda|$  des Wortes  $\lambda$ .  $\circ$  bezeichne die Konkatenation von Wörtern.

### 3 Hierarchische Graphen

**Definition 1** Gegeben seien ein Graph  $G = (V, E)$  und das Alphabet  $\Sigma = \{1, 2\}$ .

$$HG := \{G_\lambda \mid \lambda \in \Sigma^{0,k}\}$$

heißt **hierarchischer Graph mit  $k$  Stufen** von  $G$ , wenn gilt:

1. Für alle  $\lambda \in \Sigma^{0,k}$  gilt:  $G_\lambda = (R_\lambda, [T_\lambda, R_\lambda])$  ist ein Graph mit  $T_\lambda \subseteq R_\lambda$ .
2.  $R_\epsilon = V$ .
3. Für alle  $\lambda \in \Sigma^{0,k-1}$  und  $x \in \Sigma$  gilt:

$$G_\lambda = (R_\lambda, [T_\lambda, R_\lambda]) \in HG \implies G_{\lambda \circ x} = (R_{\lambda \circ x}, [T_{\lambda \circ x}, R_{\lambda \circ x}]) \in HG$$

und  $R_{\lambda \circ 1}, R_{\lambda \circ 2}, T_\lambda$  ist eine Separation von  $R_\lambda$  mit trennender Knotenmenge  $T_\lambda$  in dem von  $R_\lambda$  induzierten Graphen  $G_{R_\lambda} \subseteq G = (V, E)$ .

4. Für alle  $\lambda \in \Sigma^k$  gilt:  $T_\lambda = R_\lambda$ .

Ein Graph  $G_\lambda \in HG$  heißt **Graph der Stufe  $|\lambda|$** .  $r(v) = \lambda$  heißt **der Regionenname des Knotens  $v \in V$** , falls  $v \in T_\lambda$  gilt.

Der Aufbau eines hierarchischen Graphen  $HG$  aus einem Graphen  $G$  erfolgt sukzessive von Stufe Null bis zur Stufe  $k$ .  $V = R_\epsilon$  ist die Knotenmenge des obersten Graphen  $G_\epsilon$ , d.h. der Graph der Stufe Null aus  $HG$ . Dann wird eine Separation  $R_1, R_2, T_\epsilon$  von  $G$  mit der trennenden Knotenmenge  $T_\epsilon$  festgelegt, wodurch der Graph  $G_\epsilon = (R_\epsilon, [T_\epsilon, R_\epsilon])$  vollständig bestimmt ist. Die Graphen  $G_1 = (R_1, [T_1, R_1])$  und  $G_2 = (R_2, [T_2, R_2])$  werden durch Separation der Knotenmengen  $R_1, R_2$  in  $R_{11}, R_{12}, T_1$  bzw.  $R_{21}, R_{22}, T_2$  bzgl. der durch  $R_1, R_2$  in  $G$  induzierten Graphen bestimmt usw. Alle Graphen auf der Stufe  $k$  eines hierarchischen Graphen sind vollständig, da  $T_\lambda = R_\lambda$  ist. In Abbildung 1 ist ein planarer Graph und eine mögliche Separation in Teilgraphen dargestellt. Man beachte, daß die Graphen  $G_{R_\lambda}$  nicht zum hierarchischen Graphen gehören, sondern die von  $R_\lambda$  in dem gegebenen Graphen induzierten Teilgraphen sind. Die trennenden Mengen auf der Stufe drei sind:  $T_{111} = \{v_1\}, T_{112} = \emptyset, T_{121} = \{v_3\}, T_{122} = \{v_7\}, T_{211} = \{v_{10}\}, T_{212} = \emptyset, T_{221} = \{v_{14}\}, T_{222} = \{v_{16}\}$ . Der Aufbau des zugehörigen hierarchischen Graphen  $HG$  ist in Abbildung 2 skizziert, und die Graphen  $G_\epsilon, G_1 \in HG$  sind exemplarisch dargestellt.

Die Stufe  $0 \leq i \leq k$  eines hierarchischen Graphen mit  $k+1$  Stufen besteht aus  $2^i$  Graphen. Aus Gründen der Einfachheit haben wir nur den speziellen Fall des 'vollständig ausgeglichenen' hierarchischen Graphen definiert. Die Verallgemeinerung ergibt sich durch Entfernen von leeren Graphen.

**Definition 2** Seien  $f : \mathbb{N} \rightarrow \mathbb{R}^+$  eine monoton wachsende Funktion und  $\alpha, \beta \in \mathbb{R}^+$  feste Parameter mit  $\frac{1}{2} \leq \alpha < 1, \beta \geq 0$ . Ein hierarchischer Graph  $HG$  mit  $n$  Knoten in  $G_\epsilon \in HG$  heißt  $\alpha$ - $\beta$ - $f(n)$ - $HG$ , wenn für alle Graphen  $G_\lambda = (R_\lambda, [T_\lambda, R_\lambda]) \in HG$  gilt:

1.  $|R_\lambda| \leq \alpha^{|\lambda|} n$  und
2.  $|T_\lambda| \leq \beta f(|R_\lambda|)$ .

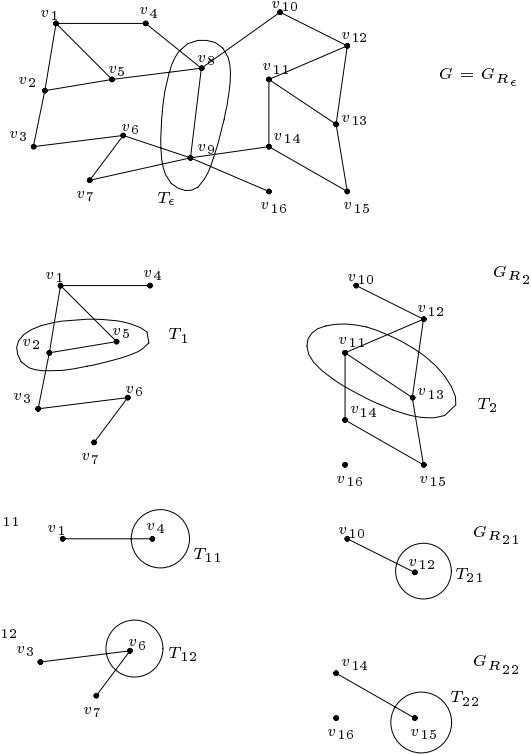


Abbildung 1: Ein planarer Graph mit Separationen in Teilgraphen.

Durch Einsetzen der ersten Bedingung in die zweite folgt:  $|T_\lambda| \leq \beta f(\alpha^{|\lambda|} n)$ . Dies werden wir bei den Komplexitätsuntersuchungen ausnutzen. Man beachte, daß ein hierarchischer  $\alpha$ - $\beta$ - $f(n)$ -HG Graph, der nur nichtleere Graphen enthält, wegen Bedingung (1) nicht mehr als  $O(\log n)$  nicht-leere Stufen besitzt. Für  $f(n)$  werden wir  $\sqrt{n}$  einsetzen, d.h. jede trennende Knotenmenge  $T_\lambda$  enthält in einem  $\alpha$ - $\beta$ - $\sqrt{n}$ -HG höchstens  $\beta\sqrt{\alpha^{|\lambda|} n}$  Knoten.

## 4 Aufbau eines hierarchischen Graphen

Das folgende Theorem wurde in einer allgemeineren Form von Lipton und Tarjan bewiesen [11].

**Theorem 1** *Gegeben sei ein planarer Graph  $G = (V, E)$  mit  $n$  Knoten. Eine Separation  $\{A, B, C\}$  von  $V$  mit trennender Knotenmenge  $C$ , die folgende Bedingungen erfüllt:*

- $|A|, |B| \leq \alpha n$  und
- $|C| \leq \beta\sqrt{n}$

*existiert für die Parameter  $\alpha = \frac{1}{2}, \beta = \frac{2\sqrt{2}}{1-\sqrt{\frac{2}{3}}} \leq 15.5$  stets und kann in  $O(n)$  Operationen berechnet werden.*

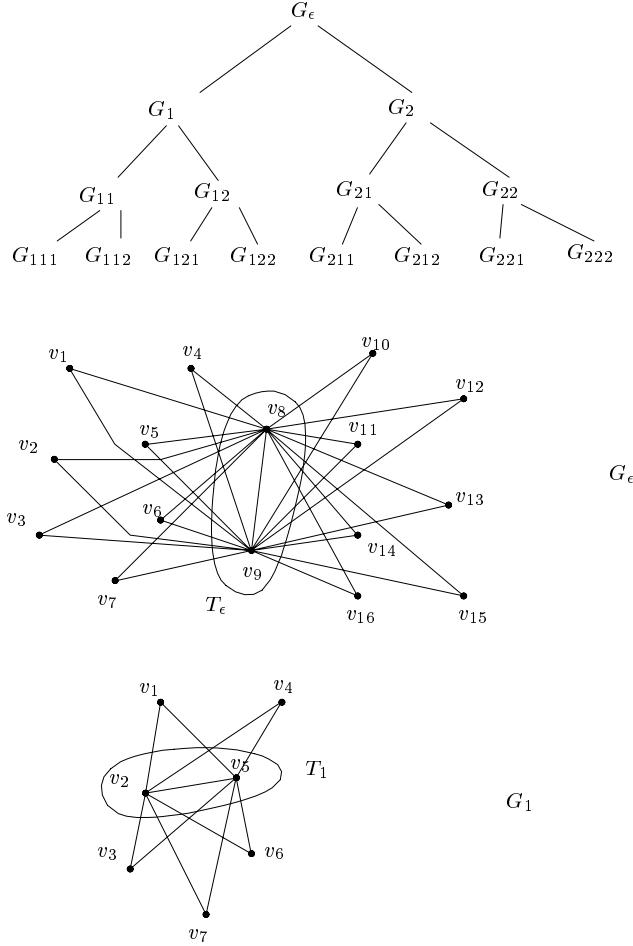


Abbildung 2: Struktur eines hierarchischen Graphen  $HG$  und  $G_\epsilon, G_1 \in HG$ .

Es wurden mehrfache Verbesserungen dieses Separatortheorems für die Parameter  $\alpha$  und  $\beta$  angegeben, die in [5] und [7] nachgelesen werden können. Für die Komplexitätsuntersuchungen, die wir durchführen, sind die genauen Werte irrelevant, so daß wir  $\alpha$  und  $\beta$  als konstant annehmen können.

**Theorem 2** *Gegeben seien obige Parameter  $\alpha$  und  $\beta$  und ein planarer Graph  $G = (V, E)$ . Zu  $G$  läßt sich ein hierarchischer  $\alpha$ - $\beta$ - $\sqrt{n}$ -HG Graph der Stufe  $O(\log n)$  in  $O(n^{1.5})$  Operationen konstruieren.*

**Beweis:** Der hierarchische Graph habe  $k + 1$  Stufen. Er wird sukzessive von Stufe Null bis zur Stufe  $k$  durch Anwendung des Separatortheorems 1 aufgebaut. Da die Separierung nur linearen Aufwand in der Anzahl der Knoten benötigt und in jeder Stufe weniger als  $n$  verschiedene Knoten existieren, ist insgesamt höchstens mit  $O(n \log n)$  Operationen für die Separierung zu rechnen. Da die Knotenmengen  $T_\lambda$  für alle  $\lambda \in \Sigma^{0,k}$  paarweise diskunkt sind, ist zum Aufbau der Kantenmengen  $[T_\lambda, R_\lambda]$  in jedem Graph  $G_\lambda \in HG$

folgender Aufwand notwendig:

Ein hierarchischer  $\alpha\text{-}\beta\text{-}\sqrt{n}\text{-HG}$  Graph zu einem planaren Graph benötigt nach Theorem 2 höchstens den Speicheraufwand  $O(n^{1.5})$ . In dem Beispiel ist ein hierarchischer  $\frac{1}{2}\text{-}1\text{-}\sqrt{n}\text{-HG}$  Graph dargestellt (siehe Abbildung 2).

## 5 Vorbereitung zur Wegsuche im hierarchischen Graphen

**Definition 3** Sei  $HG$  ein hierarchischer Graph zum Graphen  $G = (V, E)$ . Für die Knoten  $u, v \in V$  heißt  $LLG(u, v) = \lambda$  **letzter gemeinsamer Graphname** genau dann, wenn gilt:  $\lambda$  ist das längste gemeinsame Präfix von  $r(u)$  und  $r(v)$ . Die Knotenmenge  $\text{Rand}(u, v) = \bigcup_{i=0}^{|LLG(u, v)|} T_{(LLG(u, v))i}$  heißt **Randknotenmenge** von  $u, v \in V$ .

**Lemma 1** Gegeben sei ein hierarchischer Graph  $HG$  zu einem Graph  $G$ . In jedem Weg  $w$ , der vollständig in einem Teilgraphen  $G_{R_\lambda}$  für  $G_\lambda = (R_\lambda, [T_\lambda, R_\lambda]) \in HG$  verläuft, aber für alle  $x \in \Sigma = \{1, 2\}$  nicht vollständig in  $G_{R_{\lambda \circ x}}$  für  $G_{\lambda \circ x} \in HG$  verläuft, gibt es einen Knoten  $v \in T_\lambda$ .

**Beweis:** Da  $R_{\lambda \circ 1}, R_{\lambda \circ 2}, T_\lambda$  eine Separation mit trennender Knotenmenge  $T_\lambda$  in  $G_{R_\lambda}$  ist, folgt die Aussage.

**Lemma 2** Gegeben sei ein hierarchischer Graph  $HG$  zu dem Graphen  $G$ . Ein Weg  $w$ , der vollständig in  $G_{R_\lambda}$  für  $G_\lambda \in HG$  verläuft, verläuft auch vollständig in  $G_{R_{(\lambda)_i}}$  für jedes  $0 \leq i \leq |\lambda|$ .

**Beweis:** Die Aussage folgt aus  $G_{R_{(\lambda)}j} \subseteq G_{R_{(\lambda)}i}$  für  $0 \leq i \leq j \leq |\lambda|$ .

**Theorem 3** Sei  $HG$  ein hierarchischer Graph zum Graphen  $G = (V, E)$ . Für jeden Weg  $w$  von  $u$  nach  $v$  in  $G$  gibt es einen Knoten  $z \in \text{Rand}(u, v)$ , der auf dem Weg  $w$  liegt.

**Beweis:** Sei  $LGG(u, v) = \lambda$ . Für den Fall  $u \in T_\lambda$  oder  $v \in T_\lambda$  sind wir bereits fertig, da  $u$  oder  $v$  der gesuchte Knoten ist. Im anderen Fall sei o.B.d.A.  $u \in R_{\lambda \circ 1}$  und  $v \in R_{\lambda \circ 2}$ . Der Weg  $w$  verläuft vollständig in  $G$ , aber nicht vollständig in  $G_{R_{\lambda \circ 1}}$  und nicht vollständig in  $G_{R_{\lambda \circ 2}}$ . Nach Lemma 2 gibt es ein maximales  $0 \leq i \leq |\lambda|$ , so daß  $w$  vollständig in  $G_{R_{(\lambda)_i}}$  verläuft. Nach Lemma 1 gibt es dann einen Knoten  $z \in T_{(\lambda)_i}$ , der auf  $w$  liegt. Aus  $\text{Rand}(u, v) = \bigcup_{i=0}^{|\lambda|} T_{(\lambda)_i}$  folgt das Theorem. ■

In dem Beispiel ist der letzte gemeinsame Graphname  $LGG(v_3, v_4) = 1$ . Jeder Weg vom Knoten  $v_3$  zum Knoten  $v_4$  führt also über einen der Knoten  $\text{Rand}(v_3, v_4) = T_\epsilon \cup T_1 = \{v_2, v_5, v_8, v_9\}$  (siehe Abbildung 1).

**Lemma 3** Sei  $\alpha$ - $\beta$ - $\sqrt{n}$ - $HG$  ein hierarchischer Graph zum Graphen  $G = (V, E)$ . Es gilt für alle Knotenpaare  $u, v \in V$ :  $|\text{Rand}(u, v)| \in O(\sqrt{n})$ .

**Beweis:** Der hierarchische Graph habe  $k + 1$  Stufen. Nach Definition 2 und 3 gilt:

$$|\text{Rand}(u, v)| \leq \sum_{i=0}^{|LGG(u, v)|} |T_{(LGG(u, v))_i}| \leq \sum_{i=0}^k \beta \sqrt{\alpha^i n} \leq \beta \sqrt{n} \sum_{i=0}^{\infty} (\sqrt{\alpha})^i \in O(\sqrt{n})$$
■

## 6 Gewichtete hierarchische Graphen

Wir betrachten nun gewichtete planare Graphen und erweitern alle Graphen eines hierarchischen Graphen um eine Kantengewichtsfunktion.

**Definition 4** Der um die Kantengewichtsfunktion erweiterte hierarchische Graph  $HG$  zu dem Graphen  $G = (V, E, \delta)$  heißt gewichteter hierarchischer Graph zu  $G$ , wenn für alle Graphen  $G_\lambda = (R_\lambda, [T_\lambda, R_\lambda], \delta_\lambda) \in HG$  gilt:

$$\delta_\lambda(\{u, v\}) = \delta(u, v) \quad \forall \{u, v\} \in [T_\lambda, R_\lambda].$$

**Theorem 4** Gegeben sei ein planarer Graph  $G = (V, E, \delta)$ . Ein gewichteter hierarchischer Graph  $\alpha$ - $\beta$ - $\sqrt{n}$ - $HG$  zu  $G$  lässt sich in  $O(n^2)$  Schritten berechnen.

**Beweis:** Ohne die Kantengewichtsfunktion lässt sich der hierarchische Graph  $HG$  nach Theorem 2 in  $O(n^{1.5})$  Schritten berechnen. Mit dem von Klein u.a. entwickelten Verfahren zur Berechnung der kürzesten Entfernung in  $O(n)$  Zeit in planaren Graphen lassen sich in  $O(n^2)$  Zeit für alle Knotenpaare die kürzesten Entfernungen bestimmen [10]. ■

Zwar benötigen wir im hierarchischen Graphen nicht die Entfernung für jedes Knotenpaar, aber zur Berechnung der kürzesten Entfernungen auf den Stufen größer als Null im hierarchischen Graphen reicht die Betrachtung der induzierten Graphen nicht aus, sondern es muß der gesamte gegebene planare Graph einbezogen werden. Auch die Einbeziehung bereits berechneter Entfernungen bringt keine Laufzeitverbesserung, da die gesamte Randknotenmenge betrachtet werden muß (siehe auch [12]).

**Theorem 5** *Gegeben sei ein gewichteter hierarchischer  $\alpha$ - $\beta$ - $\sqrt{n}$ -HG Graph zu dem planaren Graphen  $G = (V, E, \delta)$ . Die Länge des kürzesten Weges  $\delta(u, v)$  kann für je zwei beliebige Knoten  $u, v \in V$  in  $O(\sqrt{n})$  Operationen berechnet werden.*

**Beweis:** In  $O(\log n)$  Schritten wird  $LGG(u, v)$  berechnet (vgl. Abschnitt 8). Dann wird ein Knoten  $z \in Rand(u, v)$  mit folgender Eigenschaft gesucht:

$$\delta(u, z) + \delta(z, v) = \min\{\delta(u, y) + \delta(y, v) \mid y \in Rand(u, v)\}$$

Nach Theorem 3 gilt:  $\delta(u, v) = \delta(u, z) + \delta(z, v)$ , und nach Lemma 3 ( $|Rand(u, v)| \in O(\sqrt{n})$ ) folgt die Behauptung. ■

In dem Beispielgraphen aus Abschnitt 3 sei das Kantengewicht  $\delta(\{v_1, v_4\}) = 3$  und alle anderen Kantengewichte seien eins. Dann ergibt sich der in Abbildung 3 dargestellte Teilgraph  $G_1 \in HG$  für den gewichteten hierarchischen Graphen. Die kürzeste Entfernung  $\delta(v_3, v_4) = 4$  läßt sich durch die Betrachtung von  $Rand(v_3, v_4) = \{v_2, v_5, v_8, v_9\}$  ermitteln.

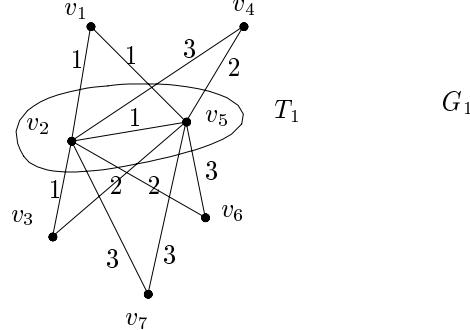


Abbildung 3: Gewichteter Teilgraph  $G_1 \in HG$ .

Wenn ein kürzester Weg mit  $l$  Knoten ausgegeben werden soll, sind dazu weitere  $O(l)$  Operationen aufzuwenden. Für jede Kante in einem Graphen  $G_\lambda$  des hierarchischen Graphen, an der nach Definition 4 der kürzeste Weg vermerkt ist, muß dazu ein Knoten auf diesem kürzesten Weg gespeichert werden. Da auch die Teilwege von kürzesten Wegen kürzeste Wege sind, kann anhand dieser Knoten durch erneuten Zugriff auf den Graphen  $G_\lambda$  in  $O(l)$  Operationen der kürzeste Weg zusammengesetzt werden.

## 7 Regionengewichtete hierarchische Graphen

Die Kantengewichte in gewichteten hierarchischen Graphen entsprechen immer den kürzesten Entfernungen im zugrundeliegenden Graphen. Es reicht jedoch für die Wegberechnung mit

dem hierarchischen Graphen aus, die kürzeste Entfernung in einem induzierten Teilgraphen des zugrundeliegenden Graphen zu vermerken. Für eine Kante des Graphen  $G_\lambda = (R_\lambda, [T_\lambda, R_\lambda]) \in HG$  werden wir die kürzeste Entfernung bezogen auf den von  $R_\lambda$  in  $G$  induzierten Graphen  $G_{R_\lambda}$  bestimmen.

**Definition 5** Der hierarchische Graph  $HG$  zu  $G = (V, E, \delta)$  heißt regionengewichteter hierarchischer Graph, falls für alle  $G_\lambda = (R_\lambda, [T_\lambda, R_\lambda], \delta_\lambda) \in HG$  und für alle Kanten  $\{u, v\} \in [T_\lambda, R_\lambda]$  gilt:

$$\delta_\lambda(\{u, v\}) = \delta_{R_\lambda}(u, v)$$

Dabei ist  $\delta_{R_\lambda}$  die auf Knoten fortgesetzte Kantengewichtsfunktion in dem von  $R_\lambda \subseteq V$  in  $G$  induzierten Graphen.

Der regionengewichtete hierarchische Graph zum Beispielgraphen aus Abschnitt 3 als gewichteter Graph mit  $\delta(\{v_1, v_4\}) = 3$  (alle anderen Kantengewichte seien eins) ist wiederum exemplarisch für  $G_1 \in HG$  in Abbildung 4 dargestellt.

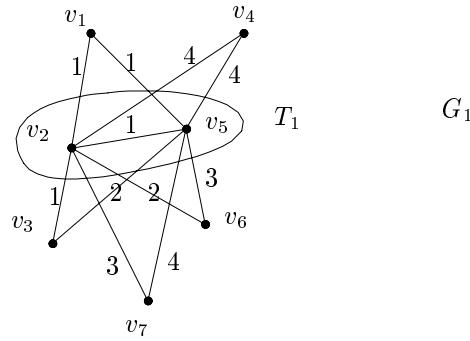


Abbildung 4: Regionengewichteter Teilgraph  $G_1 \in HG$ .

**Theorem 6** Gegeben sei ein planarer Graph  $G = (V, E, \delta)$ . Ein regionengewichteter hierarchischer  $\alpha$ - $\beta$ - $\sqrt{n}$ -HG Graph zu  $G$  lässt sich in  $O(n^{1.5})$  Operationen berechnen.

**Beweis:** Nach Theorem 2 wird der hierarchische Graph  $HG$  ohne die Kantengewichtsfunktion in  $O(n^{1.5})$  Operationen berechnet. Um die Kantengewichte  $\delta_\lambda(\{u, v\})$  für einen Knoten  $u \in T_\lambda$  und alle Knoten  $v \in R_\lambda$  zu bestimmen, genügt es, den Algorithmus aus [10] in dem planaren Teilgraphen  $G_{R_\lambda} \subseteq G$  einzusetzen. Im worst-case wird somit höchstens der Aufwand  $O(|R_\lambda|)$  benötigt. Zur Berechnung von allen Kantengewichten von  $[T_\lambda, R_\lambda]$  für alle Graphen  $G_\lambda \in HG$  ergibt sich nach den selben Gleichungsumformungen wie im Beweis zu Theorem 2 für einen hierarchischen Graphen mit  $k + 1$  Stufen:

$$\sum_{\lambda \in \Sigma^{0,k}} |T_\lambda| O(|R_\lambda|) \in O(n^{1.5})$$

■

Um nachzuweisen, daß die Bestimmung der kürzesten Entfernung mit regionengewichteten hierarchischen Graphen in  $O(\sqrt{n})$  Zeit möglich ist, beziehen wir uns auf das Theorem 3: Jeder Weg von einem Knoten  $u \in V$  zu einem Knoten  $v \in V$  enthält einen Knoten aus  $\text{Rand}(u, v)$ . Die kurze Argumentation im Beweis zu Theorem 5 reicht hier nicht aus, da im regionengewichteten Graphen die Kantengewichte i.A. nicht den kürzesten Entfernungen im zugrundeliegenden Graphen entsprechen.

**Theorem 7** *Sei  $\alpha\text{-}\beta\text{-}\sqrt{n}\text{-HG}$  ein regionengewichteter hierarchischer Graph zu einem Graphen  $G$ . Die Länge des kürzesten Weges  $\delta(u, v)$  kann für je zwei beliebige Knoten  $u, v \in V$  in  $O(\sqrt{n})$  Operationen berechnet werden.*

**Beweis:** Wir verfolgen zunächst dieselbe Argumentation wie in Theorem 3. Sei  $LGG(u, v) = \lambda$ , und o.B.d.A. seien  $u \in R_{\lambda \circ 1}$  und  $v \in R_{\lambda \circ 2}$ . Nach Lemma 2 gibt es ein maximales  $0 \leq i \leq |\lambda|$ , so daß der kürzeste Weg  $w$  von  $u$  nach  $v$  vollständig in  $G_{R_{(\lambda)}i}$  verläuft. Nach Lemma 1 gibt es dann einen Knoten  $z \in T_{(\lambda)i}$ , der auf  $w$  liegt. Es gilt also:

$$\delta(w) = \delta(u, z) + \delta(z, v) = \delta_{R_{(\lambda)}i}(u, z) + \delta_{R_{(\lambda)}i}(z, v)$$

Da im regionengewichteten hierarchischen Graphen für alle Kanten  $\{p, q\} \in [T_{(\lambda)i}, R_{(\lambda)}i]$  gilt:

$$\delta_{(\lambda)i}(\{p, q\}) = \delta_{R_{(\lambda)}i}(p, q)$$

genügt die Betrachtung von  $\text{Rand}(u, v)$ , um die Länge des kürzesten Weges von  $u$  nach  $v$  zu bestimmen. Nach Lemma 3 ( $|\text{Rand}(u, v)| \in O(\sqrt{n})$ ) folgt das Theorem. ■

In dem gewichteten Beispiel ( $\delta(\{v_1, v_4\}) = 3$ ) gibt es zwei kürzeste Wege zwischen  $v_3$  und  $v_4$ , die beide über den Knoten  $v_8$  führen (siehe Abbildung 1). In dem Graphen  $G_\epsilon$  des regionengewichteten hierarchischen Graphen  $HG$  gibt es die Kanten  $\{v_4, v_8\}, \{v_3, v_8\}$  mit folgenden Kantengewichten:  $\delta_\epsilon(\{v_4, v_8\}) = \delta_{R_\epsilon}(v_4, v_8) = 1$  und  $\delta_\epsilon(\{v_3, v_8\}) = \delta_{R_\epsilon}(v_3, v_8) = 3$ . Die Betrachtung der Randknotenmenge  $\text{Rand}(v_3, v_4)$  genügt wiederum, um die kürzeste Entfernung zwischen  $v_3$  und  $v_4$  zu bestimmen.

In regionengewichteten hierarchischen Graphen wird nach Theorem 7 analog zu den gewichteten hierarchischen Graphen ein kürzester Weg mit  $l$  Knoten in  $O(\sqrt{n} + l)$  Operationen berechnet.

## 8 Implementierung

Der zugrundeliegende Graph sei in Form von Adjazenzlisten gespeichert. Bei der Separierung sind zum einen die induzierten Teilgraphen des zugrundeliegenden Graphen ebenfalls in Form von Adjazenzlisten zu speichern. Dies ist in jeder Stufe in  $O(n)$  Zeit möglich. Zum anderen wird die Kantengewichtsfunktion jedes Graphen  $G_\lambda = (R_\lambda, [T_\lambda, R_\lambda], \delta_\lambda)$  eines hierarchischen Graphen in einer Tabelle mit  $|R_\lambda|$  Zeilen und  $|T_\lambda|$  Spalten gespeichert, so daß der Zugriff für ein Knotenpaar in konstanter Zeit möglich ist. Jeder Tabelleneintrag  $\text{Tab}(u, v)$  enthält neben dem Kantengewicht den nächsten Knoten nach  $u$ , der auf dem kürzesten Weg von  $u$  nach  $v$  liegt. Dieser Knoten wird gebraucht, um den kürzesten Weg ausgeben zu können. Außerdem wird bei der Separation ein Baum  $B$  mit den Wörtern  $\lambda \in \Sigma^{0,k}$  als Knoten aufgebaut, bei

dem für jeden Knoten  $\lambda$  ein Verweis auf die trennende Menge  $T_\lambda$  mitgeführt wird. Für jeden Knoten  $v \in V$  des zugrundeliegenden Graphen wird weiterhin der Regionenname  $r(v)$  (falls  $v \in T_\lambda$ ) vermerkt.

Die Berechnung des kürzesten Weges zwischen  $u, v \in V$  erfolgt dann in zwei Schritten:

1. Berechne  $LGG(u, v)$ . Dies ist das längste gemeinsame Präfix von  $r(u)$  und  $r(v)$ , d.h. es werden höchstens  $O(\log n)$  Operationen benötigt.
2. Minimumsbildung über  $Rand(u, v)$ . Im Baum  $B$  werden vom Knoten  $LGG(u, v)$  (direkter Zugriff) auf dem Pfad bis zur Wurzel die trennenden Mengen aufgesucht. Das Kantengewicht kann jeweils durch einen Zugriff auf die angelegten Tabellen ermittelt werden.

## 9 Experimente

Da Straßennetze in der Regel wegen Brücken und Untertunnelungen nur fast planar sind, lässt sich das Separatortheorem nicht direkt anwenden. Stattdessen haben wir eine Heuristik zur Separation verwendet. Die uns vorliegenden Straßendaten sind im GDF-Format gespeichert, in dem die Straßen in neun Geschwindigkeitsklassen eingeteilt sind. Das Straßennetz vom Großraum Stuttgart mit 15224 Knoten und 38032 Kanten wurde sukzessive von der Stufe  $i = 0$  bis zur Stufe  $i = 8$  entlang der Straßen der Klasse  $i$  separiert. Da bei jeder Separation mindestens fünf verschiedene Zusammenhangskomponenten aufgetreten sind, haben wir einen hierarchischen Graphen zu  $\Sigma = \{1, 2, 3, 4, 5\}$  konstruiert. Der Graph besitzt neun Stufen und benötigt 42.8 MBytes Speicherplatz. Schon das Stuttgarter Straßennetz benötigt 3.5 MBytes. Dieser spezielle hierarchische Graph ist ein  $\alpha$ - $\beta$ - $\sqrt{n}$ -HG mit  $\alpha = \frac{1}{2}$  und  $\beta = \frac{5}{2}$ .

Auf einer SPARCstation 10/40 mit 128 MBytes Hauptspeicher ergaben sich unter dem Betriebssystem SunOS 5.4 folgende Laufzeiten: Der gewichtete hierarchische Graph wurde in 104 Minuten aufgebaut, und die Berechnung der kürzesten Entfernung benötigt im average-case 0.305 Millisekunden. Im Vergleich dazu brauchte der Dijkstra Algorithmus 586 Millisekunden [6]. Eine vollständige Tabelle für die Entfernung aller Knotenpaare braucht fast 0.5 GBytes an Speicherplatz, die nicht im Hauptspeicher untergebracht werden konnten. Die reine CPU-Zeit zur Entfernungs berechnung beträgt unter 0.3 Millisekunden, aber da ein Plattenzugriff notwendig ist, ergibt sich eine gesamte Laufzeit von durchschnittlich 60 Millisekunden pro Tabellenzugriff.

## 10 Ausblick

Wir werden versuchen, die Ergebnisse von planaren Graphen auf dünne Graphen ( $|E| \in O(n)$ ) zu verallgemeinern. Zwar haben Bui und Jones in [2] nachgewiesen, daß bereits die Separation von Graphen mit Grad drei auch für Näherungen NP-hart ist, aber uns genügen Separationen der Größe  $O(\sqrt{n})$ .

Für Verkehrsinformationssysteme ist es entscheidend, daß ein Update auf dem zugrundeliegenden Straßennetz nur zu kleinen Änderungen in der hierarchischen Struktur führt. Da sich Straßennetze jährlich durchschnittlich um 5 % ändern, wäre eine Neuberechnung der hierarchischen Struktur bei jeder Änderung inakzeptabel. Es sind die Bedingungen festzustellen, unter denen ein Update nur kleine Änderungen des hierarchischen Graphen nach sich zieht.

## Literatur

- [1] R. Agrawal and H.V. Jagadish. Algorithms for searching massive graphs. *IEEE Transactions on knowledge and data engineering*, 9(2):225 – 238, april 1994.
- [2] T. N. Bui and C. Jones. Finding good approximate vertex and edge partitions is NP-hard. *Information Processing Letters*, 42(3):153–159, may 1992.
- [3] A. Car and A.U. Frank. Modelling a hierarchy of space applied to large road networks. In *Proceedings of IGIS'94, International Workshop on Advanced Research in Geographic Information Systems*, pages 15–24, Ascona, Switzerland, 1994.
- [4] Adrijana Car. *Hierarchical Spatial Reasoning: Theoretical Consideration and its Application to Modeling Wayfinding*. PhD thesis, Department of Geoinformation der TU Wien, 1996. Geoinfo-Serie Nr. 10.
- [5] Chung. Improved separators for planar graphs. In *ICTAG: 6th Quadrennial International Conference on the Theory and Applications of Graphs*, 1991.
- [6] E.W. Dijkstra. A note on two problems in connection with graphs. *Numerical Mathematics*, 1:269 – 271, 1959.
- [7] Djidjev and Venkatesan. Reduced constants for simple cycle graph separation. *Acta Informatica*, 34:231 – 243, 1997.
- [8] K. Ishikawa, S. Azume M. Ogawa, and T. Ito. Map navigation software of the Electro Multivision of the '91 Toyota Soarer. In *Int. Conf. on Vehicle Navigation and Information Systems (VNIS IVHS)*, IEEE, pages 463–473, 1991.
- [9] S. Jung and S. Pramanik. HiTi graph model of topographical road maps in navigation systems. In *Proc. IEEE 12th Int. Conf. Data Engineering*, pages 76–84. New Orleans, LA, 1996.
- [10] Philip N. Klein, Satish Rao, Monika H. Rauch, and S. Subramanian. Faster shortest-path algorithms for planar graphs. In *Proc. 26th Symp. Theory of Computing*, pages 27–37. Assoc. for Computing Machinery, 1994.
- [11] R.J. Lipton and R.E. Tarjan. A separator theorem for planar graphs. *SIAM Journal Appl. Math.*, 36:177 – 189, 1979.
- [12] Bernhard Riedhofer. Hierarchische Strassengraphen. Diplomarbeit 1452, Universität Stuttgart, Institut für Informatik, 1997.
- [13] Jacob Shapiro, Jerry Waxman, and Danny Nir. Level graphs and approximate shortest path algorithms. *Networks*, 22:691–717, 1992.