

# Beschriften gewichteter Punkte mit verschiebbaren Labeln

## Studienarbeit

Moritz Minzloff

Betreuer: Dr. Alexander Wolff, Étienne Schramm

Universität Karlsruhe

Fakultät für Informatik

Institut für Logik, Komplexität und Deduktionssysteme (ILKD)

07. März 2005

## Zusammenfassung

Das Erstellen und Beschriften von Karten und technischen Zeichnungen ist eine aufwendige Arbeit. Besonders bei großen Datenmengen ist man an einer Automatisierung der Aufgabe interessiert. Man stellt sich das Problem, eine vorgegebene Punktmenge durch Label (in Form von Rechtecken oder anderen geometrischen Formen) zu „beschriften“. Vermöge einer Gewichtung der Punkte sind Prioritäten zwischen den Labeln festgelegt. Ein Algorithmus soll nun eine „sinnvolle“ Platzierung der Label finden, die den Prioritäten bestmöglich nachkommt. In den meisten Modellen führt dies auf ein NP-vollständiges Problem. Deswegen ist man an Approximationsalgorithmen interessiert, die in polynomieller Zeit laufen und deren Lösung beliebig nahe an der optimalen Lösung liegen, so genannten *Polynomialzeit-Approximationsschemata* (polynomial time approximation schemes, kurz PTAS).

Die vorliegende Arbeit untersucht ein einfaches Schiebemodell, in dem die Label (in Form von Rechtecken) den zugehörigen Punkt am unteren Rand berühren müssen. Es wird gezeigt, dass man unter bestimmten Voraussetzungen das Problem in polynomieller Zeit diskretisieren und auf bestehende Lösungsansätze zurückgreifen kann. Anschließende Überlegungen beweisen, dass es genügt, das Verhältnis zwischen maximaler Labellänge und -höhe zu beschränken, um diese Voraussetzungen zu erfüllen.

Im allgemeinen Fall bleibt die Frage nach einem PTAS jedoch weiterhin ungelöst. Es wird aber immerhin eine  $(2/3 - \epsilon)$ -Approximation angegeben.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
1.1	Bisherige Resultate . . . . .	2
1.2	Problemdefinition . . . . .	4
<b>2</b>	<b>Diskretisierung des Problems</b>	<b>6</b>
2.1	Reduktionsidee . . . . .	6
2.2	Der Reduktionssatz . . . . .	7
2.2.1	Normale Beschriftungen . . . . .	8
2.2.2	Schwacher Reduktionssatz . . . . .	9
2.2.3	Allgemeiner Reduktionssatz . . . . .	12
<b>3</b>	<b>Stutzen von gewichteten Graphen</b>	<b>17</b>
3.1	Stockwerkfunktionen . . . . .	17
3.1.1	Grundlagen . . . . .	17
3.1.2	Außenplanare Graphen . . . . .	19
3.1.3	Einteilungen . . . . .	25
3.2	Allgemeinere Stockwerkfunktionen . . . . .	29
3.2.1	Probleminstanzen mit Labellängenbeschränkungen . . . . .	30
3.2.2	Grenzen der Methode . . . . .	31
3.3	Weitere Ansätze . . . . .	33
3.3.1	Induktion über die Kardinalität von Einteilungen . . . . .	33
3.3.2	Baumzerlegungen . . . . .	33
	<b>Abschließende Bemerkungen</b>	<b>35</b>
	<b>Literatur</b>	<b>37</b>

# Kapitel 1

## Einleitung

Beschriftungen von Diagrammen, Karten, Graphen oder technischen Zeichnungen spielen eine Schlüsselrolle bei der Visualisierung von Information. Erst das Hinzufügen von *Labeln* an bestimmten Punkten, Gebieten und anderen Objekten ermöglicht es Information zu vermitteln. Dabei steigt bei einer wachsenden Menge an Information, die veranschaulicht werden sollen, auch das Interesse, das Platzieren von Beschriftungen zu automatisieren.

### 1.1 Bisherige Resultate

Bei den verschiedenen Merkmalen, die beschriftet werden müssen, liegt der Hauptfokus der algorithmischen Geometrie auf der Beschriftung von einzelnen Punkten. Label werden meist als achsenparallele Rechtecke modelliert [PSS<sup>+</sup>03, AvKS98]. Man unterscheidet zwischen Varianten mit Rechtecken beliebiger Höhen und Längen oder Beschränkungen für diese Größen. So werden im Fall des vorliegenden Textes Rechtecke mit gleicher Höhe betrachtet. Auch weitere geometrische Formen wie beispielsweise Kreisscheiben finden Verwendung [DMM02, EJS01]. Üblicherweise wird gefordert, dass sich zwei Label nicht überschneiden, den jeweils zu beschriftenden Punkt jedoch mit ihrem Rand berühren. Im Rahmen dieser Ausarbeitung sind die einzelnen Punkte zusätzlich mit Gewichten versehen. Das Problem, das sich stellt, ist eine gewichtsmaximale Beschriftung einer Teilmenge der Punkte zu finden, deren Labels den genannten Bedingungen genügen. Andere Zielfunktionen, auf die hier nicht näher eingegangen wird, arbeiten ohne Gewichte und es gilt möglichst viele Label zu platzieren [vKSW99] oder das Ziel ist jeden Punkt mit einem möglichst großen Label zu versehen [DMM02].

Generell unterscheidet man zwischen *Festpositionsmodellen* und *Schiebemodellen*, die in der Literatur fixed-point models und slider-models genannt werden. In Festpositionsmodellen wird für jedes Label eine feste Menge von Positionen auf seinem Rand vorgegeben, an denen sich der zugehörige Punkt befinden muss. Beispielsweise sind das die vier Ecken der Label oder auch deren Seitenmitten. Schiebemodelle erlauben einem Label den Punkt an beliebigen Stellen ihrer Seiten zu berühren. Die gängigsten Varianten beider Modelltypen

sind in Abbildung 1.1 aufgeführt. Dabei steht jedes Rechteck für eine zulässige Position des Labels. Bei den Schiebemodellen deuten die Pfeile an, dass das Rechteck entlang der angezeigten Richtung verschoben werden darf. Poon et al. liefern in [PSS<sup>+</sup>03] einen Überblick über die Verhältnisse der gewichtsmaximalen Beschriftungen unter den verschiedenen Modellen. Abbildung 1.2 zeigt das Festpositionsmodell \*P, das für die vorliegende Arbeit von Interesse ist: In diesem wird jedem Label eine individuelle (endliche) Menge von Positionen für den zugehörigen Punkt vorgegeben.

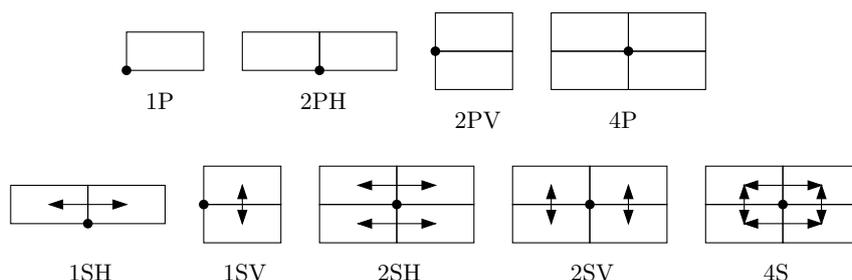


Abbildung 1.1: Die gängigsten Beschriftungsmodelle.

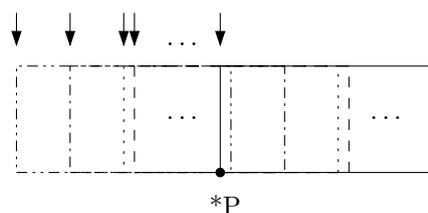


Abbildung 1.2: Im Modell \*P wird für jedes Label eine individuelle (endliche) Menge an Positionen für den zugehörigen Punkt vorgegeben.

Unglücklicherweise stellt sich die Suche nach einer gewichtsmaximalen Beschriftung bereits in einfachen Modellen als NP-vollständig heraus. So beweisen Formann und Wagner [FW91] die NP-Vollständigkeit im 4P-Modell. Für die Modelle 2PH und 2PV konnten Formann und Wagner hingegen einen Algorithmus angeben, der das entsprechende Entscheidungsproblem (Gibt es eine überschneidungsfreie Beschriftung für eine gegebene Menge von Labeln?) in polynomieller Zeit löst. Entsprechende Beweise für die Komplexität unter den Schiebemodellen 1SH und 4S finden sich in [PSS<sup>+</sup>03] bzw. [vKSW99]. Auch weitere Varianten, die beispielsweise das 1P-Modell verwenden aber Höhe und Breite der Label bei gleichbleibendem Flächeninhalt variieren lassen, sind NP-schwer [IL03]. Einen guten Überblick über den aktuellen Stand der Forschung liefert die *Map-Labeling Bibliography* [WS96].

Für die Festpositionsmodelle (inklusive \*P) ist zwar ein PTAS bekannt [AvKS98], nicht jedoch für die Schiebemodelle. Da das Beschriftungsproblem bereits dann NP-vollständig ist, wenn die Label nur entlang der horizontalen Achse verschoben werden dürfen, beschränkt sich die folgende Ausarbeitung auf die Suche nach einem PTAS für das Modell 1SH. Zu Beginn wird im ersten Abschnitt eine formale Definition des Problems angegeben. Der zweite Abschnitt

zeigt, wie man unter bestimmten Voraussetzungen Beschriftungen im Modell 1SH durch Beschriftungen im Modell \*P approximieren kann. Die Voraussetzungen werden im dritten Abschnitt näher untersucht. In diesem wird gezeigt, dass eine große Klasse von Probleminstanzen die Voraussetzungen für Approximierbarkeit erfüllt. Ein PTAS für den allgemeinen Fall wird in dieser Arbeit jedoch nicht gefunden.

## 1.2 Problemdefinition

Bevor die Idee für einen Approximationsalgorithmus in den folgenden Kapiteln genauer ausgeführt wird, soll zuerst das gestellte Problem formalisiert werden: Gegeben seien im Weiteren eine Punktmenge  $P = \{p_1, \dots, p_n\} \subseteq \mathbb{R}^2$  sowie eine Funktion  $w : P \rightarrow \mathbb{R}_{\geq 0}$ , die jedem Punkt ein Gewicht zuordnet und eine Funktion  $l : P \rightarrow \mathbb{R}_{>0}$ , die die Labellängen festlegt. Dabei wird  $l_i$  verkürzt für  $l(p_i)$  geschrieben. Für jeden Punkt  $p_i = (x_i, y_i)$  sei  $L_i := (x_i - l_i, x_i) \times (y_i, y_i + 1)$  und  $L_i(r) := \{(x, y) + (r, 0) \mid (x, y) \in L_i\}$  das um  $r$  nach rechts verschobene Label  $L_i$ . Das Tupel  $(P, w, l)$  heißt *Probleminstanz*.

In diesem Kontext gelangt man zu folgender

**Definition 1.1.** Eine *Beschriftung*  $B$  einer Probleminstanz  $(P, w, l)$  ist eine Teilmenge

$$B \subseteq \{L_i(r) \mid i \in \{1, \dots, n\}, r \in [0, l_i]\},$$

für die zusätzlich gilt:

$$\forall L_i(r), L_j(s) \in B, L_i(r) \neq L_j(s) : i \neq j, (L_i(r)) \cap (L_j(s)) = \emptyset.$$

*Bemerkung.* Anschaulich gesprochen bestehen Beschriftungen aus höchstens  $n$  (paarweise verschiedenen) Labels  $L_i$ , die parallel zur  $x$ -Achse verschoben in die Ebene  $\mathbb{R}^2$  gelegt werden, sodass

1. das verschobene Label  $L_i$  den Punkt  $p_i$  am Rand berührt,
2. sich keine zwei verschobenen Labels schneiden.

Abbildung 1.3 verdeutlicht die vorangegangenen Formalismen mit einem Beispiel.

**Definition 1.2.** Sei  $B$  eine Beschriftung. Das *Gewicht* eines Elements  $L_i(r) \in B$  ist durch  $w(i) := w(p_i)$  gegeben. Das *Gewicht*  $w(B)$  einer Beschriftung  $B$  ist das Gesamtgewicht aller enthaltenen Elemente:

$$w(B) := \sum_{L_i(r) \in B} w(i)$$

*Problemstellung.* Gesucht ist eine gewichtsmaximale Beschriftung.

Im weiteren Verlauf dieses Textes wird dieses Problem mit 2d-1SH bezeichnet.

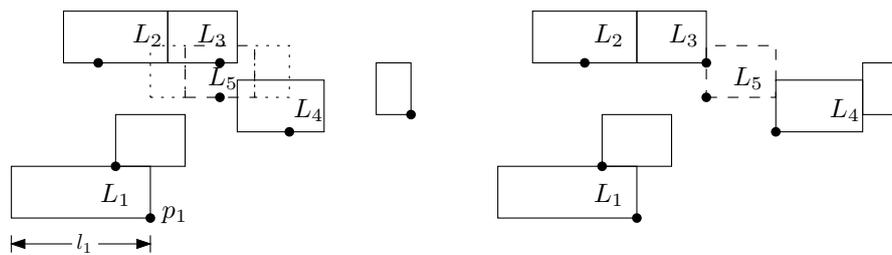


Abbildung 1.3: Zwei Beispielbeschriftungen. Das Label  $L_5$  konnte in die linke Beschriftung nicht zusätzlich aufgenommen werden, da es in jeder möglichen Position mindestens eines der Labels  $L_2$ – $L_4$  geschnitten hätte. Durch Verschieben der Labels kann man aber genug Platz für  $L_5$  schaffen, siehe rechte Beschriftung.

## Kapitel 2

# Diskretisierung des Problems

In diesem Kapitel wird das Problem 2d-1SH auf ein Beschriftungsproblem mit \*P-Modell zurückgeführt. In diesem Modell wird für jedes Label eine endliche Menge von möglichen Positionen vorgegeben, an denen sich das Label befinden darf. Das so diskretisierte Problem kann dann mittels eines bereits bekannten PTAS gelöst werden, siehe [AvKS98] (oder [EJS01], falls die maximale Labellänge beschränkt ist). Somit ist die grundlegende Idee, eine solche Menge von Positionen zu berechnen, dass die Einschränkung auf das diskretisierte Problem keinen großen Verlust beim Optimalgewicht nach sich zieht.

Auf dem Weg dorthin wird der Begriff des *Beschriftungsgraphen* eingeführt, der es ermöglicht graphentheoretische Überlegungen im Kontext von Beschriftungen anzustellen. Mit dessen Hilfe wird dann der *Reduktionssatz* bewiesen, der für einen Spezialfall die Existenz von Mengen der eben beschriebenen Gestalt garantiert.

### 2.1 Reduktionsidee

Zu gegebener Probleminstanz  $(P, w, l)$  soll für fest vorgegebenes  $\epsilon > 0$  eine Konstruktionsvorschrift angegeben werden, die eine Familie von (endlichen) Mengen  $M_i \subseteq [0, l_i]$  mit folgenden Eigenschaften liefert:

- (M1)  $m := \sum_{i \in I} |M_i|$  ist polynomiell in  $n$ ,
- (M2)  $\{M_i \mid i \in I\}$  lässt sich in polynomieller Zeit  $M(n)$  berechnen und
- (M3) für jede Beschriftung  $B$  von  $(P, w, l)$  existiert eine Beschriftung  $B'$  mit:  
 $(i, r) \in B' \Rightarrow r \in M_i$  und  $w(B') \geq (1 - \epsilon)w(B)$ .

Die Berechtigung für diese Zielvorgabe liefert folgende

**Proposition 2.1.** Existieren Mengen mit Eigenschaften (M1), (M2) und (M3), so gibt es für beliebige  $\alpha, \epsilon > 0$  auch einen  $(1 - \alpha)(1 - \epsilon)$ -Approximationsalgorithmus

für das Problem 2d-1SH, der in  $M(n) + A_\alpha(m)$  Zeit läuft. Dabei ist  $A_\alpha(m)$  die Laufzeit eines  $(1 - \alpha)$ -Approximationsalgorithmus für das diskrete Beschriftungsproblem mit Modell \*P.

*Beweis.* Seien also Mengen  $M_i$  mit (M1), (M2) und (M3) gegeben. Diese definieren ein diskretes Beschriftungsproblem mit Modell \*P: Anstatt wie bei 2d-1SH alle verschobenen Labels  $L_i(r)$  für  $r \in [0, l_i]$  in Betracht zu ziehen, berücksichtige nur noch Positionen  $r$  aus  $M_i$ . Ist nun  $B_{\text{opt}}$  die gewichtsmaximale Beschriftung (im Schiebemodell), so besagt Eigenschaft (M3), dass eine Beschriftung des diskreten Problems existiert, deren Gewicht maximal um einen Faktor  $\epsilon$  von  $w(B_{\text{opt}})$  abweicht. Folglich gilt dies insbesondere auch für die optimale Lösung des diskreten Problems.

Betrachte nun folgende Vorgehensweise: Konstruiere in einem ersten Schritt die Mengen  $M_i$  in  $M(n)$  Zeit. Nach (M2) ist dies möglich. Verwende anschließend die Label  $L_i$  zusammen mit den Mengen  $M_i$  als Eingabe eines PTAS für das diskrete Beschriftungsproblem. Nach  $A_\alpha(m)$  Zeit liefert dessen Ausgabe eine Approximation der optimalen Lösung des diskreten Problems, die wiederum eine Approximation an die optimale Lösung des Ursprungsproblems darstellt. (M1) garantiert, dass  $m$  polynomiell in  $n$  ist und somit auch  $A_\alpha(m)$ .  $\square$

Beispielsweise beschreiben Agarwal, van Kreveld und Suri in [AvKS98] einen  $k/(k+1)$ -Approximationsalgorithmus für \*P, der in  $O(m^{2k-1})$  Zeit läuft, wobei  $m$  die Anzahl aller Positionen aller Labels ist. Das Verfahren verwendet *stabbing lines* (siehe Abschnitt 3.1.3) um eine gegebene Menge von Label zu unterteilen und berechnet mit Hilfe von dynamischem Programmieren jeweils für die Teilmengen eine gewichtsmaximale überschneidungsfreie Teilmenge aller Label<sup>1</sup>. Wenn man (M1), (M2) und (M3) erfüllen kann, erhält man also eine  $(k/(k+1))(1 - \epsilon)$ -Approximation des 2d-1SH Problems mit Gesamtlaufzeit  $M(n) + O(m^{2k-1})$ .

## 2.2 Der Reduktionssatz

In diesem Abschnitt sollen im Rahmen des Reduktionssatzes Voraussetzungen präzisiert werden, unter denen man Mengen mit den Eigenschaften (M1), (M2) und (M3) berechnen kann. Der Beweis wird durch explizite Angabe eines Algorithmus erfolgen. Für den Beweis des Reduktionssatzes beschränkt man zuerst den Kontext auf *normale Beschriftungen*. Anschließend Überlegungen zeigen, dass der Satz auch in einem allgemeineren Rahmen Gültigkeit besitzt.

Für den weiteren Verlauf dieses Abschnitts sei eine Probleminstanz  $(P, w, l)$  fest vorgegeben.

---

<sup>1</sup>Dieser Algorithmus verwendet im Original topologisch abgeschlossene Rechtecke als Eingabe. Im Rahmen dieser Ausarbeitung wird hingegen mit offenen Rechtecken gearbeitet. Mit kleinen Änderungen funktioniert der Algorithmus aus [AvKS98] aber auch in diesem Fall. Es muss bei der Berechnung der Schnittmenge von zwei Rechtecken allerdings durch einen zusätzlichen symbolischen Vergleich verhindert werden, dass zwei verschobene Label des gleichen Punktes ausgewählt werden.

## 2.2.1 Normale Beschriftungen

**Definition 2.2.** Sei  $B$  eine Beschriftung und seien  $L_i(r), L_j(s) \in B$ . Das Label  $L_j(s)$  folgt aus dem Label  $L_i(r)$  in der Beschriftung  $B$ , falls die beiden Bedingungen

1.  $\nexists p = (x, y) \in P : x = x_j - l_j + s$
2.  $x_i + r = x_j - l_j + s$  und  $(y_i, y_i + 1) \cap (y_j, y_j + 1) \neq \emptyset$

erfüllt sind. In diesem Fall schreibt man  $L_i(r) \succ_B L_j(s)$  oder, falls klar ist, welche Beschriftung  $B$  gemeint ist, auch kurz  $L_i(r) \succ L_j(s)$ .

*Bemerkung.* Anschaulich folgt  $L_j(s)$  aus  $L_i(r)$  genau dann, wenn der rechte Rand des verschobenen Labels  $L_i(r)$  den linken Rand von  $L_j(s)$  berührt und ferner diese Berührung weder an der  $x$ -Koordinate eines Punktes aus  $P$  noch ausschließlich an zwei Labelecken stattfindet. Offensichtlich ist die Relation  $\succ$  auf der Menge der Label einer Beschriftung asymmetrisch und weder reflexiv noch transitiv.

**Definition 2.3.** Sei  $B$  eine Beschriftung und  $L_i(r) \in B$ . Dann heißt  $L_i(r)$  *normal* in  $B$ , wenn eine der folgenden Bedingungen zutrifft:

- (L1)  $r = 0$ ,
- (L2)  $\exists p = (x, y) \in P : x = x_i - l_i + r$  oder
- (L3)  $\exists L \in B : L \succ L_i(r)$ .

*Bemerkung.*  $L_i(r)$  ist in  $B$  also genau dann normal, wenn entweder sein rechter Rand den Punkt  $p_i$  berührt (L1) oder der linke Rand des Labels die  $x$ -Koordinate eines Punktes (L2) oder den rechten Rand eines anderen Labels (L3) berührt, siehe Abbildung 2.1

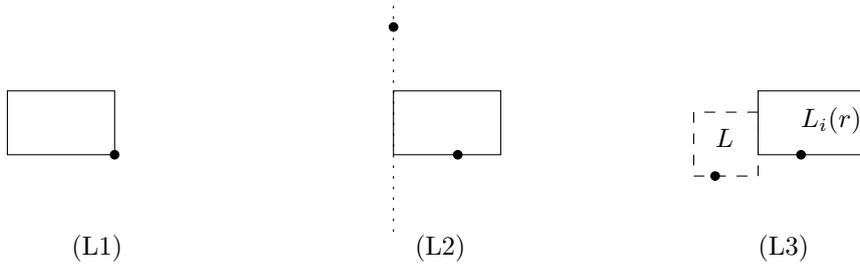


Abbildung 2.1: Beispiel zur Definition von normalen Label einer Beschriftung.

**Definition 2.4.** Eine Beschriftung  $B$  heißt *normal*, falls alle  $L_i(r) \in B$  in  $B$  normal sind.

*Bemerkung.* Betrachtet man auf der Menge aller Beschriftungen folgende Äquivalenzrelation

$$B \cong B' \Leftrightarrow (\forall i : \exists r \in [0, l_i] : L_i(r) \in B \Leftrightarrow \exists s \in [0, l_i] : L_i(s) \in B'),$$

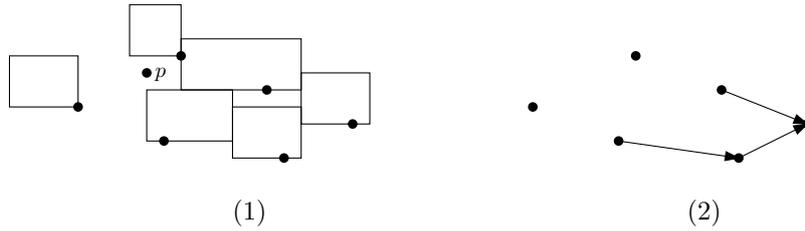


Abbildung 2.2: (1) zeigt eine (normale) Beschriftung und (2) den dazugehörigen Beschriftungsgraph. Man beachte, dass der Punkt  $p$  *keinem* Knoten im Graphen entspricht, weil er nicht beschriftet wurde.

d.h. zwei Beschriftungen werden genau dann als äquivalent angesehen, wenn sie genau die gleichen Punkte beschriften, so ist durch Definition 2.3 tatsächlich eine Normalform gegeben. (Zur Definition von Normalformen siehe beispielsweise [Cal96].) Eine Normalform einer Beschriftung kann durch folgendes Vorgehen berechnet werden: Solange es nicht-normale Label gibt, wähle ein  $L_i(r)$  von diesen und ersetze es durch ein normales Label  $L_i(s)$ , wobei  $s < r$ . Dies ist immer möglich und das Verfahren terminiert, da die Label strikt nach links gerückt werden und es sonst ein Label  $L_i$  gäbe, dass für unendlich viele  $r \in [0, l_i]$  normal wäre (und das ist nicht möglich). Für eine Beschriftung  $B$  bezeichne im weiteren  $F(B)$  eine äquivalente normale Beschriftung, deren Label sich mindestens so weit links befinden wie die entsprechenden in  $B$ . Abbildung 2.2 (1) zeigt ein Beispiel einer normalen Beschriftung.

Beachte: Die normalen Beschriftungen sind keine kanonischen Formen, da es zu gegebener Menge beschrifteter Punkte verschiedene normale Beschriftungen geben kann.

## 2.2.2 Schwacher Reduktionssatz

**Definition 2.6.** Sei  $B$  eine Beschriftung. Der *Beschriftungsgraph*  $G_B = (V, E)$  von  $B$  ist gegeben durch  $V := B$  und  $E := \{(v, w) \in E \mid v, w \in V : v \succ w\}$ .

*Bemerkung.* Nach Definition ist jeder Beschriftungsgraph  $G_B$  gerichtet und lässt sich auf den Knoten gewichten: Ordne dem Knoten  $v = L_i(r)$  das Gewicht  $w(v) := w(i)$  zu. Das *Gewicht* des Graphen sei dann durch  $w(G_B) := \sum_{v \in V} w(v)$  gegeben.

**Satz 2.7 (Schwacher Reduktionssatz).** Sei  $g : \mathbb{N} \rightarrow \mathbb{N}$  eine Funktion, sodass für jedes  $t \in \mathbb{N}$  und jede Beschriftung  $B$  eine normale Beschriftung  $B'$  mit  $w(B') \geq (1 - 1/t)w(B)$  existiert, in deren Beschriftungsgraph  $G_{B'}$  die Pfade kürzer als  $g(t)$  sind, dann gilt:

Für alle  $i = 1, \dots, n$  existieren Mengen  $M_i \subseteq [0, l_i]$ , die die Eigenschaften (M1), (M2) und (M3) erfüllen. Genauer ist  $|M_i| \in O(n^{g(t)})$  und die Mengen lassen sich in  $O(g(t)n^{O(g(t))})$  Zeit berechnen.

*Beweis.* Sei  $\epsilon > 0$  fest vorgegeben. Wähle  $t$ , sodass  $1/t < \epsilon$  und betrachte Algorithmus 1.

```

Eingabe : Die Mengen  $P$  sowie  $n (= |P|)$ ,  $l_1, \dots, l_n \in \mathbb{R}_{>0}$  und  $t \in \mathbb{N}$ 
Ausgabe : Mengen  $M_i$  mit Eigenschaften (M1), (M2) und (M3)
1  $T :=$  Mengen-Array[ $n, g(t)$ ];
   // Induktionsanfang
2 für  $i = 1, \dots, n$  tue
3    $T[i, 0] \leftarrow \{0\}$ ;
4   für  $j = 1, \dots, n$  tue
5     wenn  $x_j \leq x_i$  dann
6        $T[i, 0] \leftarrow T[i, 0] \cup (\{x_j - x_i + l_i\} \cap [0, l_i])$ ;
7     Ende
8   Ende
9 Ende
   // Induktionsschritt
10 für  $\tau = 1, \dots, g(t) - 1$  tue
11   für  $i = 1, \dots, n$  tue
12      $T[i, \tau] \leftarrow \emptyset$ ;
13     für  $j = 1, \dots, n$  tue
14       wenn  $x_j \leq x_i$  dann
15          $T[i, \tau] \leftarrow T[i, \tau] \cup ((T[j, \tau - 1] + x_j - x_i + l_i) \cap [0, l_i])$ ;
16       Ende
17     Ende
18   Ende
19 Ende
   // Zusammenfassung
20 für  $i = 1, \dots, n$  tue
21    $M_i \leftarrow \emptyset$ ;
22   für  $\tau = 0, \dots, g(t) - 1$  tue
23      $M_i \leftarrow M_i \cup T[i, \tau]$ ;
24   Ende
25 Ende

```

**Algorithmus 1** : Berechnung von Mengen mit Eigenschaften (M1)-(M3) unter den Voraussetzungen von Satz 2.6 in polynomieller Zeit.

*Korrektheit.* Eigenschaft (M2) folgt aus der Analyse der Laufzeit des Algorithmus. Bleibt zu zeigen, dass die Mengen  $M_i$ , wie sie der Algorithmus liefert, Eigenschaften (M1) und (M3) erfüllen. Aufgrund der Voraussetzung reicht es nachzuweisen, dass für eine beliebige normale Beschriftung  $B$  mit maximaler Pfadlänge  $t$  und  $L_i(r) \in B$  auch  $r \in M_i$  erfüllt ist. Ferner sollen die Mengen  $M_i$  natürlich von polynomieller Größe sein.

Sei also  $B$  eine normale Beschriftung und  $L_i(r) \in B$ . Der längste Pfad, dessen Ende bei  $L_i(r)$  ist, habe die Länge  $\tau$ . Behauptung:

$$r \in T[i, \tau] \text{ und } |T[i, \tau]| \in O(n^\tau)$$

Daraus folgt sofort, dass  $M_i = \bigcup_{\tau=0}^{g(t)-1} T[i, \tau]$  die gewünschten Eigenschaften besitzt. Der Beweis der Behauptung erfolgt per Induktion über  $\tau$ :

*Induktionsanfang,  $\tau = 0$ .* In diesem Fall gilt

$$\nexists L \in B : L \succ L_i(r).$$

Also folgt, da  $B$  normal ist, dass

$$r = 0 \text{ oder } r = x - x_i + l_i$$

für ein  $(x, y) \in P$ . Da  $r \in [0, l_i]$  nach Definition von Beschriftungen, reicht es im zweiten Fall nur die  $(x, y) \in P$  in Betracht zu ziehen, für die  $x \leq x_i$  gilt. Letztlich gilt offensichtlich  $|T[i, 0]| \leq n + 1 \in O(n)$ .

*Induktionsschritt,  $\tau - 1 \rightarrow \tau$ .* Jetzt gilt

$$\exists L_j(s) \in B : L_j(s) \succ L_i(r),$$

d.h. (wiederrum da  $B$  normal ist)  $x_i - l_i + r = x_j + s$  bzw.

$$r = x_j + s - x_i + l_i.$$

Der längste Pfad auf dem  $x_j$  liegt, hat Länge  $\tau - 1$  und nach Induktionsvoraussetzung gilt  $s \in T[j, \tau - 1]$ . Aus den gleichen Gründen wie beim Induktionsanfang müssen auch hier nur  $L_j(s) \in B$  mit  $x_j \leq x_i$  in Betracht gezogen werden. Also folgt:

$$T[i, \tau] = \bigcup_{(j,s) \in B, x_j \leq x_i} ((T[j, \tau - 1] + x_j - x_i + l_i) \cap [0, l_i])$$

Nach Induktionsvoraussetzung haben die Mengen  $T[j, \tau - 1]$  eine Größe in  $O(n^{\tau-1})$  und somit folgt  $|T[i, \tau]| \in nO(n^{\tau-1}) = O(n^\tau)$ .

*Laufzeitanalyse und Speicherbedarf.* Die Tabelle  $T$  enthält  $g(t)n$  Mengen der Größe höchstens  $O(n^{g(t)})$ . Ingesamter Speicherbedarf ist also in  $O(g(t)n^{O(g(t))})$ . Die Vereinigungen in Zeilen 6, 15 und 23 lassen sich als Vereinigungen sortierter Listen der Größe höchstens  $O(n^{g(t)})$  implementieren und beanspruchen so  $O(n^{g(t)})$  Zeit. Ingesamt werden diese Zeilen maximal  $n^2$ ,  $(g(t)-1)n^2$  und  $n^2$  mal durchlaufen, folglich liegt die Gesamtlaufzeit des Algorithmus in  $O(g(t)n^{O(g(t))})$ .  $\square$

### 2.2.3 Allgemeiner Reduktionssatz

Beschränkt man sich auf Probleminstanzen, die die Voraussetzung des Schwachen Reduktionssatzes erfüllen, liefert also der Satz zusammen mit Proposition 2.0 ein PTAS für 2d-1SH. Gelingt es zu zeigen, dass diese Voraussetzung immer erfüllt ist, oder, dass allgemeinere Voraussetzungen ausreichen und diese immer erfüllt sind, so erhält man für den generellen Fall ein PTAS. Dieser Aufgabe widmet sich der weitere Verlauf der vorliegenden Arbeit.

In diesem Abschnitt wird gezeigt, dass man im Schwachen Reduktionssatz die Forderung nach Normalität der approximierenden Beschriftung fallen lassen kann, wenn man den Begriff der Beschriftungsgraphen verallgemeinert. Die Idee ist, eine Beschriftung  $B$  durch  $F(B)$  zu ersetzen. (Man erinnere sich daran, dass  $F(B)$  eine zu  $B$  äquivalente Beschriftung ist, deren Label sich mindestens so weit links befinden wie in  $B$ .) Dies ist aber nicht ohne weitere Überlegungen möglich, will man die Voraussetzungen des Schwachen Reduktionssatzes verändern: Beim Wechsel von  $B$  zu  $F(B)$  können unter Umständen neue, zu lange Pfade entstehen. Dies ist genau dann der Fall, wenn zwei Label existieren, sodass das eine in  $F(B)$  aus dem anderen folgt, nicht aber in  $B$ . Abbildung 2.3 illustriert diese Situation und die folgende Proposition erfasst dies in formal.

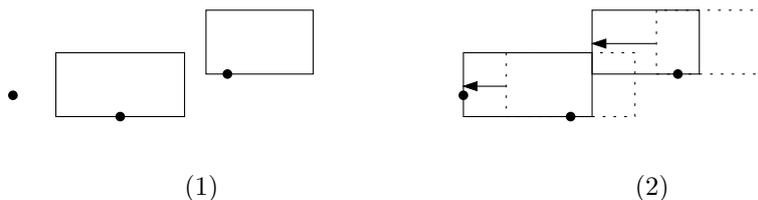


Abbildung 2.3: (a) Situation vor Überführung in eine normale Beschriftung: Keine Label folgt aus einem anderen. (b) Situation nach Überführung in eine normale Beschriftung: Jetzt folgt das rechte Label aus dem linken.

**Proposition 2.11.** Sei  $B$  eine Beschriftung. Existieren  $L_i(r), L_j(s)$  in  $B$  und  $L_i(r')$  und  $L_j(s')$  in  $F(B)$ , sodass

$$L_i(r) \not\prec_B L_j(s) \text{ und } L_i(r') \succ_{F(B)} L_j(s'),$$

dann gilt:

- (B1)  $x_i < x_j$
- (B2)  $\nexists(x, y) \in P : x_i + r \leq x \leq x_j - l_j + s$
- (B3)  $x_i + r \geq x_j - l_j$
- (B4)  $(y_i, y_i + 1) \cap (y_j, y_j + 1) \neq \emptyset$

*Beweis.* (B1)–(B4) folgen direkt aus den Definitionen von Beschriftungen und normalen Beschriftungen.  $\square$

Nun kann die ursprüngliche Definition von Beschriftungsgraphen ersetzt werden:

**Definition 2.12.** Sei  $B$  eine Beschriftung. Der *Beschriftungsgraph*  $G_B = (V, E)$  von  $B$  ist gegeben durch  $V := B$  und  $(L_i(r), L_j(s)) \in E$  genau dann, wenn eine der folgenden Bedingungen zutrifft:

1.  $L_i(r) \succ L_j(s)$  oder
2.  $L_i(r)$  und  $L_j(s)$  erfüllen (in dieser Reihenfolge) die Bedingungen (B1)-(B4)

*Bemerkung.* Eigentlich ist nur noch die zweite Bedingung relevant, da die erste die zweite impliziert.

Alle Folgerungen von Labels, die beim Wechsel einer Beschriftung  $B$  zu  $F(B)$  entstehen könnten, sind durch die neue Definition im Beschriftungsgraphen vorweggenommen. Damit ist es tatsächlich möglich in der Voraussetzung des Schwachen Reduktionssatzes „normal“ fallen zu lassen. Beachte dabei, dass der Beweis des Satzes nur benötigte, dass Beschriftungsgraphen gewichtet und gerichtet sind. Das ist bei der neuen Definition offensichtlich noch erfüllt und somit kann der Beweis direkt übernommen werden.

**Satz 2.13 (Reduktionssatz).** Sei  $g : \mathbb{N} \rightarrow \mathbb{N}$  eine Funktion, sodass für jedes  $t \in \mathbb{N}$  und jede Beschriftung  $B$  eine Beschriftung  $B'$  mit  $w(B') \geq (1 - 1/t)w(B)$  existiert, in deren Beschriftungsgraph  $G_{B'}$  die Pfade nicht länger als  $g(t)$  sind, dann gilt:

Für alle  $i = 1, \dots, n$  existieren Mengen  $M_i \subseteq [0, l_i]$ , die Eigenschaften (M1), (M2) und (M3) erfüllen. Genauer ist  $|M_i| \in O(n^{g(t)})$  und die Mengen lassen sich in  $M(n) = O(g(t)n^{O(g(t))})$  berechnen.

*Beweis.* Die Behauptung folgt aus den vorangegangenen Überlegungen sowie dem Schwachen Reduktionssatz.  $\square$

**Korollar 2.14.** Beschränkt man sich auf Beschriftungen, die die Voraussetzung des Reduktionssatzes erfüllen und sei  $g$  die daraus entnommene Funktion, so existiert ein  $(1 - \alpha)(1 - 1/t)$ -Approximationsalgorithmus für (das so eingeschränkte Problem) 2d-1SH, der in  $O(g(t)n^{O(g(t))}) + A_\alpha(O(n^{O(g(t))}))$  Zeit läuft.

*Beweis.* Folgt direkt aus dem Reduktionssatz sowie Proposition 2.0.  $\square$

Das Kapitel wird abgeschlossen, indem noch eine einige Eigenschaften von Beschriftungsgraphen notiert werden.

**Lemma 2.16.** Jeder Beschriftungsgraph  $G_B = (V, E)$  erfüllt folgende Eigenschaften:

- (G1)  $G_B$  ist planar,
- (G2)  $G_B$  ist azyklisch,
- (G3) Ist  $(v, w)$  ein Kante in  $G_B$ , dann existiert kein weiterer Pfad von  $v$  nach  $w$ .
- (G4) Alle Knoten in  $G_B$  haben Ein- und Ausgangsgrad höchstens 2,

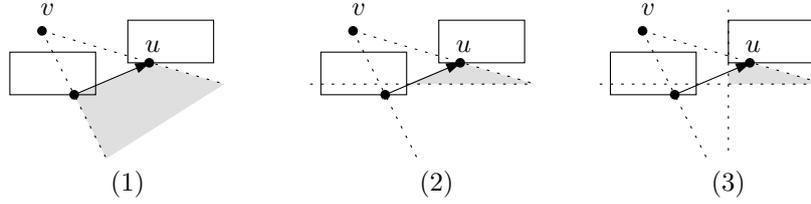


Abbildung 2.4: Um die bestehende Kante zu kreuzen, muss  $v$  offensichtlich mit einem Knoten  $w$  im grauen Bereich verbunden werden (1). Dieser darf aber nicht „zu tief“ liegen, ansonsten überschneiden sich nicht die  $y$ -Abschnitte der zugehörigen Labels (2). Auch darf  $w$  nicht vor linken Rand des  $u$  zugehörigen Labels liegen. Andererseits aber auch nicht hinter diesem Label, ansonsten überschneidet sein Label dieses oder liegt komplett dahinter, was auch nicht sein darf (3). An der übriggebliebenen Stelle ist aber kein Platz für ein weiteres Label!

*Beweis.* Ad (G1). Setzt man den Knoten  $L_i(r)$  an die Stelle  $p_i$  und stellt Kanten als Geradenstücke dar, dann erhält man eine planare Einbettung des Graphen. Anschaulich ist die Behauptung klar, siehe auch Abbildung 2.4.

Es lässt sich aber ebenso ein rigoroser Beweis führen: Nehme an, es würden sich zwei Kanten  $(L_i(r), L_j(s))$  und  $(L_{i'}(r'), L_{j'}(s'))$  schneiden. Es folgt direkt

$$(x_i, x_j) \cap (x_{i'}, x_{j'}) \neq \emptyset \text{ und } (y_i, y_j) \cap (y_{i'}, y_{j'}) \neq \emptyset \quad (2.1)$$

Ferner kann ohne Einschränkung vorausgesetzt werden, dass  $y_{i'} \leq y_i$ . Unterscheide nun folgende Fälle:

1.  $y_{i'} \leq y_i < y_{i'} + 1$ .

Mit Hilfe der Definition von Beschriftungsgraphen folgt dann:

$$x_i \notin [x_{i'} - l_{i'} + r', x_{i'} + r'] \cup [x_{i'} + r', x_{j'} - l_{j'} + s'] \cup [x_{j'}, \infty). \quad (2.2)$$

Denn ansonsten würden sich die Label  $L_i(r)$  und  $L_{i'}(r')$  überkreuzen oder die Kante zwischen  $L_{i'}(r')$  und  $L_{j'}(s')$  würde nicht existieren oder es gäbe einen Widerspruch zu 2.1. Unterscheide also die Fälle

- (a)  $x_i < x_{i'} - l_{i'} + r'$ .

Einerseits muss  $x_j - l_j + s < x_{i'}$  erfüllt sein, denn ansonsten könnte es keine Kante zwischen  $L_i(r)$  und  $L_j(s)$  geben, aber andererseits auch  $x_{i'} < x_j$ . Also gilt  $y_j \geq y_{i'} + 1 > y_{i'}$  und damit auch  $y_j > y_{j'}$ , da  $y_{j'} \in (y_{i'} - 1, y_{i'} + 1)$ . Somit gilt  $x_j > x_{j'}$ . (Ansonsten können sich die gegebenen Kanten nicht überschneiden:  $L_i(r)$  liegt links oberhalb von  $L_{i'}(r')$  und  $L_j(s)$  läge ebenfalls links oberhalb von  $L_{j'}(s')$ .) Dann gilt aber  $y_j \geq y_{j'} + 1$  und somit  $y_i > y_j - 1 \geq y_{j'}$ . Widerspruch zu 2.1!

- (b)  $x_{j'} - l_{j'} + s' < x_i < x_{j'}$ .

Es folgt sofort, dass  $y_i \geq y_{j'} + 1 > y_{j'}$ . Somit auch  $y_j \geq y_i - 1 \geq y_{j'} + 1 - 1 = y_{j'}$ . Da aber 2.1 erfüllt sein muß, gilt  $y_{j'} < y_j < y_{i'}$ . Somit gilt  $x_j - l_j + s > x_{j'} + s'$ , da sich sonst die Label  $L_j(s)$  und  $L_{j'}(s')$  schneiden würden. Dann können sich die Kanten aber nicht

schneiden, denn:  $L_i(r)$  liegt rechts oberhalb von  $L_{i'}(r')$  und ebenso liegt  $L_j(s)$  rechts oberhalb von  $L_{j'}(s')$ . Damit können sich die Kanten aber nicht schneiden!

2.  $y_i \geq y_{i'} + 1$ .

Da  $y_j \in (y_i - 1, y_i + 1)$  nach Voraussetzung, gilt in diesem Fall also  $y_j \in (y_{i'}, y_i + 1)$  und mit 2.1 sogar  $y_j \in (y_{i'}, y_{i'} + 1)$ , da  $y_{j'} \in (y_{i'} - 1, y_{i'} + 1)$ . Ferner folgt daraus und der Definition von Beschriftungsgraphen, dass

$$x_j \notin (-\infty, x_{i'} - l_{i'} + r'] \cup (x_{i'} - l_{i'} + r', x_{i'} + r') \cup [x_{i'} + r', x_{j'} - l_{j'} + s'). \quad (2.3)$$

Andernfalls würden sich die Kanten nicht kreuzen oder eine der beiden Kanten würde nicht existieren oder zwei Label der Beschriftung würden sich überschneiden. Unterscheide die Fälle

(a)  $x_i + r < x_{j'}$ .

In diesem Fall gilt  $x_j - l_j + s < x_{j'}$  und mit 2.3 folgt, dass  $y_j \geq y_{j'} + 1 > y_{i'}$  und damit  $y_i \geq y_{i'} + 1 > y_j \geq y_{j'} + 1 > y_{i'}$ . Das ist ein Widerspruch zu 2.1!

(b)  $x_i + r \geq x_{j'}$ .

Nach Voraussetzung kann  $L_i(r)$  nicht rechts von  $L_{j'}(s')$  liegen. Also gilt  $y_i \geq y_{j'} + 1 > y_{j'}$ , ansonsten würden sich die Label  $L_i(r)$  und  $L_{j'}(s')$  überschneiden. Daraus folgt sofort, dass  $y_j > y_i - 1 \geq y_{j'}$ . Nach Voraussetzung bereits  $y_i, y_j > y_{i'}$ , im Widerspruch zu 2.1!

*Ad (G2).* Offensichtlich, da aus  $(L_i(r), L_j(s)) \in E$  folgt, dass  $x_i < x_j$ .

*Ad (G3).* Sei also  $(L_i(r), L_j(s)) \in E$  und es existiere ein Pfad  $P$  von  $L_i(r)$  nach  $L_j(s)$ , der diese Kante nicht benutzt. Dann enthält  $P$  einen Knoten  $L_{j'}(s')$  der zwischen  $L_i(r)$  und  $L_j(s)$  liegt. Somit gilt:  $x_i + r \leq x_{j'} \leq x_j - l_j + s$ . Das ist aber ein direkter Widerspruch zur Kante zwischen  $L_i(r)$  und  $L_j(s)$ !

*Ad (G4).* Seien  $(L_i(r), L_{j_1}(s_1)), (L_i(r), L_{j_2}(s_2)), (L_i(r), L_{j_3}(s_3)) \in E$ . Daraus folgt:

$$(y_i, y_i + 1) \cap (y_{j_k}, y_{j_k} + 1) \neq \emptyset \quad \forall k = 1, 2, 3$$

und somit ohne Einschränkung

$$(y_{j_1}, y_{j_1} + 1) \cap (y_{j_2}, y_{j_2} + 1) \neq \emptyset.$$

Wiederum ohne Einschränkung folgt daraus sofort, dass  $x_{j_1} + s_1 < x_{j_2} - l_{j_2} + s_2$ . Aber dann auch:

$$x_i + r \leq x_{j_1} - l_{j_1} + s_1 < x_{j_1} + s_1 < x_{j_2} - l_{j_2} + s_2 \leq x_{j_2}$$

Das darf aber nicht sein, denn dann wäre für  $L_i(r)$  und  $L_{j_2}(s_2)$  Bedingung (B2) verletzt! Analog zeigt man, dass auch der Eingangsgrad jedes Knoten höchstens 2 ist.

□

*Bemerkung.* Eigenschaft (G4) ist hier nur der Vollständigkeit halber mit aufgelistet. Sie wird im weiteren Verlauf nie verwendet. Allerdings charakterisiert Lemma 2.15 selbst mit (G4) Beschriftungsgraphen nicht eindeutig. Man denke

sich beispielsweise einen Graphen, der zwei Punkte  $p_i$  und  $p_j$  der gegebenen Punktmenge verbindet, deren  $y$ -Koordinaten sich um mehr als 1 unterscheiden. Offensichtlich erfüllt dieser Graph die Eigenschaften (G1)-(G4), kann aber keiner Beschriftung entsprechen!

## Kapitel 3

# Stutzen von gewichteten Graphen

Im vorangegangenen Kapitel wurde gezeigt, dass unter den Voraussetzungen des Reduktionssatzes ein PTAS für 2d-1SH existiert. Genauer wurde bewiesen, dass dann ein  $(1 - \alpha)(1 - 1/t)$ -Approximationsalgorithmus mit Gesamtlaufzeit  $O(g(t)n^{O(g(t))}) + A_\alpha(O(n^{O(g(t))}))$  existiert. Dabei ist  $g$  die Funktion, die im Reduktionssatz verwendet wird.

In diesem Kapitel wird für einige Klassen von Beschriftungsgraphen bewiesen, dass die enthaltenen Graphen so „gestutzt“ werden können, dass sie die Voraussetzungen des Reduktionssatzes erfüllen: Die Idee ist, eine Menge von Knoten mit genügend geringem Gewicht aus dem Graphen zu entfernen, sodass die maximale Pfadlänge beschränkt ist. In den Fällen, in denen dies möglich ist, wird eine konkrete Funktion  $g$  zu Verwendung in Algorithmus 1 angegeben. Weiterhin werden Negativbeispiele aufgeführt, bei denen der verwendete Ansatz zumindest ohne weitere Änderungen nicht funktioniert.

### 3.1 Stockwerkfunktionen

Alle Fälle in der vorliegenden Ausarbeitung, in denen das Stutzen der Beschriftungsgraphen erfolgreich ist, basieren auf *Stockwerkfunktionen*. Diese teilt jedem Knoten eines Graphen ein bestimmtes „Stockwerk“ zu. Der grundlegende Gedanke wird sein, dass einerseits das Stutzen *innerhalb* der Stockwerke ein vergleichsweise leichtes Problem darstellt, andererseits die Anzahl, wie oft ein Pfad *zwischen* den Stockwerken wechselt, ebenfalls beschränkt ist.

#### 3.1.1 Grundlagen

**Definition 3.1.** Sei  $G = (V, E)$  ein Graph und  $f : V \rightarrow \mathbb{Z}$  eine Funktion, die folgende Eigenschaften erfüllt:

$$(S1) \quad \forall (v, w) \in E : f(w) - f(v) \in \{0, 1\},$$

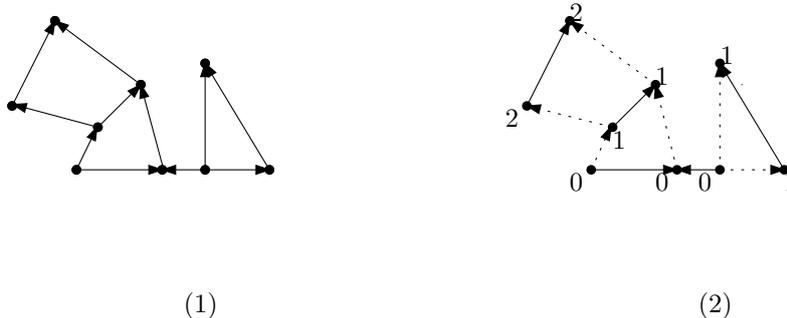


Abbildung 3.1: Ein Beispiel für einen Graphen (1) und den Werten einer Stockwerkfunktion auf seinen Knoten (2). Die gestrichelten Pfeile deuten Wechsel von einem Stockwerk zum nächsten an.

$$(S2) \quad \forall v \in V : |\{w \in V \mid (v, w) \in E \text{ und } f(v) = f(w)\}| \leq 1$$

In diesem Fall heißt  $f$  *Stockwerkfunktion von  $G$*  und die Mengen  $V_{i,f} := \{v \in V \mid f(v) = i\}$  heißen *Stockwerke von  $G$  bezüglich  $f$* .

Ist aus dem Kontext klar, welche Stockwerkfunktion gemeint ist, werden im Weiteren die Stockwerke  $V_{i,f}$  auch kurz mit  $V_i$  bezeichnet.

*Bemerkung.* Sei  $G$  ein Graph mit Stockwerkfunktion  $f$ . Die Eigenschaft (S2) sorgt dafür, dass die Subgraphen  $G[V_i]$  aus Wäldern von Bäumen bestehen. Dabei sind die Kanten zur jeweiligen Wurzel gerichtet. Ist nun  $(v, w)$  eine Kante in  $G$  und  $f(v) = i$ , so besagt (S1), dass die Kante entweder vollständig innerhalb des Stockwerks  $V_i$  liegt oder von  $V_i$  nach  $V_{i+1}$  führt. Ein Pfad in  $G$  der Länge  $L$  wechselt also höchstens  $L$  mal das Stockwerk und kann dabei pro Wechsel immer nur von einem Stockwerk zum direkten Nachfolger gelangen.

**Proposition 3.2.** Sei  $G = (V, E)$  ein knotengewichteter Wald. Dann existiert eine Teilmenge  $V'$  von  $V$  mit Gewicht höchstens  $1/t$ -mal dem von  $V$ , sodass alle Pfade in  $G[V \setminus V']$  maximal  $2t - 3$  lang sind.

*Beweis.* Sei  $G$  zunächst nur ein Baum. Die Graphen  $G_\tau$ ,  $\tau = 0, \dots, t - 1$  seien dadurch gegeben, dass man aus  $G$  alle Knoten der Stockwerke  $\tau, \tau + t, \tau + 2t, \dots$  entfernt. Dann sind die Längen aller Pfade in  $G_\tau$  durch  $2t - 3 = (t - 1) + (t - 2)$  begrenzt: Jeder Pfad kann zwischen maximal  $t - 1$  Stockwerken wechseln und diese kann ein Pfad höchstens einmal vollständig herauf- und wieder herunterlaufen.

Nach dem Schubfachprinzip gilt ferner für mindestens ein  $\tau_0$ , dass  $w(G_{\tau_0}) \geq (1 - 1/t)w(G)$ . Nimm als  $V'$  die Menge aller Knoten in den Stockwerken  $\tau_0, \tau_0 + t, \tau_0 + 2t, \dots$

Ist  $G$  im Allgemeinen ein Wald, verfähre in jedem einzelnen Baum wie beschrieben.  $\square$

**Satz 3.3.** Sei  $G$  ein knotengewichteter Graph mit Stockwerkfunktion  $f$ . Dann existiert eine Teilmenge  $V'$  von  $V$  mit Gewicht höchstens  $1/t$ -mal dem von  $V$ , sodass alle Pfade in  $G[V \setminus V']$  maximal  $(2t - 1)(4t - 6)$  lang sind.

*Beweis.* Man kann sehr ähnlich zum letzten Beweis vorgehen: Betrachte die Graphen

$$G_\tau = G[V \setminus \bigcup_{i \in 2t\mathbb{Z}} V_{\tau+i}], \quad \tau = 0, \dots, 2t - 1.$$

Jeder Pfad in  $G_\tau$  kann höchstens  $2t - 1$ -mal das Stockwerk wechseln. Wiederum muss für mindestens ein  $\tau_0$  die Aussage  $w(G_{\tau_0}) \geq (1 - 1/2t)w(G)$  erfüllt sein. Setze

$$V^{(1)} := \bigcup_{i \in 2t\mathbb{Z}} V_{\tau_0+i}.$$

Nun sind die einzelnen Stockwerke in  $G[V \setminus V^{(1)}]$  Wälder und wendet man auf diese Proposition 3.1 an, erhält man für jedes Stockwerk  $V_j$ ,  $j \in \mathbb{Z} \setminus (\tau_0 + 2t\mathbb{Z})$  eine Menge  $V_j^{(2)}$ , sodass die Pfade in  $G[V_j \setminus V_j^{(2)}]$  höchstens Länge  $(2t - 1)(4t - 6)$  besitzen und  $w(G[V_j \setminus V_j^{(2)}]) \geq (1 - 1/(2t))w(G[V_j])$  erfüllt ist. Setzt man

$$V^{(2)} = \bigcup_{j \in \mathbb{Z} \setminus (\tau_0 + 2t\mathbb{Z})} V_j^{(2)} \quad \text{und} \quad V' = V^{(1)} \cup V^{(2)}$$

so erhält man:

$$w(G[V \setminus V']) \geq (1 - 1/(2t))(1 - 1/(2t))w(G) \geq (1 - 1/t)w(G)$$

und

Alle Pfade in  $G[V \setminus V']$  haben maximale Pfadlänge  $(2t - 1)(4t - 6)$ . □

**Korollar 3.5.** Beschränkt man das Problem 2d-1SH auf Probleminstanzen, sodass zu jedem möglichen Beschriftungsgewicht eine Beschriftung existiert, deren Beschriftungsgraph eine Stockwerkfunktionen besitzt, so existiert ein  $(1 - \alpha)(1 - 1/t)$ -Approximationsalgorithmus, der in  $O(t^2 n^{O(t^2)}) + A_\alpha(O(n^{O(t^2)}))$  Zeit läuft.

*Beweis.* Folgt direkt aus Satz 3.2 und dem Reduktionssatz mit  $g(t) = (2t - 1)(4t - 6) - 1 \in O(t^2)$ . □

Wälder sind ein erstes triviales Beispiel von (Beschriftungs-)Graphen, für die Stockwerkfunktionen existieren. In den folgenden Abschnitten werden weitere konkrete Beispiele behandelt.

### 3.1.2 Außenplanare Graphen

Außenplanare Graphen sind planare Graphen, die so in die Ebene eingebettet werden können, dass alle Knoten auf einer einzigen Facette liegen. In diesem Abschnitt wird gezeigt, dass Graphen dieser Klasse Stockwerkfunktionen besitzen. Anschließend werden  $k$ -außenplanare Graphen definiert, die – informell gesprochen – durch Schachtelung von  $k$  außenplanaren Graphen entstehen. Der naheliegende Gedanke, dass auch  $k$ -außenplanare Graphen zumindest für begrenztes  $k > 1$  Stockwerkfunktionen besitzen, erweist sich leider als nicht richtig, wie ein Gegenbeispiel für  $k = 2$  zeigen wird.

**Definition 3.7.** Sei  $G$  ein planarer Graph.  $G$  heißt *außenplanar*, falls eine planare Einbettung existiert, sodass alle Knoten auf einer einzigen Facette liegen. Diese wird *äußere Facette* genannt.

**Lemma 3.8.** Sei  $G$  ein planarer Graph und  $F$  ein Teilgraph von  $G$  der aus den Kanten einer Facette besteht. Ferner erfülle  $F$  Eigenschaften (G1)-(G3). Dann existiert eine Stockwerkfunktion für  $F$ , wobei deren Werte auf einer Kante der Facette vorgegeben werden können.<sup>1</sup>

*Beweis.* Sei  $(v, w)$  eine Kante von  $F$ . Ohne Einschränkung gilt für jede Stockwerkfunktion  $f$  von  $F$ , dass  $f(v) = 0$ . Zu zeigen ist, dass Stockwerkfunktionen existieren, die an Knoten  $w$  den Wert 0 bzw. 1 annehmen. Der Beweis erfolgt per Induktion über die Anzahl  $m$  der Kanten der Facette. Aufgrund von Eigenschaft (G3) besitzt  $F$  mindestens vier Kanten.

*Induktionsanfang,  $m = 4$ .* Alle möglichen Fälle sind in Abbildung 3.2 dargestellt.

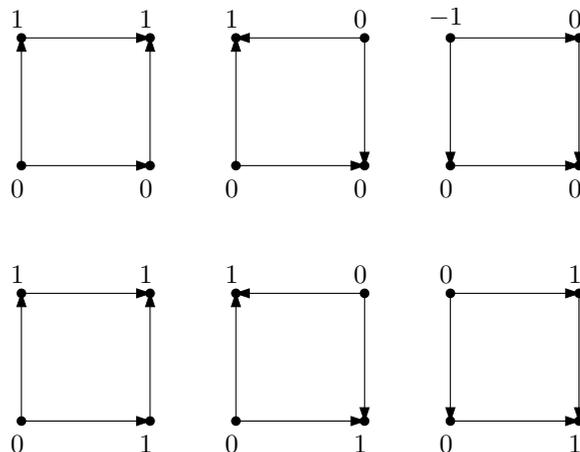


Abbildung 3.2: Eine Facette aus vier Kanten mit Eigenschaften (G1)-(G3) besitzt immer eine Stockwerkfunktion. Dabei kann der Wert an zwei benachbarten Knoten beliebig vorgegeben werden.

*Induktionsschritt,  $m \rightarrow m + 1$ .* Sei also  $F$  eine Facette aus  $m + 1$  Kanten. Abbildung 3.3 zeigt, wie aus  $F$  eine Facette aus  $m$  Kanten gewonnen werden kann, sodass der neu entstandene Graph  $F'$  nach wie vor planar und azyklisch ist. Unterscheide nun drei Fälle:

1. Zur Kante  $(v, u)$  gibt es einen alternativen Pfad in  $F'$ . Dann ist jede Kante von  $F'$  also Teil eines Pfades von  $v$  nach  $u$ . Weist man  $v$  einen beliebigen Wert  $z$  zu und allen anderen Knoten  $z + 1$ , so ist dadurch offensichtlich eine Stockwerkfunktion für  $F'$  gegeben. Da  $F$  Eigenschaft (G3) erfüllt, muss die Kante zwischen  $w$  und  $u$  von  $w$  nach  $u$  gerichtet sein. (Andernfalls gäbe es in  $F$  einen alternativen Pfad von  $v$  nach  $w$ !) Betrachte nun Abbildung 3.4 (1).

<sup>1</sup>Natürlich nur insofern, solange man nicht direkt die Eigenschaften einer Stockwerkfunktion verletzt.

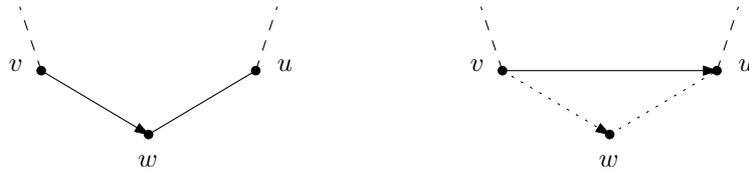


Abbildung 3.3: Verkürzung einer Facette mit Eigenschaften (G1)-(G3) um eine Kante unter Beibehaltung von (G1) und (G2). (Denn wäre (G2) verletzt, wäre vorher (G3) nicht erfüllt gewesen.)

2. Zu einer Kante  $\neq (v, u)$  gibt es einen alternativen Pfad in  $F'$ . Durch umdrehen der Kante zwischen  $v$  und  $u$  erhält man eine neue Facette, die zusätzlich zu (G1) und (G2) nun auch (G3) erfüllt. Beachte weiter, dass die Kante zwischen  $w$  und  $u$  von  $u$  nach  $w$  gerichtet sein muss. Andernfalls verletzt  $F$  die Eigenschaft (G3). Der Induktionsschritt ist in Abbildung 3.4 (2) zu sehen.

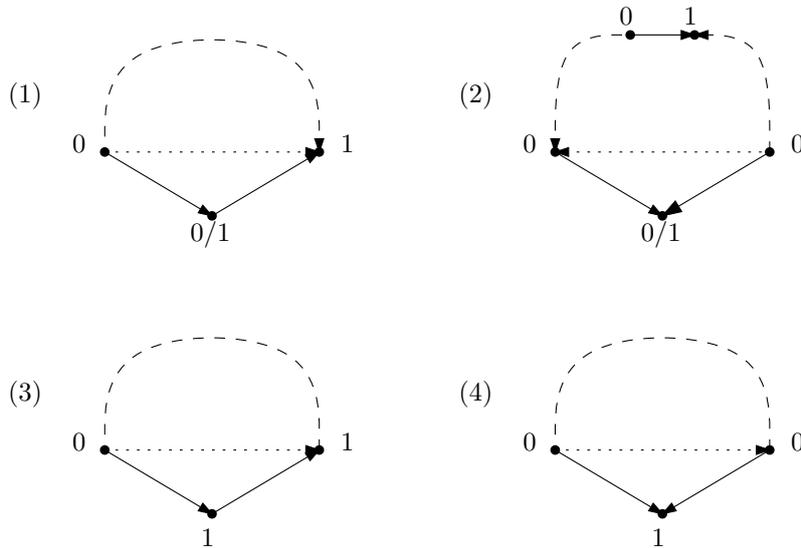


Abbildung 3.4: Erweiterung einer Facette mit Stockwerkfunktion um eine Kante unter Vorgabe der Werte an zwei benachbarten Knoten.

3.  $F'$  erfüllt Eigenschaft (G3). Soll  $w$  den Wert 1 bekommen, kann die Induktionsvoraussetzung direkt angewendet werden, siehe Abbildung 3.4 (3) und (4).

Andernfalls soll  $w$  den Wert 0 erhalten. Man muss nun zwischen den verschiedenen Möglichkeiten der Ausgangsgrade von  $v$  und  $u$  unterscheiden. Alle Varianten sind in Abbildung 3.5 dargestellt. In Diagrammen (1),(2) und (3) besitzen sowohl  $v$  als auch  $u$  – abgesehen von der Kante mit  $w$  – eine Kante, die von ihnen ausgeht. Im Fall (4) hat keiner von beiden Knoten eine weitere Kante, deren Ausgangsknoten sie sind. In den letzten beiden Fällen (5) und (6) hat jeweils einer von den beiden eine solche Kante. Man beachte, dass durch gestrichelte Pfeile eine maximal lange,

evtl. leere Folge von Kanten angedeutet werden soll. Durch den zweiten gepunkteten Pfeil wird die Facette ein weiteres Mal verkürzt, um letztlich die Induktionshypothese anwenden zu können.

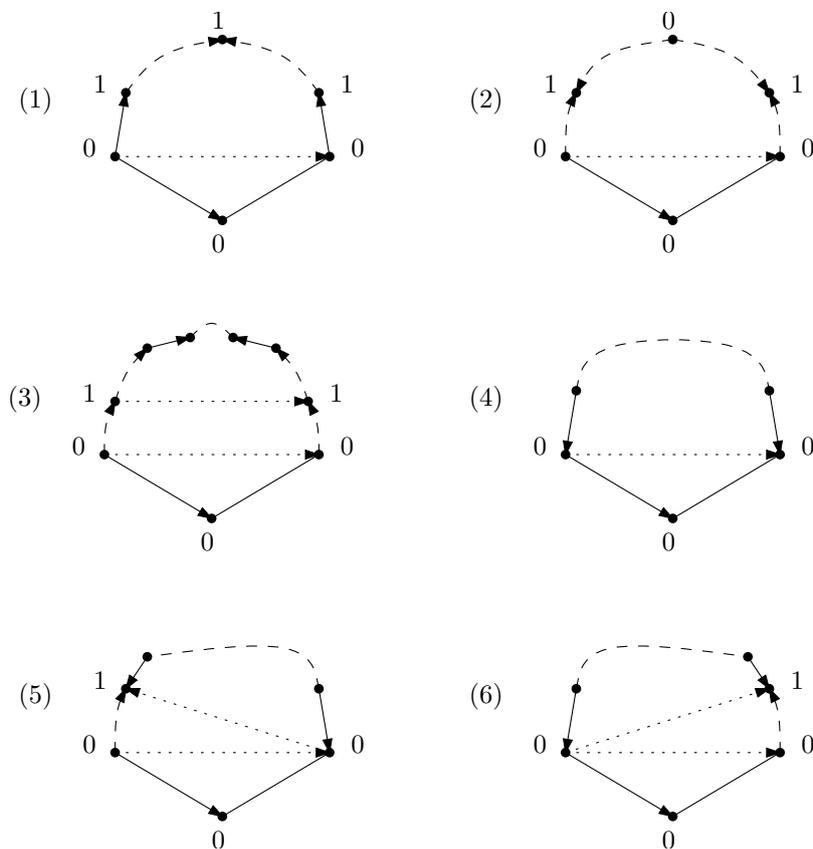


Abbildung 3.5: Erweiterung einer Facette mit Stockwerkfunktion um eine Kante unter Vorgabe der Werte an zwei benachbarten Knoten.

□

**Proposition 3.9.** Sei  $G$  ein außenplanarer Graph mit Eigenschaften (G1)-(G3). Dann existiert eine Stockwerkfunktion für  $G$ .

*Beweis.* Ohne Einschränkung sei  $G$  zusammenhängend. Der Beweis erfolgt per Induktion über  $|E| = m$ , wobei  $G = (V, E)$ .

*Induktionsanfang,  $m = 0$ .* Trivial.

*Induktionsschritt,  $m \rightarrow m + 1$ .* Wähle eine außenplanare Einbettung des Graphen und betrachte  $G$  zunächst als ungerichteten Graphen.

Angenommen  $G$  ist nicht zweifach kantenzusammenhängend. Dann existiert eine Kante  $(v, w)$ , sodass  $G[E \setminus \{(v, w)\}]$  in zwei Teilgraphen  $G_1, G_2$  zerfällt, die durch keine Kante verbunden sind. Nach Induktionsvoraussetzung existiert für  $G_1$  eine Stockwerkfunktion  $f_1$  und für  $G_2$  eine Stockwerkfunktion  $f_2$ . Ohne

Einschränkung ist  $v$  in  $G_1$  und  $w$  in  $G_2$  und es gilt  $f_2(w) = f_1(v) + 1$ . Dann lassen sich die beiden Funktionen aber zu einer Stockwerkfunktion  $f$  auf ganz  $G$  zusammensetzen vermöge

$$f(u) := \begin{cases} f_1(u), & u \text{ Knoten aus } G_1 \\ f_2(u), & u \text{ Knoten aus } G_2 \end{cases} .$$

Sei also  $G$  zweifach kantenzusammenhängend: Zu beliebigem Knotenpaar  $v, w$  existieren zwei kantendisjunkte Pfade, die  $v$  und  $w$  in  $G$  (immernoch ungerichtet betrachtet) verbinden. Im Folgenden sei mit Facette immer eine Facette gemeint, die nicht die äußere ist. Mit Lemma 3.7 und mit Hilfe von Algorithmus 2 läßt sich die Existenz einer Stockwerkfunktion für  $G$  nachweisen.

**Eingabe** : außenplanarer, zweifach kantenzusammenhängender Graph  $G$   
mit Eigenschaften (G1)-(G3)

**Ausgabe** : Stockwerkfunktion  $f$  für  $G$

- 1  $G' \leftarrow \text{Graph}(\emptyset)$ ;
- 2  $f \leftarrow \text{Stockwerkfunktion}(G')$ ;
- 3  $\mathcal{F} \leftarrow \{F \mid F \text{ Facette von } G\}$ ;
- 4 Wähle eine Facette  $F$  aus  $\mathcal{F}$ ;
- 5 Entferne  $F$  aus  $\mathcal{F}$  und füge sie  $G'$  hinzu;
- 6 Erweitere  $f$  um  $F$  auf  $G'$ ;
- 7 **wenn** eine Facette  $F$  in  $\mathcal{F}$  existiert, die mit  $G'$  genau eine Kante  
gemeinsam hat **dann**
- 8 | Wähle dieses  $F$  und gehe zu Zeile 5;
- 9 **Ende**
- 10 **wenn** eine Facette  $F$  in  $\mathcal{F}$  existiert, die mit  $G'$  genau einen Knoten  
gemeinsam hat **dann**
- 11 | Wähle dieses  $F$  und gehe zu Zeile 5;
- 12 **Ende**
- 13 Gib  $f$  zurück;

**Algorithmus 2** : Konstruktion einer Stockwerkfunktion Facette für Facette.

Offensichtlich terminiert der Algorithmus, da  $G$  endlich und somit  $\mathcal{F}$  endlich ist und bei jeder Wiederholung von Zeile 5 die Menge  $\mathcal{F}$  kleiner wird. Bleibt die Korrektheit zu zeigen.

Beim ersten Aufruf von Zeile 6 ist  $f$  die Stockwerkfunktion auf der leeren Menge und Lemma 3.7 garantiert, dass eine Stockwerkfunktion auf einer einzelnen Facette existiert. Sei diese  $f$ . Bei wiederholten Aufrufen von Zeile 6 soll  $f$  um eine Facette erweitert werden, die entweder genau einen Knoten oder genau eine Kante mit  $G'$  (der Domäne von  $f$ ) teilt. Das ist wiederum nach Lemma 3.7 möglich, da dieses sogar besagt, dass einer Stockwerkfunktion auf einer Facette die Werte an zwei Knoten einer Kante beliebig vorgegeben werden dürfen.

Beindet man sich nun direkt vor Ausführung von Zeile 10, so ist folgendes zu zeigen: Falls  $G$  keine Facette besitzt, die mit  $G'$  genau einen Knoten teilt, dann gilt  $G = G'$ . Nehme also an, es existiert noch eine Kante  $e$ , die nicht Teil von  $G'$  ist. Da  $G$  zusammenhängend ist, gibt es eine Verbindung zwischen  $e$  und  $G'$ . Sei

$v$  ein Knoten auf dieser Verbindung. Dann gibt es zwei kantendisjunkte Pfade von  $v$  nach  $G'$ , da  $G$  2-fach kantenzusammenhängend ist und somit existiert eine Facette  $F$ , die mit  $G'$  mindestens einen Knoten teilt. Zu zeigen ist:  $F$  hat mit  $G'$  genau einen Knoten gemeinsam.

Angenommen  $F$  und  $G'$  haben zwei Knoten  $v, w$  gemeinsam.

*Hilfsbehauptung 1.* Dann existiert eine Facette  $F^*$ , die mit  $G'$  genau eine Kante gemeinsam hat.

Offensichtlich ist dies ein direkter Widerspruch zu der Tatsache, dass man sich in Zeile 10 des Algorithmus befindet. Bleibt die Hilfsbehauptung 1 zu beweisen.

*Beweis der Hilfsbehauptung 1.* Es existieren Pfade von  $v$  nach  $w$ , die in  $G'$  bzw.  $F$  verlaufen. Sind diese beiden Pfade kantendisjunkt, so definieren sie eine weitere Facette  $F'$ , die mit  $G'$  eine Kante gemeinsam hat. Andernfalls haben  $F$  und  $G'$  eine Kante gemeinsam.

*Hilfsbehauptung 2.* Zwei Facetten, die eine Kante  $(v, w)$  gemeinsam haben, haben keine gemeinsamen Knoten außer  $v$  und  $w$ .

Damit haben  $F'$  und  $G'$  bzw.  $F$  und  $G'$  genau eine Kante gemeinsam. Setze  $F^* := F'$  bzw.  $F^* := F$ .

*Beweis der Hilfsbehauptung 2.* Angenommen  $F_1$  und  $F_2$  seien Facetten, die sich eine Kante  $(v, w)$  teilen und  $u$  ein weiterer gemeinsamer Knoten. (Also  $v \neq u \neq w$ .) Dann gibt es Pfade  $s_1$  und  $s_2$ , die  $w$  und  $u$  verbinden und Teil von  $F_1$  bzw.  $F_2$  sind.

*Fall 1,  $s_1 = s_2$ .* Dann liegt  $w$  ausschließlich an den Facetten  $F_1$  und  $F_2$  und insbesondere nicht an der äußeren Facette im Widerspruch dazu, dass  $G$  außenplanar ist.

*Fall 2,  $s_1 \neq s_2$ .* Dann umschließen die Pfade  $s_1$  und  $s_2$  einen Bereich, in dem die äußere Facette liegen muß. (Sonst liegt wiederum  $w$  nicht an der äußeren Facette.) Dann kann aber  $v$  nicht an der äußeren Facette liegen. Widerspruch!  $\square$

**Definition 3.13.** Sei  $G$  ein planarer Graph.  $G$  heißt 1-außenplanar, falls  $G$  außenplanar ist. Für  $k \geq 2$  heißt  $G$   $k$ -außenplanar, falls  $G$  bei gegebener planarer Einbettung durch Entfernen aller Knoten auf der äußeren Facette<sup>2</sup>  $(k - 1)$ -außenplanar wird.

Die Definition wird in Abbildung 3.6 durch ein Beispiel veranschaulicht.

*Bemerkung.* Proposition 3.8 besagt also, dass gerichtete, azyklische 1-außenplanare Graphen Stockwerkfunktionen besitzen. Dies ist bestmöglich: Bereits für gerichtete, azyklische 2-außenplanare Graphen existiert im Allgemeinen keine Stockwerkfunktion mehr. In Abbildung 3.7 ist ein Beispiel für diesen Fall anhand eines 2-außenplanaren Beschriftungsgraphen zu sehen.

Ohne Einschränkung weist jede Stockwerkfunktion dem am weitesten links liegenden Knoten den Wert 0 zu. Die Zahlen in der Abbildung deuten nun an, welchen Wert eine Stockwerkfunktion an diesen Labels mindestens annehmen

<sup>2</sup>Die äußere Facette bezeichne in diesem Fall die Facette mit unbeschränktem Flächeninhalt.

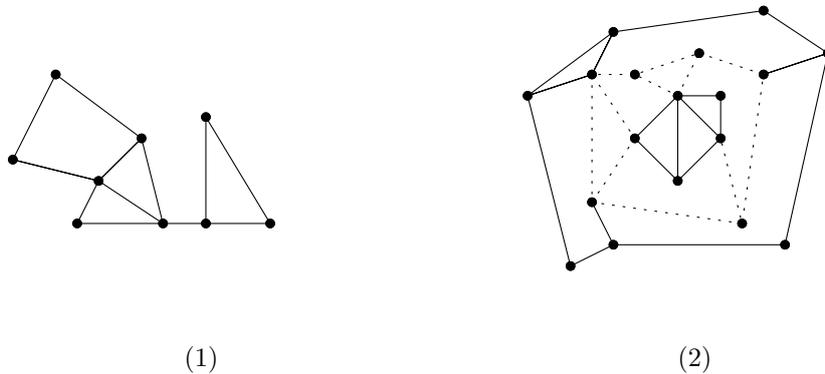


Abbildung 3.6: (1) Ein außenplanarer Graph, (2) ein 3-außenplanarer Graph. Durch die gepunkteten und durchgezogenen Kanten in (2) sollen die ineinandergeschichteten außenplanaren Graphen angedeutet werden.

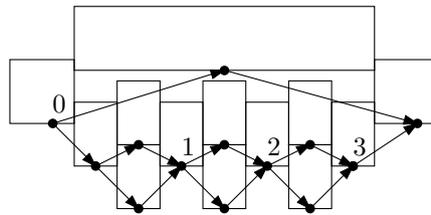


Abbildung 3.7: Eine Beschriftung, deren Graph 2-außenplanar ist und keine Stockwerkfunktion besitzt.

müsste. Wie man sieht, müsste dann aber ein „Sprung“ von Stockwerk 0 auf 3 mit nur einem Zwischenschritt erlaubt sein!

Das Problem, das hier auftritt, kann umgangen werden, indem man den Quotienten zwischen maximaler und minimaler Labellänge beschränkt. Tatsächlich wird in Abschnitt 3.2.1 gezeigt, dass in diesem Fall ein PTAS für 2d-1SH existiert.

### 3.1.3 Einteilungen

Da alle Label die gleiche Höhe besitzen, läßt sich durch jede gegebene Beschriftung derart Geraden legen, dass jedes Label von genau einer Gerade geschnitten wird und jede dieser Geraden mindestens ein Label schneidet. Ähnlich der Idee der Stockwerkfunktionen werden dadurch Beschriftungen in „Ebenen“ eingeteilt, innerhalb deren keine Verzweigungen erlaubt sind. Ebenso dürfen nur Kanten zwischen „benachbarten“ Ebenen existieren, die jedoch im Gegensatz zu Kanten zwischen Stockwerken in beliebige Richtung zeigen können.

Dieses Konzept findet in der Literatur häufig Verwendung (u.a. [AvKS98, PSS<sup>+</sup>03]) und wird im Englischen mit *stabbing lines* bezeichnet. Im vorliegenden Text wird der Begriff der *Einteilung* für eine Menge von *stabbing lines* verwendet.

Offensichtlich existiert eine Stockwerkfunktion für einen Beschriftungsgraphen, wenn eine nur einelementige Einteilung für die zugehörige Beschriftung existiert. Auch im Fall einer zweielementigen Einteilung ist dies noch der Fall, da die entsprechenden Graphen außenplanar sind. Aber schon ab drei Schnittgeraden existiert im Allgemeinen keine Stockwerkfunktion mehr. Im Folgenden sollen diese Überlegungen präzisiert werden.

**Definition 3.15.** Sei  $B$  eine Beschriftung. Eine *Einteilung*  $\mathcal{K}_B$  von  $B$  ist eine endliche Teilmenge von  $\mathbb{R}$ , sodass

$$(E1) \quad \forall L_i(r) \in B : \exists y \in \mathcal{K}_B : y \in (y_i, y_i + 1) \quad \text{und}$$

$$(E2) \quad \forall y, y' \in \mathcal{K}_B, y \neq y' : |y - y'| > 1.$$

Ist  $\mathcal{K}_B$  minimal (bzgl. Inklusion) mit (E1) und (E2), dann heißt  $\mathcal{K}_B$  *minimale Einteilung* von  $B$ .

**Proposition 3.16.** Für jede Beschriftung existiert eine minimale Einteilung aus maximal  $n$  Elementen.

*Beweis.* Betrachte die Menge aller Einteilungen. Ist diese nicht leer, existiert in dieser auch eine Einteilung mit minimal vielen Elementen. Bleibt zu zeigen, dass die Menge aller Einteilungen nicht leer ist. Dies garantiert die Korrektheit von Algorithmus 3.

*Korrektheit und Laufzeitanalyse.* Es ist leicht zu sehen, dass die berechnete Menge  $\mathcal{K}$  die Eigenschaften (E1) und (E2) erfüllt: In Zeile 20 werden nur Elemente aus der Liste entfernt, für die Eigenschaft (E1) erfüllt ist und Zeilen 15 und 18 sorgen dafür, dass je zwei Elemente aus  $\mathcal{K}$  einen Abstand größer 1 haben. Letztlich wird in jedem (äußeren) Schleifendurchlauf mindestens ein Element aus  $L$  entfernt, d.h. der Algorithmus terminiert nach höchstens  $n$  Durchläufen. Insbesondere ist somit  $|\mathcal{K}| \leq n$ . Die Laufzeit liegt in  $O(n^2)$ , da pro äußerem Schleifendurchlauf die innere Schleife höchstens  $n$  mal angewendet wird.  $\square$

*Bemerkung.* Sei  $B$  eine Beschriftung. Eine Einteilung  $\mathcal{K}_B$  definiert auf natürliche Weise Teilbeschriftungen  $B_j$ ,  $j = 1, \dots, K$ , wobei  $K = |\mathcal{K}_B|$ . Die Zuteilung für ein Label  $L_i(r)$  aus  $B$  erfolgt vermöge:

$$L_i(r) \in B_j \iff y^{(j)} \in (y_i - 1, y_i),$$

Die Eigenschaften (E1) und (E2) garantieren, dass jedes Label genau einer Teilbeschriftung zugeteilt wird. Offensichtlich besitzen die Beschriftungen  $B_j$  einelementige minimale Einteilungen.

*Bemerkung.* Existieren einelementige Einteilungen für eine Beschriftung  $B$ , dann ist der Ein- und Ausgangsgrad aller Knoten des Beschriftungsgraphen  $\leq 1$ . Andernfalls existieren ohne Einschränkung Knoten  $L_i(r), L_{j_1}(s_1), L_{j_2}(s_2)$  in  $G_B$  mit  $(L_i(r), L_{j_1}(s_1)), (L_i(r), L_{j_2}(s_2)) \in E$ . (Den Fall, dass der Eingangsgrad eines Knoten größer 1 ist, kann man analog behandeln.) Gilt nun

$$(y_{j_1}, y_{j_1} + 1) \cap (y_{j_2}, y_{j_2} + 1) \neq \emptyset$$

dann auch

$$(y_i, y_i + 1) \cap (y_{j_1}, y_{j_1} + 1) \cap (y_{j_2}, y_{j_2} + 1) = \emptyset,$$

```

Eingabe : Eine Beschriftung  $B$  und die Menge der Punkte  $P$ 
Ausgabe : Eine Einteilung  $\mathcal{K}$ 
1  $L \leftarrow$  Liste( $B$ , aufsteigend sortiert nach  $y_i$ -Werten);
2  $\mathcal{K} \leftarrow \emptyset$ ;
3  $y_{\text{alt}} \leftarrow -\infty$ ;
4 solange  $L \neq \emptyset$  tue
5    $y \leftarrow y_i + 1$ , wobei  $L_i(r) = L[1]$ ;
6    $y^* \leftarrow y_i$ , wobei  $L_i(r) = L[1]$ ;
7    $j \leftarrow 1$ ;
8   solange  $y^* < y$  und  $j < |L|$  tue
9      $j \leftarrow j + 1$ ;
10     $y^* \leftarrow y_i$ , wobei  $L_i(r) = L[j]$ ;
11  Ende
12  wenn  $y^* \geq y$  dann
13     $j \leftarrow j - 1$ ;
14  Ende
15   $y^* \leftarrow \max(y_{\text{alt}}, y_i - 1)$ , wobei  $L_i(r) = L[j]$ ;
16   $k \leftarrow (y + y^*)/2$ ;
17   $\mathcal{K} \leftarrow \mathcal{K} \cup \{k\}$ ;
18   $y_{\text{alt}} \leftarrow k + 1$ ;
19  für  $l = 1, \dots, j - 1$  tue
20    Entferne( $L, l$ );
21  Ende
22 Ende

```

**Algorithmus 3** : Berechnung einer Einteilung aus maximal  $n$  Elementen für eine gegebene Beschriftung.

da die beiden Kanten den gleichen Ausgangspunkt besitzen. Andererseits sieht man mit Hilfe der Definition von Beschriftungsgraphen, dass dies ein Widerspruch ist: Denn in diesem Fall müsste entweder  $L_{j_1}s_1$  vor  $L_{j_2}s_2$  liegen oder andersherum und das darf nicht sein!

Also muss

$$(y_{j_1}, y_{j_1} + 1) \cap (y_{j_2}, y_{j_2} + 1) = \emptyset$$

erfüllt sein, im Widerspruch zu der Tatsache, dass  $B$  einelementige Einteilung besitzt.

**Proposition 3.17.** Sei  $B$  eine Beschriftung mit Einteilung  $\mathcal{K}_B$  und  $K := |\mathcal{K}_B|$ . Dann ist  $G_B = (V, E)$  höchstens  $\lceil K/2 \rceil$ -außenplanar.

*Beweis.* Es reicht die Behauptung für gerade  $K$  zu zeigen: Ist  $K$  ungerade, füge  $\mathcal{K}_B$  ein beliebiges Element hinzu, sodass Eigenschaften (E1) und (E2) nicht verletzt werden und beachte, dass  $\lceil K/2 \rceil = \lceil (K+1)/2 \rceil$ .

Gehe nun per Induktion über  $K$  vor:

*Induktionsanfang,*  $K = 0$ . Dieser Fall entspricht der leeren Beschriftung und somit dem leeren Graph. Dieser ist offensichtlich außenplanar.

*Induktionsschritt,*  $K \rightarrow K + 2$ . Sei  $\mathcal{K}_B = \{y^{(1)}, \dots, y^{(k+2)}\}$  und  $y^{(i)} < y^{(j)}$  falls  $i < j$ . Entferne alle Knoten der Teilbeschriftungen  $B_1$  und  $B_{K+2}$ . Offensichtlich ist  $\{y^{(2)}, \dots, y^{(k+1)}\}$  eine Einteilung für  $B' := B_2 \cup \dots \cup B_{k+1}$ . Nach Induktionsvoraussetzung ist  $B'$  höchstens  $\lceil K/2 \rceil$ -außenplanar. Nach der vorangegangenen Bemerkung beinhalten die Beschriftungen  $B_1, B_{k+2}$  keine Verzweigungen. Insbesondere liegen damit alle Knoten aus  $B_1$  und  $B_{k+2}$  auf einem einzigen Fenster. Somit ist  $G_B$  höchstens  $\lceil (K+2)/2 \rceil$ -außenplanar.  $\square$

**Korollar 3.18.** Sei  $B$  eine Beschriftung mit zweielementiger Einteilung. Dann existiert eine Stockwerkfunktion für den Beschriftungsgraphen  $G_B$ .

*Beweis.* Folgt direkt aus Propositionen 3.8 und 3.16.  $\square$

**Korollar 3.21.** Es existiert ein  $(2/3)(1-\alpha)(1-1/t)$ -Approximationsalgorithmus, der in  $O(t^2 n^{O(t^2)}) + A_\alpha(O(n^{O(t^2)}))$  Zeit läuft.

*Beweis.* Berechne mittels Algorithmus 3 eine Einteilung, die für jede potentiell mögliche Beschriftung eine Einteilung ist. Das erreicht man, indem man als Eingabe neben der gegebenen Punktmenge  $P$  die Menge  $B^*$  aller Labels  $L_i$  verwendet.

Entsprechend einer Beschriftung  $B$ , die durch eine Einteilung in Teilbeschriftungen  $B_j$  aufgeteilt wird, kann  $B^*$  in disjunkte Mengen  $B_j^*$  geteilt werden, sodass jedes Label in einer dieser Mengen vorkommt. Man erhält nun drei neue Probleminstanzen  $P_i$ ,  $i = 1, 2, 3$  indem man aus  $P$  alle Punkte  $p$  entfernt, deren Label  $L_p$  in  $\bigcup_{j \in i+3\mathbb{N}} B_j$  vorkommen.

Da sich aufgrund von Eigenschaft (E2) nur Labels direkt aufeinanderfolgender  $B_j^*$  berühren können, stellt der Löschvorgang sicher, dass sich jede Beschriftung der Probleminstanzen  $P_i$  aus Teilbeschriftungen zusammensetzt, die voneinander unabhängig sind und höchstens zweielementige Einteilungen besitzen.

Somit garantiert Korollar 3.17, dass alle Beschriftungsgraphen der Probleminstanzen  $P_i$  Stockwerkfunktionen besitzen.

Nach Korollar 3.4 lässt sich also in insgesamt  $O(t^2 n^{O(t^2)}) + A_\alpha(O(n^{O(t^2)}))$  je eine  $(1 - \alpha)(1 - 1/t)$ -Approximation der Optimallösung der Probleminstanz  $P_i$  finden. Sei  $B'_{\max}$  die gewichtsmaximale Beschriftung der drei Approximationen. Bezeichnet dann  $B_{\max}$  die gewichtsmaximale Optimallösung der  $P_i$ , dann gilt offensichtlich folgende Ungleichung:

$$w(B'_{\max}) \geq (1 - \alpha)(1 - 1/t)w(B_{\max}) \geq (2/3)(1 - \alpha)(1 - 1/t)w(B_{\text{opt}}),$$

wobei  $B_{\text{opt}}$  die Optimallösung der Originalprobleminstanz ist. Damit ist die Behauptung gezeigt.  $\square$

*Bemerkung.* Bereits bei Beschriftungen mit dreielementigen minimalen Einteilungen stößt das Konzept der Stockwerkfunktionen an seine Grenzen: Das gleiche Beispiel, das bereits für 2-außenplanare Graphen die Existenz von Stockwerkfunktionen im Allgemeinform wiederlegt hat, kann auch hier verwendet werden. Abbildung 3.8 verdeutlicht die Situation nochmals.

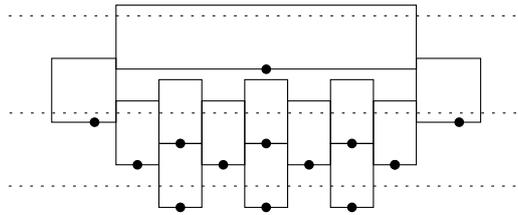


Abbildung 3.8: Eine Beschriftung mit dreielementiger minimaler Einteilung, für deren Graphen keine Stockwerkfunktion existiert.

## 3.2 Allgemeinere Stockwerkfunktionen

Der letzte Abschnitt hat den Begriff der Stockwerkfunktionen eingeführt und erste Resultate vorgestellt. Die mächtigste Aussage gab Korollar 3.4: Beschränkt man sich auf Probleminstanzen, in denen alle Beschriftungsgraphen Stockwerkfunktionen besitzen, existiert ein PTAS. Allerdings ist dies nicht sehr konkret. Im weiteren Verlauf wurden explizite Beispiele für Graphen, die Stockwerkfunktionen besitzen, vorgestellt: Außenplanare Beschriftungsgraphen und Beschriftungsgraphen, die von Beschriftungen herrühren, die zweielementige Einteilungen besitzen.

Der Versuch diese Resultate auf  $k$ -außenplanare Beschriftungsgraphen oder Beschriftungen mit mehrelementigen Einteilungen zu verallgemeinern, scheiterte in beiden Fällen. Das wirft die Frage auf, ob das Konzept der Stockwerkfunktionen an seine Grenzen gestoßen ist. Im Folgenden sollen daher zwei Methoden zur Erweiterung von Stockwerkfunktionen untersucht werden.

Die Negativbeispiele bezüglich der Existenz von Stockwerkfunktionen hatten eines gemeinsam: In beiden Fällen konnten die längeren Labels im Verhältnis

zu den kürzeren „zu lang“ werden und so ein zu großen Sprung zwischen Stockwerken erzwingen. Der naheliegende Gedanke das Verhältnis zu beschränken und entsprechend große Sprünge in der Stockwerkfunktion zu erlauben, führt direkt auf ein PTAS. Andererseits kann man auch auf den bisherigen Resultaten aufbauen und statt Wäldern (Noch dazu von spezieller Natur) beliebige Graphen als Stockwerke verwenden, die Stockwerkfunktionen besitzen: Durch die Existenz der Stockwerkfunktionen für die einzelnen Stockwerke ist dann nach wie vor garantiert, dass die Pfadlänge innerhalb von Stockwerken beschränkt werden kann (ohne zu viel Gewicht zu entfernen). Beispielsweise lassen sich die beschriebenen negativen Beispiele auch bewältigen, indem (azyklische, gerichtete) außenplanare Graphen als Stockwerke zugelassen werden. Allerdings wird anhand einer Beispielbeschriftung beschrieben, dass außenplanare Stockwerke nicht ausreichen, um die Existenz von Stockwerkfunktionen zu garantieren.

### 3.2.1 Probleminstanzen mit Labellängenbeschränkungen

**Definition 3.25.** Sei  $G = (V, E)$  ein Graph und  $f : V \rightarrow \mathbb{Z}$  eine Funktion, die folgende Eigenschaften erfüllt:

$$(S1') \quad \forall (v, w) \in E : f(w) - f(v) \in \{0, 1, \dots, \lambda\},$$

$$(S2) \quad \forall v \in V : |\{w \in V \mid (v, w) \in E \text{ und } f(v) = f(w)\}| \leq 1$$

In diesem Fall heißt  $f$  eine  $\lambda$ -Stockwerkfunktion von  $G$  und die Mengen  $V_i := \{v \in V \mid f(v) = i\}$  heißen *Stockwerke* von  $G$ .

**Proposition 3.26.** Sei  $G$  ein knotengewichteter Graph mit  $\lambda$ -Stockwerkfunktion  $f$ . Dann existiert eine Teilmenge  $V'$  von  $V$  mit Gewicht höchstens  $\lambda/t$ -mal dem von  $V$ , sodass alle Pfade in  $G[V \setminus V']$  maximal  $(4t - 2)^2$  lang sind.

*Beweis.* Der Beweis verläuft analog zum Beweis von Satz 3.2 mit der einzigen Änderung, dass von  $t$  aufeinanderfolgenden Stockwerken nicht nur ein Stockwerk entfernt wird, sondern  $\lambda$  aufeinanderfolgende. Man sucht dann von den  $t - \lambda + 1$  Möglichkeiten dies zu tun die gewichtsoptimale Variante aus.  $\square$

Seien  $m := \inf\{l_i \in \mathbb{R}_{>0} \mid i = 1, \dots, n\}$  und  $M := \sup\{l_i \in \mathbb{R}_{>0} \mid i = 1, \dots, n\}$ .

**Proposition 3.28.** Ist  $m \neq 0$  und gilt  $\lfloor \frac{M}{m} \rfloor = \lambda$ , so besitzt der Beschriftungsgraph  $G_B$  jeder Beschriftung  $B$  der Probleminstanz eine  $\lambda$ -Stockwerkfunktion.

*Beweis.* Sei also  $\lfloor \frac{M}{m} \rfloor = l$  und  $G_B$  der Beschriftungsgraph einer Beschriftung  $B$ . Sei  $(j^*, s^*) \in B$ , sodass  $x_{j^*} - l_{j^*} + s^* \leq x_j - l_j + s$  für alle  $(j, s) \in B$ , d.h. die linke Seite des Labels  $L_{j^*}$  ist die am weitest links liegende. Ohne Einschränkung gilt  $0 = x_{j^*} - l_{j^*} + s^*$ . Definiere nun eine Funktion  $f : V \rightarrow \mathbb{Z}$  wie folgt:

$$f((j, s)) := \left\lfloor \frac{x_j - l_j + s}{m} \right\rfloor \quad \text{für } (j, s) \in V = B$$

Offensichtlich ist  $f$  wohldefiniert. Nun betrachte  $(i, r), (j, s) \in E$ : Einerseits folgt aus Bedingungen (B1) und (B4), dass

$$x_i + r \leq x_j - l_j + s$$

da  $B$  eine Beschriftung ist. Somit

$$\begin{aligned} f((j, s)) - f((i, r)) &\geq \left\lfloor \frac{x_i + r}{m} \right\rfloor - \left\lfloor \frac{x_i - l_i + r}{m} \right\rfloor \\ &\geq \left\lfloor \frac{l_i}{m} \right\rfloor \geq \left\lfloor \frac{m}{m} \right\rfloor = 1 \end{aligned}$$

Andererseits gilt aufgrund von Bedingungen (B2) und (B3)

$$x_j - l_j + s < x_i + r + l_j$$

und somit

$$\begin{aligned} f((j, s)) - f((i, r)) &< \left\lfloor \frac{x_i + r + l_j}{m} \right\rfloor - \left\lfloor \frac{x_i - l_i + r}{m} \right\rfloor \leq \\ &\leq \left\lfloor \frac{x_i + r + l_j}{m} - \frac{x_i - l_i + r}{m} \right\rfloor + 1 = \left\lfloor \frac{l_j + l_i}{m} \right\rfloor + 1 \\ &\leq \left\lfloor \frac{2M}{m} \right\rfloor + 1. \end{aligned}$$

Daraus folgt, dass

$$f((j, s)) - f((i, r)) \leq \left\lfloor \frac{2M}{m} \right\rfloor \leq 2l + 2,$$

d.h.  $f$  erfüllt (S1') sowie (S2). □

**Korollar 3.29.** Beschränkt man das Problem 2d-1SH auf Probleminstanzen, deren Quotient zwischen maximal und minimal zugelassener Labellänge höchstens  $\lambda$  ist, so existiert ein  $(1-\alpha)(1-\lambda/t)$ -Approximationsalgorithmus, der in  $O(t^2 n^{O(t^2)}) + A_\alpha(O(n^{O(t^2)}))$  Zeit läuft.

### 3.2.2 Grenzen der Methode

**Definition 3.30.** Sei  $G = (V, E)$  ein Graph und  $f : V \rightarrow \mathbb{Z}$  eine Funktion, die folgende Eigenschaften erfüllt:

$$(S1) \quad \forall (v, w) \in E : f(w) - f(v) \in \{0, 1\},$$

$$(S2') \quad \forall i \in \mathbb{Z} : G[V_i] \text{ ist ein außenplanarer Graph mit Eigenschaften (G1)-(G3),}$$

wobei wieder  $V_i := \{v \in V \mid f(v) = i\}$  die *Stockwerke von  $G$*  sind. In diesem Fall heißt  $f$  eine *erweiterte Stockwerkfunktion von  $G$*  und die Mengen  $V_i := \{v \in V \mid f(v) = i\}$  heißen *Stockwerke von  $G$* .

Mit dieser Definition von erweiterten Stockwerkfunktionen lassen sich das negative Beispiel aus den Abschnitten 3.1.2 und 3.1.3 bewältigen. Abbildung 3.9 gibt eine Stockwerkfunktion an.

Diese Definition von erweiterten Stockwerkfunktionen verschiebt das Problem allerdings nur um eine Ebene: Indem man einen außenplanaren Graphen in einer anderen setzt und diese mit Kanten in beide Richtungen verbindet, erhält



### 3.3 Weitere Ansätze

Bisher hat Kapitel 3 ausschließlich Stockwerkfunktionen behandelt. Das Ziel, die Existenz einer beliebig leichten Knotenmenge in Beschriftungsgraphen nachzuweisen, deren Löschen aus dem Graphen Pfade in ihrer Länge beschränkt, konnte in einigen Spezialfällen erreicht werden. – Es wurde aber auch klar, dass ohne weitere Änderungen der Allgemeinfall mit Stockwerkfunktionen nicht lösbar ist. Ohne zu formell zu werden, sollen nun zwei weitere Ansätze vorgestellt werden, mit denen man versuchen kann Graphen zu stutzen. In beiden Fällen trifft man auf Probleme, die ihm Rahmen dieser Arbeit nicht mehr gelöst wurden.

#### 3.3.1 Induktion über die Kardinalität von Einteilungen

In Abschnitt 3.1.3 wurden Einteilungen eingeführt und gezeigt, dass für Beschriftungen mit ein- und zweielementige Einteilungen Stockwerkfunktionen existieren. Es bietet sich an dieses Resultat als Basisfall einer Induktion über die Mächtigkeit von Einteilungen zu verwenden. Die Induktionsbehauptung lautet dann wie folgt: *Sei  $B$  eine Beschriftung mit Beschriftungsgraph  $G_B = (V, E)$  und  $\mathcal{K}_B$  eine Einteilung. Ist  $|\mathcal{K}_B| = m$ , dann existiert eine Teilmenge  $V'$  von  $V$  mit Gewicht höchstens  $1/t$ -mal dem von  $V$ , sodass die maximale Pfadlänge in  $G[V \setminus V']$  durch eine Funktion  $g(t)$  beschränkt ist.*

Bei der Durchführung des Induktionsschrittes befindet man sich folgender Situation: Es ist eine Beschriftung  $B$  mit Einteilung  $\mathcal{K}_B$  aus  $m + 1$  Elementen gegeben. Die Einteilung  $\mathcal{K}_B$  induziert eine Einteilung der Beschriftung in Teilbeschriftungen  $B_1, \dots, B_{m+1}$ , so wie es in Abschnitt 3.1.3 formuliert wurde. Da  $B_1 \cup \dots, B_n$  eine Beschriftung mit  $m$ -elementiger Einteilung ist, kann auf den Subgraphen  $G' := G[B_1 \cup \dots \cup B_m]$  des Beschriftungsgraphen  $G$  die Induktionshypothese angewendet werden, d.h.  $G'$  lässt sich im Sinne der Induktionsbehauptung stutzen. Alle Pfade in  $G$ , die noch unbeschränkt sind, müssen folglich Knoten aus  $G'' := G[B_{m+1}]$  verwenden.

Sei  $H := H_{m+1}$  der Graph mit Knoten  $B_{m+1}$  und zwischen zwei Knoten  $u, v$  sei genau dann eine Kante  $(u, v)$ , wenn  $(u, v)$  eine Kante in  $G$  ist, oder es in  $G$  einen Pfad von  $u$  nach  $v$  gibt. Beschränkt man sich nun darauf ausschließlich Knoten aus  $B_{m+1}$  zu entfernen, um die Länge der Pfade in  $G$  zu begrenzen, so ist folgendes Problem zu lösen: Finde eine Teilmenge  $V_H$  von Knoten aus  $H$ , die höchstens  $1/t$ -mal das Gewicht des gesamten Graphen  $H$  besitzt, sodass die maximale Pfadlänge in  $H$  ohne  $V_H$  durch eine Funktion in  $t$  beschränkt ist. – Dies war aber genau die Ausgangssituation! Zwar kann  $H$  höchstens  $n - m$  Knoten enthalten (vorausgesetzt man formuliert die Induktion auf minimalen Einteilungen), aber es ist nicht ohne weiteres klar, ob  $H$  im Gegensatz zu  $G$  planar ist, und die Ein- und Ausgangsgerade der Knoten sind nicht mehr wie bei Beschriftungsgraphen durch 2 beschränkt.

#### 3.3.2 Baumzerlegungen

Ein weiterer Ansatz Beschriftungsgraphen zu stutzen, ist Baumstrukturen zu verwenden. Dass Bäume gestutzt werden können, war eine der ersten Aussagen, die ihm Rahmen von Stockwerkfunktionen bewiesen wurde (siehe Proposition

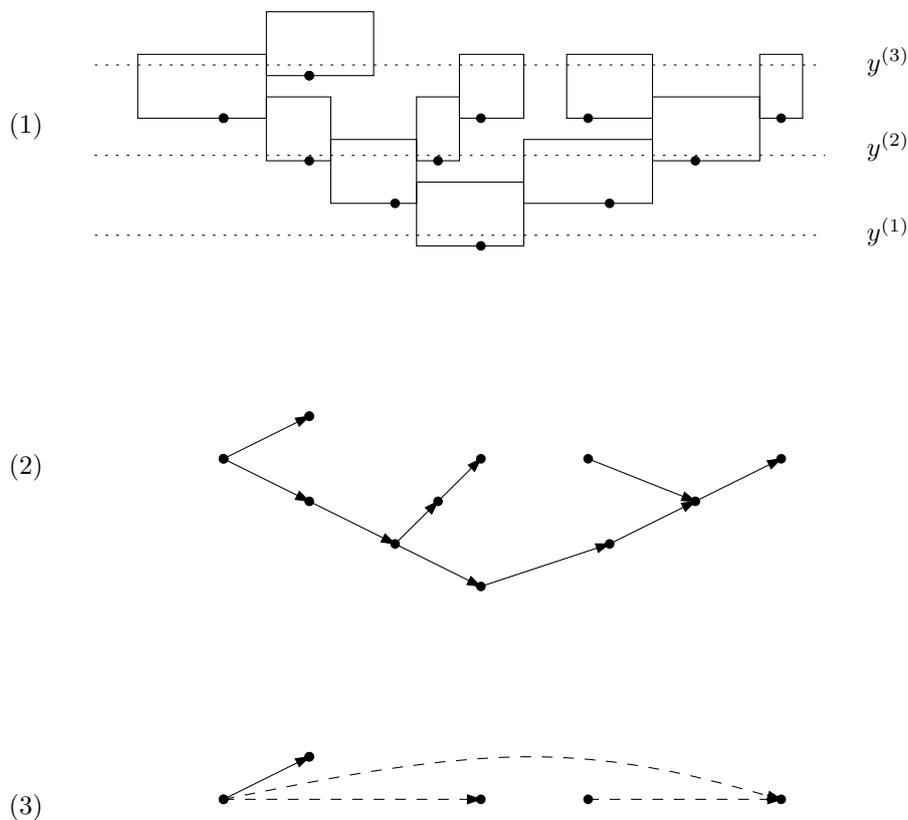


Abbildung 3.12: Beschriftung mit Einteilung (1), der zugehörige Beschriftungsgraph (2) und der Graph  $H_3$  (3).

3.1). Die Idee ist nun, für beliebige Beschriftungsgraphen einen Baum zu definieren, der gestutzt wird. Dabei ist wichtig, dass es möglich ist von den Knoten, die im Baum gelöscht werden, Rückschlüsse auf den Ursprungsgraphen zu ziehen. – Anhand von Baumzerlegungen soll dies erst vorgestellt und dann das Problem erläutert werden, welches verhindert direkt Rückschlüsse zu ziehen.

**Definition 3.32.** Eine *Baumzerlegung* eines Graphen  $G = (V, E)$  ist ein Paar  $(\mathcal{S}, T)$ , wobei  $\mathcal{S} = \{S_i \mid i \in I, S_i \subseteq V\}$  eine Familie von Teilmengen von  $V$  ist und  $T$  ein Baum auf  $I$ , sodass die folgenden Eigenschaften erfüllt sind<sup>3</sup>:

1. Die Mengen  $S_i$  überdecken ganz  $V$ :  $\bigcup_{i \in I} S_i = V$
2. Die Knoten jeder Kante befinden sich in einer gemeinsamen Menge:  $\forall (u, v) \in E \exists i \in I : \{u, v\} \subseteq S_i$
3. Für jeden Knoten  $v \in V$  induziert die Menge  $\{i \in I \mid v \in S_i\}$  einen Teilbaum auf  $T$ .

<sup>3</sup>Somit ist eine Baumzerlegung eigentlich ein Tripel  $(\mathcal{S}, T, I)$ . Die Indexmenge  $I$  wird aber im Weiteren verschwiegen.

Die *Weite* einer Baumzerlegung ist  $\max_{i \in I} |S_i| - 1$ . Die *Baumweite* eines Graphen  $G$  ist die minimale Weite einer Baumzerlegung von  $G$ .

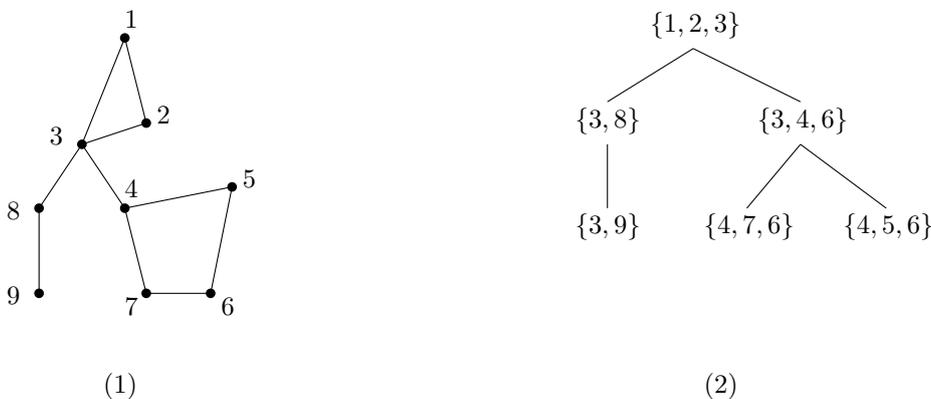


Abbildung 3.13: Ein Graph (1) und eine Baumzerlegung des Graphen (2)

*Bemerkung.* Für jeden Graphen existiert mindestens eine Baumzerlegung: Wähle  $I = \{1\}$ ,  $S_1 = V$  und  $T$  sei der triviale Graph auf einem Knoten.

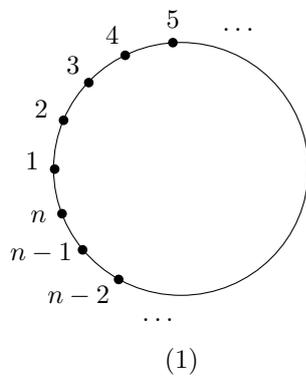
Eine Möglichkeit, nun zu verfahren, wäre die folgende<sup>4</sup>: Man beschränkt sich auf Graphen mit Baumweite  $k$ . Wähle dann für einen gegebenen Graphen  $G = (V, E)$  eine Baumzerlegung  $(\mathcal{S}, T)$  mit Weite  $k$ . Wende auf  $T$  Proposition 3.1 an und erhalte eine Menge  $I' \subseteq I$  mit höchstens  $1/t$ -mal dem Gewicht des gesamten Baumes  $T$ , deren Entfernen aus dem Baum die Pfadlänge beschränken würde. Löscht man die Knoten des Ursprungsgraphen, die in einer der Mengen  $S_i$  für  $i \in I'$  liegen, dann ist auch die Pfadlänge in  $G$  begrenzt: Jeder Pfad aus  $G$  induziert einen Weg in  $T$ .

Schon an Abbildung 3.13 lässt sich erkennen, welches Problem auftritt, wenn man Baumzerlegungen benutzen will: Die Knoten der Baumzerlegung bestehen aus Knotenmengen des Ursprungsgraphen. Die Frage ist, wie sich aus den Gewichten der Knoten des Ursprungsgraphen sinnvoll Gewichte auf den Knoten des Baumes festlegen lassen. Denn werden Knoten aus dem Baum gelöscht, müssen Rückschlüsse möglich sein, welche Knoten im Originalgraphen gelöscht werden sollen und welchen Anteil am Gesamtgewicht sie ausmachen. Abbildung 3.14 verdeutlicht die Situation anhand eines weiteren Beispiels nochmals.

Hat in diesem Fall Knoten 1 ein viel größeres Gewicht als alle anderen Knoten, stehen die Gewichte in keiner Relation mehr den Gewichten des Kreises. Beschließt man beispielsweise bei Löschung eines Knoten aus dem Baum, *alle* Knoten des ursprünglichen Graphen aus dieser Knotenmenge zu entfernen, würde man hier zu viel Gewicht entfernen.

Natürlich hätte auch eine andere Baumzerlegung gewählt werden, sodass der schwere Knoten 1 nicht in jedem Knoten des Baumes vertreten ist. Das wirft aber die Frage auf, ob man diese Wahlmöglichkeit in jedem Fall hat und wie man eine geeignete Baumzerlegung findet.

<sup>4</sup>Die grundlegende Idee hierzu stammt aus einer E-Mail von Torben Hagerup, die ich von Alexander Wolff erhielt.



$$\{1, 2, 3\} — \{1, 3, 4\} — \{1, 4, 5\} — \dots — \{1, n-1, n\}$$

(2)

Abbildung 3.14: Ein Kreis (1) und eine mögliche Baumzerlegung (2).

# Abschließende Bemerkungen

Für gegebene Beschriftungen wurde der Beschriftungsgraph definiert, der es ermöglichte graphentheoretische Überlegungen im Rahmen des Beschriftungsproblems anzustellen. Mit dieser Hilfe wurde gezeigt, wie das Problem 2d-1SH durch Lösungen einer Variante mit diskretem Beschriftungsmodell approximiert werden kann. Voraussetzung dafür war jedoch, dass ohne wesentlichen Gewichtsverlust die Weglänge in den Beschriftungsgraphen beschränkt werden kann.

Für Graphen mit Stockwerkfunktionen kann dies erreicht werden. Zu diesen gehören Wälder, außenplanare Beschriftungsgraphen und Beschriftungsgraphen, deren zugehörige Beschriftungen eine Einteilung aus zwei stabbing lines besitzen. Ferner konnte gezeigt werden, dass für *jeden* Beschriftungsgraph eine Stockwerkfunktion existiert, solange man den Quotienten zwischen maximaler und minimaler Labellänge beschränkt. Alle aufgeführten Gegenbeispiele für Graphen ohne Stockwerkfunktion machten sich zu nutze, dass dieser Quotient unbeschränkt ist. In diesem allgemeinen Fall konnte immerhin eine  $(2/3 - \epsilon)$ -Approximation des Problems 2d-1SH erreicht werden.

Eine weiterführende Arbeit könnte den Versuch unternehmen, Graphen mit Stockwerkfunktionen sinnvoll zu charakterisieren und die Frage klären, ob man im Falle eines unbeschränkten Quotienten zwischen maximaler und minimaler Labellänge tatsächlich an eine Grenze gelangt ist. Ein weiterer Ansatz wäre, das Gewicht beliebiger Beschriftungsgraphen durch das Gewicht von Beschriftungsgraphen mit Stockwerkfunktionen zu approximieren. Letztlich wäre es von Interesse die beiden Ansätze des letzten Abschnitts – Induktion über die Kardinalität von Einteilungen und Baumzerlegungen – und die dabei angesprochenen Probleme noch genauer untersuchen.

# Literaturverzeichnis

- [AvKS98] AGARWAL, PANKAJ K., MARC VAN KREVELD und SUBHASH SURI: *Label placement by maximum independent set in rectangles*. Computational Geometry: Theory and Applications, 11:209–218, 1998. 2, 3, 6, 7, 25
- [Ca196] CALMET, JACQUES: *Skriptum zur Computer-Algebra*, 1996. 9
- [DMM02] DODDI, SRINIVAS, MADHAV V. MARATHE und BERNARD M.E. MORET: *Point set labeling with specified positions*. International Journal of Computational Geometry and Applications, 12(1–2):29–66, 2002. 2
- [EJS01] ERLEBACH, THOMAS, KLAUS JANSEN und EIKE SEIDEL: *Polynomial-time approximation schemes for geometric graphs*. In: *Proc. 12th ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*, Seiten 671–679, Washington, DC, 7.–9. Januar 2001. 2, 6
- [FW91] FORMANN, MICHAEL und FRANK WAGNER: *A packing problem with applications to lettering of maps*. In: *Proc. 7th Annual ACM Symposium on Computational Geometry (SoCG'91)*, Seiten 281–288, 1991. 3
- [IL03] ITURRIAGA, CLAUDIA und ANNA LUBIW: *Elastic labels around the perimeter of a map*. Journal of Algorithms, 47(1):14–39, 2003. 3
- [PSS<sup>+</sup>03] POON, SHEUNG-HUNG, CHAN-SU SHIN, TYCHO STRIJK, TAKEAKI UNO und ALEXANDER WOLFF: *Labeling points with weights*. Algorithmica, 38(2):341–362, 2003. 2, 3, 25
- [vKSW99] KREVELD, MARC VAN, TYCHO STRIJK und ALEXANDER WOLFF: *Point labeling with sliding labels*. Computational Geometry: Theory and Applications, 13:21–47, 1999. 2, 3
- [WS96] WOLFF, ALEXANDER und TYCHO STRIJK: *The Map-Labeling Bibliography*. <http://i11www.ira.uka.de/map-labeling/bibliography/>, 1996. 3