

# Bachelorarbeit

## Optimale Zuordnungen auf Geometrischen Graphen

Sebastian Koletzko

Datum: 28.12.23

Fakultät für Mathematik  
Ruhr-Universität Bochum

Prof. Dr. Maike Buchin  
Dr. Daniela Kacso



**Eigenständigkeitserklärung:**

Hiermit erkläre ich, dass ich die heute eingereichte Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe. Bei der vorliegenden Bachelorarbeit handelt es sich um in Wort und Bild völlig übereinstimmende Exemplare. Ich erkläre weiterhin, dass die vorliegende Arbeit noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Witten, den 28.12.23

Sebastian Koletzko

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>6</b>
<b>2</b>	<b>Grundlagen</b>	<b>8</b>
2.1	Abstände auf geometrischen Graphen . . . . .	9
2.2	Lokale Optimierungen . . . . .	13
2.3	Optimale Graph-Zuordnungen . . . . .	14
<b>3</b>	<b>Ein polynomieller Algorithmus</b>	<b>16</b>
3.1	Einschränkungen an die Graph-Zuordnungen . . . . .	16
3.1.1	Erste Einschränkung . . . . .	17
3.1.2	Zweite Einschränkung . . . . .	18
3.2	Ansätze und Beschreibung eines polynomiellen Algorithmus . . . . .	19
3.2.1	$\varepsilon$ -Platzierungen der Knoten . . . . .	19
3.2.2	Schritt 1 . . . . .	19
3.2.3	Schritt 2 . . . . .	20
3.2.4	Existenz einer (schwachen) $\varepsilon$ -Platzierung von $G_1$ . . . . .	21
3.2.5	Schritt 3 . . . . .	22
3.2.6	Optimale (schwache) Kanten-Platzierungen . . . . .	23
3.2.7	Konstruktion einer optimalen $\varepsilon$ -Platzierung von $G_1$ . . . . .	24
3.2.8	Schritt 4 . . . . .	25
3.3	Ein Beispiel . . . . .	29
<b>4</b>	<b>Komplexität</b>	<b>34</b>
4.0.1	NP-Schwerheit von $\text{Min-Sum}_{(w)F}$ . . . . .	34
<b>5</b>	<b>Ausblick</b>	<b>41</b>



# 1 Einleitung

Die geometrische Graphentheorie ist ein mit der topologischen Graphentheorie verwandter Zweig der Graphentheorie, welcher sich mit Graphen beschäftigt, deren Knoten und Kanten mit geometrischen Objekten identifiziert werden. Graphen, denen eine bestimmte Darstellung in einem Anschauungsraum - zum Beispiel dem euklidischen Raum - zugewiesen wird, bezeichnen wir hier auch als eingebettete Graphen.

In dieser Arbeit wollen wir uns speziell mit geometrischen Graphen beschäftigen. Hierbei handelt es sich um eingebettete Graphen, bei denen ein Knoten jeweils als ein Punkt und eine Kante jeweils als (zwei adjazente Punkte verbindendes) Geradenstück in der euklidischen Ebene dargestellt wird. Geometrische Graphen eignen sich in der Praxis insbesondere zur Darstellung von Netzwerken wie beispielsweise von Straßennetzen. Da wir geometrische Graphen als Objekte in einem metrischen Raum betrachten, bietet es sich grundsätzlich an, sie unter Verwendung von bekannten sowie auch neuen Methoden miteinander zu vergleichen. Dabei wollen wir Ansätze zur Messung der Ähnlichkeit zwischen geometrischen Graphen etablieren, um im Anwendungsfall konkrete Aussagen über die Güte der durch den Graphen beschriebenen Repräsentation (zum Beispiel von einem bestimmten Netzwerk) treffen zu können. Dafür bedienen wir uns verschiedener Abstandsmaße zur Messung der Ähnlichkeit von geometrischen Objekten.

Für den Vergleich von geometrischen Graphen existieren dabei bereits einige Abstandsmaße in der Literatur. Diese haben allgemein unterschiedliche Eigenschaften bezüglich der Art, wie sie den Abstand zwischen geometrischen Graphen messen und welche Eigenschaften der Graphen und ihrer Beziehung zueinander in welchem Maße zum berechneten Abstand zwischen ihnen beitragen (siehe dafür zum Beispiel Akitaya et. al [2], Seite 2-3). Die metrischen Eigenschaften eines Abstandsmaßes finden dabei in dieser Ausarbeitung zwar keine besondere Erwähnung, werden aber grundsätzlich in der Literatur untersucht.

Eine zentrale Fragestellung ist mitunter, wie man neue Abstandsmaße auf geometrischen Graphen definieren kann und unter welchen Bedingungen sich diese auch für potentiell komplexe Graphen noch effizient berechnen lassen. Idealerweise erlangen wir bei der Auseinandersetzung mit einem Abstandsmaß eine gewisse Heuristik oder Anschauung dafür, wie ein Abstandsmaß die Ähnlichkeit zwischen Graphen bestimmt.

Wir werden zunächst kurz das Abstandsmaß behandeln, welches durch Akitaya et. al [2] eingeführt und untersucht wurde. Dieses ist eng mit dem in dieser Arbeit primär behandeltem Abstandsmaß verwandt und hat die Besonderheit, dass neben der geo-

metrischen Eigenschaften der Graphen auch ihre jeweilige (abstrakte) Konnektivität einen bedeutenden Einfluss auf das Ausmaß ihrer durch das Abstandsmaß beschriebenen Ähnlichkeit hat. Dafür etablieren wir initial in Kapitel 2 die nötigen Begriffe und motivieren die Einführung des Abstandsmaßes, mit welchem wir uns für den Rest der Arbeit befassen werden. Dieses Maß liefert uns einen weiteren Betrachtungswinkel, um konkretere Aussagen über die Ähnlichkeit zwischen geometrischen Graphen treffen zu können. Dabei befassen wir uns in Kapitel 3 mit der algorithmischen Herleitung und Berechnung dieses Abstandsmaßes auf Bäumen und in Kapitel 4 befassen wir uns mit der Komplexität des Abstandsmaßes für allgemeine, geometrische Graphen.

## 2 Grundlagen

In diesem Kapitel wollen wir die grundlegenden Definitionen wiederholen, die für das in Kapitel 3 behandelte Abstandsmaß auf geometrischen Graphen bedeutsam sind. Wir folgen dabei im Wesentlichen den Literaturquellen [1], [2] und [3].

Wir betrachten  $\mathbb{R}^n$  grundsätzlich als metrischen Raum, versehen mit der euklidischen Norm  $(\|\cdot\|)$ .

### Definition 2.1

Ein *Weg* ist eine stetige Abbildung  $\phi : [0, 1] \rightarrow \mathbb{R}^n$  von einem geschlossenen Intervall  $I = [0, 1]$  nach  $\mathbb{R}^n$ . Das Bild von  $I$  unter  $\phi$  nennen wir auch die *Kurve* von  $\phi$ .

Wir bezeichnen  $\phi(0)$  als den *Startpunkt*,  $\phi(1)$  als den *Endpunkt* und die Menge  $\{\phi(0), \phi(1)\}$  als die *Randpunkte* des Weges. Dabei ist grundsätzlich  $\phi(0) \neq \phi(1)$  anzunehmen.

Ein *einfacher Weg* ist ein Weg, welcher injektiv (auf  $I$ ) ist.

### Definition 2.2

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

Unter einem *eingebetteten Graphen* (*embedded graph*) verstehen wir den Graphen  $G$  zusammen mit seiner *Einbettung*  $\omega : G \rightarrow \mathbb{R}^n$ , welche jedem Knoten  $v \in V$  einen eindeutigen Punkt  $v_0 \in \mathbb{R}^n$  zuordnet und jeder Kante  $e = \{u, v\} \in E$  einen einfachen Weg  $\phi_e : [0, 1] \rightarrow \mathbb{R}^n$  mit den Randpunkten  $\omega(u)$  und  $\omega(v)$  zuordnet.

Ist jede Kante von  $G$  in den euklidischen Raum als Geradenstück mit den zur jeweiligen Kante inzidenten Knoten als Randpunkte realisiert, so nennen wir  $G$  einen *geradlinig eingebetteten Graphen* (*straight-line embedded graph*).

Dabei verlangen wir bezüglich der Einbettung grundsätzlich nicht, dass die eingebetteten Kanten von  $G$  *kreuzungsfrei* sind, das heißt Kanten schneiden sich mitunter auch in mehr Punkten als in ihren gemeinsamen Knoten.

Einen in die euklidische Ebene geradlinig eingebetteten Graphen bezeichnen wir als *geometrischen Graphen*.

Die konkrete Parametrisierung einer Kante ist dabei für unsere weiteren Betrachtungen nicht sonderlich relevant. Wovon wir aufgrund der Injektivität jedoch stets ausgehen ist, dass die zugrundeliegende Parametrisierung einer Kurve streng monoton ist und damit nicht die Orientierung ändert. Im weiteren Verlauf werden wir die konkrete Ein-



bettung einer Kante und ihre Repräsentation als Kurve in Form eines Geradenstücks in der euklidischen Ebene daher mitunter ohne explizite Unterscheidung betrachten.

### Definition 2.3

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

Für einen Knoten  $v \in V$  definieren wir seinen  $\varepsilon$ -Ball  $B_\varepsilon(v)$  als die Menge  $\{x \in \mathbb{R}^n : \|v - x\| \leq \varepsilon\}$  und für eine Kante  $e \in E$  definieren wir ihren  $\varepsilon$ -Schlauch  $T_\varepsilon(e)$  als die Menge  $\{x \in \mathbb{R}^n : \min_{a \in e} \|a - x\| \leq \varepsilon\}$ .

Wir unterscheiden grundsätzlich nicht zwischen dem Graphen bezüglich seiner inherenten abstrakten Beschreibung und der durch seine Einbettung induzierten Darstellung. Erwähnen wir zum Beispiel die Kante eines eingebetteten Graphen, so beziehen wir uns stets auf die abstrakte Beschreibung sowie das durch die Einbettung zugeordnete geometrische Objekt zugleich.

### Definition 2.4

Sei  $G$  ein geradlinig eingebetteter Graph.

Ein *einfacher Weg* (*simple path*) in  $G$  ist eine stetige Abbildung  $W : [0, 1] \rightarrow G$ , welche die Knoten und Kanten von  $G$  maximal einmal besucht. Falls die Kanten von  $G$  kreuzungsfrei eingebettet sind - also  $G$  planar ist - so ist  $W$  also injektiv. Prinzipiell und insbesondere in Bezug auf die hier erwähnte *Stetigkeit* verstehen wir die durch die Einbettung von  $G$  in den euklidischen Raum induzierte Punktmenge als *topologischen Raum*  $G \subset \mathbb{R}^n$ .

Die Kurve - beziehungsweise das Bild von  $W$  - ist anschaulich die Zeichnung einer Teilkomponente bestehend aus Kanten beziehungsweise Teilkanten von  $G$ .

Für einen Weg  $W$  in einem eingebetteten Graphen definieren wir seine *Länge* als die Anzahl der Geradenstücke, aus denen sich  $W$  zusammensetzt. Wir kennzeichnen die Länge von  $W$  durch  $|W|$ . Ist  $m$  die Anzahl der Kanten von  $G$  und ist  $W$  einfach, so gilt  $|W| \leq m$ .

## 2.1 Abstände auf geometrischen Graphen

Seien  $G_1 = (V_1, E_1)$  und  $G_2 = (V_2, E_2)$  für den gesamten weiteren Verlauf geometrische Graphen mit  $n_1 = |V_1|, m_1 = |E_1|, n_2 = |V_2|$  und  $m_2 = |E_2|$ . Dabei bezeichnet  $|V_1|, |V_2|$  jeweils die Anzahl der Knoten und  $|E_1|, |E_2|$  jeweils die Anzahl der Kanten von  $G_1$  beziehungsweise von  $G_2$ .

**Definition 2.5**

Wir nennen eine Abbildung  $s : G_1 \rightarrow G_2$  eine *Graph-Zuordnung* (*graph mapping*), falls

- 1)  $s$  jeden Knoten  $v \in V_1$  auf einen Punkt innerhalb von  $G_2$  abbildet und
- 2)  $s$  jede Kante  $\{u, v\} \in E_1$  auf einen einfachen Weg in  $G_2$  abbildet, wobei  $s(u)$  und  $s(v)$  die Randpunkte des Weges sind.

Eine Graph-Zuordnung definiert somit eine stetige Abbildung (siehe 2.4 bezüglich der Stetigkeit) von  $G_1$  nach  $G_2$  und ist im Allgemeinen weder injektiv noch surjektiv.

Um zwei geometrische Graphen zu vergleichen, werden wir den Abstand zwischen den Bildern und Urbildern beliebiger Graph-Zuordnungen zwischen ihnen messen. Dafür bedienen wir uns eines gängigen Abstandsmaßes für Wege beziehungsweise Kurven im euklidischen Raum.

**Definition 2.6**

Für zwei Wege  $f, g : [0, 1] \rightarrow \mathbb{R}^n$  definieren wir den *Fréchet-Abstand* (*Fréchet distance*) von  $f$  und  $g$  als

$$\delta_F(f, g) = \inf_{\sigma: [0,1] \rightarrow [0,1]} \max_{t \in [0,1]} \|f(t) - (g(\sigma(t)))\|,$$

wobei sich  $\sigma$  über alle orientierungserhaltenen Homöomorphismen erstreckt.

Wir definieren den *schwachen Fréchet-Abstand* (*weak Fréchet distance*) von  $f$  und  $g$  als

$$\delta_{wF}(f, g) = \inf_{\alpha, \beta: [0,1] \rightarrow [0,1]} \max_{t \in [0,1]} \|f(\alpha(t)) - (g(\beta(t)))\|,$$

wobei sich  $\alpha$  und  $\beta$  über alle stetigen Abbildungen erstrecken, welche die Randpunkte fixieren, also  $\alpha(0) = \beta(0) = 0$  und  $\alpha(1) = \beta(1) = 1$ .

Die Klammer-Notation (zum Beispiel  $\delta_{(w)F}$ ) benutzen wir zugunsten der Kompaktheit im weiteren Verlauf, wenn wir im jeweiligen Kontext den Fréchet-Abstand sowie den schwachen Fréchet-Abstand zugleich adressieren wollen. Wir schreiben in diesem Fall auch (*schwacher*) *Fréchet-Abstand*.

Zur Anschauung des Fréchet-Abstands stelle man sich  $f$  und  $g$  als Kurven im  $\mathbb{R}^2$  vor. Der (euklidische) Abstand zwischen  $f$  und  $g$  zum Zeitpunkt  $t_0 \in [0, 1]$  entspricht gerade der Länge der Strecke mit den Randpunkten  $f(t_0)$  und  $g(t_0)$ . Bei dem Fréchet-Abstand dürfen wir eine Kurve reparametrisieren, um den maximal angenommenen euklidischen Abstand zwischen  $f$  und  $g$  zu minimieren. Da der (schwache) Fréchet-Abstand

symmetrisch ist (siehe dazu [3], Seite 77), spielt die genaue Wahl der Kurve für die Reparametrisierung keine Rolle. Die Reparametrisierung selbst muss hingegen für den Fréchet-Abstand stets monoton und orientierungserhaltend sein sowie die Randpunkte fixieren.

Für den schwachen Fréchet-Abstand wird die Bedingung an die Monotonie der Reparametrisierung fallen gelassen. Zusätzlich dürfen beide Kurven entsprechend reparametrisiert werden. Wir bezeichnen allgemein einen solchen maximal angenommen Abstand zwischen zwei geometrischen Objekten auch als *Flaschenhals-Abstand*.

Ein wichtiges Ergebnis aus Alt et. al [3] (Seiten 82-84) ist, dass wir den (schwachen) Fréchet-Abstand für polygonale Kurven - d.h. Kurven, welche sich aus der Konkate-nation von Geradenstücken zusammensetzen - in polynomieller Speicher - und Laufzeitkomplexität berechnen können. Ein weiteres Ergebnis ist, dass wir zur Berechnung des (schwachen) Fréchet-Abstands zwischen zwei polygonalen Kurven ihre jeweilige gegebene Parametrisierung nicht explizit kennen müssen (siehe Alt et. al [3], Seiten 78-82). Es lassen sich bereits bei der Betrachtung der jeweiligen Kurven in der euklidischen Ebene Aussagen über ihren (schwachen) Fréchet-Abstand treffen. Abbildung 1 liefert dazu ein paar Beispiele für den (schwachen) Fréchet-Abstand zwischen polygonalen Kurven, da uns dieses Szenario im weiteren Verlauf noch häufiger begegnen wird. Aufgrund dieser Tatsache werden wir auch mitunter Aussagen über den (schwachen) Fréchet-Abstand zwischen zwei Kurven treffen, ohne ihre jeweilige Parametrisierung zu erwähnen. Grundsätzlich ist die initiale Orientierung der Kurven dabei nicht unerheblich und es ist daher stets anzunehmen, dass besagte Orientierungen für den jeweiligen Fall zugunsten des (schwachen) Fréchet-Abstands eingerichtet sind.

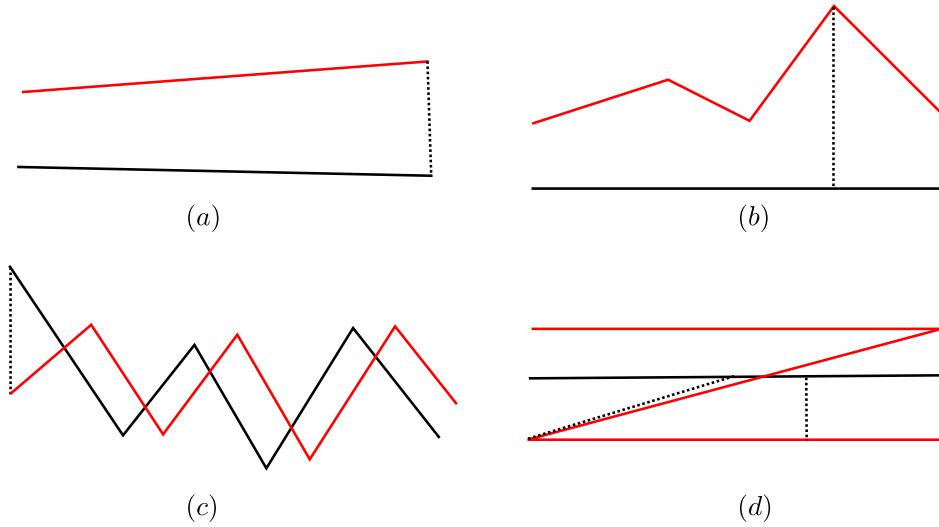


Abbildung 1: Beispiele für den (schwachen) Fréchet-Abstand zwischen jeweils einer in rot und einer in schwarz gezeichneten Kurve. Die gestrichelten Linien repräsentieren dabei die (schwachen) Fréchet-Abstände zwischen den Kurven. Für Die Abbildungen (a), (b), (c) ist der Fréchet-Abstand und der schwache Fréchet-Abstand identisch. Für zwei Geradenstücke wie in Abbildung (a) tritt der Flaschenhals-Abstand in den meisten Fällen genau zwischen zwei Randpunkten auf (abhängig von eventueller Kolinearität und der Lage der Randpunkte zueinander). Wäre jeweils eine der beiden Kurven anders orientiert, so würde der (schwache) Fréchet-Abstand entsprechend deutlich größer ausfallen. In Abbdilung (d) kennzeichnet die längere gestrichelte Linie den Fréchet-Abstand und die kürzere gestrichelte Linie den schwachen Fréchet-Abstand der beiden Kurven.

Quelle: In Anlehnung an Alt et. al [3] (Seite 83, Figure 7) bezüglich Abbildung (d), ansonsten eigene Darstellung.

Aufgrund seiner Eigenschaften eignet sich der (schwache) Fréchet-Abstand insbesondere zum Messen der Ähnlichkeit von geometrischen Graphen. In diesem Sinne definieren wir nun das zentrale Abstandsmaß aus Akitaya et. al [2].

### Definition 2.7

Wir definieren den *gerichteten Graph-Abstand*  $\vec{\delta}_G$  als

$$\vec{\delta}_G(G_1, G_2) = \inf_{s: G_1 \rightarrow G_2} \max_{e \in E_1} \delta_F(e, s(e))$$

und den *gerichteten schwachen Graph-Abstand*  $\vec{\delta}_{wG}$  als

$$\vec{\delta}_{wG}(G_1, G_2) = \inf_{s: G_1 \rightarrow G_2} \max_{e \in E_1} \delta_{wF}(e, s(e)),$$

wobei  $s$  sich über alle Graph-Zuordnungen erstreckt.

Neben den geometrischen Eigenschaften hat insbesondere auch die (abstrakte) Kon-

nektivität von  $G_1$  und  $G_2$  signifikante Implikationen für das konkrete Ausmaß des gerichteten (schwachen) Graph-Abstands (siehe Abbildung 2 für ein Beispiel).

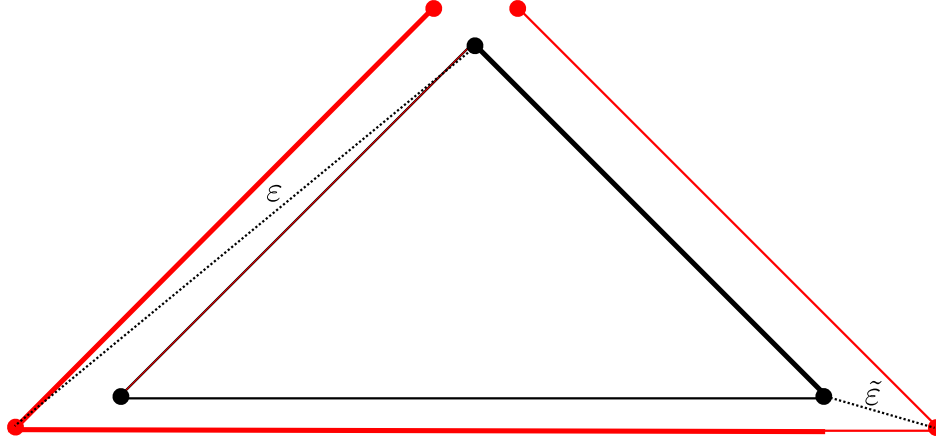


Abbildung 2: Skizzieren wir zwei Graphen  $G_1, G_2$ , so zeichnen wir die Knoten und Kanten von  $G_1$  stets in schwarz und die Knoten und Kanten von  $G_2$  stets in rot. Aufgrund der Konnektivität von  $G_1$  induziert die Zuordnung zweier Kanten von  $G_1$  auf  $G_2$  das Bild der ausbleibenden Zuordnung für die dritte Kante von  $G_1$ . Die stärker gezeichnete rote Kurve sei hier das Bild einer solchen Graph-Zuordnung von  $G_1$  nach  $G_2$ . Der Flaschenhals-Abstand wird hier durch den (schwachen) Fréchet-Abstand zwischen den stärker gezeichneten Kurven angenommen und ist durch  $\varepsilon$  gekennzeichnet. Damit ist  $\vec{\delta}_{(w)F}(G_1, G_2) = \varepsilon$ . Andersherum tritt ein so großer Flaschenhals-Abstand zum Vergleich nicht auf. Es gilt  $\vec{\delta}_{(w)F}(G_2, G_1) = \tilde{\varepsilon}$ . Im Vergleich zu der *Traversal Distance* von  $G_1$  und  $G_2$  - einem alternativen Abstandsmaß für eingebetteten Graphen - fällt  $\vec{\delta}_{(w)F}(G_1, G_2)$  hier aufgrund der jeweiligen Konnektivität von  $G_1$  und  $G_2$  deutlich größer aus (siehe dazu Akitaya et. al [2], Seite 6).

Quelle: In Anlehnung an Akitaya et. al [2], Seite 6, Figure 2 (c).

Ähnlich wie der (schwache) Fréchet-Abstand beschreibt auch der gerichtete (schwache) Graph-Abstand einen Flaschenhals-Abstand.

## 2.2 Lokale Optimierungen

Generell können für einen konkreten Fall viele Graph-Zuordnungen  $s : G_1 \rightarrow G_2$  existieren, die den gerichteten (schwachen) Graph-Abstand einhalten. Im Anwendungsfall kann man sich zum Beispiel mehrere Rekonstruktionen  $G_{1,1}, \dots, G_{1,n}$  eines Netzwerkes  $G_2$  vorstellen. Anschließend stelle man sich nun die Frage, welche der Rekonstruktionen dem ursprünglichen Netzwerk  $G_2$  am ähnlichsten ist beziehungsweise dieses entsprechend eines konkreten Abstandsmaßes für Graphen am besten beschreibt. Der gerichtete (schwache) Graph-Abstand zwischen einem  $G_{1,i}$  ( $i \in \{1, \dots, n\}$ ) und  $G_2$  beschreibt dabei in gewisser Weise eine obere Schranke für die größte Abweichung zwischen einer Rekonstruktion  $G_{1,i}$  und  $G_2$ . Unter Umständen haben evtl. sogar alle  $G_{1,i}$  denselben Flaschenhals-Abstand zu  $G_2$ , da sie dieselbe entscheidende Abweichung bei der Re-

konstruktion von  $G_2$  haben. Insbesondere interessieren wir uns hier nun entsprechend für die lokale Güte der jeweiligen Rekonstruktion, welche sich durch den gerichteten (schwachen) Graph-Abstand nicht hinreichend beschreiben lässt. Für ein anschauliches Beispiel möchten wir an dieser Stelle zusätzlich auf Buchin et. al [1] (Figure 1) verweisen.

Dies motiviert die Einführung eines weiteren Abstandsmaßes auf Graphen, welches auch lokale Optimierungen bei den Zuordnungen der Kanten unter einer Graph-Zuordnung mit in Betracht zieht.

Zuvor merken wir an, dass im Allgemeinen eine Graph-Zuordnung  $\tilde{s} : G_1 \rightarrow G_2$ , die jede Kante  $e \in E_1$  optimal bezüglich des (schwachen) Fréchet-Abstands auf  $G_2$  abbildet - so dass  $\delta_{(w)F}(e, \tilde{s}(e)) \leq \delta_{(w)F}(e, s(e))$  gilt - nicht existiert. Wir verweisen dazu auf Abbildung 3 für ein konkretes Beispiel.

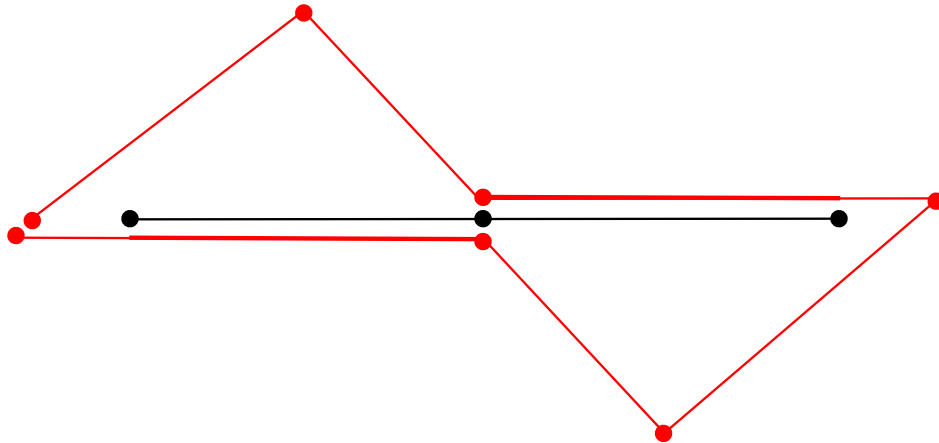


Abbildung 3: Eine Graph-Zuordnung von  $G_1$  nach  $G_2$  kann hier immer nur genau eine einzige Kante von  $G_1$  bezüglich des (schwachen) Fréchet-Abstands optimal zuordnen. Für die beiden Kanten von  $G_1$  sind die Bilder ihrer optimalen Zuordnungen dick gezeichnet. Allerdings existiert offensichtlich keine Graph-Zuordnung, die beiden lokal optimalen Kanten-Zuordnungen simultan realisieren kann.

Quelle: In Anlehnung an Figure 2 aus Buchin et. al [1].

Alternativ bietet es sich an, die summierten (schwachen) Fréchet-Abstände zwischen den Kanten in  $G_1$  und ihren Bildern in  $G_2$  unter der Graph-Zuordnung zu optimieren.

## 2.3 Optimale Graph-Zuordnungen

Damit widmen wir uns nun dem Abstandsmaß, welches für uns im weiteren Verlauf das Optimalitätskriterium zur Charakterisierung optimaler Zuordnungen auf Graphen beschreibt.

**Definition 2.8**

Wir definieren  $Min\text{-}Sum_{dist}(G_1, G_2)$  als

$$\min_{s:G_1 \rightarrow G_2} \sum_{e \in E_1} dist(e, s(e)),$$

wobei sich  $s$  über alle Graph-Zuordnungen von  $G_1$  nach  $G_2$  erstreckt und  $dist$  ein unspezifisches Abstandsmaß (für den Vergleich von Wegen) in der euklidischen Ebene ist.

Wählen wir den (schwachen) Fréchet-Abstand als zugrundeliegendes Abstandsmaß, so definieren wir entsprechend  $Min\text{-}Sum_{(w)F}(G_1, G_2)$  als

$$\min_{s:G_1 \rightarrow G_2} \sum_{e \in E_1} \delta_{(w)F}(e, s(e)).$$

Ferner bezeichnen wir mit

$$Sum_{(w)F}(s) = \sum_{e \in E_1} \delta_{(w)F}(e, s(e))$$

die Summe der (schwachen) Fréchet-Abstände zwischen den Kanten in  $G_1$  und ihren durch  $s$  zugeordneten Wegen in  $G_2$ . Wir nennen  $s$  eine  $Min\text{-}Sum_{(w)F}$  Zuordnung, falls

$$Sum_{(w)F}(s) = Min\text{-}Sum_{(w)F}(G_1, G_2).$$

### 3 Ein polynomieller Algorithmus

In diesem Kapitel werden wir zur Berechnung des  $\text{Min-Sum}_{(w)F}$  Abstands den Algorithmus aus [1] unter Berücksichtigung einiger relevanter Begriffe und Ergebnisse aus [2] herleiten und diesen Algorithmus schließlich anhand eines konkreten Beispiels demonstrieren. Sei  $G_1$  von nun an ein kreisfreier und zusammenhängender Graph, also ein *Baum*.

#### 3.1 Einschränkungen an die Graph-Zuordnungen

Für die Berechnung von  $\text{Min-Sum}_{(w)F}(G_1, G_2)$  kommen definitionsgemäß (siehe 2.8) grundsätzlich alle möglichen Graph-Zuordnungen von  $G_1$  nach  $G_2$  in Frage. Im weiteren Verlauf werden wir die Menge der Graph-Zuordnungen jedoch mit zwei zusätzlichen Einschränkungen versehen. Die erste Einschränkung ist dabei optional aber sinnvoll. Die zweite Einschränkung ist notwendig, um die polynomielle Laufzeit des Algorithmus zu gewährleisten.

Bevor wir die Einschränkungen im Detail motivieren und beschreiben, wollen wir zunächst weitere Begriffe zur Charakterisierung der Eigenschaften von Graph-Zuordnungen bezüglich der (schwachen) Fréchet-Abstände zwischen den Kanten von  $G_1$  und ihren Bildern in  $G_2$  definieren.

##### Definition 3.1

Sei  $\varepsilon \geq 0$ . Eine  $\varepsilon$ -Platzierung ( *$\varepsilon$ -placement*) eines Knotens  $v \in V_1$  ist eine maximal zusammenhängende Komponente von  $G_2$  eingeschränkt auf  $B_\varepsilon(v)$ . Eine  $\varepsilon$ -Platzierung einer Kante  $e = \{u, v\} \in E_1$  ist ein Weg  $W$  in  $G_2$ , welcher eine  $\varepsilon$ -Platzierung  $p_u$  von  $u$  mit einer  $\varepsilon$ -Platzierung  $p_v$  von  $v$  so verbindet, dass  $\delta_F(e, W) \leq \varepsilon$ .

In diesem Fall nennen wir  $p_u$  und  $p_v$  *voneinander erreichbar* (*mutually reachable*).

Eine  $\varepsilon$ -Platzierung von  $G_1$  ist eine Graph-Zuordnung  $s : G_1 \rightarrow G_2$ , so dass  $s$  jede Kante  $e \in G_1$  auf eine  $\varepsilon$ -Platzierung von  $e$  abbildet.

Eine *schwache  $\varepsilon$ -Platzierung* (*weak  $\varepsilon$ -placement*) einer Kante  $e = \{u, v\} \in E_1$  ist ein Weg  $W$  in  $G_2$ , welcher eine  $\varepsilon$ -Platzierung  $p_u$  von  $u$  mit einer  $\varepsilon$ -Platzierung  $p_v$  von  $v$  so verbindet, dass  $\delta_{wF}(e, W) \leq \varepsilon$ .

In diesem Fall nennen wir  $p_u$  und  $p_v$  *schwach voneinander erreichbar* (*weakly mutually reachable*).

Eine schwache  $\varepsilon$ -Platzierung von  $G_1$  ist eine Graph-Zuordnung  $s : G_1 \rightarrow G_2$ , so dass  $s$  jede Kante  $e \in G_1$  auf eine schwache  $\varepsilon$ -Platzierung von  $e$  abbildet. Siehe Abbildung 4 für ein Beispiel.



Synonym für  $\varepsilon$ -Platzierung eines Knotens beziehungsweise einer Kante benutzen wir mitunter auch die Begriffe *Knoten-Platzierung* und *Kanten-Platzierung*.

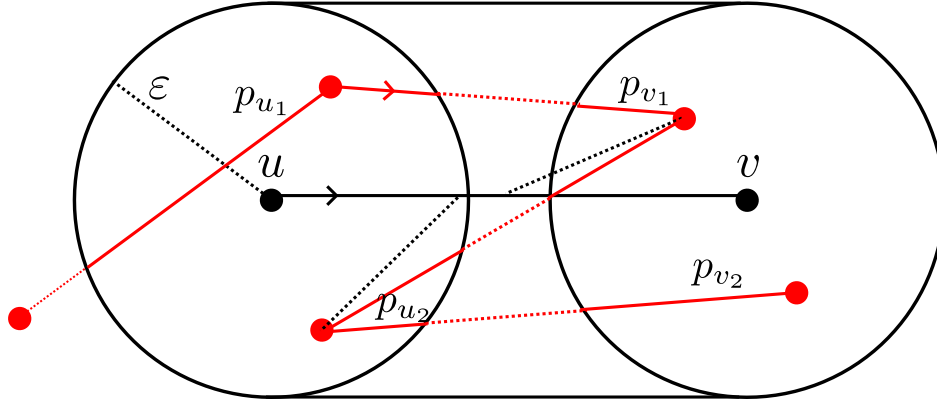


Abbildung 4:  $B_\varepsilon(u)$  enthält zwei  $\varepsilon$ -Platzierungen von  $u$ ,  $p_{u_1}$  und  $p_{u_2}$ . Analog sind  $p_{v_1}$  und  $p_{v_2}$  die  $\varepsilon$ -Platzierungen von  $v$  innerhalb von  $B_\varepsilon(v)$ . Jede Knoten-Platzierung enthält hier genau einen Knoten von  $G_2$ . Hierbei sind die Paare von Knoten-Platzierungen  $(p_{u_1}, p_{v_1})$ ,  $(p_{u_2}, p_{v_1})$  und  $(p_{u_2}, p_{v_2})$  voneinander erreichbar beziehungsweise schwach voneinander erreichbar. Die Bilder möglicher Wege, welche die besagten Paare verbinden, könnten zum Beispiel die gezeichneten Kanten in  $G_2$  zwischen den (eindeutigen) Knoten innerhalb der entsprechenden Platzierungen sein.

Die schwarz-gepunkteten Geradenstücke haben jeweils Länge  $\varepsilon$  und veranschaulichen, dass  $p_{u_1}$  und  $p_{v_2}$  lediglich schwach voneinander erreichbar sind. Denn sei  $W$  ein Weg auf  $\{u, v\}$  von  $u$  nach  $v$  und  $X$  ein einfacher Weg auf  $G_2$  von  $p_{u_1}$  nach  $p_{v_2}$  (die initiale Orientierung von  $W$  und  $X$  ist durch jeweils einen Pfeil gekennzeichnet). Sei  $t_0 \in [0, 1]$  so gewählt, dass  $X(t_0)$  auf dem Knoten in  $p_{v_1}$  liegt. Damit  $\|W(t_0) - X(t_0)\| \leq \varepsilon$  ist, muss  $W$  den  $\varepsilon$ -Ball um  $u$  verlassen. Sobald  $X$  am Knoten innerhalb von  $p_{u_2}$  angekommen ist, muss  $W$  zur Einhaltung des  $\varepsilon$ -Abstands auf  $\{u, v\}$  offensichtlich die Orientierung gewechselt haben, da der Abstand zwischen dem Knoten bei  $p_{u_2}$  und einem beliebigen Punkt auf  $\{u, v\}$  außerhalb von  $B_\varepsilon(u)$  größer als  $\varepsilon$  ist.

Quelle: Eigene Darstellung.

### 3.1.1 Erste Einschränkung

Wir haben  $\text{Min-Sum}_{dist}$  allgemein vor dem Hintergrund motiviert, optimale Graph-Zuordnungen auf geometrischen Graphen zu betrachten, die bereits einen Flaschenhals-Abstand wie den gerichteten (schwachen) Graph-Abstand einhalten.

Um diesen Zusammenhang zu bewahren, werden wir die in Betrachtung zu ziehenden Graph-Zuordnungen zunächst auf die Menge

$$\{s : G_1 \rightarrow G_2 \mid s \text{ ist eine (schwache) } \varepsilon\text{-Platzierung von } G_1\}$$

einschränken.

Generell könnten wir die Bedingung abschwächen und  $\varepsilon > \vec{\delta}_{(w)F}(G_1, G_2)$  annehmen, für größere  $\varepsilon$  vergrößert sich entsprechend auch die Menge der in Betrachtung zu ziehenden Graph-Zuordnungen, wodurch  $\text{Min-Sum}_{(w)F}(G_1, G_2)$  potentiell kleiner wird.

### 3.1.2 Zweite Einschränkung

Da wir hier zur Berechnung der Abstände zwischen einer Kante  $\{u, v\} \in E_1$  und ihrem Bild  $s(\{u, v\})$  unter einer Graph-Zuordnung  $s$  den (schwachen) Fréchet-Abstand verwenden, hat die Wahl der Bilder  $s(u)$  und  $s(v)$  (innerhalb der entsprechenden  $\varepsilon$ -Platzierungen von  $u$  und  $v$ ) eine direkte Auswirkung auf die Summe der Abstände. Anders ausgedrückt: Im Gegensatz zu  $\vec{\delta}_{(w)F}(G_1, G_2)$  ist  $\text{Min-Sum}_{(w)F}(G_1, G_2)$  nicht invariant unter dem Freiheitsgrad, den  $s$  bei der Wahl der Bildpunkte  $s(u)$  und  $s(v)$  unter Einhaltung des (schwachen) Fréchet-Abstands hat. Siehe Abbildung 5 für ein konkretes Beispiel.

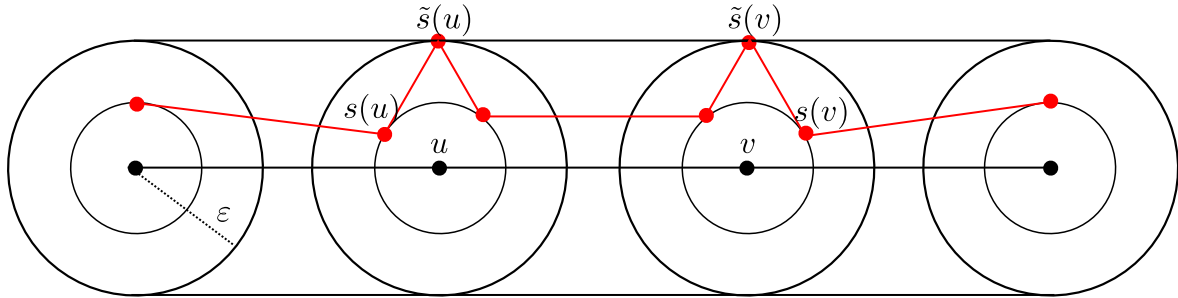


Abbildung 5: Die Wahl der Randpunkte  $\tilde{s}(u), \tilde{s}(v)$  für das Bild der mittleren Kante  $\{u, v\}$  ermöglicht zwar eine (schwache)  $\varepsilon$ -Platzierung für den skizzierten Graphen  $G_1$ , allerdings ist diese nicht optimal bezüglich des  $\text{Min-Sum}_{(w)F}$  Abstands. Die (schwachen) Fréchet-Abstände zwischen einer Kante und ihrem Bild betragen unter  $\tilde{s}$  jeweils  $\varepsilon$ . Unter  $s$  hingegen wird der Flaschenhals-Abstand von  $\varepsilon$  nur durch die Platzierung der mittleren Kante angenommen. Die äußeren Kanten lassen sich durch die Wahl der Randpunkte  $s(u)$  und  $s(v)$  so platzieren, dass der (schwache) Fréchet-Abstand zwischen ihnen und den entsprechenden Kanten von  $G_1$  nur jeweils  $\frac{\varepsilon}{2}$  beträgt.

Quelle: In Anlehnung an eine zum Zeitpunkt des Schreibens der Arbeit (nach aktuellem Kenntnisstand) unveröffentlichte Bemerkung und Skizze der Autoren von [1].

Um diesen Freiheitsgrad und die dadurch implizierte Komplexität bezüglich der Berechnung von  $\text{Min-Sum}_{(w)F}(G_1, G_2)$  auszuschließen, werden wir für einen Knoten  $u \in V_1$  sein Bild unter einer Graph-Zuordnung für jede seiner  $\varepsilon$ -Platzierungen fixieren. Als Fixpunkt wählen wir dafür einen Punkt innerhalb der  $\varepsilon$ -Platzierung von  $u$  mit minimalem (euklidischen) Abstand zu  $u$ . Wir behaupten (ohne formalen Beweis), dass die Berechnung von  $\text{Min-Sum}_{(w)F}(G_1, G_2)$  ohne eine solche Einschränkung in diesem Fall NP-schwer ist.

## 3.2 Ansätze und Beschreibung eines polynomiellen Algorithmus

Um unter den Einschränkungen 3.1.1 und 3.1.2 an die zu betrachtenden Graph-Zuordnungen von  $G_1$  nach  $G_2$   $\text{Min-Sum}_{(w)F}(G_1, G_2)$  zu berechnen, wollen wir eine (schwache)  $\varepsilon$ -Platzierung  $s : G_1 \rightarrow G_2$  konstruieren, die das Optimalitätskriterium (siehe 2.3) erfüllt. Für eine solche Abbildung  $s$  gilt dann entsprechend

$$\sum_{e \in E_1} \delta_{(w)F}(e, s(e)) = \text{Min-Sum}_{(w)F}(G_1, G_2).$$

Der algorithmische Ansatz zur Lösung des Entscheidungsproblems, ob eine (schwache)  $\varepsilon$ -Platzierung von  $G_1$  existiert (siehe [2], Seiten 9-11), liefert das Grundgerüst für die Konstruktion von  $s$  zur Berechnung von  $\text{Min-Sum}_{(w)F}(G_1, G_2)$ . Der finale Algorithmus gliedert sich letztlich in insgesamt 4 Einzelschritte, die wir im Verlauf des Kapitels entsprechend kennzeichnen und beschreiben werden.

### 3.2.1 $\varepsilon$ -Platzierungen der Knoten

Die Zuordnung der Knoten  $u, v \in V_1$  auf einen Punkt jeweils innerhalb ihrer  $\varepsilon$ -Platzierungen implizieren die Randpunkte  $W(0), W(1)$  für einen einfachen Weg  $W : [0, 1] \rightarrow G_2$ , auf den die Kante  $\{u, v\} \in E_1$  durch eine Graph-Zuordnung  $s$  abgebildet werden kann. Dass diese Randpunkte innerhalb der  $\varepsilon$ -Platzierung ihrer jeweiligen Knoten liegen, ist ein notwendiges Kriterium dafür, dass der (schwache) Fréchet-Abstand zwischen der Kante und ihrem Bild unter  $s$  nicht größer als  $\varepsilon$  ist.

Aus der Definition des (schwachen) Fréchet-Abstands (siehe 2.6) folgt direkt, dass  $\delta_{(w)F}(e, W) \geq \max\{\|u - W(0)\|, \|v - W(1)\|\}$ .

Daher berechnen wir zunächst alle  $\varepsilon$ -Platzierungen der Knoten in  $G_1$  und bezeichnen für einen Knoten  $v \in V_1$  mit  $P(v)$  die Menge seiner  $\varepsilon$ -Platzierungen.

### 3.2.2 Schritt 1

Wir initialisieren  $\varepsilon$  mit  $\varepsilon \geq \vec{\delta}_{(w)F}(G_1, G_2)$ . Da  $G_1$  ein Baum ist, können wir  $\vec{\delta}_{wF}(G_1, G_2)$  mit einer Laufzeit - und Speicherkomplexität von  $O(n_1 \cdot m_1)$  und  $\vec{\delta}_F(G_1, G_2)$  mit einer Laufzeit - und Speicherkomplexität von  $O(n_1 \cdot m_2^2)$  berechnen (siehe Akitaya et al [2], Seite 15). Anschließend berechnen wir die  $\varepsilon$ -Platzierungen aller Knoten von  $G_1$ . Jeder Knoten in  $G_1$  hat  $O(m_2)$   $\varepsilon$ -Platzierungen, womit insgesamt  $(n_1 \cdot m_2)$   $\varepsilon$ -Platzierung zu berechnen sind. Für jeden Knoten lassen sich seine  $\varepsilon$ -Platzierungen über

einen Standard-Algorithmus zur Berechnung von zusammenhängenden Komponenten in Graphen ermitteln, dessen Laufzeit linear in der Größe von  $G_2$  ist. Die Laufzeit - und Speicherkomplexität beträgt somit  $O(n_1 \cdot m_2)$ .

### 3.2.3 Schritt 2

Für jede Kante  $\{u, v\} \in E_1$  berechnen wir nun die (schwachen) Erreichbarkeiten zwischen den  $\varepsilon$ -Platzierungen  $P(u)$  und  $P(v)$ . Dabei ist für jede Kombination  $p_u \in P(u), p_v \in P(v)$  von Knoten-Platzierungen jeweils individuell zu entscheiden, ob sie (schwach) voneinander erreichbar sind.

Zur Berechnung der schwachen Erreichbarkeit ist die Existenz eines einfachen Weges zwischen zwei Knoten-Platzierungen  $p_u, p_v$ , welcher vollständig innerhalb von  $T_\varepsilon(\{u, v\})$  liegt, bereits ein hinreichendes Kriterium (siehe dazu [3], Seite 83). Schränken wir  $G_2$  auf  $T_\varepsilon(\{u, v\})$  ein, so sind alle Knoten-Platzierungen innerhalb derselben zusammenhängenden Komponente von  $G_2$  schwach voneinander erreichbar. Entsprechend speichern wir für eine zusammenhängende Komponente von  $G_2$  innerhalb  $T_\varepsilon(e)$  jeweils zwei Listen mit den zusammenhängenden Knoten-Platzierungen von  $u$  und  $v$ .

Die schwachen Erreichbarkeiten zwischen allen Knoten-Platzierungen entlang einer Kante in  $G_1$  lassen sich somit mit einer Laufzeit von  $O(m_2)$  berechnen. Insgesamt beträgt die Zeit - und Speicherkomplexität zur Berechnung der Erreichbarkeiten aller Kanten in  $G_1$  damit  $O(m_1 \cdot m_2)$ .

Für die nicht-schwachen Erreichbarkeiten ist die Existenz eines einfachen Weges  $W$  innerhalb von  $T_\varepsilon(\{u, v\})$ , welcher  $p_u$  und  $p_v$  miteinander verbindet, zwar ein notwendiges aber noch kein hinreichendes Kriterium dafür, dass  $p_u$  und  $p_v$  voneinander erreichbar sind. Wir müssen für  $W$  explizit entscheiden, ob  $\delta_F(\{u, v\}, W) \leq \varepsilon$ .

Dabei können wir für zwei Wege  $Y$  und  $Z$  das Entscheidungsproblem  $\delta_F(Y, Z) \leq \varepsilon$  in  $O(|Y| \cdot |Z|)$  lösen (siehe dazu [3], Seite 77). Da  $\{u, v\}$  ein Geradenstück ist, hat  $\{u, v\}$  die Länge 1. Damit können wir die Erreichbarkeit zwischen  $\{u, v\}$  und  $W$  in  $O(|W|)$  entscheiden. Da  $W$  per Bedingung ein einfacher Weg ist, besteht  $W$  aus maximal  $m_2$  Kanten (beziehungsweise Teilkanten) von  $G_2$ . Zur Speicherung der Erreichbarkeiten speichern wir für jede  $\varepsilon$ -Platzierung von  $u$  eine Liste mit  $\varepsilon$ -Platzierungen von  $v$ , die über einen Weg voneinander erreichbar sind. Um die Liste für ein  $p_u \in P(u)$  zu berechnen, starten wir eine Graph-Suche zur Ermittlung eines Weges in  $G_2$  beginnend bei  $p_u$ . Verlassen wir während der Suche entweder  $T_\varepsilon(\{u, v\})$  beziehungsweise überschreitet der Fréchet-Abstand  $\varepsilon$ , so entfernen wir den Suchzweig aus der Liste der möglichen Abzweigungen in  $G_2$ . Für eine Kante in  $G_1$  beträgt die Laufzeit der Suche somit  $O(m_2^2)$  und pro Kante zählen wir bis zu  $m_2^2$  Paare von potentiell voneinander erreichbarer

Knoten-Platzierungen. Für alle Kanten in  $G_1$  liegt die Speicher - und Laufzeitkomplexität damit bei  $O(m_1 \cdot m_2^2)$ .

### 3.2.4 Existenz einer (schwachen) $\varepsilon$ -Platzierung von $G_1$

Bis zu diesem Schritt haben wir die (schwachen) Erreichbarkeiten zwischen Knoten-Platzierungen paarweise ermittelt. Nun wollen wir unseren Wortschatz um den Begriff der Knoten-Platzierungen um eine Eigenschaft erweitern, um anschließend eine Aussage über die Existenz einer (schwachen)  $\varepsilon$ -Platzierung von  $G_1$  treffen zu können.

Hat nämlich ein Knoten  $u \in V_1$  neben  $\{u, v\} \in E_1$  noch eine weitere inzidente Kante  $\{u, w\}$ , so muss neben einem  $p_v \in P(v)$  insbesondere auch ein  $p_w \in P(w)$  existieren, so dass die Paare  $(p_u, p_v)$  und  $(p_u, p_w)$  (schwach) voneinander erreichbar sind. Die (schwache) Erreichbarkeit zwischen einer Knoten-Platzierungen mit mindestens einer weiteren Knoten-Platzierung pro benachbartem Knoten stellt somit ein notwendiges Kriterium für die Existenz einer (schwachen)  $\varepsilon$ -Platzierung von  $G_1$  dar.

#### Definition 3.2

Eine  $\varepsilon$ -Platzierung  $p_v$  eines Knotens  $v$  nennen wir *gültig (valid)*, falls für jeden Nachbarn  $u$  von  $v$  eine  $\varepsilon$ -Platzierung  $p_u$  existiert, so dass  $p_v$  und  $p_u$  voneinander erreichbar sind. Ansonsten nennen wir  $p_v$  *ungültig (invalid)*.

Analog nennen wir eine  $\varepsilon$ -Platzierung  $p_v$  eines Knotens  $v$  *schwach gültig (weakly valid)*, falls für jeden Nachbarn  $u$  von  $v$  eine  $\varepsilon$ -Platzierung  $p_u$  existiert, so dass  $p_v$  und  $p_u$  schwach voneinander erreichbar sind. Ansonsten nennen wir  $p_v$  *schwach ungültig (weakly invalid)*.

Für die Konstruktion von  $s$  können entsprechend nur (schwach) gültige  $\varepsilon$ -Platzierungen der Knoten von  $G_1$  in Betracht gezogen werden. Denn die Wahl einer (schwach) ungültigen Knoten-Platzierung für die Zuordnung eines Knotens von  $G_1$  hat definitionsgemäß (3.2) direkt zur Folge, dass die zugrundeliegende Graph-Zuordnung keine (schwache)  $\varepsilon$ -Platzierung von  $G_1$  sein kann. Entsprechend löschen wir unter den in *Schritt 1* (3.2.2) ermittelten Knoten-Platzierungen alle diejenigen, die (schwach) ungültig sind.

Dabei ist zu beachten, dass nach dem Löschen einer (schwach) ungültigen  $\varepsilon$ -Platzierung  $p_v \in P(v)$  von  $v$  jede vor der Löschung noch (schwach) gültige  $\varepsilon$ -Platzierung eines Nachbarn von  $v$  nun (schwach) ungültig geworden sein kann. Unter Umständen war zum Beispiel die (schwache) Gültigkeit der Knoten-Platzierung eines Nachbarn von  $v$  gerade durch die (schwache) Erreichbarkeit zu der zuvor gelöschten  $\varepsilon$ -Platzierung von  $v$  bedingt. In dem Sinne löschen wir rekursiv solange (schwach) ungültige  $\varepsilon$ -Platzierungen

der Knoten in  $G_1$ , bis keine (schwach) ungültigen Knoten-Platzierungen mehr existieren. Als Abbruchbedingung der Rekursion gilt daher, dass die Löschung der letzten bekannten (schwach) ungültigen Knoten-Platzierung keine weitere Löschung mehr impliziert.

### 3.2.5 Schritt 3

In diesem Schritt werden, wie oben beschrieben, alle Knoten-Platzierungen (rekursiv) bereinigt. Um für eine Kante  $\{u, v\} \in E_1$  zu entscheiden, welche Platzierungen der Knoten  $u$  und  $v$  (schwach) voneinander erreichbar sind, bedienen wir uns dafür der in *Schritt 2* (3.2.3) berechneten Erreichbarkeiten. In *Schritt 1* (3.2.2) haben wir  $O(n_1 \cdot m_2)$  Knoten-Platzierungen ermittelt. Für die schwache Erreichbarkeit haben wir in *Schritt 2* für eine Kante  $\{u, v\} \in E_1$  die zusammenhängende Komponente innerhalb von  $T_\varepsilon(\{u, v\})$  und ihre enthaltenen Knoten-Platzierungen von  $u$  und  $v$  gespeichert. Ist nun eine Knoten-Platzierung  $p_v \in P(v)$  ungültig, so muss  $p_v$  aus allen entsprechenden Listen von Knoten-Platzierungen gelöscht werden, in denen  $p_v$  enthalten ist. Dies betrifft potentiell alle zu  $v$  adjazenten Knoten, daher beträgt die Laufzeit zum Bereinigen von  $p_v$   $O(\deg(v))$ . Mit der Identität  $\sum_{v \in V_1} (\deg(v)) = 2m_1$  beträgt die Laufzeit zum Bereinigen aller ungültigen Knoten-Platzierungen  $O(m_1 \cdot m_2)$ .

Für die nicht-schwache Erreichbarkeit haben wir für jede Knoten-Platzierung  $p_v$  eine Liste mit erreichbaren Knoten-Platzierungen gespeichert. Sollte  $p_v$  ungültig sein, so müssen wir  $p_v$  für jeden zu  $v$  adjazenten Knoten aus den Listen der Knoten-Platzierungen entfernen, in denen  $p_v$  enthalten ist. Für jeden zu  $v$  adjazenten Knoten sind dies  $O(m_2)$  Listen, daher beträgt die Laufzeit für die Löschung von  $p_v$  insgesamt  $O(\deg(v) \cdot m_2)$ . Das Bereinigen aller Knoten-Platzierungen beträgt entsprechend  $O(m_1 \cdot m_2^2)$ .

Nachdem die *Schritte 1-3* durchgeführt wurden, liefert uns ein Hilfssatz aus Akitaya et al. [2] (siehe dazu Seite 13, Lemma 6) eine konkrete Aussage über die Existenz einer (schwachen)  $\varepsilon$ -Platzierung von  $G_1$ . Da der Beweis dazu konstruktiver Natur ist, führen wir ihn hier mit auf.

### Lemma 3.3

Sei  $G_1$  ein Baum und für  $v \in V_1$  sei  $P(v)$  die Menge der (schwach) gültigen  $\varepsilon$ -Platzierungen von  $v$ , nachdem alle (schwach) ungültigen  $\varepsilon$ -Platzierungen der Knoten in  $G_1$  rekursiv gelöscht wurden (analog wie in 3.2.5). Enthält  $P(v)$  für jedes  $v \in V_1$  mindestens eine (schwach) gültige  $\varepsilon$ -Platzierung, so existiert eine (schwache)  $\varepsilon$ -Platzierung von  $G_1$ .

*Beweis.* Für den Beweis konstruieren wir eine  $\varepsilon$ -Platzierung von  $G_1$ . Wir ernennen einen beliebigen Knoten  $v_r \in V_1$  zur Wurzel und betrachten  $G_1$  als gerichteten Baum (*Out-Tree*) ausgehend von  $v_r$ . Wir ordnen  $v_r$  einen beliebigen Punkt innerhalb einer seiner (schwachen)  $\varepsilon$ -Platzierungen zu und bearbeiten anschließend iterativ alle Knoten von  $G_1$  ausgehend von der Wurzel  $v_r$ .

Dabei wählen wir stets für einen Knoten  $v \in V_1$ , dessen Vorgänger bereits zugewiesen wurde, die  $\varepsilon$ -Platzierung  $p_v \in P(v)$  so, dass  $p_v$  und die bereits ausgewählte (beziehungsweise zugeordnete)  $\varepsilon$ -Platzierung  $p_a$  des Vorgängers von  $v$  voneinander (schwach) erreichbar sind. Für die Kante zwischen  $v$  und seinem Vorgänger können wir nun mindestens eine passende Kanten-Platzierung wählen. Denn nach Voraussetzung ist  $p_a$  (schwach) gültig und somit muss ein  $p_v$  mit der geforderten Erreichbarkeit zu  $p_a$  existieren. Per Definition der (schwachen) Erreichbarkeit gilt ferner, dass mindestens ein einfacher Weg in  $G_2$  existiert, welcher  $p_v$  und  $p_a$  verbindet und dessen (schwacher) Fréchet-Abstand mit der Kante zwischen  $v$  und seinem Vorgänger nicht größer als  $\varepsilon$  ist.

Da wir  $G_1$  als gerichteten Baum betrachten, sind nach endlich vielen Bearbeitungsschritten von der Wurzel ausgehend alle Knoten in  $G_1$  entsprechend zugewiesen und ihre  $\varepsilon$ -Platzierungen wurden mit den  $\varepsilon$ -Platzierungen ihrer Vorfahren durch einen einfachen Weg in  $G_2$  verbunden. Da  $G_1$  als Baum insbesondere kreisfrei ist, konnte so jede Kante explizit in  $G_1$  einer  $\varepsilon$ -Platzierung zugeordnet werden, ohne dass jemals der (schwache) Fréchet-Abstand größer als  $\varepsilon$  ist. Die durch die Auswahl dieser  $\varepsilon$ -Platzierungen implizierte Graph-Zuordnung ist damit insgesamt eine  $\varepsilon$ -Platzierung von  $G_1$ .  $\square$

Über die Konstruktion aus dem Beweis von Lemma 3.3 erhalten wir einen ersten Ansatz zur Konstruktion einer (schwachen)  $\varepsilon$ -Platzierung gemäß der erwähnten Einschränkungen und des Optimalitätskriteriums.

### 3.2.6 Optimale (schwache) Kanten-Platzierungen

Im Beweis von 3.3 konnten wir zur Konstruktion einer  $\varepsilon$ -Platzierung von  $G_1$  stets eine beliebige (schwache)  $\varepsilon$ -Platzierung einer Kante wählen.

Diesen Ansatz werden wir erweitern, indem wir statt einer beliebigen jeweils nur eine lokal optimale (schwache)  $\varepsilon$ -Platzierung einer Kante für die Graph-Zuordnung in Betracht ziehen. Dabei minimiert eine lokal optimale (schwache) Kanten-Platzierung  $p_{\{u,v\}}$  den (schwachen) Fréchet-Abstand zu der Kante  $\{u, v\}$ . Für ein Paar von Knoten-Platzierungen  $p_u, p_v$  sei

$$\Delta(p_u, p_v) = \min_{p_{\{u,v\}}} \delta_{(w)F}(\{u, v\}, p_{\{u,v\}}),$$

wobei sich  $p_{\{u,v\}}$  über alle (schwachen)  $\varepsilon$ -Platzierungen von  $\{u, v\}$  erstreckt. Die unter 3.1.2 geforderten Fixpunkte werden dafür vor der Berechnung von  $\Delta$  gesetzt.

Wir erweitern *Schritt 2* des Algorithmus um die Berechnung optimaler (schwacher) Kanten-Platzierungen und speichern für jede Kante  $e = \{u, v\} \in G_1$  eine Liste  $L_e$  mit Einträgen  $(p_v, p_u, \Delta(p_v, p_u))$  für alle  $p_v \in P(v)$ ,  $p_u \in P(u)$  die (schwach) voneinander erreichbar sind. Dies lässt sich grundsätzlich mit polynomieller Speicher- und Laufzeitkomplexität berechnen. Wir werden im weiteren Verlauf noch Verweise bezüglich der Komplexitäten für diesen Schritt liefern, diese jedoch hier nicht im Detail behandeln.

Sofern  $L_e$  für alle Knoten in  $G_1$  berechnet ist und alle ungültigen Knoten-Platzierungen nach Ausführung von *Schritt 3* (3.2.5) bereinigt sind, reduziert sich die Konstruktion einer optimalen (schwachen)  $\varepsilon$ -Platzierung von  $G_1$  (beziehungsweise die Berechnung von  $\text{Min-Sum}_{(w)F}(G_1, G_2)$ ) auf eine optimale Gesamt-Auswahl der jeweiligen Paare von Knoten-Platzierungen für jedes Paar adjazenter Knoten in  $G_1$ . So eine Konstruktion ließe sich analog wie im Beweis zu Lemma 3.3 realisieren und eine so ermittelte Graph-Zuordnung ist insbesondere auch eine  $\varepsilon$ -Platzierung von  $G_1$ . Letztlich stellt dann die Frage, unter welcher der möglichen Auswahlen von Knoten-Platzierungen bei der Konstruktion von  $s$  die Summe der Gewichte der durch  $s$  implizierten Kanten-Platzierungen minimiert und damit das Optimalitätskriterium erfüllt wird.

### 3.2.7 Konstruktion einer optimalen $\varepsilon$ -Platzierung von $G_1$

Ein naiver Ansatz zur Berechnung von  $\text{Min-Sum}_{(w)F}(G_1, G_2)$  wäre nun, alle möglichen Kombinationen der  $\varepsilon$ -Platzierungen der Knoten von  $G_1$  über beispielsweise einen *Backtracking-Algorithmus* auszuschöpfen. Dabei würden wir - beginnend bei dem ausgewiesenen Wurzelknoten  $v_r$  - die Knoten und Kanten von  $G_1$  wie in der Konstruktion der Graph-Zuordnung (siehe 3.3) zuordnen. An jedem Knoten, wo es für die Auswahl einer Knoten-Platzierung mehr als eine Möglichkeit gibt, hätte der Backtracking-Algorithmus eine Abzweigung pro Auswahlmöglichkeit. Sobald wir erstmalig  $G_1$  vollständig zugeordnet haben, speichern wir die Zuordnung  $s$  sowie ihr Gewicht  $\text{Sum}_{(w)F}(s)$ , welches sich als Summe der paarweise gewählten Knoten-Platzierungen und ihren  $\Delta$ -Gewichten ergibt. Für einen bereits zugeordneten Teilgraphen könnten wir sein Gewicht unter den bereits zugeordneten Kanten mitführen und backtracken, sobald dieses das bis zu diesem Iterationsschritt minimalste Gewicht einer vollständigen Zuordnung von  $G_1$  überschreitet. Nachdem der Algorithmus terminiert, hätten wir so  $\text{Min-Sum}_{(w)F}(G_1, G_2)$  berechnet.

Allerdings ist trotz der Einschränkungen an die zu betrachtenden Graph-Zuordnungen der Möglichkeitsraum im Allgemeinen zu groß für einen solchen (brute-force) Ansatz. Jeder Knoten in  $G_1$  hat potentiell bis zu  $m_2$   $\varepsilon$ -Platzierungen und für zwei benach-



barte Knoten in  $G_1$  gibt somit bis zu  $m_2^2$  Möglichkeiten, die Knoten-Platzierungen zu kombinieren. Insgesamt wächst die Anzahl der potentiellen Kombinationsmöglichkeiten exponentiell mit der Anzahl der Kanten beziehungsweise der Anzahl der Knoten ( $G_1$  ist ein Baum, daher gilt  $m_1 = n_1 - 1$ ) von  $G_1$ .

Stattdessen konstruieren wir nun im letzten Schritt des Algorithmus eine geeignete Suchstruktur für die Zuweisung und Gewichtung von Knoten-Platzierungen. Anders als bei dem naiven Ansatz werden wir die Knoten-Platzierungen nun stets *greedy* - also möglichst optimal - berechnen und zuweisen. Dabei werden wir eine (schwache)  $\varepsilon$ -Platzierung  $s$  konstruieren, die ebenfalls eine  $\text{Min-Sum}_{(w)F}$  Zuordnung ist, also das Optimalitätskriterium erfüllt.

### 3.2.8 Schritt 4

Wir interpretieren  $G_1$  nach wie vor (analog wie im Beweis zu Lemma 3.3) als gerichteten Wurzelbaum mit beliebiger Wurzel  $v_r$  und konstruieren einen abstrakten Graphen  $H = (V_H, E_H)$  als Suchstruktur für die Berechnung. Jede (schwach) gültige  $\varepsilon$ -Platzierung  $p_v$  eines Knotens  $v \in V_1$  wird durch einen entsprechenden Knoten  $i_{p_v} \in V_H$  repräsentiert. Für zwei Knoten-Platzierungen  $p_v \in P(v)$  und  $p_u \in P(u)$  gilt  $(i_{p_u}, i_{p_v}) \in E_H$  genau dann, wenn

- (i)  $G_1$  die (gerichtete) Kante  $(v, u)$  enthält und
- (ii)  $p_v$  und  $p_u$  (schwach) voneinander erreichbar sind.

Insgesamt ist  $H$  somit ein gerichteter, azyklischer Graph. Die Gerichtetheit folgt dabei direkt aus der Konstruktion der Kanten in  $H$  und ein Kreis in  $H$  würde direkt der Voraussetzung widersprechen, dass  $G$  ein (gerichteter) Baum ist. Da jeder Knoten in  $G_1$  bis zu  $m_2$  Knoten-Platzierungen hat, gilt  $|V_H| \in O(n_1 \cdot m_2)$  und  $|E_H| \in O(n_1 \cdot m_2^2)$ . Wir induzieren durch  $w_H : E_H \rightarrow \mathbb{R}$  mit  $w_H(i_{p_v}, i_{p_u}) = \Delta(p_v, p_u)$  eine Gewichtung der Kanten von  $H$  und initialisieren  $w(p_v) = 0$  für alle Knoten-Platzierungen. Wir werden nun die Knoten-Platzierungen iterativ und greedy einander zuweisen und so eine optimale (schwache)  $\varepsilon$ -Platzierung von  $G_1$  berechnen. Im Gegensatz zu der Konstruktion von Lemma 3.3 berechnen wir  $s$  und die Gewichte der Knoten-Platzierungen *bottom-up*. Für einen Knoten  $v \in V_1$ , dessen Nachfahren ausschließlich Blätter in  $G_1$  sind, setzen wir

$$w(p_v) = \sum_{u : u \text{ ist Nachfahre von } v} \min_{p_u \in P(u)} w(p_u) + w_H(p_v, p_u) \quad (1)$$

Dabei speichern wir die Zuordnung  $s$  welche  $w(p_v)$  realisiert für alle  $p_v \in P(v)$  und löschen alle Nachfahren von  $v$  aus  $G_1$ , nachdem alle Knoten-Platzierungen von  $v$  bearbeitet wurden. Diese Bearbeitung eines Knotens in  $G_1$  bezeichnen wir als einen Iterationsschritt.

**Lemma 3.4**

Schritt 4 berechnet in  $O(n_1)$  Iterationsschritten (siehe 3.2.8)  $\text{Min-Sum}_{(w)F}(G_1, G_2)$  und eine optimale  $\varepsilon$ -Platzierung von  $G_1$ .

*Beweis.* Wir beweisen die Aussage über die Invariante, dass nach jedem Iterationsschritt der Teilgraph (eingeschränkt auf den zuletzt bearbeiteten Knoten) optimal zugeordnet ist.

Vor dem ersten Iterationsschritt ist das Gewicht aller Knoten-Platzierungen 0. Wir bezeichnen den aktualisierten Graphen  $G_1$  nach dem  $i$ -ten Iterationsschritt mit  $G_1^i$ . Für einen beliebigen Knoten  $v \in V_1$ , dessen Nachfahren Blätter in  $G_1^0$  sind, wird das Gewicht aller Knoten-Platzierungen nun durch die Berechnung von Gleichung (1) für alle Nachfahren von  $v$  minimiert. Insbesondere entspricht  $w(v)$  gerade der Summe aller gewichteten Kanten-Platzierungen. Betrachten wir nun  $s$  als die Abbildung, die  $\min w(p_v) : p_v \in P(v)$  realisiert, so ist  $s$  offensichtlich eine Graph-Zuordnung, die  $\text{Min-Sum}_{(w)F}(G_v, G_2)$  auf dem Teilgraphen  $G_v$  - mit  $v$  als Wurzel - einhält.

Seien nun alle Teilgraphen in  $G_1^i$  für ein beliebiges  $i > 1$  an allen bereits bearbeiteten Knoten optimal zugeordnet und sei  $u \in V_1$  ein Knoten, dessen Nachfahren Blätter in  $G_1^i$  sind. Die Graph-Zuordnung, die  $\min w(p_u)$  für  $p_u \in P(u)$  realisiert, wählt unter Umständen nach der Berechnung von Gleichung (1) für einen Nachfahren  $\tilde{u}$  von  $u$  eine andere Knoten-Platzierung als  $\min w(p_{\tilde{u}})$  ( $p_{\tilde{u}} \in P(\tilde{u})$ ). Nach Voraussetzung sind jedoch wiederum alle (potentiellen) Nachfahren von  $\tilde{u}$  für jede Knoten-Platzierung von  $\tilde{u}$  optimal zugeordnet. Insgesamt ist damit die Graph-Zuordnung, die  $\min w(p_u) : p_u \in P(u)$  realisiert, optimal in dem geforderten Sinne.

Da zu Beginn von Schritt 4 weniger als  $n_1$  Knoten mindestens einen Nachfahren haben und pro Iteration genau ein Knoten bearbeitet wird, endet Schritt 4 nach  $O(n_1)$  Iterationsschritten mit der Berechnung der Wurzel  $v_r$  und es gilt nach obiger Argumentation:

$$\min_{p_{v_r} \in P(v_r)} w(p_{v_r}) = \text{Min-Sum}_{(w)F}(G_1, G_2) \quad (2)$$

Die Graph-Zuordnung, welche Gleichung (2) realisiert, ist (nach 3.3) eine (schwache)  $\varepsilon$ -Platzierung von  $G_1$ .

□

Insgesamt setzt sich der Algorithmus also aus den folgenden Einzelschritten zusammen:

- 1) *Schritt 1:* Berechnung der  $\varepsilon$ -Platzierungen der Knoten von  $G_1$ .
- 2) *Schritt 2:* Berechnung der (schwachen) Erreichbarkeiten und optimaler (schwachen) Kanten-Platzierungen.
- 3) *Schritt 3:* Rekursive Löschung aller ungültigen  $\varepsilon$ -Platzierungen der Knoten von  $G_1$ .
- 4) *Schritt 4:* Konstruktion der Suchstruktur  $H$  und iterative Berechnung von  $\text{MinSum}_{(w)F}(G_1, G_2)$  und einer optimalen  $\varepsilon$ -Platzierung von  $G_1$ .

**Satz 3.5**

Sei  $G_1$  ein Baum und sei  $\varepsilon \geq \vec{\delta}_{(w)F}(G_1, G_2)$ . So lässt sich unter Einhaltung der Einschränkung 3.1.2 eine (schwache)  $\varepsilon$ -Platzierung  $s$  von  $G_1$  mit einer polynomiellen Laufzeitkomplexität von  $O(n_1 \cdot m_2^3) + T_{(w)F}(g_1, g_2)$  und einer polynomiellen Speicherkomplexität von  $O(n_1 \cdot m_2^2) + S_{(w)F}(g_1, g_2)$  berechnen, so dass für jede andere (schwache)  $\varepsilon$ -Platzierung  $\tilde{s}$  von  $G_1$  gilt:

$$\sum_{e \in E_1} \delta_{(w)F}(e, s(e)) \leq \sum_{e \in E_1} \delta_{(w)F}(e, \tilde{s}(e)).$$

Dabei beschreibt  $T_{(w)F}(g_1, g_2)$  die Laufzeitkomplexität und  $S_{(w)F}(g_1, g_2)$  die Speicherkomplexität zur Berechnung der minimalen (schwachen) Fréchet-Abstände, in Abhängigkeit der durch  $g_1$  und  $g_2$  symbolisierten Komplexitäten von  $G_1$  und  $G_2$ .

*Beweis.* Die Korrektheit des Algorithmus folgt direkt mit Lemma 3.4. Zur Berechnung des minimalen (schwachen) Fréchet-Abstands existieren in der Literatur diverse Algorithmen mit unterschiedlichen polynomiellen Laufzeit - und Speicherkomplexitäten. Diese wollen wir hier lediglich referenzieren. Entscheidend für uns ist lediglich, dass entsprechende Laufzeit - und Speicherkomplexitäten zum Berechnen des Problems tatsächlich polynomiell sind. Für den Fréchet-Abstand verweisen wir hierfür konkret auf Alt et. al [4] (Seite 272). Weitere polynomielle Ansätze finden sich zusätzlich in [5] und [6].

Ansonsten wird die sonstige Speicherkomplexität der Schritte 1-4 durch die Speicherung von  $H$  und die sonstige Laufzeitkomplexität der Schritte 1-4 durch die Berechnung von Gleichung (2) dominiert.

$H$  speichert alle Paare (schwach) voneinander erreichbarer Knoten-Platzierungen sowie die minimalen (schwachen) Fréchet-Abstände zwischen einem sie verbindenden Weg

und der assoziierten Kante in  $G_1$ . Für zwei (schwach) voneinander erreichbare Knoten-Platzierungen enthält  $H$  eine Kante. Für jeden Knoten in  $G_1$  gibt es  $O(m_2^2)$  solcher Paare und insgesamt liegt die Speicherkomplexität von  $H$  für alle Knoten in  $G_1$  damit bei  $O(n_1 \cdot m_2^2)$ .

Zur Berechnung von Gleichung (2) betrachten wir für eine feste Knoten-Platzierung  $p_v \in P(v)$  den zur Lösung der Gleichung relevanten Teilgraphen von  $H$ , bestehend aus allen Knoten-Platzierungen von  $v$  sowie deren (schwach) erreichbaren Nachfahren. Dieser Teilgraph weist somit für eine fixierte Knoten-Platzierung eine Komplexität von  $O(m_2^2)$  auf. Sofern  $v$  nicht die Wurzel ist, gibt es  $\deg(v) - 1$  Nachfahren von  $v$ . Ferner gilt  $|P(v)| \in O(m_2)$ . Damit beträgt die Zeitkomplexität zur Berechnung von Gleichung (2) für alle Knoten-Platzierungen von  $v$  insgesamt

$$O(\deg(v) \cdot m_2^3). \quad (3)$$

Mit (2) und der Identität

$$\sum_{v \in V_1} \deg(v) = 2 \cdot m_1 = 2(n - 1) \quad (4)$$

ergibt sich schließlich die Laufzeitkomplexität von  $O(n_1 \cdot m_2^3)$ .  $\square$

### 3.3 Ein Beispiel

Anhand eines kleinen Beispiels (siehe Abbildung 6) wollen wir den oben beschriebenen Algorithmus nun demonstrieren.

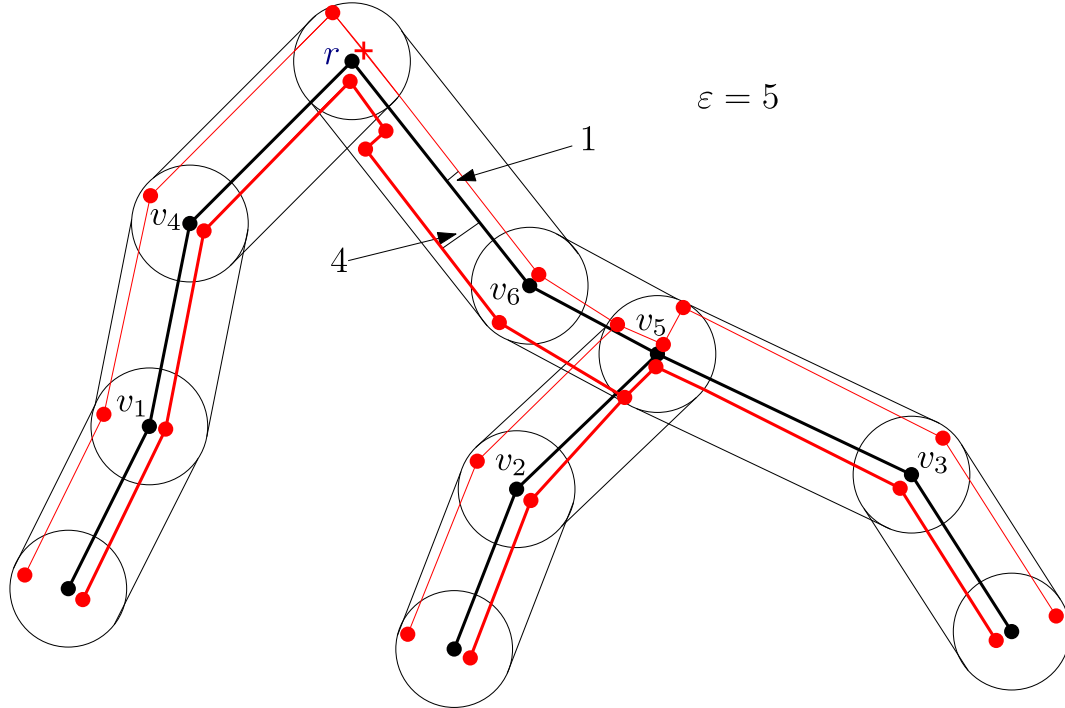


Abbildung 6:  $G_1$  ist in schwarz und  $G_2$  in rot gezeichnet. Der Wurzelknoten für  $G_1$ , welcher die Bearbeitungsreihenfolge der Knoten für den Algorithmus sowie die Struktur des Suchgraphen  $H$  bestimmt, ist mit  $r$  gekennzeichnet. Die sonstigen Knoten von  $G_1$ , die keine Blätter sind, sind mit  $v_i$  ( $1 \leq i \leq 6$ ) gekennzeichnet.

Quelle: In Anlehnung an Figure 3 aus Buchin et. al [1].

$G_2$  besteht hier (Abbildung 6) aus 2 disjunkten Teilgraphen - deren Kanten wir mit unterschiedlicher Stärke skizziert haben. Wir bezeichnen mit  $\widehat{G}_2$  den Teilgraphen von  $G_2$ , dessen Kanten dick skizziert sind und mit  $\widetilde{G}_2$  den Teilgraphen von  $G_2$ , dessen Kanten dünn skizziert sind. Jeder Knoten von  $G_1$  hat dabei genau 2 gültige  $\varepsilon$ -Platzierung, jeweils eine für  $\widehat{G}_2$  und  $\widetilde{G}_2$ . Es ist leicht zu sehen, dass jede durch die skizzierten  $\varepsilon$ -Bälle induzierten Knoten-Platzierung (schwach) gültig ist und genau einen Knoten enthält. Als Randpunkte der Kanten-Platzierungen fixieren wir hier stets - mit Ausnahme des Wurzelknotens  $r$  - für alle Knoten-Platzierungen die entsprechenden Knoten von  $G_2$  innerhalb der jeweiligen Platzierung. Dies machen wir, da diese mit einer Ausnahme bereits so eingebettet sind, dass sie die Punkte mit minimalem Abstand zu dem jeweiligen Knoten in  $G_1$  sind. Für eine Knoten-Platzierung von  $r$  innerhalb von  $\widetilde{G}_2$  haben wir den einzigen Fixpunkt, der hier kein Knoten von  $G_2$  ist, mit einem roten

Kreuz gekennzeichnet. Der Fréchet-Abstand zwischen einer Kante in  $G_1$  und einer ihrer Kanten-Platzierungen liegt stets in  $\{1, 4\}$ .

Die Suchstruktur  $H$ , skizziert in Abbildung 7, besteht aus 2 disjunkten, gerichteten Bäumen.

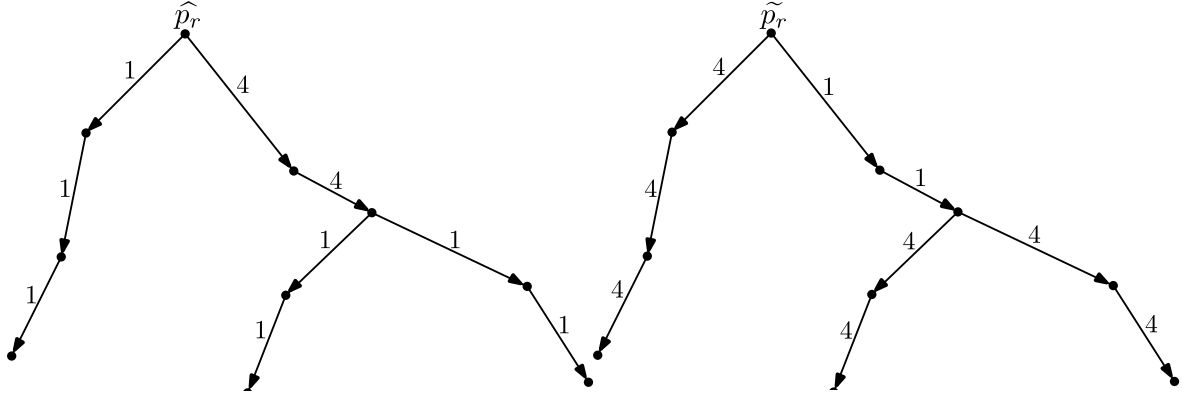


Abbildung 7: Eine Skizze von  $H$ . Jeder Knoten in  $H$  repräsentiert genau eine Knoten-Platzierung. Dabei repräsentiert  $\hat{p}_r$  konkret die  $\varepsilon$ -Platzierung von  $r$ , welche in  $\widehat{G}_2$  liegt und analog repräsentiert  $\tilde{p}_r$  die  $\varepsilon$ -Platzierung von  $r$ , welche in  $\widetilde{G}_2$  liegt. Wir haben  $H$  hierbei so skizziert, dass sich die durch  $H$  repräsentierten Knoten-Platzierungen aus ihrer Lage ergeben, die der Einbettung von  $G_1$  aus Abbildung 6 nachempfunden ist. Die Kanten von  $H$  sind bereits gewichtet und kennzeichnen die minimalen Kosten - gemäß des (schwachen) Fréchet-Abstands - einer Kanten-Platzierung für eine Auswahl zweier (schwach) voneinander erreichbarer Knoten-Platzierungen.

Quelle: In Anlehnung an Figure 3 aus Buchin et. al [1].

Nun beginnen wir, die Knoten in  $G_1$  zu bearbeiten, deren Nachfahren allesamt Blätter sind. Initial sind dies die Knoten  $v_1, v_2, v_3$  (siehe Abbildung 6). Bei den  $\varepsilon$ -Platzierungen der Knoten behalten wir die Konvention bei, dass wir für einen Knoten  $v_i \in V_1$  mit  $\widehat{p}_{v_i}$  die  $\varepsilon$ -Platzierung von  $v_i$  innerhalb von  $\widehat{G}_2$  und analog mit  $\widetilde{p}_{v_i}$  die  $\varepsilon$ -Platzierung von  $v_i$  innerhalb von  $\widetilde{G}_2$  kennzeichnen. Da das Gewicht aller Knoten-Platzierungen initial auf 0 gesetzt ist, sind zur Berechnung der Gewichte der Knoten-Platzierungen von  $v_1, v_2, v_3$  lediglich die entsprechenden gewichteten Kanten in  $H$  zu betrachten (siehe Abbildung 8).

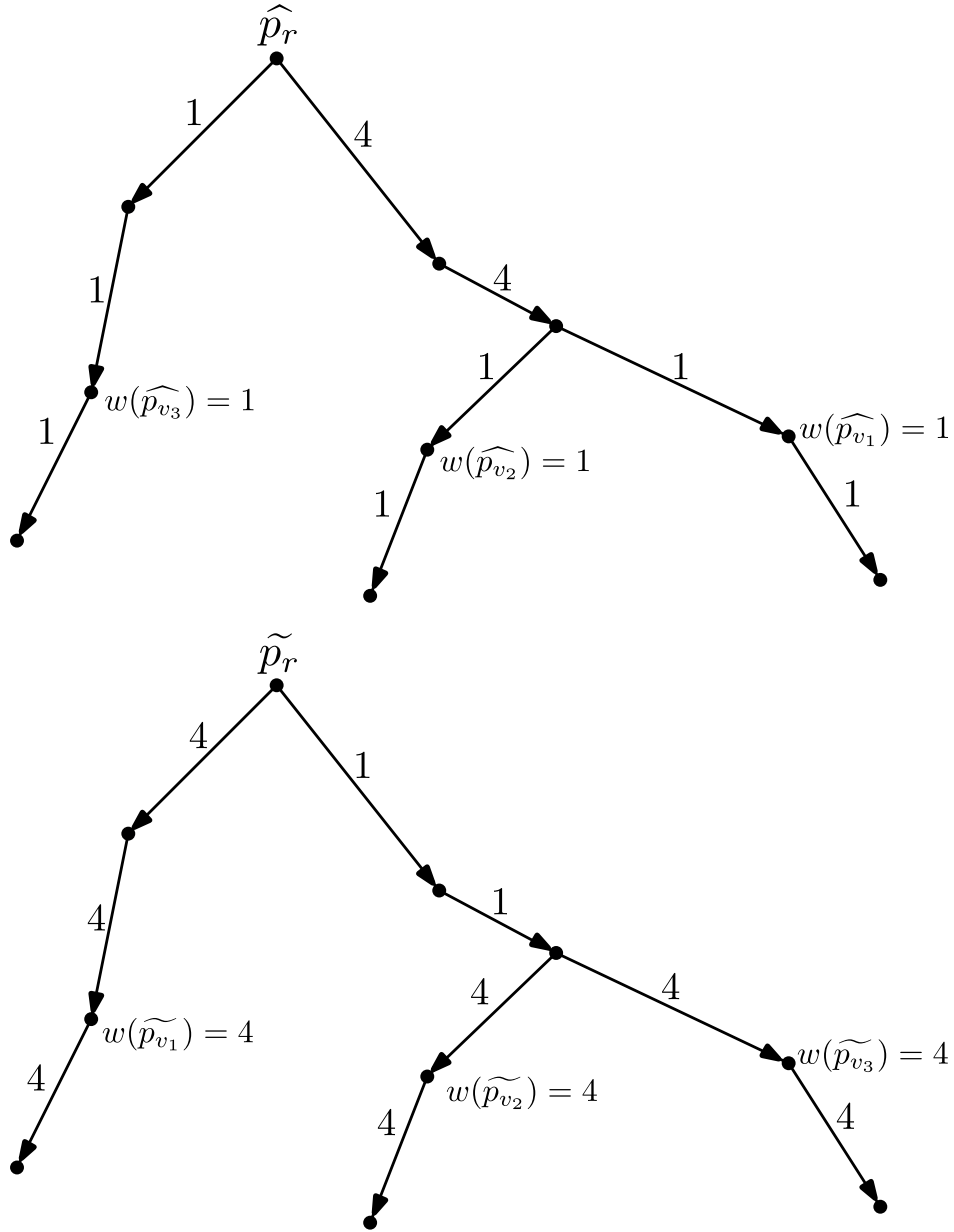


Abbildung 8: Nachdem die Knoten  $v_1, v_2, v_3$  bearbeitet wurden, sind die Gewichte aller ihrer  $\varepsilon$ -Platzierungen berechnet.

Anschließend sind die Knoten  $v_4$  und  $v_5$  in dem aktualisierten Graphen  $G_1^3$  diejenigen Knoten, deren Nachfahren Blätter sind.

$w(\widehat{p}_{v_5})$  berechnet sich zum Beispiel konkret entsprechend der Gleichung (1) über

$$\begin{aligned}
 w(\widehat{p}_{v_5}) &= w(\widehat{p}_{v_2}) + w_H(\widehat{p}_{v_5}, \widehat{p}_{v_2}) + w(\widehat{p}_{v_3}) + w_H(\widehat{p}_{v_5}, \widehat{p}_{v_3}) \\
 &= 1 + 1 + 1 + 1 \\
 &= 4
 \end{aligned} \tag{5}$$

Da in diesem Beispiel für alle Knoten-Platzierungen jeweils genau eine (schwach) er-

reichbare Knoten-Platzierung existiert, brauchen wir grundsätzlich - und explizit in in Gleichung (5) - kein Minimum aus mehreren Möglichkeiten berechnen.

Nachdem alle inneren Knoten von  $G_1$  bearbeitet wurden, steht nur noch die Bearbeitung des Wurzelknotens  $r$  aus. Die bis dahin berechneten Gewichte der Knoten-Platzierungen sind in Abbildung 9 notiert.

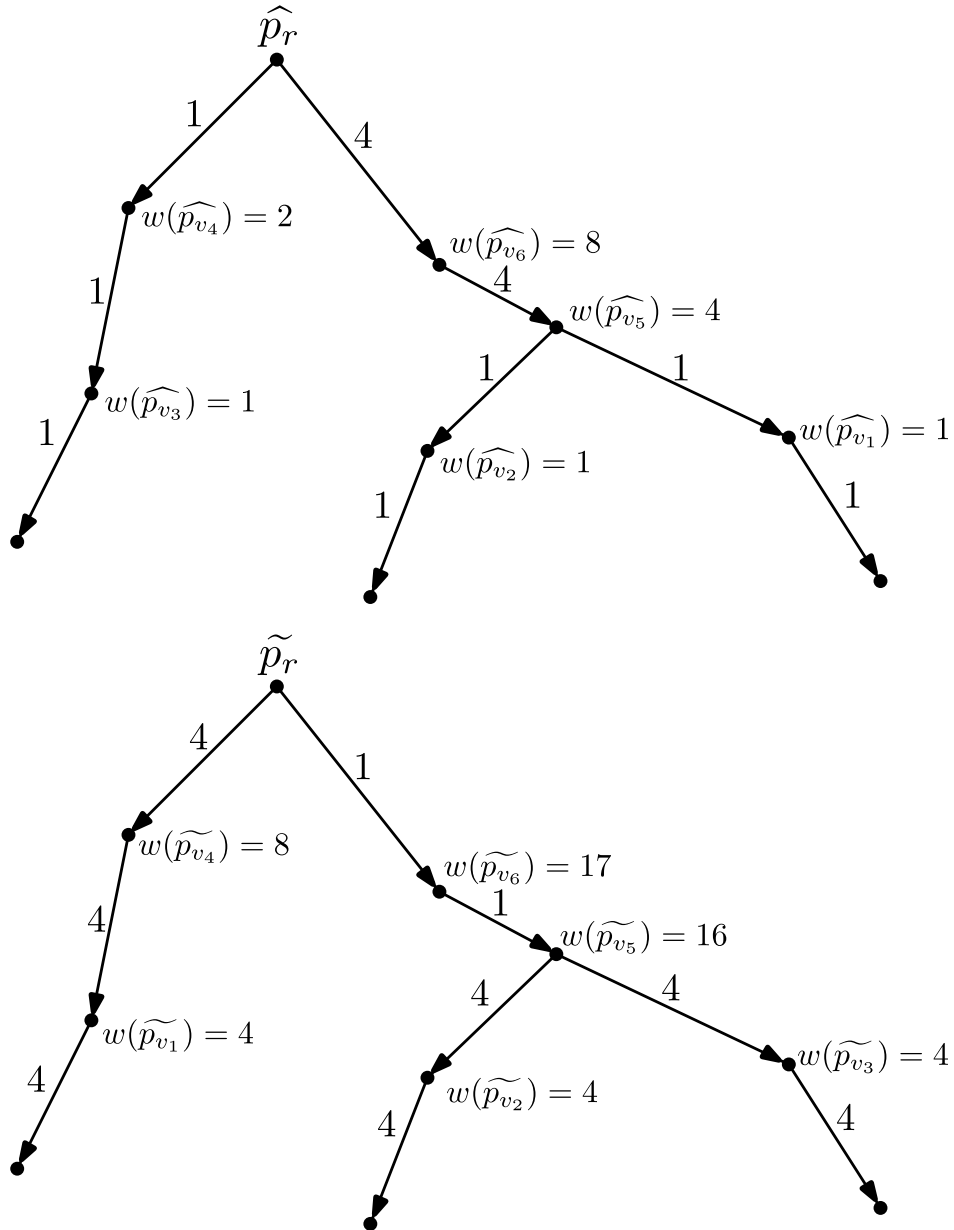


Abbildung 9: Alle berechneten Gewichte von  $\varepsilon$ -Platzierungen der inneren Knoten ( $v_i$ ) von  $G_1$ .



Für die  $\varepsilon$ -Platzierungen von  $r$  berechnen wir

$$\begin{aligned}
w(\widehat{p_r}) &= w(\widehat{p_{v_4}}) + w_H(\widehat{p_r}, \widehat{p_{v_4}}) + w(\widehat{p_{v_6}}) + w_H(\widehat{p_r}, \widehat{p_{v_6}}) \\
&= 2 + 1 + 8 + 4 \\
&= 15
\end{aligned} \tag{6}$$

und

$$\begin{aligned}
w(\widetilde{p_r}) &= w(\widetilde{p_{v_4}}) + w_H(\widetilde{p_r}, \widetilde{p_{v_4}}) + w(\widetilde{p_{v_6}}) + w_H(\widetilde{p_r}, \widetilde{p_{v_6}}) \\
&= 8 + 4 + 17 + 1 \\
&= 30
\end{aligned} \tag{7}$$

Damit ist  $\text{Min-Sum}_{(w)F}(G_1, G_2) = 15$  und die Graph-Zuordnung  $s$ , die  $w(\widetilde{p_r}) = 15$  realisiert, ist eine  $\text{Min-Sum}_{(w)F}$  Zuordnung. In unserem Beispiel ist das Bild von  $s$  gerade  $\widehat{G_2}$ .

## 4 Komplexität

### 4.0.1 NP-Schwerheit von $\text{Min-Sum}_{(w)F}$

Abschließend wollen wir zeigen, dass die Berechnung des  $\text{Min-Sum}_{(w)F}$  Abstands ohne Bedingungen an die Graphen  $G_1$  und  $G_2$  im Allgemeinen NP-schwer ist. Ein für die Konstruktion des Beweises ähnliches Setting finden wir in Akitaya et. al [2] (Seite 12-13), wo die Autoren die NP-Schwerheit des Graph-Abstands über eine Reduktion von *Binary Constraint Satisfaction Problem (CSP)* beweisen. Hier orientieren wir uns im Wesentlichen an der oben genannten Konstruktion aus [2].

#### Definition 4.1

Ein *Binary Constraint Satisfaction Problem* beschreibt folgendes Entscheidungsproblem:

Gegeben eine Instanz  $\langle X, D, C \rangle$  bestehend aus

einer Menge von *Variablen*  $X = \{x_1, x_2, \dots, x_n\}$ ,

einer Menge von *Domänen*  $D = \{D_1, D_2, \dots, D_n\}$ ,

und einer Menge von *Bedingungen*  $C = \{C_1, C_2, \dots, C_k\}$ .

Für jede Variable  $x_i \in X$  beschreibt ihre Domäne  $D_i \in D$  die Menge ihrer möglichen Wertzuweisungen. Eine Bedingung  $C_{i,j} \in C$  spezifiziert für je zwei unterschiedliche Variablen  $x_i, x_j \in X$  eine Relation  $R_{C_{i,j}} \subseteq D_i \times D_j$ . Wir nennen ein Wertepaar  $(d_i, d_j) \in D_i \times D_j$  für eine vorhandene Bedingung  $C_{i,j}$  an die Variablen  $x_i, x_j$  *zulässig*, wenn  $(d_i, d_j) \in R_{C_{i,j}}$ , falls  $(d_i, d_j)$  jedoch kein Element von  $R_{C_{i,j}}$  ist, *verletzt* das Wertepaar  $(d_i, d_j)$  die Bedingung  $C_{i,j}$  und wir nennen  $(d_i, d_j)$  *unzulässig*.

Die Fragestellung ist nun, ob alle Variablen einem Wert aus ihrer Domäne zugewiesen werden können, so dass alle mit Bedingungen versehenen Wertepaare zulässig sind. In diesem Falle nennen wir  $\langle X, D, C \rangle$  *lösbar*.

Für die allgemeine Komplexität der Klasse von *Constraint Satisfaction Problems* und insbesondere des *Binary Constraint Satisfaction Problems* siehe zum Beispiel [7].

#### Satz 4.2

Seien  $G_1$  und  $G_2$  geometrische Graphen und sei  $\varepsilon \geq 0$ . Das Entscheidungsproblem  $\text{Min-Sum}_{(w)F}(G_1, G_2) \leq \varepsilon$  ist NP-schwer.

*Beweis.* Sei  $\langle X, D, C \rangle$  ein beliebiges Binary Constraint Satisfaction Problem.

Wir repräsentieren jede Variable  $x_i \in X$  durch einen Knoten  $v_i \in V_1$  in  $G_1$  und für jede Bedingung  $C_{i,j}$  an die Variablen  $x_i, x_j$  hat  $G_1$  die Kante  $\{v_i, v_j\}$ . Wir setzen  $\varepsilon = |E_1|$ . Wir betten  $G_1$  so in die euklidische Ebene ein, dass sich für je zwei Knoten  $v_i, v_j$  ihre  $\varepsilon$ -Bälle nicht berühren und sich der  $\varepsilon$ -Schlauch einer Kante  $e \in E_1$  genau dann mit dem  $\varepsilon$ -Ball eines Knotens  $v \in V_1$  überlappt, wenn  $e$  und  $v$  inzident sind.

Eine solche Einbettung für  $G_1$  lässt sich zum Beispiel realisieren, indem wir die Knoten in  $G_1$  (gleichmäßig) auf einem Kreis mit entsprechend großem Radius verteilen.

Jeder Wert  $d_{i,a} \in D_i$  wird durch einen Knoten  $u_{i,a}$  in  $G_2$  repräsentiert und wir betten  $u_{i,a}$  beliebig innerhalb des 1-Balles  $B_1(v_i)$  von  $v_i$  ein. Für jedes Wertepaar  $d_{i,a} \in D_i, d_{j,b} \in D_j$  enthält  $G_2$  genau dann die Kante  $\{u_{i,a}, u_{j,b}\}$ , wenn die Wertekombination  $(d_{i,a}, d_{j,b})$  zulässig ist.

Alle einzubettenden Kanten von  $G_1$  und  $G_2$  ergeben sich explizit (als Geradenstücke) zwischen den entsprechenden Knoten-Einbettungen. Insgesamt behaupten wir, dass sich die oben beschriebene Einbettung von  $G_1$  und  $G_2$  in Polynomialzeit realisieren lässt. Für ein kleines Beispiel der beschriebenen Konstruktion referenzieren wir auf Abbildung 10.

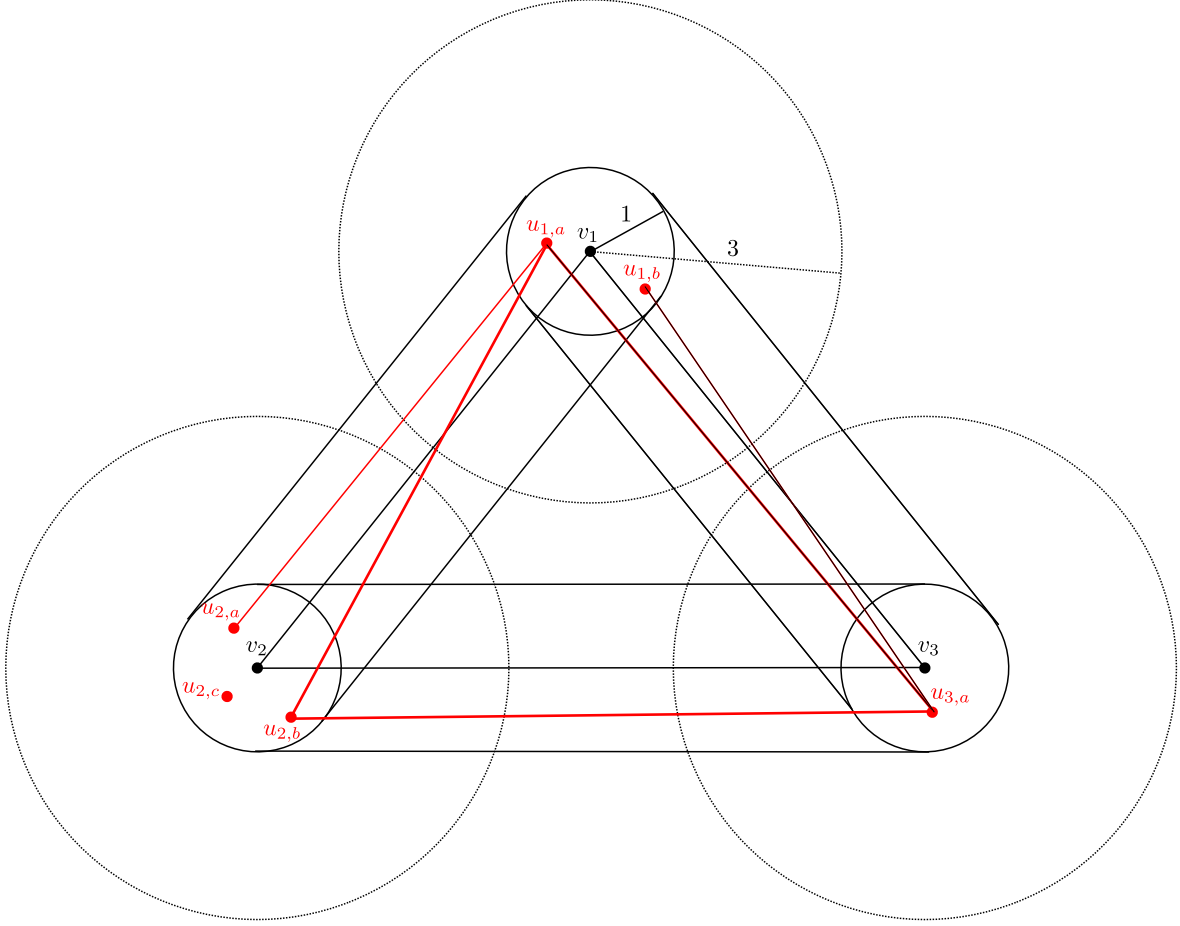


Abbildung 10: Eine Einbettung von  $G_1$  und  $G_2$  nach oben beschriebener Konstruktion. Zugrunde liegt die Instanz  $\langle X, D, C \rangle$  mit:  $X = \{x_1, x_2, x_3\}$ ,  $D_1 = \{d_{1,a}, d_{1,b}\}$ ,  $D_2 = \{d_{2,a}, d_{2,b}, d_{2,c}\}$ ,  $D_3 = \{d_{3,a}\}$  und den Bedingungen  $C_{1,2}, C_{1,3}, C_{2,3}$  mit den Relationen  $R_{C_{1,2}} = \{(d_{1,a}, d_{2,a}), (d_{1,a}, d_{2,b})\}$ ,  $R_{C_{1,3}} = \{(d_{1,a}, d_{3,a}), (d_{1,b}, d_{3,a})\}$  und  $R_{C_{2,3}} = \{(d_{2,b}, d_{3,a})\}$ . Es ist  $\varepsilon = |E_1| = 3$  und die hervorgehobene Kurve kennzeichnet das Bild einer (schwachen) 1-Platzierung von  $G_1$ . Offensichtlich ist damit  $\text{Min-Sum}_{(w)F}(G_1, G_2) \leq \varepsilon$  und die durch die  $\text{Min-Sum}_{(w)F}$  Zuordnung repräsentierte Auswahl von Werten  $\{d_{1,a}, d_{2,b}, d_{3,a}\}$  ist eine Lösung von  $\langle X, D, C \rangle$ .

Quelle: Eigene Darstellung

Wir wollen nun zeigen, dass

$$\min_{s: G_1 \rightarrow G_2} \sum_{e \in E_1} \delta_{(w)F}(e, s(e)) \leq \varepsilon \iff \langle X, D, C \rangle \text{ ist lösbar.}$$

” $\Leftarrow$ ”:

Sei  $\langle X, D, C \rangle$  lösbar.

Dann existiert eine Wertezuweisung  $(\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_n) \in D_1 \times D_2 \times \dots \times D_n$ , welche keine der existierenden Bedingungen in  $C$  verletzt.

Wir definieren eine Abbildung  $\tilde{s} : G_1 \rightarrow G_2$  mit  $\tilde{s}(v_i) = u_{\tilde{d}_i}$ . Nach Konstruktion von  $G_2$  existiert für jedes  $\tilde{d}_i$  ein assoziierter Knoten  $u_{\tilde{d}_i} \in V_2$ . Ferner existiert für eine beliebige Kante  $\{v_i, v_j\} \in E_1$  die Kante  $\{u_{\tilde{d}_i}, u_{\tilde{d}_j}\} \in E_2$  und da das mit  $\{u_{\tilde{d}_i}, u_{\tilde{d}_j}\}$  assoziierte Wertepaar  $(\tilde{d}_i, \tilde{d}_j)$  per Voraussetzung zulässig ist, ist  $\{u_{\tilde{d}_i}, u_{\tilde{d}_j}\}$  innerhalb von  $T_1(\{v_i, v_j\})$  eingebettet und die Randpunkte liegen respektive innerhalb von  $B_1(v_i)$  und  $B_1(v_j)$ . Damit ist  $\delta_{(w)F}(\{v_i, v_j\} \{u_{\tilde{d}_i}, u_{\tilde{d}_j}\}) \leq 1$ .

Wir erweitern  $\tilde{s}$  nun insofern, dass  $\{v_i, v_j\}$  durch  $\tilde{s}$  so auf  $\{u_{\tilde{d}_i}, u_{\tilde{d}_j}\}$  abgebildet wird, dass  $\tilde{s}$  den Flaschenhals-Abstand von  $\delta_{(w)F}(\{v_i, v_j\}, s(\{v_i, v_j\})) \leq 1$  realisiert. Bilden wir jede Kante von  $G_1$  auf diese Weise auf  $G_2$  ab, so ist  $\tilde{s}$  eine Graph-Zuordnung und  $\delta_{(w)F}(e, \tilde{s}) \leq 1$  für jedes  $e \in E_1$ , also ist  $\tilde{s}$  eine 1-Platzierung von  $G_1$ .

Da  $G_1$  insgesamt  $\varepsilon = |E_1|$  Kanten hat, ist

$$\sum_{e \in E_1} \delta_{(w)F}(e, \tilde{s}(e)) \leq \varepsilon. \quad (8)$$

Über die Identitäten

$$\min_{s: G_1 \rightarrow G_2} \sum_{e \in E_1} \delta_{(w)F}(e, s(e)) \leq \sum_{e \in E_1} \delta_{(w)F}(e, \tilde{s}(e))$$

und (8) erhalten wir insgesamt

$$\min_{s: G_1 \rightarrow G_2} \sum_{e \in E_1} \delta_{(w)F}(e, s(e)) \leq \varepsilon.$$

” $\Rightarrow$ ”:

Sei

$$\min_{s: G_1 \rightarrow G_2} \sum_{e \in E_1} \delta_{(w)F}(e, s(e)) \leq \varepsilon.$$

So ist  $s(e)$  offensichtlich für alle  $e \in E_1$  eine (schwache)  $\varepsilon$ -Platzierung von  $e$ , da ansonsten  $\delta_{(w)F}(\tilde{e}, s(\tilde{e})) > \varepsilon$  für ein  $\tilde{e} \in E_1$  und damit insbesondere  $\sum_{e \in E_1} \delta_{(w)F}(e, s(e)) > \varepsilon$ .

Ferner wollen wir argumentieren, dass  $s(e)$  für alle  $e \in E_1$  auch eine (schwache) 1-Platzierung von  $e$  ist.

Da jedes Bild einer Kante von  $G_1$  unter  $s$  zumindest eine  $\varepsilon$ -Platzierung ist, liegen die Randpunkte von  $s(e)$  insbesondere innerhalb der  $\varepsilon$ -Platzierungen der zu  $e$  inziden-

ten Knoten. Dann existiert aber für  $e$  insbesondere ein einfacher Weg in  $G_2$ , welcher vollständig innerhalb von  $T_1(e)$  liegt und mit den  $\varepsilon$ -Platzierungen der zu  $e$  inzidenten Knoten verbunden ist. Innerhalb dieser  $\varepsilon$ -Platzierungen liegen jeweils per Konstruktion auch die entsprechenden 1-Platzierungen der zu  $e$  inzidenten Knoten. Da  $s$  eine  $\text{Min-Sum}_{(w)F}$  Zuordnung ist, werden mögliche Optimierungen bei der Platzierung der Kante  $e$  stets realisiert. Eine Optimierung des  $\text{Min-Sum}_{(w)F}$  Abstands über die Nicht-Realisierung einer ansonsten möglichen 1-Platzierung ist aufgrund der entsprechenden Einbettung von  $G_1$  und  $G_2$  ausgeschlossen. Ist das Bild  $s(e)$  also eine  $\varepsilon$ -Platzierung von  $e$ , so ist  $s(e)$  insbesondere auch eine 1-Platzierung von  $e$ .

Da die Bilder aller Kanten von  $G_1$  unter  $s$  per Voraussetzung  $\varepsilon$ -Platzierungen sein müssen, ist für eine beliebige Kante  $\{v_i, v_j\} \in E_1$  ihr Bild  $s(\{v_i, v_j\})$  auch stets eine 1-Platzierung von  $\{v_i, v_j\}$ .

Im Falle des Fréchet-Abstands gilt damit, dass wir durch eine  $\text{Min-Sum}_F$  Zuordnung  $s : G_1 \rightarrow G_2$  das Bild  $s(\{v_i, v_j\})$  einer Kante eindeutig mit einer Kante  $\{u_{i,a}, u_{j,b}\} \in E_2$  identifizieren können. Gemeint ist damit genau die Kante, welche die entsprechenden 1-Platzierungen der Knoten  $v_i$  und  $v_j$  verbindet und in  $T_1(\{v_i, v_j\})$  liegt. Die Eindeutigkeit dieser Kante folgt direkt aus der Konstruktion von  $G_2$ .

Dabei ist gewährleistet, dass für die Kante  $\{u_{i,a}, u_{j,b}\} \in E_2$ , ihren assoziierten Wertezuweisungen  $(d_{i,a}, d_{j,b}) \in D_i \times D_j$  und die Bedingung  $C_{i,j} \in C$  an die Variablen  $x_i, x_j$  gilt, dass  $(d_{i,a}, d_{j,b}) \in R_{C_{i,j}}$ .

Ist also  $\text{Min-Sum}_F(G_1, G_2) \leq \varepsilon$ , so ist  $\langle X, D, C \rangle$  erfüllbar.

Für den schwachen Fréchet-Abstand gilt das Argument, dass wir das Bild einer Kante in  $G_1$  unter  $s$  eindeutig mit einer Kante in  $G_2$  identifizieren können, allgemein nicht.

Dies liegt daran, dass grundsätzlich jedes beliebige Knotenpaar  $u_i, u_j \in V_2$  mit  $u_i \in B_1(v_i)$ ,  $u_j \in B_1(v_j)$  eine Kante in  $G_2$  haben kann.

So existieren potentiell einfache Wege in  $G_2$  welcher über mehr als 2 Knoten von  $G_2$  innerhalb von  $T_1(\{v_i, v_j\})$  führen, deren Startpunkt in  $B_1(v_i)$  und deren Endpunkt in  $B_1(v_j)$  liegt. Ein solcher Weg  $W$  ist dabei - da er bedingt durch die Konstruktion von  $G_2$  vollständig innerhalb von  $T_1(\{v_i, v_j\})$  liegt - grundsätzlich als schwache 1-Platzierung der Kante  $\{v_i, v_j\}$  realisierbar und damit ist  $\delta_{wF}(\{v_i, v_j\}, W) \leq 1$ . Siehe Abbildung 11 für ein Beispiel.

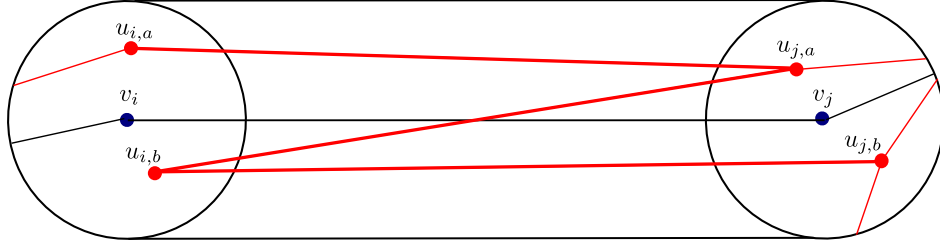


Abbildung 11: Durch die Hervorhebung haben wir das Bild eines Weges  $W$  in  $G_2$  markiert, welcher zwei 1-Platzierungen schwach miteinander verbindet. Für den schwachen Fréchet-Abstand genügt dabei die Tatsache, dass  $W$  vollständig in  $T_1(\{v_i, v_j\})$  liegt. Die mit der Kanten-Platzierung assoziierten Knoten  $u_{i,a}, u_{j,b}$  repräsentieren allerdings offensichtlich keine gültige Wertezuweisung für die Variablen  $x_i, x_j$ . Im Allgemeinen garantiert uns für den schwachen Fréchet-Abstand somit  $\text{Min-Sum}_{wF}(G_1, G_2) \leq \varepsilon$  noch keine Lösbarkeit von  $\langle X, D, C \rangle$ .

Quelle: Eigene Darstellung

Um diesen Fall auszuschließen und ein ähnliches Szenario wie für den Fréchet-Abstand zu gewährleisten, fügen wir mittig auf jeder Kante  $\{v_i, v_j\} \in E_1$  einen zusätzlichen Knoten  $v_{ij}$  in  $G_1$  ein, so dass  $B_1(v_{ij}) \cap B_1(v_i) = B_1(v_{ij}) \cap B_1(v_j) = \emptyset$  gilt. Abbildung 12 zeigt dies anhand eines Beispiels. Dafür ist es unter Umständen notwendig, initial ein größeres  $\varepsilon$  zu wählen, damit die beschriebene Konstruktion (auch für  $m_1 \leq 3$ ) stets gelingt.

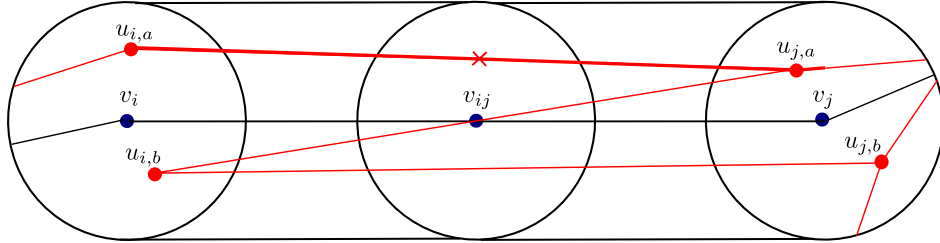


Abbildung 12: Die Kante  $\{v_i, v_j\}$  zerfällt nach dem Hinzufügen von  $v_{ij}$  in die zwei Kanten  $\{v_i, v_{ij}\}, \{v_{ij}, v_j\}$ . Ein mögliches Bild von  $v_{ij}$  unter  $s$  ist in der Abbildung durch ein rotes Kreuz gekennzeichnet. Jede Kante in  $G_2$  von  $B_1(v_i)$  nach  $B_1(v_j)$  induziert eine eindeutige 1-Platzierung des Knotens  $v_{ij}$ . Da  $B_1(v_{ij})$  keine Knoten von  $G_2$  enthält, ist die Konkatenation der Bilder von  $\{v_i, v_{ij}\}$  und  $\{v_{ij}, v_j\}$  unter  $s$  eindeutig mit einer Kante in  $G_2$  identifizierbar.

Quelle: Eigene Darstellung

Wir bezeichnen mit  $\tilde{G}_1 = (\tilde{V}_1, \tilde{E}_1)$  den Graphen, den wir durch die oben beschriebene Transformation erhalten.

Gilt nun

$$\min_{s: \tilde{G}_1 \rightarrow G_2} \sum_{e \in E_1} \delta_{wF}(e, s(e)) \leq 2m_1,$$

so erhalten wir für jede Kante  $\{v_i, v_j\} \in E_1$  und ihre Zerlegung  $\{v_i, v_{ij}\}, \{v_{ij}, v_j\} \in \tilde{E}_1$  über  $s(\{v_i, v_{ij}\})$  und  $s(\{v_{ij}, v_j\})$  die eindeutige, assoziierte Kante  $\{u_{i,a}, u_{i,b}\} \in E_2$ . Die durch  $\{u_{i,a}, u_{j,b}\} \in E_2$  repräsentierte Wertezuweisung  $(d_{i,a}, d_{j,b})$  verletzt dabei nicht die Bedingung an die durch  $v_i$  und  $v_j$  repräsentierten Variablen  $x_i$  und  $x_j$ , und ist damit gültig.

Insgesamt ist  $\langle X, D, C \rangle$  somit lösbar. □



## 5 Ausblick

In dieser Arbeit haben wir uns, motiviert durch das Ziel, optimale Zuordnungen auf geometrischen Graphen zu untersuchen, mit dem  $\text{Min-Sum}_{(w)F}$  Abstand beschäftigt. Dabei haben wir die Komplexität dieses Abstandsmaßes für allgemeine Graphen geklärt und - falls  $G_1$  ein Baum ist und wir die Menge der Graph-Zuordnungen (wie in 3.1 beschrieben) einschränken - einen Polynomialzeit-Algorithmus zur Berechnung beschrieben.

Dabei verbleiben noch einige offene Fragen bezüglich der Komplexität unter anderen als den hier untersuchten Bedingungen. Buchin et al. [1] gehen zum Beispiel davon aus dass die Berechnung des  $\text{Min-Sum}_{(w)F}$  Abstands auch auf planaren Graphen und auf Bäumen ohne die notwendige Einschränkung (aus 3.1.2) NP-schwer ist. Formale Beweise dieser Annahmen stehen dabei noch aus.

Allgemein beschreibt der  $\text{Min-Sum}_{dist}$  Abstand eine ganze Klasse von Abstandsmaßen auf Graphen, deren Mitglieder sich durch die Wahl eines geeigneten Abstandsmaßes für die Messung der Kanten-Zuordnungen unterscheiden. Grundsätzlich ließen sich also die Eigenschaften des  $\text{Min-Sum}_{dist}$  Abstands unter der Verwendung von anderen Abstandsmaßen betrachten. Hier bietet sich zum Beispiel für geometrische Graphen insbesondere der diskrete Fréchet-Abstand (siehe dazu zum Beispiel Aronov et. al [8]) als möglicher Kandidat an. Wir gehen davon aus, dass wir bei der Wahl des diskreten Fréchet-Abstands die in 3.1.2 erwähnte Einschränkung zur Gewährleistung der polynomialen Laufzeit nicht mehr benötigen. Eine formale Auseinandersetzung mit diesem Ansatz steht dabei allerdings noch aus.

Letzlich haben wir bei unserer Auseinandersetzung mit dem  $\text{Min-Sum}_{(w)F}$  Abstand nur ein Mitglied dieser oben beschriebenen Klasse betrachtet und damit insgesamt nur einen kleinen Exkurs in den Raum der noch offenen Möglichkeiten gewagt.



# Literatur

- [1] Maike Buchin, Bernhard Kilgus. Distance Measures for Embedded Graphs - Optimal Graph Mappings. *European Workshop on Computational Geometry*, 2020.
- [2] Hugo Akitaya, Maike Buchin, Bernhard Kilgus, Stef Sijben, Carola Wenk. Distance measures for embedded graphs *Computational Geometry*, Volume 95, 2021.
- [3] Hemlut Alt, Michael Godau. Computing the Fréchet distance between two polygonal curves. *Int. Journal of Computational Geometry and Applications*, 5:75-91, 1995.
- [4] Helmut Alt, Alon Efrat, Günter Rote, and Carola Wenk. Matching planar maps. *Journal of Algorithms*, 49(2):262 – 283, 2003.
- [5] Joachim Gudmundsson, Martin P. Seybold, Sampson Wong. Map matching queries on realistic input graphs under the Fréchet distance. *In Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) (pp. 1464-1492). Society for Industrial and Applied Mathematics, 2023.*
- [6] Kevin Buchin, Maike Buchin, Joachim Gudmundsson, Aleksandr Popov, Sampson Wong. Map Matching Queries Under Fréchet Distance on Low-Density Spanners. *In The 39th European Workshop on Computational Geometry, 2023.*
- [7] Richard M. Karp, Reducibility among combinatorial problems. *Springer Berlin Heidelberg, 2010.*
- [8] Boris Aronov, Sariel Har-Peled, Christian Knauer, Yusu Wang, Carola Wenk. Fréchet distance for curves, Revisited. *In Proc. 14th Annu. European Sympos. Algorithms (ESA), pages 52–63, 2006.*