Automatische Zusammenstellung von Fahrgemeinschaften

Volker Claus, Friedhelm Buchholz
Institut für Informatik, Universität Stuttgart

Zusammenfassung

Es fahren zu viele Autos auf den Straßen. Oft werden sie nur als Transportmittel zur Arbeitsstätte eingesetzt, wo sie unproduktiv den ganzen Tag herumstehen und anderweitig nutzbare Flächen belegen. Fahrgemeinschaften (abgekürzt FGMs), die aus drei oder vier Personen bestehen, könnten eine deutliche Entlastung bewirken.
In diesem Artikel wird ein System zur Erstellung und Verwaltung von Fahrgemeinschaften entworfen. Eingegeben werden Adressen und Eigenarten von Personen; ausgegeben werden Vorschläge für Fahrgemeinschaften, die bzgl. der Umwege oder der Gesamtwegstrecke zusammengestellt wurden; bereits als nicht sinnvoll erkannte Vorschläge werden beim nächsten Mal vermieden. Zur Lösung des Problems müssen viele Gebiete der Informatik herangezogen werden. So muß Verkehrswissen dargestellt, eingegeben und verarbeitet werden, es sind schnelle Algorithmen für kürzeste Wege in Straßengraphen zu entwickeln, Zuordnungen von Personen zu Fahrgemeinschaften müssen aufgestellt werden, und es ist eine interaktive Datenverwaltung zu realisieren.
Fast überall stößt man auf grundsätzliche Fragen, die vorab eine theoretische Klärung der Sachverhalte erfordern. Vor allem hierüber soll in diesem Artikel berichtet werden. Speziell werden mögliche Definitionen für Straßengraphen vorgestellt, die Berechnung von Umwegen diskutiert und die NP-Härte des Zuordnungsproblems für Fahrgemeinschaften mit 3 oder mehr Personen nachgewiesen.

0.1 Formalismen

In diesem Artikel seien $\mathbb{N}_0$ die Menge der natürlichen Zahlen (einschl. der Null), $\mathbb{R}$ die Menge der reellen Zahlen und $\mathbb{R}_+$ die Menge der nichtnegativen reellen Zahlen (einschl. der Null). Ein knoten- und kantenmarkierter gerichteter *Graph* $G = (V, E, \mu, \delta)$ besteht aus einer endlichen Menge von Knoten V, einer Menge von Kanten $E \subseteq V \times V$ und den Abbildungen $\mu : V \to M_1$ und $\delta : E \to M_2$ für zwei Mengen M_1 und M_2. Ein *Weg* der Länge j vom Knoten u_0 zum Knoten u_j im Graphen G ist eine Folge von Knoten $(u_0, u_1, \ldots, u_j)$ mit $(u_{i-1}, u_i) \in E$ für $i = 1, \ldots, j$. Eine Kantenmarkierung δ wird durch

$$\delta(w) = \sum_{i=1}^{j} \delta((u_{i-1}, u_i))$$

auf beliebige Wege $w = (u_0, u_1, \ldots, u_j)$ von u_0 nach u_j und durch

$$\delta(u, v) = \min\{\delta(w) \mid w \text{ ist ein Weg von } u \text{ nach } v \text{ in } G\}$$

auf beliebige Knotenpaare (u, v) fortgesetzt.
Wenn $P = \{p_1, \ldots, p_n\}$ eine n-elementige Menge ist, dann heißt eine Menge $\{F_1, F_2, \ldots, F_r\}$ mit

für $i = 1, \ldots, r$ gilt: $\emptyset \neq F_i \subseteq P$,

für $i \neq j$ gilt: $F_i \cap F_j = \emptyset$,

$\bigcup_{i=1}^{r} F_i = P$

eine *Partition* von P.

0.2 Motivation und Systemaufbau

Das zu erstellende System wird durch seine Ein-Ausgabe-Relation charakterisiert. Eingegeben werden in das System für jede Person fogende Daten:

- Rahmenbedingungen (feste oder gleitende Arbeitszeiten, Schichtdienst, wechselnder Arbeitsplatz usw.),
- Adresse der Wohnung,
- Adresse des Arbeitsplatzes oder der Arbeitsplätze,

- (späteste) Ankunftszeit für jede Arbeitsstelle,
- (geplante) Rückkehrzeit zur Wohnung, bzw. Dauer der Arbeitszeit(en),
- Eigenarten der Person (Geschlecht, nur als Fahrer oder als Mitfahrer geeignet, unterhält sich gern, raucht, möchte reine Frauenfahrgemeinschaft, obere Zeitschranke c in Minuten, die diese Person bereit ist, für die Fahrgemeinschaft zusätzlich aufzubringen, usw.).

Ausgegeben werden Vorschlagslisten von Fahrgemeinschaften, wobei früher bereits fehlgeschlagene Vorschläge zu beachten sind. Die genannten Personen sollten dann versuchen, Fahrgemeinschaften zu bilden. Scheitern diese oder müssen sie neu zusammengesetzt werden, dann soll das System die nicht vermittelten Personen erneut zuordnen.

Eine Fahrgemeinschaft (FGM), bestehend aus j Personen, wird nur dann vorgeschlagen, wenn die j Personen folgendes FGM-Kriterium erfüllen: $j \leq 4$, genau eine Person ist Fahrer, und der Zeitverlust, den der Fahrer in Kauf nehmen muß, um die $j-1$ Mitfahrer abzuholen und abzusetzen, beträgt höchstens c Minuten (siehe oben unter Eigenarten). Zusätzlich dürfen keine Konflikte bei den Eigenarten der j Personen vorliegen.

Die Schranke c kann auch global vorgegeben werden. Alternativ zum Zeitverlust kann auch der Umweg in Kilometern beschränkt werden.

Auf Grund dieser Ein-Ausgabe-Beziehung und des FGM-Kriteriums muß das System folgendes leisten können:

- Aus den Adressen müssen die geografischen Positionen auf der Erde oder im Straßengraphen, der das Verkehrswissen des Systems enthält, errechnet werden können.
- Hieraus wiederum läßt sich der kürzeste Weg (bzgl. Zeit oder Entfernung) ermitteln, der vermutlich von der jeweiligen Person benutzt wird; zusätzlich sollten eventuell der zweitkürzeste, der drittkürzeste usw. Weg berechet werden können.
- Es ist ein Nachbarschaftsbegriff festzulegen. Benachbart sollen zwei Personen genau dann sein, wenn sie von ihrer Wohnung die Wohnung des anderen innerhalb einer Zeitschranke erreichen können, bzw. wenn die Wohnungen weniger als eine gegebene Kilometerzahl voneinander entfernt sind.

- Es ist eine Halbordnung „kann abholen“ zu definieren: Eine Person P kann eine Person Q abholen, wenn die Wohnung und die Arbeitsstätte von Q in der Nähe des Weges liegen, den P bei der Fahrt von seiner Wohnung zum Arbeitsplatz zurücklegt.

- Die Menge der Personen ist nun in möglichst wenige Teilmengen zu zerlegen, so daß für die Personen jeder Teilmenge das FGM-Kriterium erfüllt ist. Da diese Zerlegung sehr aufwendig ist, werden nur Personen betrachtet, die benachbart sind oder die ein Fahrer abholen kann. Diese Teilmengen werden als Vorschläge für mögliche Fahrgemeinschaften ausgegeben.

- Die einmal gemachten Vorschläge werden mitprotokolliert, um gescheiterte FGMs nicht erneut vorzuschlagen. Weiterhin sollte erfaßt werden, ob früher gemachte Vorschläge tatsächlich zu FGMs geführt haben.

Das System sollte darüber hinaus

- auf Einzelanfragen reagieren können, d.h.: wenn eine Person rasch eine FGM sucht, so sollte schnell ein Vorschlag gemacht werden, in welcher nicht voll besetzten FGM sie mitfahren kann;

- eine grafische Oberfläche besitzen, in der Stadtpläne, Straßengraphen, Wege und Umwege dargestellt werden können,

- Zusatzkomponenten für Plausibilitätstests, manuellen Eingriff und Änderungsdienste besitzen,

- Schnittstellen für Erweiterungen und Anbindung an andere Systeme haben.

Das System kann entweder einer Vermittlungsfirma gehören oder als online-Dienst allgemein zugänglich sein.

Es gibt diverse Untersuchungen über die Vermittlung von Fahrgemeinschaften [Reinke 85] [Reinkober 94]. Wie so oft erwies sich hierbei nicht ein technisches System, sondern das menschliche Verhalten als der entscheidende Faktor. FGMs kommen meist deshalb nicht zustande, weil Menschen einander nicht mögen, ihre (scheinbare) Ungebundenheit, die ihnen das Auto suggeriert, erhalten wollen, nicht bereit sind, Gewohnheiten aufzugeben, und das

Prestigeobjekt Auto nicht mit anderen teilen wollen. Auch von außen wird kein Druck zur Bildung von FGMs erzeugt: Der Benzinpreis ist noch relativ gering, die steuerliche Abschreibung ist gegeben, ein Verkehrsinfarkt und Staus finden noch relativ selten auf dem Weg zur Arbeit statt, und die Politik und Wirtschaft ist nicht an der Verringerung der Autos, sondern an der Erhöhung des PKW-Bestandes bei gleichzeitiger besserer Auslastung der vorhandenen Straßen interessiert. Durch FGMs könnten viele Nachteile beseitigt werden, wie Streßerhöhung, Unpünktlichkeit, Verschwendung von Flächen für Parkplätze, Lärmbelästigung, gesundheitliche Auswirkungen und Umweltschäden. Das Bewußtsein für die verheerenden Auswirkungen, die durch den sorglosen Umgang mit Energieressourcen und die Zerstörung von Lebensgrundlagen wie Luft und Wasser heute schon klar abzusehen sind, ist in Deutschland bisher kaum vorhanden. Ein Fahrgemeinschaften-Vermittlungs-System könnte aber schon bald zu einem Bündel von Maßnahmen gehören, um die Zahl der Autos auf den Straßen (nicht der Autos im Besitz der Bürger!) zu den Stoßzeiten mindestens zu halbieren.

0.3 Verkehrswissen und Straßengraphen

Da der genaue Straßenverlauf und die Nutzung der Landschaft für FGMs nicht interessieren, wird die gesamte Verkehrsinformation in Graphen eingefangen. Kreuzungen werden hierbei durch Knoten, Straßenstücke zwischen den Kreuzungen durch Kanten dargestellt. Die Knoten und Kanten sind markiert: Den Knoten werden die Erdkoordinaten (als Gauß-Krüger-Koordinaten), der Name der Kreuzung, die Namen der kreuzenden Straßen, die Größe, die Ampelschaltfolge, der Zustand, der Durchsatz der Kreuzung usw. zugeordnet, den Kanten der Name der Straße, die Länge des Straßenstücks, die Hausnummern, Bezirksangaben, Postleitzahl, Straßendaten wie Belag, Zustand, Zahl der Spuren, Geschwindigkeitsbegrenzungen usw.

Eine Überschlagsrechnung ergibt, daß man für das FGM-Problem mit 100 Byte je Knoten und 150 Byte je Kante auskommt. Bei höchstens 3 Millionen Kreuzungen in Deutschland und 6 Millionen Straßenstücken zwischen ihnen reicht eine Speicherkapazität von 2 Gigabyte für die ganze Bundesrepublik aus.

Zwei Probleme ergeben sich nun sofort:

- Wie stellt man das Verkehrswissen maschinenunabhängig dar, so daß man es für jede neue Anwendung wieder nutzen kann?

- Wie muß man die Straßengraphen definieren, damit die späteren Algorithmen effizient arbeiten können?

Auf die erste Frage soll hier nicht eingegangen werden. Naheliegend ist es, alle Straßen als Liste ihrer Kreuzungen, Plätze, Sackgassenenden usw. mit allen erforderlichen Hilfsinformationen zu erfassen [BMV], [OFFIS 93]. Für die zweite Frage sind die später einzusetzenden Algorithmen für kürzeste Wege wichtig. Meist verwendet man den Dijkstra-Algorithmus [Dijkstra 59] oder Varianten wie den A*-Algorithmus [Dechter, Pearl 85]. Deren Aufwand beträgt $O(m \cdot \log(n))$, wobei n die Zahl der Knoten und m die Zahl der Kanten des Graphen sind (Fibonacci-Heaps, [Ottmann, Widmayer 93]). Da in der Praxis n und m oft zwischen 20 000 und 50 000 liegen, benötigt jede Wegsuche über 10 Minuten, wenn der Rechner 1 Millisekunde pro Knotenbearbeitung braucht. Dies ist indiskutabel hoch. Man wird daher hierarchisch vorgehen. Hierzu sollen zwei Ansätze vorgestellt werden. Zum einen kann man einen Straßengraphen als Überlagerung von Graphen verschiedener Level, zum anderen als hierarchischen Graphen, in dem jeder Knoten einem Graph entspricht, einführen.

In Straßenkarten gibt es meist vier Level von Straßen: Siedlungs-, Kreis-, Land- und Fernstraßen. Formal kann man dann einen k-Level-Graphen wie folgt definieren:

Definition 1 (k-Level-Straßengraph der Granularität c)
Seien $k \geq 1$ eine natürliche Zahl und $c = (c_1, \ldots, c_{k-1}) \in \mathbb{R}_+^{k-1}$ ein positiver reeller Vektor mit $c_1 \leq c_2 \leq \cdots \leq c_{k-1}$. $(G_1, G_2, \ldots, G_k)$ heißt k-Level-Straßengraph der Granularität c über der Knotenmenge V_1 und der Kantenmenge E_1, wenn gilt:

1. *Für $i = 1, \ldots, k$ gilt: $G_i = (V_i, E_i, \mu_i, \delta_i, \tau_i, \varphi_i)$ ist ein gerichteter, schlingenfreier, knoten- und kantenmarkierter Graph mit*

   $\mu_i : V_i \rightarrow \mathbb{R} \times \mathbb{R}$ *(gibt die Koordinaten der Knoten an)*

   $\delta_i : E_i \rightarrow \mathbb{R}_+$ *(gibt die Länge der Kanten an)*

   $\tau_i : E_i \rightarrow \mathbb{R}_+$ *(gibt die Durchfahrtzeit der Kanten an)*

   $\varphi_i : E_i \rightarrow \{0, 1\}$ *(gibt an, ob die Kante echt (0)oder virtuell (1) ist).*

$E_i^0 = \{(u,v) \in E_i | \varphi_i((u,v)) = 0\}$ *heißt die Menge der* echten Kanten *des Graphen* G_i *und* $E_i^1 = \{(u,v) \in E_i | \varphi_i((u,v)) = 1\}$ *die Menge der* virtuellen Kanten *von* G_i.

2. $V_k \subseteq V_{k-1} \subseteq \ldots \subseteq V_2 \subseteq V_1$ *und* $E_k^0 \subseteq E_{k-1}^0 \subseteq \ldots \subseteq E_2^0 \subseteq E_1^0$. *Die Kanten* $(u,v) \in E_i \setminus E_{i+1}$ *heißen* Kanten vom Level i.

3. *Für alle* $1 \le i,j \le k$ *und für alle* $u \in V_i \cap V_j$: $\mu_i(u) = \mu_j(u)$.
Für alle $1 \le i,j \le k$ *und für alle* $(u,v) \in E_i^0 \cap E_j^0$: $\delta_i((u,v)) = \delta_j((u,v))$ *und* $\tau_i((u,v)) = \tau_j((u,v))$.

4. *Für alle* i*, für alle Knoten* $u,v \in V_i$ *und für alle* $1 \le j < i$ *muß aus* $\delta_j(u,v) \ge c_j$ *folgen:* $\tau_j(u,v) \ge \tau_i(u,v)$.

5. *Für alle* i *und für alle Kanten* $(u,v) \in E_i$ *gilt, daß die Länge dieser Kante mindestens so groß wie der euklidische Abstand der Punkte* u *und* v *sein muß. Formal:* $\sqrt{(\mu_{i,1}(u) - \mu_{i,1}(v))^2 + (\mu_{i,2}(u) - \mu_{i,2}(v))^2} \le \delta_i((u,v))$. *Hierbei bedeuten* $\mu_{i,1}(u)$ *die erste und* $\mu_{i,2}(u)$ *die zweite Komponente von* $\mu_i(u)$.

6. *Für alle* i *und für alle virtuellen Kanten* $(u,v) \in E_i^1$ *gilt:*
$\delta_i((u,v)) = \min\{\delta_j(u,v) \mid j \le i\}$,
$\tau_i((u,v)) = \min\{\tau_j(u,v) \mid j \le i\}$.

Die Bedingungen *1.* bis *3.* in dieser Definition beschreiben den Aufbau der Graphen G_i. Durch Weglassen der (echten) Kanten niedrigen Levels entstehen die Graphen, mit denen immer größere Entfernungen schneller überbrückt werden können. Bedingung *4.* stellt hierbei sicher, daß auf den Kanten höheren Levels die Entfernungen tatsächlich *schneller* überbrückt werden, sofern diese Entfernungen nicht zu kurz, d.h., größer als die Granularität in dem Level sind. Da den Knoten durch die Funktion μ Punkte auf der Erdoberfläche entsprechen, muß ihre durch δ gegebene Entfernung größer als der euklidische Abstand sein. Schließlich können auf höherem Level zusätzlich virtuelle Kanten eingeführt werden, die nichts anderes als Abkürzungen für kürzeste Wege auf einem tieferen Level zwischen zwei Knoten sind. Bedingung *6.* formalisiert diese Aussage. Mit dem Konzept der virtuellen Kanten lassen sich Streckenabschnitte, die oft betrachtet werden müssen und mehrere Kreuzungen aufweisen, auf einem höheren Level vereinfacht durch eine Kante darstellen.

In Abbildung 1 ist ein 2-Level-Straßengraph (G_1, G_2) mit Granularität (3) in der ungerichteten Version dargestellt. Alle Kanten liegen in E_1. Die Kante $\{v_1, v_{13}\}$ soll die einzige virtuelle Kante sein; sie liegt in E_2^1, d.h. $E_2^1 = \{\{v_1, v_{13}\}\}$ $(\varphi_2(\{v_1, v_{13}\}) = 1)$. Die Kantenmarkierung für die virtuelle Kante ergibt sich entsprechend Bedingung 6. Alle übrigen gestrichelten Linien bilden die Kantenmenge E_2^0, also die echten Kanten auf Level 2. Im Graphen G_1 stimmen hier die Gewichtsfunktionen δ_1 und τ_1 genau überein $(\forall (u,v) \in E_1^0\ \delta_1((u,v)) = \tau_1((u,v)))$, daher ist an den Kanten nur eine Zahl angegeben. Man beachte, daß der 2-Level-Straßengraph nicht der Granularität (2) genügt. Dies scheitert an den Knoten v_4, v_6 :

$$\delta_1(v_4, v_6) = \tau_1(v_4, v_6) = 2.5\ <\ \tau_2(v_4, v_6) = 6.$$

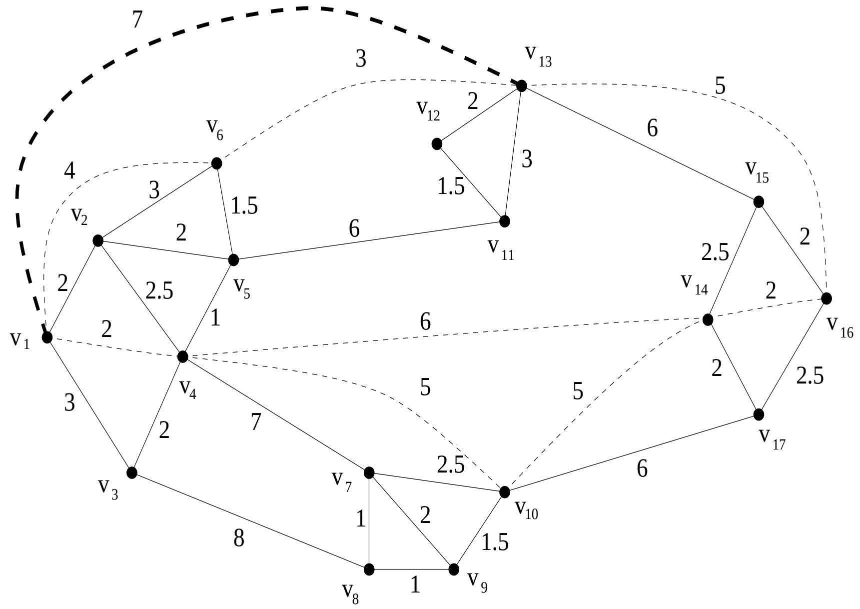

Abbildung 1: 2-Level-Straßengraph mit Granularität (3).

Man beachte, daß die Bedingung 4. für größere Graphen äuërst schwierig nachzuprüfen ist. Anstatt die Bedinung für alle Knotenpaare aus dem Level i zu fordern, kann man sie auch nur für alle bestehenden Kanten aus dem

Level i verlangen. Dadurch ist nicht mehr zugesichert, daß es zwischen zwei Knoten $u, v \in V_i$ im Level i einen kürzeren Weg als in einem kleineren Level $(1 \leq j < i)$ gibt, auch wenn die Entfernung groß genug ist $(\delta_j(u,v) \geq c_j)$.

Der zweite Ansatz verwendet hierarchische Graphen. Dies sind Graphen, deren Knoten mit Graphen (den 'Verfeinerungen' $\kappa(v)$ eines Knotens v) markiert sind. In jeder Verfeinerung zeichnet man einen Knoten $\zeta(v)$ aus, der anschaulich als Mittelpunkt dient; auf ihn werden alle Entfernungen im übergeordneten Graphen bezogen. Eine mögliche Definition lautet:

Definition 2 (hierarchischer Straßengraph)

- *$\emptyset_G$ bezeichne den Graphen mit der leeren Knotenmenge.*
- $\mathcal{G}_0 \overset{\text{def}}{=} \{\emptyset_G\}$.
- *Für alle $i \geq 0$ setze*

$$\begin{aligned}\mathcal{G}_{i+1} \overset{\text{def}}{=} \{G = (V, E, \kappa, \zeta, \mu, \delta, \tau) \mid (V,E) &\text{ ist ein gerichteter Graph,}\\ &\text{auf dem zusätzliche Komponenten } \mu, \delta, \tau \text{ wie in}\\ &\text{Definition 1 definiert sind, } \kappa : V \to \textstyle\bigcup_{j=0}^{i} \mathcal{G}_j,\\ &\text{und } \zeta \text{ ist eine Abbildung der Knotenmenge } V\\ &\text{in die Knotenmengen der durch } \kappa \text{ definierten Graphen mit:}\\ &\forall v \in V : \kappa(v) = (V_v, E_v, \ldots) \neq \emptyset_G \Rightarrow \zeta(v) \in V_v\}.\end{aligned}$$

  Zusätzlich muß gelten, daß die Knotenmenge V und alle Knotenmengen $\kappa(v)$ für $v \in V$ paarweise disjunkt sind, ebenso die Knotenmengen der tieferen Stufen untereinander und mit allen höheren Stufen.

- *Die Menge $\bigcup_{i=0}^{\infty} \mathcal{G}_i$ heißt Menge der hierarchischen Straßengraphen. Die Graphen aus $\mathcal{G}_i$ heißen hierarchische Straßengraphen der Stufe i.*

Falls stets $\kappa : V \to \mathcal{G}_i$ für die Knotenmengen V der Graphen aus $\mathcal{G}_{i+1}$ festgelegt wird, so spricht man von vollständig ausgeglichenen hierarchischen Straßengraphen. Allgemein kann man auch verlangen, daß die Graphen zusammenhängend oder stark zusammenhängend sein müssen. Weiterhin müssen die Funktionen μ, δ und τ auf den verschiedenen Stufen gewissen Nebenbedingungen genügen, die in Definition 2 noch nicht aufgenommen wurden. In der Praxis wird man alle auftretenden Graphen so klein wählen,

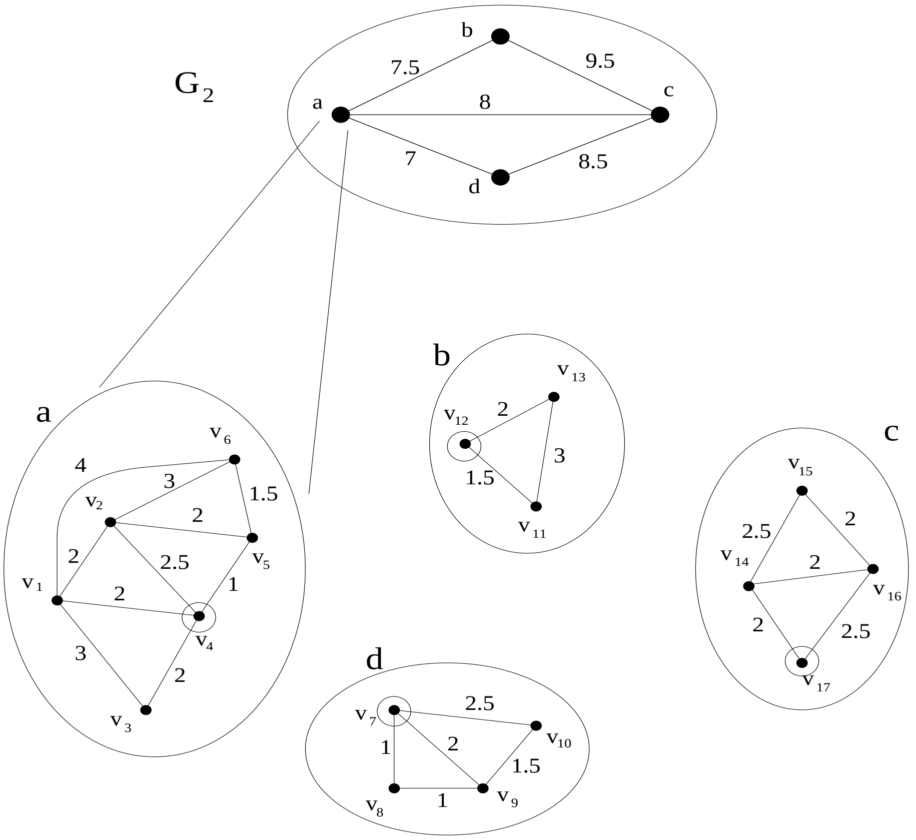

Abbildung 2: Hierarchischer Straßengraph mit zwei Stufen.

daß die Funktionen μ, δ und τ durch Tabellen dargestellt werden können, wodurch Entfernungen und benötigte Zeiten auch über mehrere Stufen hinweg sehr rasch ermittelt werden können.

In Abbildung 2 ist der 2-Level-Straßengraph aus Abbildung 1 als hierarchischer Straßengraph mit 2 Stufen dargestellt, wobei nur die Zeitabstände τ_1, τ_2 an den Kanten angegeben sind. Virtuelle Kanten werden hier nicht beachtet; die Kanten zwischen den durch κ definierten Graphen verschwinden auf unterem Level und fließen in die Kanten auf höherem Level mit ein. D.h. der hierarchische Straßengraph gibt die reale Straßensituation i.A. nicht korrekt wieder. Dies hängt von der Modellierung ab. Der Graph G_2

auf dem höchsten Level besitzt die Knotenmenge $\{a, b, c, d\}$. Jedem Knoten ist durch die Funktion κ ein Graph auf der unteren Stufe zugeordnet, wobei der durch die Abbildung ζ ausgewählte Knoten mit einem Kreis markiert ist ($\zeta(a) = v_4, \zeta(b) = v_{12}, \zeta(c) = v_{17}, \zeta(d) = v_7$). Die Kantengewichte auf der untersten Stufe entsprechen genau den Kantengewichten im 2-Level-Straßengraphen, während die Kantengewichte in der zweiten Stufe die kürzesten Entfernungen zwischen den durch ζ ausgewählten Knoten sind.

Was sind die Vor- und Nachteile der beiden Ansätze? Führt man eine Wegsuche mit Straßengraphen nach Definition 1 durch, dann sucht man auf jedem Level die kürzesten Verbindungen zu Knoten, die sich jeweils auf dem nächsthöheren Level befinden, und setzt hieraus schrittweise einen Weg von u nach v zusammen. Auf dem obersten Level wird man alle Entfernungen in einer Tabelle speichern. Man benötigt zur Ermittlung des kürzesten Weges zwar immer noch den Aufwand $O(m \cdot \log(n))$, jedoch sind n und m nun nur noch die Anzahlen von Knoten und Kanten, die in einer gewissen Umgebung des Start-, bzw. Zielknotens liegen. Beim zweiten Ansatz tabelliert man alle Entfernungen auf der höchsten Stufe und innerhalb jeder Verfeinerung. Ein Weg von u nach v wird hier ermittelt, indem man (jeweils ausgehend von u und von v) die Entfernung von einem Knoten x zum Mittelpunktsknoten $\zeta(y)$, wobei x in der Vefeinerung $\kappa(y)$ des Knotens y der nächsthöheren Stufe liegt, abliest und dann in der darüberliegenden Stufe genauso fortfährt; dies bricht ab, wenn zwei Knoten in derselben Verfeinerung liegen oder man auf der obersten Stufe angelangt ist. Da alle Entfernungen tabelliert sind, erhält man nach $2 \cdot k - 1$ Tabellenzugriffen einen Wert, der eine obere Schranke für den kürzesten Weg von u nach v bildet, im allgemeinen aber nicht die kürzeste Entfernung, sondern nur eine gute Näherung angibt. Diesen Nachteil kann man abmildern, indem man zu jedem Knoten nicht nur den Weg zum zugehörigen Mittelpunktsknoten, sondern richtungsabhängig auch weitere Wege zu benachbarten Mittelpunktsknoten einbezieht. Sind dies im Schnitt r Stück, so muß man mit höchstens $O((\sum_{i=1}^{k-1} r^i)^2)$ Tabellenzugriffen rechnen. In der Praxis mit $k = 4$ und $r = 3$ ergeben sich höchstens 1521 Zugriffe, was für die Umwegsuche zu groß ist. Es bietet sich daher ein Kompromiß an: Für die nur einmal zu ermittelnden kürzesten Wege zwischen Wohnung und Arbeitsstätte wähle man ein exaktes Verfahren, für die Umwegsuche, bei der sehr viele Wege zu berechnen sind, verwende man die Tabellentechnik der hierarchischen Graphen.

In dem Beispielgraphen (Abbildungen 1 und 2) ergibt sich für die Wegsuche zwischen den Knoten v_2 und v_{16} mit dem 2-Level-Straßengraph als kürzester Weg $w_1 = (v_2, v_1, v_4, v_{14}, v_{16})$ mit Weglänge $2 + 2 + 6 + 2 = 12$ und mit dem hierarchischen Straßengraph der kürzeste Weg $w_2 = (v_2, v_4 = \zeta(a), a, c, \zeta(c) = v_{17}, v_{16})$ mit Weglänge $2.5 + 8 + 2.5 = 13$. Der kürzeste Weg $w_3 = (v_2, v_4, v_{14}, v_{16})$ besitzt jedoch die Weglänge $2.5 + 6 + 2 = 10.5$.

0.4 Weg- und Umwegsuche

Für jede Person muß das System den kürzesten Weg von der Wohnung zur Arbeitsstätte und zurück finden. Falls es mehrere Wege gibt, muß durch Zusatzbedingungen festgelegt werden, welchen Weg das System weiterhin verwenden soll. In der Praxis wird dieser Weg noch von der Uhrzeit und dem Wochentag abhängen (Stoßzeiten, hohe Verkehrsdichten am Freitag Nachmittag usw.). Diese Wege lassen sich entweder durch wiederholte Anwendung des Dijkstra-Algorithmus (von einem zu allen Knoten) oder des Floyd-Algorithmus (von allen zu allen Knoten) berechnen. Eine brauchbare Variante zum Dijkstra-Algorithmus ist der A*-Algorithmus, der sich mit Hilfe des euklidischen Abstands schnell in die richtige Richtung bewegt. Er ist aber wenig geeignet, wenn die Zeit zu optimieren ist, da hierbei eine Anfangsbewegung in die entgegengesetzte Richtung oft zum kürzesten Weg führen kann. In [OFFIS 93] wurde der Halmakegel als Strategie vorgeschlagen, der aber im allgemeinen nicht den kürzesten Weg liefert. Alle diese Algorithmen haben ein $O(m \cdot \log(n))$-Verhalten, weshalb sie nur für eine einmalige Berechnung geeignet sind.

Wegen der Staugefahren kennen alle Fahrer in der Regel neben dem kürzesten auch den zweit- und drittkürzesten Weg, auf den ggf. ausgewichen wird. Um die k-kürzesten Wege zu bestimmen, kann man eine Variante des Floyd-Algorithmus ([Minieka 74]), die in kubischer Zeit läuft oder spezielle Algorithmen, die den Dijkstra-Algorithmus verwenden, benutzen [Yen 71] [Dreyfus 69].

Will man mehrere Personen zu einer FGM zusammenfassen, dann muß man berechnen, wie groß der kleinste Umweg ist, den eine Person fahren muß, um zusätzlich zum eigenen Weg die Beifahrer von deren Wohnung zur Arbeitsstätte zu befördern. Wenn k Personen $p_1, p_2, \ldots, p_k$ mit den Startpunkten (=Wohnungen) $s_1, s_2, \ldots, s_k$ und den Zielpunkten (=Arbeitsstätten) $z_1, z_2, \ldots, z_k$ gegeben sind, dann muß man für jeden potentiellen Fahrer $p_i (i = 1, \ldots, k)$ alle Wege betrachten, die mit s_i beginnen, mit z_i enden und alle anderen Start- und Zielpunkte enthalten, wobei aber jedes z_j im Weg erst nach dem zugehörigen s_j auftreten darf. Für $k = 2$ gibt es nur 2 solcher zulässiger Wege, für $k = 3$ sind es 18 und für $k = 4$ bereits 360. Die Anzahl dieser Wege wächst exponentiell, sie läßt sich nach unten leicht durch $k! \cdot (k-1)!$ abschätzen.

Wir beschreiben nun, welche Umwege berechnet werden müssen.

Definition 3

1. *Es seien $P = \{p_1, \ldots, p_n\}$ eine Menge von n Personen, $S = \{s_1, \ldots, s_n\}$ die zugehörige Menge der Startpunkte und $Z = \{z_1, \ldots, z_n\}$ die Menge der Zielpunkte. Es sei $V = S \cup Z$ die Menge aller Punkte, $E = V \times V$. $\delta : V \times V \to \mathbb{R}_+$ erfülle die Axiome einer Abstandsfunktion: $\forall x, y, z \in V : (\delta(x,y) = 0 \Leftrightarrow x = y) \wedge (\delta(x,y) = \delta(y,x)) \wedge (\delta(x,z) \leq \delta(x,y) + \delta(y,z))$. Der vollständige kantenmarkierte Graph $G = (V, E, \delta)$ heißt dann ein* FGM-Graph *zu P.*

2. *Sei $F \subseteq P$, o.B.d.A. $F = \{p_1, \ldots, p_k\}$ für ein k mit $1 \leq k \leq n$. Ein Weg $w = (x_1, \ldots, x_{2k})$ der Länge $2k - 1$ heißt* zulässiger Weg *für F, wenn gilt:*

   (a) *In w kommt jedes $s_1, \ldots, s_k, z_1, \ldots, z_k$ genau einmal vor.*

   (b) *$\forall i, j, l : (x_i = s_l \wedge x_j = z_l) \Rightarrow i < j$.*

   (c) *$x_1 = s_r \Rightarrow x_{2k} = z_r$. Die Person p_r heißt dann Fahrer für diesen Weg w.*

3. *Die Funktion U, die jeder Teilmenge $F \subseteq P$ den minimalen Umweg zuordnet, also $U(F) = \min\{\delta(w) - \delta(s_i, z_i) \mid 1 \leq i \leq k, w$ ist ein zulässiger Weg für F, der mit s_i beginnt $\}$, heißt* Umweg für F.

4. *Die Funktion K, die jeder Teilmenge $F \subseteq P$ die minimalen Kosten zuordnet, also $K(F) = \min\{\delta(w) \mid w$ ist ein zulässiger Weg für F $\}$, heißt* Kostenfunktion für F.

Bemerkung: Es gibt Beispiele, in denen $U(F)$ und $K(F)$ zu verschiedenen Wegen mit verschiedenen Fahrern führen.

Wir müssen Fahrgemeinschaften F so zusammenstellen, daß $U(F)$ 'zumutbar' bleibt oder daß die Gesamtkosten minimal werden. Hierzu muß der Umweg zu einem $F \subseteq P$ möglichst rasch berechnet werden können. Dies ist aber im allgemeinen nicht effizient möglich, da im Falle, daß jeder Startpunkt s_i sehr dicht bei seinem Zielpunkt z_i für $i = 1, \ldots, k$ liegt, die Berechnung des Umwegs gleich der Lösung des TSP (Travelling Salesperson Problem) ist. Beschränkt man die Anzahl der Personen in F auf 4, dann genügt es, 360 Möglichkeiten durchzuprobieren (s.o.). Da diese Konstante recht groß ist, wird man Heuristiken einsetzen; z.B. kann man als Fahrer diejenige Person i wählen, für die $\delta(s_i, z_i)$ maximal ist, wodurch sich die Zahl der zu testenden Möglichkeiten auf 90 verringert. Von hier aus könnte man nach einem Entfernungskriterium die zuerst abzuholende Person fest bestimmen usw. Dadurch würde man nicht immer, aber ziemlich oft den minimalen Umweg ermitteln. Geeignete Heuristiken, die im Mittel einem Zufallsverfahren deutlich überlegen sind und die möglichst wenige Fälle durchrechnen müssen, werden derzeit entwickelt.

0.5 Zuordnungen

Die Aufgabe lautet: Berechne zu einem $k \in \mathbb{N}_0$ mit $k \geq 2$, einer Personenmenge P, einem FGM-Graphen G und einem $c \in \mathbb{R}^+$ Fahrgemeinschaften F der Größe $\leq k$, deren Umweg $U(F)$ oder deren gesamte Kosten $K(F)$ jeweils $\leq c$ sind. Wir definieren die zu lösenden Probleme formal:

Definition 4 *Es seien $k \in \mathbb{N}_0$ mit $k \geq 2$, eine Personenmenge P, ein FGM-Graph G zu P und ein $c \in \mathbb{R}^+$ gegeben.*

1. **Problem FGM-k-Umweg**: *Gesucht ist eine Partition $\{F_1, F_2, \ldots, F_r\}$ von P, so daß*

   (a) *für $j = 1, \ldots, r$ gilt: $\mid F_j \mid \leq k$,*

   (b) *für $j = 1, \ldots, r$ gilt: $U(F_j) \leq c$ und*

   (c) *r minimal ist.*

2. **Problem FGM-k-Kosten**: *Gesucht ist eine Partition $\{F_1, F_2, \ldots, F_r\}$ von P, so daß*

   (a) *für $j = 1, \ldots, r$ gilt: $\mid F_j \mid \leq k$,* *(b)* $\sum_{j=1}^{r} K(F_j) \leq c$.

3. **Problem FGM-k-Umweg-$\mathbb{R}^2$, Problem FGM-k-Kosten-$\mathbb{R}^2$**: *Wie unter 1., bzw. 2., jedoch ist der FGM-Graph in die Ebene eingebettet, und die Kantenmarkierung δ ist der euklidische Abstand zwischen den Punkten.*

Theorem 1 *[Buchholz 94]*
Sei $|P| = n$. Dann gilt

- *FGM-2-Umweg ist in $O(n^{2.5})$ lösbar.*
- *FGM-2-Kosten ist in $O(n^6)$ lösbar.*

Beweis: Man kann diese Probleme leicht auf das maximale Matching, bzw. auf das gewichtete maximale Matching zurückführen. Hierzu konstruiere zu der Personenmenge P dem FGM-Graph $G = (V, E, \delta)$ und der Umwegschranke $c \in \mathbb{R}^+$ den Graphen $G' = (P, E')$ mit $E' = \{\{p_i, p_j\} \mid U(\{p_i, p_j\}) \leq c\}$. Dies läßt sich in der Zeit $O(n^2)$ durchführen. Offenbar gilt: $\{F_1, F_2, ..., F_r\}$ ist genau dann ein maximales Matching in G', wenn $\{F_1, F_2, ..., F_r\} \cup \{\{p\} \mid p \in P$ und für alle $1 \leq i \leq r : p \notin F_i\}$ ist FGM-2-Partition für P mit minimalem r. Die Konstruktion für die Kosten ist etwas umfangreicher; hierbei werden $n \times (n-1)$ Knoten hinzugefügt und dann das gewichtete maximale Matching mit kubischem Aufwand gelöst [Papadimitriou, Steiglitz 82].

Theorem 2 *[Buchholz 94]*
Sei $|P| = n$. Dann gilt

- *FGM-k-Umweg ist für jedes $k \geq 3$ NP-hart.*
- *FGM-k-Umweg-$\mathbb{R}^2$ ist für jedes $k \geq 3$ NP-hart.*

Beweis: Die erste Aussage ist sogar richtig, wenn man zusätzlich fordert, daß alle Mengen F_i genau drei Elemente enthalten und daß alle Zielpunkte identifiziert werden, also $z_1 = ... = z_n$ gilt. Im Beweis führt man eine Reduktion auf das 3-dimensionale Matching durch, das NP-vollständig ist. Die zweite Aussage gilt ebenfalls für den Fall, daß alle Mengen F_i genau drei Elemente enthalten. Der Beweis dafür wird durch Rückführung auf das Exact-3-Cover-Problem gezeigt, welches NP-vollständig ist. Dieses lautet:

Gegeben: $X = \{x_1, ..., x_{3r}\}$, $C = \{c_1, ..., c_m\}$
mit $c_i \subseteq X$ und $\mid c_i \mid= 3$ für $i = 1, ..., m$.

Gesucht: $C' \subseteq C$ mit $\mid C' \mid= r$ und $\forall x\ \exists c \in C'$ mit $x \in c$.

Wir führen den Nachweis an einem typischen Beispiel. Die Einzelheiten sind in [Buchholz 94] ausgeführt.

Gegeben sind folgende Mengen.

$$\begin{aligned}
X &= \{x_1, x_2, x_3, x_4, x_5, x_6\} \\
C &= \{c_1, c_2, c_3, c_4\} \\
c_1 &= \{x_1, x_3, x_4\} \\
c_2 &= \{x_2, x_4, x_6\} \\
c_3 &= \{x_2, x_5, x_6\} \\
c_4 &= \{x_3, x_4, x_6\}
\end{aligned}$$

Gesucht ist eine Teilmenge C' von C mit zwei Elementen. Wir konstruieren die Start- und Zielorte im euklidischen Raum $\mathbb{R}^2$ und lösen dann FGM-3-Umweg für einen Umweg der Länge $c = 0$.

In einem 2-dimensionalen Gitter, in dem die Gitterpunkte horizontal und vertikal jeweils den Abstand eins voneinander haben, werden die Start- und Zielorte von sogenannten Trittbrettfahrern wie in Abbildung 3 positioniert. Ein Trittbrettfahrer zeichnet sich dadurch aus, daß sein Start- und Zielort genau übereinstimmen.

Die Spalten entsprechen den Elementen von X und die Zeilen den Elementen von C. Es werden die Start- und Zielorte von $|X||C|$ Trittbrettfahrern eingefügt, wobei die Orte von Trittbrettfahrer x genau auf dem Gitterpunkt der Zeile i liegen, wenn gilt: $x \in c_i$. Im anderen Fall $x \notin c_i$ sind die Start- und Zielorte um $\frac{1}{2}$ nach unten verschoben. Für den Umweg $c = 0$ können noch keine Fahrgemeinschaften gebildet werden. Wir fügen nun Spalten- und Zeilenfahrer ein, die jeweils die Trittbrettfahrer in einer Spalte oder Zeile mitnehmen (und sofort wieder absetzen) können.

In jeder Zeile i werden drei Zeilenfahrer eingefügt, die jeweils den Startort links von der Zeile i und den Zielort rechts von der Zeile i haben. Insgesamt werden also $3|C|$ Zeilenfahrer eingefügt. Dies führt dazu, daß in jeder Zeile entweder eine Fahrgemeinschaft der Größe drei oder zwei Fahrgemeinschaften der Größe drei gebildet werden können. Es kann kein Zeilenfahrer einen Zeilenfahrer oder einen Trittbrettfahrer einer anderen Zeile mitnehmen, da

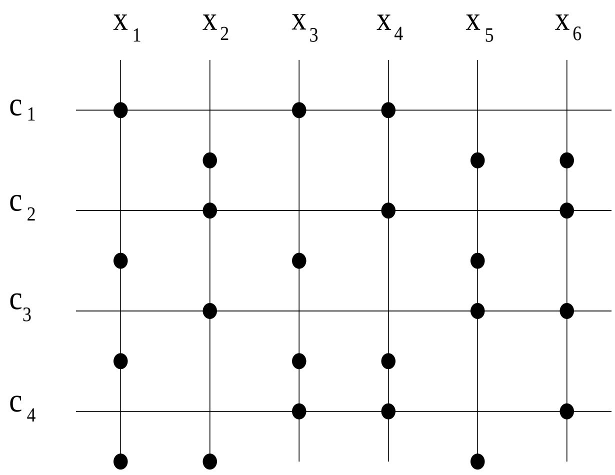

Abbildung 3: 24 Gittertrittbrettfahrer

dafür ein Umweg $c > 0$ gefahren werden müßte. Wenn in einer Zeile nur eine Fahrgemeinschaft gebildet wird, müssen die drei Zeilenfahrer zusammen fahren (sonst wäre es keine exakte Einteilung). Im anderen Fall werden genau die drei Trittbrettfahrer auf dieser Zeile mitgenommen. In Abbildung 4 sind die Start- und Zielorte für das Beispiel eingezeichnet. Man beachte, daß s_i der Startort und z_i der Zielort von den drei Zeilenfahrern aus Zeile i ist.

Wenn in der Zeile i zwei Fahrgemeinschaften gebildet werden, so korrespondiert dies genau damit, daß die Menge c_i in die Partition C' aufgenommen wird. Ohne weitere Fahrer würden in jeder Zeile zwei Fahrgemeinschaften gebildet analog dazu, daß $C' = C$ wäre. Es ist aber gefordert, daß jedes Element von X in genau einer Menge $c \in C'$ liegt. In jeder Spalte werden daher die Start- und Zielorte von $|C| - 1$ Spaltenfahrern und $|C| - 1$ Spaltenbeifahrern positioniert, so daß in einer exakten Partitionierung in Fahrgemeinschaften genau $|C| - 1$ Fahrgemeinschaften der Größe drei in jeder Spalte gebildet werden. Die Spaltenbeifahrer haben in jeder Spalte einen gemeinsamen Startort b_i oberhalb und einen gemeinsamen Zielort b_i' unterhalb des Basisgitters ($i = 1, \ldots, |X|$). Die Startorte s_{ij} der $|C| - 1$ Spaltenfahrer liegen oberhalb und die Zielorte z_{ij} unterhalb des Basisgitters, wobei die Start- und Zielorte genau so versetzt sind, daß sich die Spaltenfahrer nicht gegenseitig mitnehmen können ($j = 1, \ldots, |C| - 1$). In Abbildung 5 sind

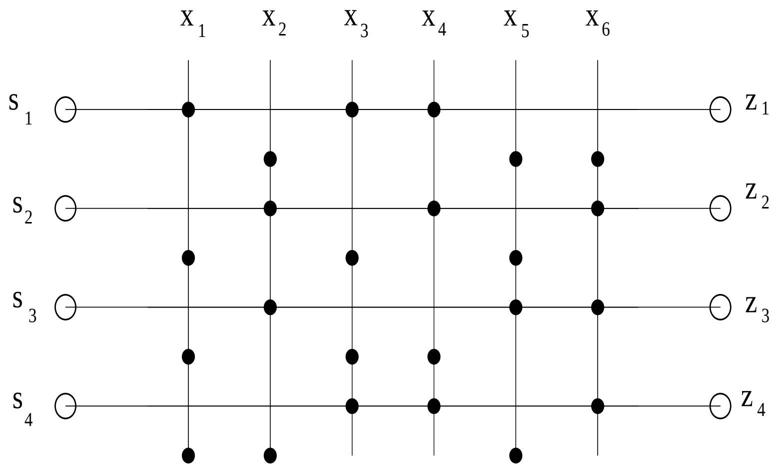

Abbildung 4: Start und Zielorte von $4 \cdot 3 = 12$ Zeilenfahrern

die Start- und Zielorte der Spaltenfahrer und der Spaltenbeifahrer aus dem Beispiel dargestellt.

Damit eine exakte Partitionierung in 3'er Fahrgemeinschaften zustande kommt, müssen in jeder Spalte genau $|C| - 1$ Fahrgemeinschaften gebildet werden. Es bleibt also in jeder Spalte genau ein Trittbrettfahrer aus dem Basisgitter übrig, der nicht von Spaltenfahrern mitgenommen wird. Dieser muß nun von Zeilenfahrern mitgenommen werden, damit 3'er Fahrgemeinschaften gebildet werden können.

In unserem Beispiel haben wir insgesamt die Start- und Zielorte von $4 \cdot 6 + 3 \cdot 4 + 3 \cdot 6 + 3 \cdot 6 = 72$ Fahrern eingefügt. In einer Lösung für FGM-3-Umweg werden in jeder Spalte drei Fahrgemeinschaften, in den Zeilen eins und drei jeweils zwei Fahrgemeinschaften und in den Zeilen zwei und vier jeweils eine Fahrgemeinschaft gebildet. Dies entspricht genau der Partitionierung $C' = \{c_1, c_3\}$, die eine Lösung für das Exact-3-Cover-Problem ist. Zwar sind Variationen innerhalb der Fahrgemeinschaften einer Spalte oder einer Zeile (falls zwei Fahrgemeinschaften gebildet werden) möglich, aber aufgrund der Konstruktion sind dies die einzigen Variationen.

Allgemein werden ausgehend von $|C||X|$ Start- und Zielorten im Basisgitter, genau $3|C| + (|C| - 1)|X| + (|C| - 1)|X| = 2|C||X| + 3|C| - 2|X|$ weitere Start- und Zielorte eingefügt, d.h. die Konstruktion ist in polynomieller Zeit

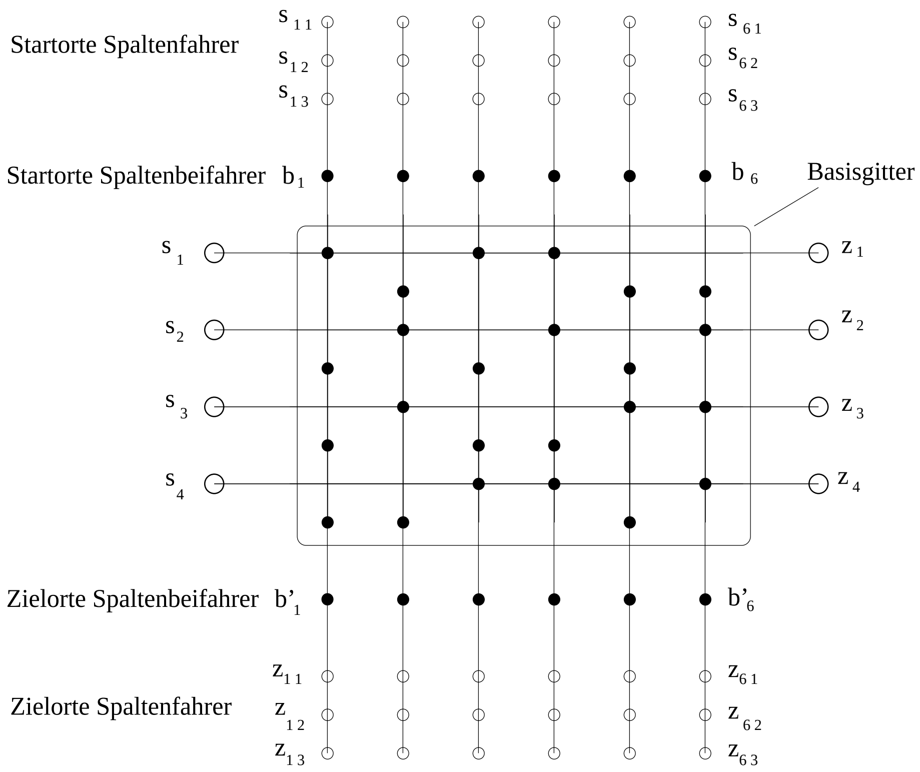

Abbildung 5: 18 Spaltenfahrer und 18 Spaltenbeifahrer

durchführbar.

■

Die Einführung von sogenannten Trittbrettfahrern, die nur auf einem Weg der Länge 0 mitgenommen werden, diente lediglich zur Vereinfachung. Natürlich können deren Start- und Zielorte diagonal so verschoben werden, daß sie nur von Spaltenfahrern (oder Zeilenfahrern) aus einer Spalte (oder einer Zeile) mitgenommen werden können. Damit hat man die NP-Härte auch für eine beliebige Umwegschranke.
Man kann nun weitere Sätze beweisen, die die Schwierigkeit des Problems zeigen, jeweils mehr als zwei Personen in Fahrgemeinschaften optimal zusammenzufassen. Beispielsweise liegt die NP-Härte auch vor, wenn der FGM-

Graph ein transitiver Graph ist $((u,v) \in E \wedge (v,w) \in E \Rightarrow (u,w) \in E)$ [Buchholz 94]. Ein Extremfall ist die (noch ungelöste) Frage, ob die NP-Härte bereits für den eindimensionalen Fall gilt, wenn also alle Start- und Zielpunkte auf einer Geraden angeordnet sind und jeder Zielpunkt rechts von seinem zugehörigen Startpunkt liegt. Wir vermuten, daß auch das Problem FGM-k-Kosten NP-hart ist, dies konnte aber noch nicht nachgewiesen werden.

Fahrgemeinschaften unter einer allgemeinen Bewertungsfunktion optimal zusammenzustellen, scheidet aus Effizienzgründen aus. Eine Alternative ist die Benutzung von Nachbarschaftsbeziehungen, die wir bereits in Abschnitt 0.2 erwähnten. Dabei kommt uns entgegen, daß Personen eher bereit sind, eine Fahrgemeinschaft zu bilden, wenn sie eine lange gemeinsame Fahrtzeit haben. D.h. die Wohn- und die Arbeitsorte müssen jeweils nah beieinander liegen. Die Eigenschaften einer Person (Wohn-, Arbeitsort, Arbeitszeiten, etc.) werden dazu entsprechend im euklidischen Raum skaliert. Jede Person wird als Punkt im euklidischen Raum in einer geometrischen Datenstruktur, die eine effiziente Nachbarschaftsuche unterstützt, aufgefasst. Da damit zu rechnen ist, daß häufig neue Personen hinzukommen bzw. bereits vermittelte Personen wieder aussteigen, muß eine dynamische Datenstruktur eingesetzt werden. Eine Kombination aus einer statischen geometrischen Datenstruktur (Quad-Bäume [Bentley, Finkel 74]) und einer dynamischen geometrischen Datenstruktur, die einen zu hohen Speicherbedarf hat (Grid File [Ottmann, Widmayer 93]), wurde entwickelt. Die sogenannten Grid-Quad-Bäume, die in [Buchholz 95] definiert sind, leisten bei dynamischem Verhalten im average case eine effiziente Nachbarschaftssuche.

Literatur

[Buchholz 94] Friedhelm Buchholz, Die Komplexität des Fahrgemeinschaften-Problems, Studienarbeit Nr. 1327, Universität Stuttgart, 1994

[Buchholz 95] Friedhelm Buchholz, Entwurf eines Systems zur Vermittlung von Fahrgemeinschaften, Diplomarbeit Nr. 1226, Universität Stuttgart, 1995

[Bentley, Finkel 74] J.L. Bentley and R.A. Finkel, Quad trees, A data structure for retrieval on composite keys, Acta Informatica 4, 1-9, 1974

[BMV] Bundesministeium für Verkehr, Anweisung für eine Straßendatenbank (ASB), Bonn, Referat StB10

[Dechter, Pearl 85] Rina Dechter and Judea Pearl, Generalized Best-First Search Strategies and the Optimality of A^* , Journal of the ACM, New York, 505 - 536, 1985

[Dijkstra 59] Dijkstra E.W., A note on two problems in connection with graphs, Numerical Mathematics, 1, 1959

[Dreyfus 69] Dreyfus S. E., An appraisal of some shortest-path algorithms, Operations Research 17, 395 - 412, 1969-1

[Minieka 74] Minieka Edward, On Computing Sets of Shortest Paths in a Graph, Communications of the ACM 17, Nr.6, 351 - 353, 1974

[OFFIS 93] V. Claus, J. Risau und M. Papenbrock, Projekt Fahrgemeinschaften - Abschlußbericht , Oldenburger Forschungs- und Entwicklungsinstitut für Informatik-Werkzeuge und -Systeme OFFIS, 1993

[Ottmann, Widmayer 93] T. Ottmann and P. Widmayer, Algorithmen und Datenstrukturen, Reihe Informatik Band 70, 1993

[Papadimitriou, Steiglitz 82] C. Papadimitriou, K. Steiglitz, Combinatorial optimization: algorithms and complexity, Englewood Cliffs, N.J. : Prentice-Hall, 1982

[Reinke 85] Volkmar Reinke, Fahrgemeinschaften im Berufsverkehr - Möglichkeiten und Grenzen der Förderung, Institut für Raumplanung Universität Dortmund (IRPUD), Dortmunder Beiträge zur Raumplanung, Band 39, 1985

[Reinkober 94] Norbert Reinkober, Fahrgemeinschaften und Mobilitätszentrale - Bestandteile eines zukunftsorientierten Öffentlichen Personennahverkehrs, Schriftenreihe für Verkehr und Technik Band 81, Erich Schmidt Verlag, 1994

[Yen 71] J. Y. Yen, Finding the K Shortest Loopless Paths in a Network, Management of Science, USA, Vol. 17, No. 11, 1971