

DIPLOMARBEIT

Umwege in Polygonen

Angefertigt am Institut für Angewandte Mathematik

Vorgelegt der

Mathematisch-Naturwissenschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

September 2002

Von

Ansgar Grüne

Aus

Wickede/Soest

Inhaltsverzeichnis

1	Einleitung	5
2	Definitionen	8
3	Rückführung auf Randpunkte	9
4	Existenz eines Umwegmaximums	13
5	Maximum auf Eckpunkt-Randpunkt-Paar	16
5.1	Nicht-paralleler Fall mit eindeutigen a, b	19
5.2	Paralleler Fall mit eindeutigen a, b	24
5.3	Sprung von $\pi(p', q')$ zwischen Ecken	25
5.4	$\pi(p', q')$ läuft über beide Kanten	26
5.5	Maximumpaar in P^C gegenseitig sichtbar	26
6	Maximale Anzahl von Umwegmaxima	28
6.1	Beschreibung der Ausgangssituation	28
6.2	Keine Berührung gegenüberliegender Wege	30
6.3	Der Stifte-Ansatz	31
6.4	Geradebiegen des dualen Vierecks	35
6.5	Schlussfolgerung	40
7	Ein exakter Algorithmus	44
7.1	Notwendiges Kriterium für ein einseitiges Maximum	44
7.2	Spezialfall des notwendigen Kriteriums	47
7.3	Maximumsuche in einem Trichter	47
7.4	Optimalität der Suche im Trichter	50
7.5	Bereitstellung der Trichter	54
7.6	Der exakte Algorithmus	57
8	Ein schneller Approximationsalgorithmus	59

9 Untere Laufzeitschranken	66
9.1 Exaktes Eckpunktmaximum in Polygonen	66
9.2 Exaktes Eckpunktmaximum auf polygonalen Ketten	70
9.3 Approximation des Eckpunktmaximums auf Ketten	72
9.4 Approximation des Eckpunktmaximums in Polygonen	74
9.5 Klassifikation der Problemstellung	77
10 Einordnung und Ausblick	81
Literatur	83
Index	85

1 Einleitung

Diese Diplomarbeit behandelt Umwege in einfachen Polygonen. Der (*relative*) *Umweg* $u_P(p, q)$ zwischen zwei Polygonpunkten p und q ist der Quotient aus der Länge der kürzesten Verbindung durch das Polygon $|\pi(p, q)|$ und der direkten Entfernung $|\overline{pq}|$ (siehe Abb. 1).

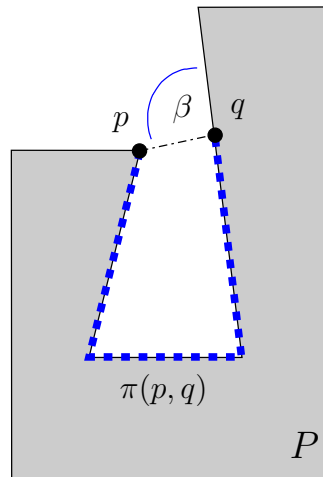


Abbildung 1: Polygon mit maximalem Umweg

Untersucht werden Punktpaare, deren Umwege im zugehörigen Polygon maximal sind. Gibt es immer solche Punktpaare? Wie viele kann es geben? Welche Eigenschaften haben sie? Und wie kann man sie effizient ermitteln?

Derartige Fragestellungen sind seit kurzem Grundlage vieler Forschungsaktivitäten (siehe [NS00], [EBKLL01], [LMS02] und [AKKS02]). Für Anwendungen ist der Fall allgemeiner geometrischer Graphen besonders interessant. Der Graph kann z.B. zu optimierende Verkehrsnetze wie Schienenverbindungen oder Straßennetze, aber auch Kommunikations- oder Versorgungsnetzwerke repräsentieren. Der maximale Umweg ist dann ein Gütemaß für das Netz, und Punktpaare maximalen Umwegs sind Kandidaten für das Einfügen neuer Verbindungen zur Verbesserung des Netzes. Allerdings ist die Komplexität dieses allgemeinen Falles sehr hoch, so dass die Betrachtung von einfacheren Spezialfällen erforderlich wird.

Einen bereits erforschten Spezialfall stellen *polygonale Ketten* (Streckenzüge) dar. So entwickeln Langerman, Morin und Soss in [LMS02] einen nicht-deterministischen Algorithmus, der in einer erwarteten Laufzeit von $O(n \log n)$ ein Punktpaar maximalen Umwegs findet. n bezeichnet hier die Anzahl der Kettenkanten. Agarwal et al. liefern in [AKKS02] durch Anwendung der „Parametric Search“-Methode von Megiddo [Meg83] für das gleiche Problem einen deterministischen Algorithmus mit einer Laufzeit von $O(n \log^4 n)$. Ein deterministischer Approximationsalgorithmus von Ebberts-Baumann et al. [EBKLL01], der Punktpaare findet, deren Umweg mit vorgegebener Genauigkeit am Maximum liegt, läuft in $O(n \log n)$.

Die im folgenden betrachtete Situation in einfachen Polygonen kann im Hinblick auf ihre algorithmische Komplexität in etwa als Zwischenstufe zwischen dem Kettenfall und dem Fall allgemeiner Graphen angesehen werden.

Sie ist komplizierter als die Situation auf Ketten, weil die kürzesten Wege prinzipiell durch das gesamte Innere des Polygons verlaufen können. Es ist nur klar, dass die kürzesten Wege polygonale Ketten sind, die ausschließlich Polygoneckpunkte als innere Knoten besitzen (sie-

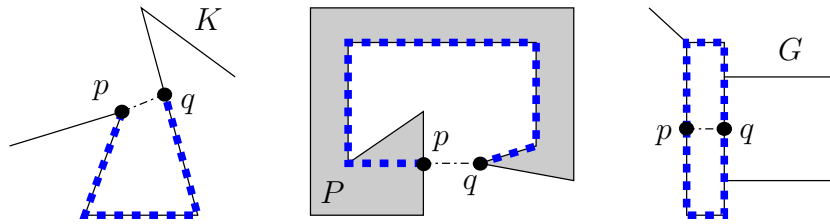


Abbildung 2: maximale Umwege einer Kette K , eines Polygons P und eines Graphen G

he [LP84]). Im Kettenfall hingegen ist der Verlauf der kürzesten Wege durch die Kette selbst eindeutig vorgegeben (vgl. Abb. 2). Und selbst im Fall eines allgemeinen geometrischen Graphen ist der Verlauf der kürzesten Wege auf die Kantenmenge des Graphen eingeschränkt. Der vergleichsweise freie Verlauf der kürzesten Wege im Polygon ist das Hauptproblem der vorliegenden Diplomarbeit, das an vielen Stellen auftaucht.

Andererseits ist der Polygonfall einfacher als der Graphenfall. Denn in einfachen Polygonen sind im Gegensatz zum Graphenfall die kürzesten Wege eindeutig. So gesehen hat die Untersuchung von einfachen Polygonen Ähnlichkeit mit der Betrachtung von Bäumen, d.h. zyklusfreien Graphen.

Unsere Untersuchungen zum Polygonfall beginnen wir nach einführenden Definitionen in Abschnitt 2 mit dem relativ problemlosen Beweis, dass in nicht-konvexen Polygonen das Umwegmaximum nur zwischen Randpunktpaaren angenommen werden kann (siehe Abschnitt 3). Diese müssen sich in P^C (Komplement von P) gegenseitig sehen können. Auch die Existenz- und die Eindeutigkeitsfrage von Punktpaaren maximalen Umwegs sind leicht zu beantworten, wie in Abschnitt 4 zu sehen ist.

Im darauf folgenden Abschnitt 5 wird zum ersten Mal die Kompliziertheit des Polygonfalles deutlich. Zwar gilt auch in Polygonen, dass das Umwegmaximum immer auch von einem Eckpunkt-Randpunkt-Paar angenommen wird. Aber der Nachweis gestaltet sich wegen des schon beschriebenen, relativ freien Verlaufs der kürzesten Wege weit mühseliger als im Kettenfall (vgl. Lemma 2 in [EBKLL01]). Um überhaupt zu einem Resultat zu gelangen, benutzen wir zur Eingrenzung der kürzesten Wege die sogenannten Sanduhren.

Auch der Beweis dafür, dass sich die direkten Verbindungen von Punktpaaren maximalen Umwegs nicht schneiden können, benötigt im Abschnitt 6 wesentlich kompliziertere Überlegungen als im Kettenfall. Eine leichte Folgerung daraus ist, dass es in nicht-konvexen Polygonen höchstens $O(n)$ viele Umwegmaxima geben kann.

Die schnellen exakten Algorithmen aus [LMS02] und [AKKS02] können leider in dieser Form überhaupt nicht auf den Polygonfall übertragen werden. Schon die notwendige Bedingung für ein einseitiges lokales Umwegmaximum wird wesentlich komplizierter (vergleiche Lemma 7.1 auf Seite 46 mit Lemma 1 aus [EBKLL01]). Durch die Verwendung von Shortest-Path-Trees erhalten wir in Abschnitt 7 aber einen exakten Algorithmus mit immerhin quadratischer Laufzeit.

Die Übertragung des Approximationsalgorithmus aus [EBKLL01] gelingt hingegen. In Abschnitt 8 benötigen wir dazu einige Sonderbetrachtungen und profitieren davon, dass sich der Hershberger-Algorithmus zur Ermittlung von kürzesten Polygonwegen aus [GH87] so einbinden lässt, dass er die Gesamtlaufzeit des Approximationsalgorithmus nicht erhöht.

Im Abschnitt 9 schließlich liefern wir untere Laufzeitschranken für die Suche nach Eckpunktpaaren maximalen Umwegs. Die benutzte Rückführung auf das Problem ELEMENT-

UNIQUENESS für ganze Zahlen liefert auch untere Schranken für Algorithmen, die den maximalen Eckpunktumweg nur approximieren. Mit dem gleichen Verfahren erhalten wir neue untere Schranken für den Kettenfall. Diese Herangehensweise scheitert jedoch bei der Aufstellung einer unteren Schranke für die Suche nach allgemeinen Punktpaaren maximalen Umwegs, also bei Aufhebung der Beschränkung auf Eckpunktpaare. Am Ende des Abschnittes vergleichen wir die betrachteten Probleme noch mit verwandten Problemen und deren Laufzeitschranken.

In Abschnitt 10 schließlich liefern wir einen Ausblick auf weitergehende Fragestellungen.

Während der Arbeit an der vorliegenden Diplomarbeit ist auch ein Java-Programm namens „Polygonal Detour“ entstanden. Es eignet sich gut für die Untersuchung und Visualisierung von Umwegsituationen in Polygonen. Eine Applet-Version findet sich unter <http://www.8ung.at/a.gruene/>.

2 Definitionen

In diesem Abschnitt sollen die benötigten Begriffe genau definiert werden. Die Hauptobjekte der Diplomarbeit sind *einfache Polygone*. Preparata und Shamos definieren sie in [PS90] wie folgt (aus dem Englischen übersetzt):

Ein Polygon ist definiert durch eine endliche Menge von Liniensegmenten, so dass jeder Segmentendpunkt zu genau zwei Segmenten gehört, und keine Untermenge von Segmenten diese Eigenschaft besitzt. Die Liniensegmente sind die Kanten und ihre Endpunkte sind die Eckpunkte des Polygons. [...]

Ein Polygon ist einfach, wenn es kein Paar von nicht aufeinander folgenden Kanten gibt, die einen gemeinsamen Punkt haben. Ein einfaches Polygon teilt die Ebene in zwei disjunkte Gebiete, das Innere (beschränkt) und das Äußere (unbeschränkt), welche vom Polygon getrennt werden. [...] (Beachte: Häufig wird der Begriff Polygon auch dazu benutzt, die Vereinigung des Inneren und der Grenze zu bezeichnen.)

Im folgenden schließen wir uns dem in Klammern hinzugefügten Sprachgebrauch an. Hat man nun zwei Punkte p, q in einem einfachen Polygon P gegeben, so kann man, wie schon in der Einleitung erwähnt, folgende zwei in Abb. 1 dargestellten Verbindungskurven betrachten:

1. Die direkte Verbindungsstrecke \overline{pq} .
2. Den *kürzesten Weg* $\pi(p, q)$ von p nach q durch das Polygon.

Die Länge des Weges $\pi(p, q)$ bezeichnen wir mit $|\pi(p, q)|$. Dann gilt offensichtlich immer $|\pi(p, q)| \geq |\overline{pq}|$, also $\frac{|\pi(p, q)|}{|\overline{pq}|} \geq 1$.

Der Bruch beschreibt dabei den relativen Umweg, den man in Kauf nehmen muss, wenn man gezwungen ist, durch das Polygon zu gehen. Wir nennen diesen Wert kurz den *Umweg zwischen p und q durch P* . Genauer definieren wir:

$$(1) \quad u_P(p, q) := \begin{cases} \frac{|\pi(p, q)|}{|\overline{pq}|} & \text{für } p \neq q \\ 1 & \text{für } p = q \end{cases}$$

Die Definition von $u_P(p, q)$ für $p = q$ mag dabei etwas seltsam anmuten, aber sie ist, wenn man Grenzübergänge betrachtet, durchaus einleuchtend und erleichtert uns die Stetigkeitsargumentation im Abschnitt 4. Zu einem Polygon P ist dann der *maximale Umweg* gegeben durch:

$$(2) \quad u_P^{\max} := \sup_{p, q \in P} u_P(p, q)$$

Abbildung 1 zeigt für ein bestimmtes Polygon P ein Punktpaar (p, q) , das den maximalen Umweg u_P^{\max} annimmt. Alle weiteren notwendigen Definitionen werden wir an den Stellen angeben, an denen sie benötigt werden.

3 Rückführung auf Randpunkte

In diesem Abschnitt wird gezeigt, dass es zu zwei Polygonpunkten $p, q \in P$ immer zwei Randpunkte gibt, deren Umweg mindestens genauso groß ist wie derjenige von p und q . Dadurch können wir uns im folgenden bei der Suche nach Punktpaaren maximalen Umwegs auf Randpunktpaare beschränken.

Bei der Analyse spielt die Sichtbarkeitsbeziehung zweier Punkte eine wichtige Rolle.

Zwei Punkte p, q aus einem Polygon P heißen *gegenseitig sichtbar*, wenn ihre direkte Verbindung \overline{pq} ganz in P liegt. Offenbar sind damit p, q genau dann gegenseitig sichtbar, wenn $u_P(p, q) = 1$ ist.

Der Spezialfall eines konvexen Polygons ist deshalb sehr leicht zu beschreiben. Ein Polygon ist genau dann konvex, wenn sich alle Punkte gegenseitig sehen können. Das ist genau dann der Fall, wenn der maximale Umweg $u_P^{\max} = 1$ ist.

Interessanter ist der Fall, in dem p und q nicht gegenseitig sichtbar sind. Ein solches Punktpaar nennen wir *in P^C gegenseitig sichtbar*, wenn \overline{pq} bis auf p und q ganz im Komplement von P , also in P^C liegt.

Folgendes Lemma, dessen Aussage in Abbildung 3 veranschaulicht wird, zeigt, dass es zu jedem nicht gegenseitig sichtbaren Punktpaar ein in P^C gegenseitig sichtbares Randpunktpaar mit mindestens gleich großem Umweg gibt. Die strengere Aussage in Unterpunkt 2. macht deutlich, dass in nicht-konvexen Polygonen Punktpaare maximalen Umwegs in P^C gegenseitig sichtbar sein müssen. Denn sonst würde man durch die strikte Ungleichung einen Widerspruch zur Maximalität ihres Umwegs erhalten. Auf diese Weise wird die Aussage von Lemma 3.1 auch im Beweis von Korollar 5.1 (siehe S. 27) benutzt.

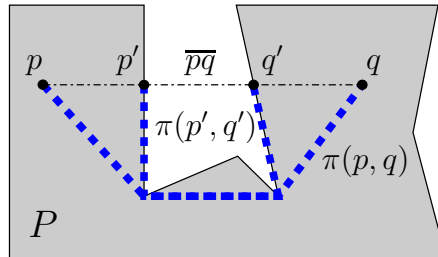


Abbildung 3: Randpunkte mit größerem Umweg

Lemma 3.1

1. Zu zwei beliebigen Punkten p, q eines Polygons P gibt es immer zwei Randpunkte $p', q' \in \partial P$ mit $u_P(p', q') \geq u_P(p, q)$.
2. Sind p und q weder in P noch in P^C gegenseitig sichtbar, dann können als p' und q' Punkte auf \overline{pq} gewählt werden, die in P^C gegenseitig sichtbar sind und für die sogar die strikte Ungleichung $u_P(p', q') > u_P(p, q)$ gilt.

Beweis. Sind p und q gegenseitig sichtbar, dann ist $u_P(p, q) = 1$ und deshalb erfüllt jedes beliebige Randpunktpaar $(p, q) \in \partial P \times \partial P$ die Behauptung. Sind p und q in P^C gegenseitig sichtbar, dann sind beide Punkte Randpunkte und sie erfüllen selbst die Behauptung.

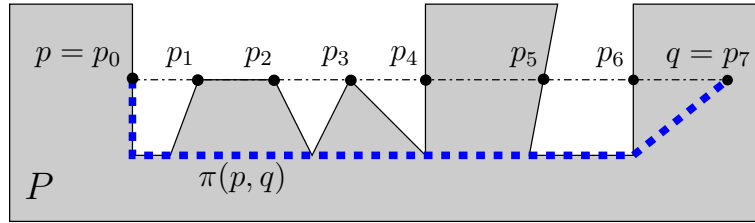


Abbildung 4: „echte“ Randschnittpunkte von \overline{pq}

Sei also $(p, q) \in P \times P$ ein Punktpaar, das weder in P noch in P^C gegenseitig sichtbar ist. Wir werden für ein solches Punktpaar die Gültigkeit der Aussage aus Unterpunkt 2. zeigen. Damit ist auch der Beweis von 1. abgeschlossen.

Seien nun $p_0 := p, p_n := q$ und p_1 bis p_{n-1} alle von p und q verschiedenen „echten“ Schnittpunkte von \overline{pq} mit dem Rand ∂P . Dabei nennen wir einen Punkt in $\overline{pq} \cap \partial P$ einen *echten Randschnittpunkt*, wenn er $\overline{pq} \cap P^C$ berührt. Dadurch schließen wir alle Randpunkte aus, die auf dem Inneren eines Stückes von \overline{pq} liegen, das genau über den Rand ∂P verläuft, wie z.B. bei der Strecke $\overline{p_1 p_2}$ in Abb.4. Da (p, q) weder in P noch in P^C gegenseitig sichtbar ist, ist $n \geq 2$.

Da alle Punkte auf einer Linie liegen, ist

$$(3) \quad |\overline{pq}| = \sum_{i=0}^{n-1} |\overline{p_i p_{i+1}}|$$

Andererseits gilt für $|\pi(\cdot, \cdot)|$ die Dreiecksungleichung, da sonst der aus den Teilwegen zusammengesetzte Weg kürzer wäre, im Widerspruch zur Definition von π . Es gilt also:

$$(4) \quad |\pi(p, q)| \leq \sum_{i=0}^{n-1} |\pi(p_i, p_{i+1})|$$

Nehmen wir zunächst an, dass sogar die strikte Ungleichung gilt:

$$(5) \quad |\pi(p, q)| < \sum_{i=0}^{n-1} |\pi(p_i, p_{i+1})|$$

Analog zum Beweis von Lemma 3 in [EBKLL01] folgt:

$$\begin{aligned}
 (6) \quad u_P(p, q) &= \frac{|\pi(p, q)|}{|\overline{pq}|} \stackrel{(3),(5)}{<} \frac{\sum_{i=0}^{n-1} |\pi(p_i, p_{i+1})|}{\sum_{i=0}^{n-1} |\overline{p_i p_{i+1}}|} \\
 &\leq \max_{0 \leq i \leq n-1} \frac{|\pi(p_i, p_{i+1})|}{|\overline{p_i p_{i+1}}|} = \max_{0 \leq i \leq n-1} u_P(p_i, p_{i+1})
 \end{aligned}$$

Um die letzte Ungleichung einzusehen, muss man nur erkennen, dass für positive Zahlen, falls $a_i/b_i \leq q$ für alle i gilt, auch $\sum_i a_i / \sum_i b_i \leq q$ ist. Dies wiederum ist leicht zu sehen, wenn man die beteiligten Ungleichungen jeweils mit dem auftretenden Nenner multipliziert.

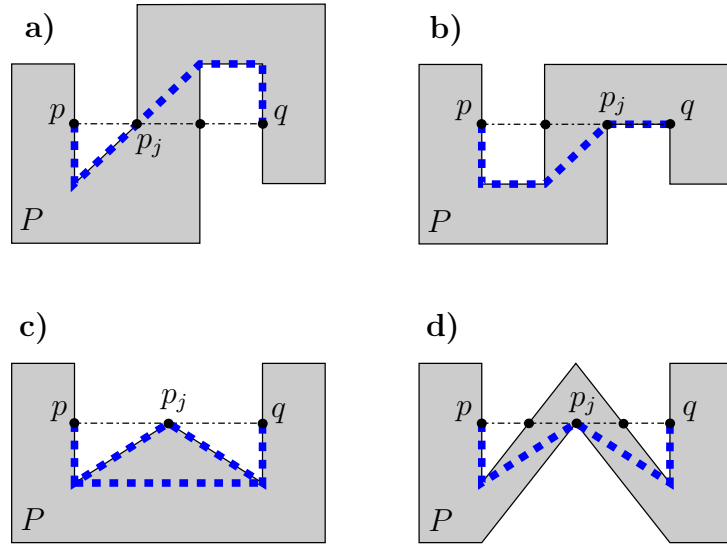


Abbildung 5: Fallunterscheidung

Im Falle der strikten Ungleichung (5) ist mit (6) schon der wichtigste Teil von Aussage 2. bewiesen. Es bleibt für diesen Fall nur noch zu zeigen, dass Punktpaare (p_i, p_{i+1}) , die das Maximum auf der rechten Seite von Gleichung (6) annehmen, für die also $u_P(p_i, p_{i+1}) = \max_{0 \leq i \leq n-1} u_P(p_i, p_{i+1})$ gilt, in P^C gegenseitig sichtbar sind. Diese leichte Aussage zeigen wir am Ende des Beweises.

Zunächst wenden wir uns dem noch nicht behandelten Fall zu, in dem die strikte Ungleichung (5) nicht gilt. Dann gilt in (4) die Gleichheit. Das ist wegen der Eindeutigkeit des kürzesten Weges (siehe z.B. Proposition 1 in [Mit00]) nur möglich, wenn $\pi(p, q)$ tatsächlich der aus den Wegen $\pi(p_i, p_{i+1})$ zusammengesetzte Weg ist. Da außerdem $n \geq 2$ ist, gibt es einen echten Randschnittpunkt $p_j \notin \{p, q\}$, der von $\pi(p, q)$ besucht wird.

Hier können die in Abbildung 5 gezeigten Fälle auftreten. Kreuzt $\pi(p, q)$ in p_j die Strecke \overline{pq} , dann muss ein angrenzender Teil von \overline{pq} zu P gehören (vgl. Abb. 5a)). Verläuft $\pi(p, q)$ auf mindestens einem angrenzenden Teilstück direkt auf \overline{pq} , dann gehört natürlich dieses Teilstück selbst zu P (siehe 5b)). Berührt $\pi(p, q)$ die Strecke \overline{pq} in p_j nur, aber kreuzt sie nicht und bleibt auch nicht auf ihr, dann muss, da man sonst $\pi(p, q)$ verkürzen könnte (siehe 5c)), auch wieder ein angrenzender Teil von \overline{pq} zu P gehören (siehe 5d)).

In allen Fällen gibt es also ein k mit $\overline{p_k p_{k+1}} \subset P$, d.h. $u_P(p_k, p_{k+1}) = 1$, und es folgt:

$$\begin{aligned}
(7) \quad u_P(p, q) &\stackrel{(3), (4)}{\leq} \frac{\sum_{i=0}^{n-1} |\pi(p_i, p_{i+1})|}{\sum_{i=0}^{n-1} |\overline{p_i p_{i+1}}|} \\
&= \frac{\left(\sum_{i \in \{0, \dots, n-1\} \setminus \{k\}} |\pi(p_i, p_{i+1})| \right) + |\pi(p_k, p_{k+1})|}{\left(\sum_{i \in \{0, \dots, n-1\} \setminus \{k\}} |\overline{p_i p_{i+1}}| \right) + |\overline{p_k p_{k+1}}|} \\
&\stackrel{\text{s.u.}}{<} \frac{\sum_{i \in \{0, \dots, n-1\} \setminus \{k\}} |\pi(p_i, p_{i+1})|}{\sum_{i \in \{0, \dots, n-1\} \setminus \{k\}} |\overline{p_i p_{i+1}}|} \\
&\leq \max_{0 \leq i \leq n-1} u_P(p_i, p_{i+1})
\end{aligned}$$

Dabei folgt die strikte Ungleichung aus der Tatsache, dass für $a > b > 0$ und $c > 0$ gilt (multipliziere zum Nachweis mit den Nennern):

$$\frac{a+c}{b+c} < \frac{a}{b}$$

Wir haben also sowohl im Fall der strikten Ungleichung (5) als auch im Fall der Gleichheit in (4) die strikte Ungleichung (6) bzw. (7) gezeigt. Über die Fallunterscheidung in Abbildung 5 hätte man alternativ zeigen können, dass es gar nicht möglich ist, dass der kürzeste Weg $\pi(p, q)$ alle echten Randschnittpunkte p_0, \dots, p_n besucht, und dass deshalb die Gleichheit in (4) wegen der Eindeutigkeit der kürzesten Wege nicht gelten kann.

Da (6) bzw. (7) nun gezeigt ist, bleibt, um die Aussage 2. zu vervollständigen, nur noch zu beweisen, dass ein Punktpaar (p_l, p_{l+1}) von aufeinander folgenden echten Randschnittpunkten von \overline{pq} , welches den maximalen Umweg unter solchen Paaren annimmt, in P^C gegenseitig sichtbar ist.

Sei also (p_l, p_{l+1}) so ein Punktpaar. Da (p, q) nicht gegenseitig sichtbar ist, gilt $u_P(p, q) > 1$, also wegen den gezeigten Ungleichungen (6) bzw. (7) auch $u_P(p_l, p_{l+1}) > 1$. D.h. (p_l, p_{l+1}) ist ebenfalls nicht gegenseitig sichtbar. Nach Konstruktion sind aber alle Paare aufeinander folgender Punkte entweder gegenseitig sichtbar oder in P^C gegenseitig sichtbar. Also ist (p_l, p_{l+1}) in P^C gegenseitig sichtbar. \square

4 Existenz eines Umwegmaximums

In diesem Abschnitt werden wir die Existenz eines Punktpaares mit maximalem Umweg beweisen. Die Argumentation läuft hauptsächlich über die Stetigkeit des Umweges $u_P(.,.)$ als Funktion auf $P \times P$ und die Kompaktheit von $P \times P$. Nur in dem Spezialfall eines spitzen Eckpunktes (siehe Abb. 6), in der eine Unstetigkeit von $u_P(.,.)$ auftritt, ist eine genauere Betrachtung notwendig.

Jedes Polygon P ist abgeschlossen und beschränkt, also kompakt in \mathbb{R}^2 . Damit ist auch $P \times P$ kompakt in $\mathbb{R}^2 \times \mathbb{R}^2$. Könnten wir nun zeigen, dass $u_P(.,.)$ überall stetig ist, so würde $u_P(.,.)$ auf $P \times P$ sein Maximum annehmen. Die Existenz eines Punktpaares mit maximalem Umweg wäre gezeigt.

Durch kanonische Abschätzungen ist leicht zu zeigen, dass $|\pi(.,.)|$ auf $P \times P$ stetig ist. Natürlich ist auch $|\overline{pq}|$ stetig. Beides ist am leichtesten bezüglich der 1-Norm $\|(p, q)\| := \|p\|_{\mathbb{R}^2} + \|q\|_{\mathbb{R}^2}$ zu zeigen, die bekanntermaßen zur euklidischen Norm äquivalent ist. Die Stetigkeit von Zähler und Nenner impliziert dann die Stetigkeit von $u_P(.,.)$ in Punktpaaren (p, q) mit $p \neq q$.

Ist $p = q$ kein Eckpunkt, dann sind auch Punktpaare in einer kleinen Umgebung gegenseitig sichtbar, also bleibt hier $u_P(p, q) = 1$. Demnach ist $u_P(.,.)$ auch in solchen Punktpaaren stetig. Gleiches gilt, falls $p = q$ ein Eckpunkt ist, der aus dem Polygon herausragt, dessen Außenwinkel also nicht kleiner als π ist.

Die Funktion $u_P(.,.)$ ist also in allen Punktpaaren $(p, q) \in P \times P$ stetig, bei denen $p \neq q$ ist, oder bei denen $p = q$ kein *spitzer Eckpunkt* ist. Ein solcher spitzer Eckpunkt ist in Abbildung 6 dargestellt. Er liegt vor, wenn die Ecke bildlich gesprochen in das Polygon hinein sticht, d.h. wenn der Außenwinkel kleiner als π ist. Ist $p = q$ ein spitzer Eckpunkt, dann ist $u_P(.,.)$ an dieser Stelle tatsächlich unstetig.

Zur Analyse dieses Falles nennen wir den Eckpunkt c und den zugehörigen Außenwinkel γ . Da die Ecke in c spitz ist, ist also $\gamma < \pi$. Weiterhin sei ε so klein, dass in der zugehörigen ε -Scheibe $B_\varepsilon(c)$ keine weitere Randkante von P auftaucht. Jetzt kann man eine Umgebung U von $(c, c) \in P \times P$ so klein wählen, dass für alle Punktpaare $(p, q) \in U$ sowohl p als auch q in $B_\varepsilon(c)$ liegen. Wir wollen für solche Punktpaare $u_P(p, q)$ nach oben abschätzen. Dazu betrachten wir nur Punktpaare aus U , die nicht gegenseitig sichtbar sind, denn sonst lässt sich der Umweg ja durch 1 abschätzen.

Nach Lemma 3.1 ist für solche Paare $u_P(p, q) \leq u_P(p', q')$, wobei p' und q' die beiden Randschnittpunkte von \overline{pq} sind (siehe Abb. 6). Es genügt folglich zur Aufstellung der gesuchten Abschätzung Randpunktpaare (p', q') zu betrachten, bei denen die beiden Punkte auf unterschiedlichen Randkanten liegen.

Verschiebt man nun p' und q' auf dem Rand so, dass die Winkel im Dreieck $cp'q'$ gleich bleiben, dann bleibt offensichtlich auch der Umweg gleich. Also hängt der Umweg von solchen sich gegenüberliegenden Randpunkten bei festem γ nur vom Winkel $\beta_{p'}$ (bzw. $\beta_{q'}$) ab. Daher genügt es bei der Suche nach einem Randpunktpaar mit maximalem Umweg, p' fest zu lassen und nur q' zu verschieben.

Da in dieser Situation alle betrachteten kürzesten Wege über die sich in c treffenden Kanten verlaufen, ist sie analog zum Kettenfall. Nach Lemma 1 von [EBKLL01] gibt es dann tatsächlich ein Maximum, und es ist charakterisiert durch (siehe auch Lemma 5.2 auf S. 26):

$$(8) \quad \cos \beta_{q'} = -\frac{|\overline{p'q'}|}{|\pi(p', q')|}$$

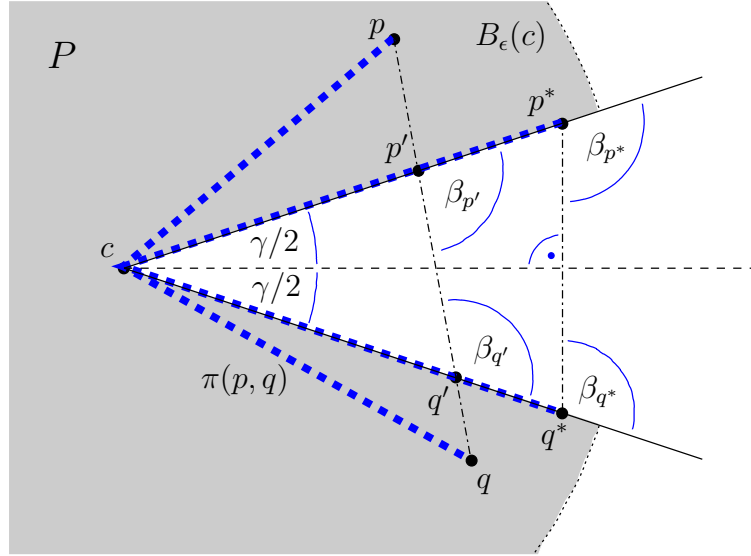


Abbildung 6: maximaler Umweg in einer Umgebung eines spitzen Eckpunktes

Diese Gleichung ist genau dann erfüllt, wenn p' und q' gleich weit von c entfernt sind, wie dies bei den Punkten p^* und q^* in Abb. 6 der Fall ist. Um dies einzusehen, kann man den Winkel $\pi - \beta_{q^*}$ in der Ecke q^* des Dreiecks cp^*q^* betrachten. Mit ihm gilt wegen des eingezeichneten rechten Winkels, und weil $|\overline{q^*c}| = \pi(p^*, q^*)/2$ ist, tatsächlich:

$$\cos \beta_{q^*} = -\cos(\pi - \beta_{q^*}) = -\frac{\frac{|\overline{p^*q^*}|}{2}}{\frac{|\pi(p^*, q^*)|}{2}} = -\frac{|\overline{p^*q^*}|}{|\pi(p^*, q^*)|}$$

Betrachtet man in dem gleichen rechtwinkligen Dreieck den Winkel $\gamma/2$, so erhält man analog:

$$\sin \frac{\gamma}{2} = \frac{|\overline{p^*q^*}|}{|\pi(p^*, q^*)|} = \frac{1}{u_P(p^*, q^*)}$$

Also ist insgesamt gezeigt:

Lemma 4.1 *Ist P ein Polygon und c eine Ecke von P mit Außenwinkel $\gamma < \pi$, also eine spitze Ecke, dann gibt es eine Umgebung U von $(c, c) \in P \times P$, so dass für alle Punktpaare $(p, q) \in U$ gilt:*

$$u_P(p, q) \leq \frac{1}{\sin(\frac{\gamma}{2})}$$

Der maximale Wert wird von Punktpaaren angenommen, dessen Punkte sich auf den Kanten an c so gegenüberliegen, dass sie gleich weit von c entfernt sind.

Zusammen mit den Stetigkeitsüberlegungen vom Anfang dieses Abschnittes folgt sofort die Existenz eines Punktpaares mit maximalem Umweg:

Satz 4.1 *In jedem Polygon P gibt es ein Punktpaar $(p, q) \in P \times P$ mit maximalem Umweg, so dass also $u_P(p, q) = u_P^{\max}$ ist.*

Beweis. Sei $(p_i, q_i)_{i \in \mathbb{N}}$ eine Maximalfolge in $P \times P$, d.h. $\lim_{i \rightarrow \infty} u_P(p_i, q_i) = u_P^{\max}$. Da $P \times P$ kompakt ist, konvergiert eine Teilfolge $(\tilde{p}_i, \tilde{q}_i)$ gegen ein Punktpaar (p, q) . Ist nun $p \neq q$ oder $p = q$ keine spitze Ecke, dann ist $u_P(\cdot, \cdot)$ in (p, q) stetig, und deshalb gilt $u_P(p, q) = \lim_{i \rightarrow \infty} u_P(\tilde{p}_i, \tilde{q}_i) = u_P^{\max}$. Ist hingegen $p = q$ eine spitze Ecke, dann liegen wegen der Teilfolgenkonvergenz fast alle Folgenglieder $(\tilde{p}_i, \tilde{q}_i)$ in der in Lemma 4.1 konstruierten Umgebung U . Daher gilt nach diesem Lemma $u_P^{\max} = \lim_{i \rightarrow \infty} u_P(\tilde{p}_i, \tilde{q}_i) \leq 1/\sin(\gamma/2)$. Andererseits besagt das Lemma auch, dass es ein Punktpaar $(p^*, q^*) \in U \subseteq P \times P$ gibt, mit $u_P(p^*, q^*) = 1/\sin(\gamma/2)$. Also folgt $u_P(p^*, q^*) = u_P^{\max}$. In beiden Fällen ist der Satz bewiesen. \square

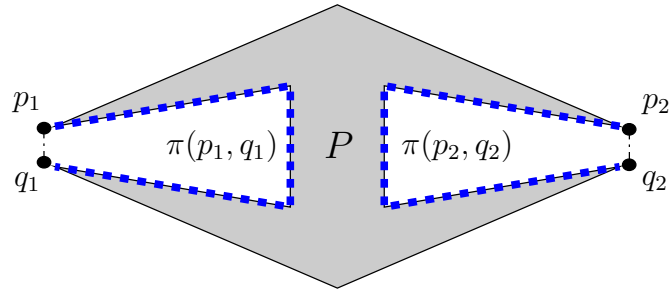


Abbildung 7: Polygon mit zwei Punktpaaren maximalen Umwegs

Neben der Existenzfrage ist auch die Frage nach der Eindeutigkeiteines Punktpaares mit maximalem Umweg zu klären. Sie ist durch Angabe von Gegenbeispielen leicht zu beantworten. Abb. 7 zeigt ein solches Gegenbeispiel.

5 Maximum auf Eckpunkt-Randpunkt-Paar

Nach Lemma 3.1 kann man sich bei der Suche nach einem Punktpaar mit maximalem Umweg auf Randpunkte beschränken. Das Anliegen dieses Abschnittes ist es, weiter zu zeigen, dass man sich sogar auf Eckpunkt-Randpunkt-Paare zurückziehen kann.

Hierzu sind einige Begriffe hilfreich. Wir nennen ein Punktpaar $(p, q) \in \partial P \times \partial P$ *Randpunktpaar*. Es handelt sich um ein *echtes Randpunktpaar*, wenn weder p noch q Eckpunkte von P sind. Analog werden wir den Begriff „echt“ auch für einzelne Randpunkte und für Eckpunkt-Randpunkt-Paare verwenden.

Weiterhin nennen wir ein Element $x \in X$ ein *lokales Maximum* einer Funktion $f : X \rightarrow \mathbb{R}$, wenn es eine Umgebung U von x gibt, so dass für alle $y \in U$ gilt, dass $f(x) \geq f(y)$ ist. Uns genügt also, dass der Funktionswert in der Umgebung nirgends größer ist. Ein Punktpaar $(p, q) \in P \times P$ ist folglich ein *lokales Umwegmaximum*, wenn es eine Umgebung U von (p, q) gibt mit $\forall (p', q') \in U : u_P(p, q) \geq u_P(p', q')$.

Könnten wir nun zeigen, dass echte Randpunktpaare kein lokales Maximum von $u_P(.,.)$ bilden, hätten wir unser Ziel erreicht. Denn nach Satz 4.1 gibt es ein Punktpaar mit maximalem Umweg, und wegen Lemma 3.1 gibt es dann ein Randpunktpaar mit maximalem Umweg. Letzteres Punktpaar bildet insbesondere ein lokales Maximum von $u_P(.,.)$, könnte also dann kein echtes Randpunktpaar sein. D.h. mindestens einer der Randpunkte wäre ein Eckpunkt.

Leider zeigt schon unsere Eckpunktanalyse im Abschnitt 4 (siehe Lemma 4.1), dass echte Randpunktpaare durchaus ein lokales Maximum von $u_P(.,.)$ bilden können. Allerdings werden wir zeigen, dass der dortige Fall, bei dem $\pi(p, q)$ über beide Kanten (die, auf der p liegt, und die, auf der q liegt) läuft, der einzige Fall ist, in dem ein lokales Maximum bei echten Randpunktpaaren auftreten kann. Und er ist unproblematisch, wie wir in Abschnitt 5.4 sehen werden.

Satz 5.1 *In jedem Polygon gibt es ein Eckpunkt-Randpunkt-Paar mit maximalem Umweg.*

Beweis. Im konvexen Fall hat jedes Punktpaar den Umweg 1, also gibt es auch ein Eckpunkt-Randpunkt-Paar mit maximalem Umweg.

Ist das Polygon P nicht konvex, dann gibt es nach Satz 4.1 und Lemma 3.1 ein in P^C gegenseitig sichtbares Randpunktpaar (p, q) mit maximalem Umweg. Handelt es sich um ein Eckpunkt-Randpunkt-Paar, dann gilt der zu zeigende Satz. Sei also (p, q) ein echtes Randpunktpaar.

Nach Guibas und Hershberger [GH87] liegt $\pi(p', q')$ für alle $(p', q') \in d \times e$ in der sogenannten *Sanduhr* (engl. „Hourglass“) von d und e bzgl. P . Es gibt offene und geschlossene Sanduhren. Beide Fälle sind in Abbildung 8 dargestellt.

Bei einer *offenen Sanduhr* sind die beteiligten Kanten durch nach außen konvexe¹ polygonale Ketten verbunden, die sich nirgendwo treffen (siehe Sanduhr von d und e in Abb. 8).

Eine *geschlossene Sanduhr* besteht aus zwei Trichtern mit d bzw. e als Oberseite, deren Fußpunkte mit einer polygonalen Kette verbunden sind. Ein *Trichter* (engl. „Funnel“) ist ein

¹D.h. die Außenwinkel sind $\leq \pi$.

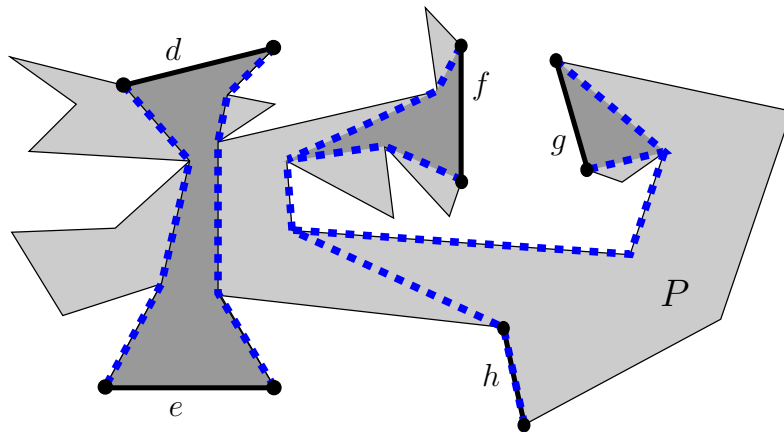


Abbildung 8: verschiedene Sanduhrarten in einem Polygon

Polygon, das durch eine einzelne Kante, die *Oberseite*, und zwei an sie ansetzende nach außen konvexe Randketten, die sich im *Fußpunkt* (in engl. Literatur „Cusp“²) treffen, begrenzt wird. In Abbildung 8 ist die geschlossene Sanduhr von f und g eingezeichnet. Abbildung 10 zeigt eine geschlossene Sanduhr ohne das sie umgebende Polygon.

Schließlich treten auch *entartete Trichter* auf. Dann führen alle kürzesten Wege $\pi(p', q')$ über denselben Eckpunkt der betreffenden Polygonkante. Der beteiligte Trichter besteht nur noch aus dieser Kante und die verbindende Kette setzt direkt am jeweiligen Kantenendpunkt an (siehe Kante h in Abb. 8 und vgl. Abschnitt 5.4).

Nach diesen Definitionen sei kurz daran erinnert, dass wir im Beweis von Satz 5.1 ein echtes Randpunktpaar (p, q) mit maximalem Umweg betrachten, das in P^C gegenseitig sichtbar ist. Es liege p auf der Kante d und q auf der Kante e .

Wegen der geometrischen Situation muss die Sanduhr von d und e geschlossen sein. Zum

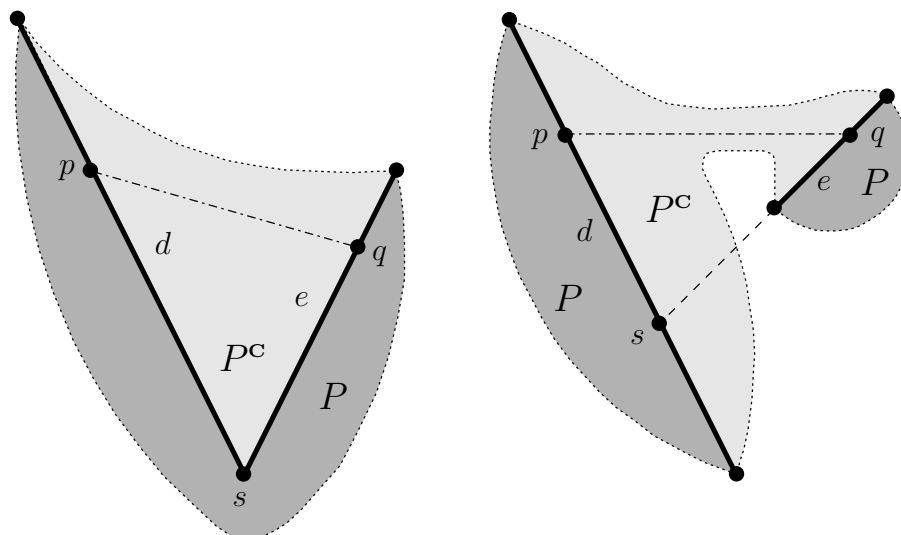


Abbildung 9: in P^C gegenseitig sichtbares, echtes Randpunktpaar (p, q)

²Die englischen Standardbezeichnungen finden sich z.B. in [GHL⁺87]

Fall wird sich aber als relativ harmlos herausstellen, so dass sich der Zusatzaufwand in Grenzen hält.

Als erstes seien die Kanten d und e nicht parallel.

5.1 Nicht-paralleler Fall mit eindeutigen a, b

Die Situation sieht dann so aus wie in Abb. 11. Wir benennen den Abstand von p zum

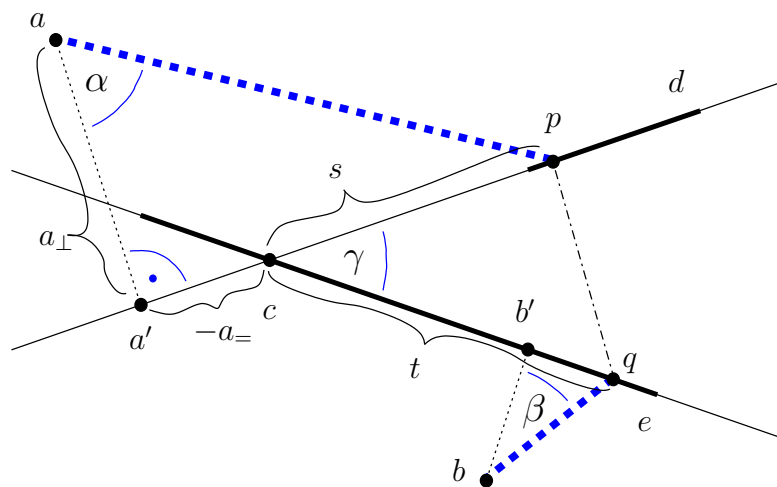


Abbildung 11: Randpunktpaar mit ersten Stücken des kürzesten Weges

Schnittpunkt c mit s , also $s := |\overline{pc}|$ und denjenigen von q zu c mit t , also $t := |\overline{qc}|$. Sei weiterhin wie in Abb. 11 zu sehen a' (bzw. b') die senkrechte Projektion von a (bzw. b) auf die Verlängerung von d (bzw. e).

Dann zerlegen wir a und b noch in eine Komponente, die senkrecht zu der entsprechenden Kante liegt und eine parallele Komponente. Genauer definieren wir $a_{\perp} := |\overline{aa'}|$, $b_{\perp} := |\overline{bb'}|$, $|a_{=}| := |\overline{a'c}|$ und $|b_{=}| := |\overline{b'c}|$, wobei das Vorzeichen von $a_{=}$ (bzw. $b_{=}$) genau dann positiv sei, wenn a' (bzw. b') auf der gleichen Seite von c liegt wie p (bzw. q).

Schließlich brauchen wir zur Vereinfachung der Formeln noch Namen für die Winkel in a und b und definieren α und β durch die Gleichungen $\sin \alpha = (s - a_{=}) / |\overline{ap}|$, $\cos \alpha = a_{\perp} / |\overline{ap}|$, $\sin \beta = (t - b_{=}) / |\overline{bq}|$, und $\cos \beta = b_{\perp} / |\overline{bq}|$. Dabei wird, falls der Trichter von d entartet ist, falls also a ein Eckpunkt von d ist, α zu $\pi/2$ oder $-\pi/2$. Analoges gilt für den Punkt b .

All diese Definitionen bringen den Vorteil, dass wir nun alle Randpunktpaare $(p', q') \in d \times e$ über s' und t' parametrisiert betrachten können. Weil nach unseren Überlegungen (p, q) ein in P^C gegenseitig sichtbares echtes Randpunktpaar ist, gilt $s, t \neq 0$, also $s, t > 0$. Weil (p, q) ein echtes Randpunktpaar ist, also weder p noch q Eckpunkte sind, gilt auch, selbst wenn a oder b auf der zugehörigen Kante d bzw. e liegen, dass $|\overline{ap}| > 0$ und $|\overline{bq}| > 0$ sind.

Der Umweg $u_P(p', q')$ wird jedenfalls eine Funktion in den zwei Variablen s' und t' . Wir

definieren:

$$\begin{aligned}
(9) \quad f(s', t') &:= u_P(p'(s'), q'(t')) \\
&= \frac{|\pi(p'(s'), q'(t'))|}{|p'(s')q'(t')|} \\
&\stackrel{\text{Kosinussatz}}{=} \frac{|p'(s')a| + |\pi(a, b)| + |bq'(s')|}{\sqrt{s'^2 + t'^2 - 2s't' \cos(\gamma)}} \\
&\stackrel{\text{Pythagoras}}{=} \frac{\sqrt{(s' - a_-)^2 + a_\perp^2} + |\pi(a, b)| + \sqrt{(t' - b_-)^2 + b_\perp^2}}{\sqrt{s'^2 + t'^2 - 2s't' \cos(\gamma)}}
\end{aligned}$$

Nun liegt in (p, q) genau dann ein lokales Maximum von $u_P(\cdot, \cdot)$, wenn in dem entsprechenden (s, t) ein lokales Maximum von $f(\cdot, \cdot)$ liegt. Die Einschränkung der Vergleichspunktpaare auf Randpunktpaare ist aufgrund des Beweises von Lemma 3.1 möglich. Ein lokales Maximum von $f(\cdot, \cdot)$ liegt schließlich genau dann vor, wenn für alle $(m, n) \in \mathbb{R}^2$ die Funktion $f(\Delta) := f(s + m\Delta, t + n\Delta)$ ein lokales Maximum in 0 hat.³ Wir berechnen zuerst:

$$\begin{aligned}
(10) \quad \frac{df}{d\Delta} &= \frac{\left(m \frac{(s+m\Delta - a_-)}{\sqrt{(s+m\Delta - a_-)^2 + a_\perp^2}} + n \frac{(t+n\Delta - b_-)}{\sqrt{(t+n\Delta - b_-)^2 + b_\perp^2}} \right)}{\sqrt{(s+m\Delta)^2 + (t+n\Delta)^2 - 2(s+m\Delta)(t+n\Delta) \cos(\gamma)}} \\
&\quad - \frac{m((s+m\Delta) - (t+n\Delta) \cos \gamma) + n((t+n\Delta) - (s+m\Delta) \cos \gamma)}{\left(\sqrt{(s+m\Delta)^2 + (t+n\Delta)^2 - 2(s+m\Delta)(t+n\Delta) \cos(\gamma)} \right)^3} \\
&\quad \cdot \left(\sqrt{(s+m\Delta - a_-)^2 + a_\perp^2} + |\pi(a, b)| + \sqrt{(t+n\Delta - b_-)^2 + b_\perp^2} \right)
\end{aligned}$$

Wir wollen diese Ableitung an der Stelle $\Delta = 0$ auswerten. Benutzt man zur Vereinfachung die sich aus den Definitionen ergebenden Gleichungen $\sin \alpha = (s - a_-) / \sqrt{(s - a_-)^2 + a_\perp^2}$, $\sin \beta = (t - b_-) / \sqrt{(t - b_-)^2 + b_\perp^2}$, $|\overline{pq}| = \sqrt{s^2 + t^2 - 2st \cos \gamma}$ und $|\pi(p, q)| = \sqrt{(s - a_-)^2 + a_\perp^2} + |\pi(a, b)| + \sqrt{(t - b_-)^2 + b_\perp^2}$, so folgt:

$$\begin{aligned}
(11) \quad \frac{df}{d\Delta}(0) &= \frac{(m \sin \alpha + n \sin \beta)}{|\overline{pq}|} \\
&\quad - |\pi(p, q)| \frac{m(s - t \cos \gamma) + n(t - s \cos \gamma)}{|\overline{pq}|^3}
\end{aligned}$$

Wenn nun f in $\Delta = 0$ ein lokales Maximum haben soll, dann muss die erste Ableitung verschwinden. Und das muss für alle $(m, n) \in \mathbb{R}^2$ der Fall sein. Also folgt bei zusätzlicher Multiplikation der Gleichung mit $|\overline{pq}|^3$:

$$(12) \quad \sin \alpha |\overline{pq}|^2 - |\pi(p, q)| (s - t \cos \gamma) \stackrel{!}{=} 0$$

und

$$(13) \quad \sin \beta |\overline{pq}|^2 - |\pi(p, q)| (t - s \cos \gamma) \stackrel{!}{=} 0$$

³Man könnte auch auf die Einführung von Δ verzichten und über die Maximumkriterien für Funktionen auf \mathbb{R}^2 argumentieren. Allerdings ermöglicht die Δ -Betrachtung die Multiplikation mit $|p'q'|^3$ in (16) und damit eine Termvereinfachung.

Es folgt eine kurze Gleichung für $|\pi(p, q)|$.

$$\begin{aligned}
(14) \quad s \sin \alpha + t \sin \beta &\stackrel{(12),(13)}{=} s |\pi(p, q)| \frac{s - t \cos \gamma}{|\overline{pq}|^2} + t |\pi(p, q)| \frac{t - s \cos \gamma}{|\overline{pq}|^2} \\
&= |\pi(p, q)| \frac{s(s - t \cos \gamma) + t(t - s \cos \gamma)}{s^2 + t^2 - 2st \cos \gamma} \\
&= |\pi(p, q)|
\end{aligned}$$

Und Gleichungen für $\sin \alpha$, $\sin \beta$:

$$\begin{aligned}
(15) \quad \sin \alpha (t - s \cos \gamma) &\stackrel{(12),(13)}{=} \sin \beta (s - t \cos \gamma) \\
t (\sin \alpha + \sin \beta \cos \gamma) &= s (\sin \beta + \sin \alpha \cos \gamma)
\end{aligned}$$

Wenn f in $\Delta = 0$ ein lokales Maximum hat, dann darf $df/d\Delta$ in 0 kein $-/+$ - Vorzeichenwechsel haben. Das Vorzeichen von $df/d\Delta$ ist aber identisch mit dem von $(df/d\Delta) \cdot |\overline{p'q'}|^3$. Also muss gelten:

$$(16) \quad \left. \frac{d}{d\Delta} \left(\frac{df}{d\Delta} |\overline{p'q'}|^3 \right) \right|_{\Delta=0} \stackrel{!}{\leq} 0$$

Nun berechnet sich diese Ableitung wie folgt (Ausgangspunkt siehe Gleichung (10)):

$$\begin{aligned}
(17) \quad \frac{d}{d\Delta} \left(\frac{df}{d\Delta} |\overline{p'q'}|^3 \right) &= \\
&\left(\frac{m^2 \sqrt{(s + m\Delta - a_{\perp})^2 + a_{\perp}^2} - m(s + m\Delta - a_{\perp}) \frac{m(s + m\Delta - a_{\perp})}{\sqrt{(s + m\Delta - a_{\perp})^2 + a_{\perp}^2}}}{(s + m\Delta - a_{\perp})^2 + a_{\perp}^2} \right. \\
&+ \left. \frac{n^2 \sqrt{(t + n\Delta - b_{\perp})^2 + b_{\perp}^2} - n(t + n\Delta - b_{\perp}) \frac{n(t + n\Delta - b_{\perp})}{\sqrt{(t + n\Delta - b_{\perp})^2 + b_{\perp}^2}}}{(t + n\Delta - b_{\perp})^2 + b_{\perp}^2} \right) \\
&\cdot ((s + m\Delta)^2 + (t + n\Delta)^2 - 2(s + m\Delta)(t + n\Delta) \cos \gamma) \\
&+ \left(\frac{m(s + m\Delta - a_{\perp})}{\sqrt{(s + m\Delta - a_{\perp})^2 + a_{\perp}^2}} + \frac{n(t + n\Delta - b_{\perp})}{\sqrt{(t + n\Delta - b_{\perp})^2 + b_{\perp}^2}} \right) \\
&\cdot 2(m(s + m\Delta) + n(t + n\Delta) - m(t + n\Delta) \cos \gamma - n(s + m\Delta) \cos \gamma) \\
&- \left(\frac{m(s + m\Delta - a_{\perp})}{\sqrt{(s + m\Delta - a_{\perp})^2 + a_{\perp}^2}} + \frac{n(t + n\Delta - b_{\perp})}{\sqrt{(t + n\Delta - b_{\perp})^2 + b_{\perp}^2}} \right) \\
&\cdot (m(s + m\Delta) + n(t + n\Delta) - m(t + n\Delta) \cos \gamma - n(s + m\Delta) \cos \gamma) \\
&- \left(\sqrt{(s + m\Delta - a_{\perp})^2 + a_{\perp}^2} + |\pi(a, b)| + \sqrt{(t + n\Delta - b_{\perp})^2 + b_{\perp}^2} \right) \\
&\cdot (m^2 + n^2 - 2mn \cos \gamma)
\end{aligned}$$

Wieder werten wir an der Stelle $\Delta = 0$ aus und benutzen dieselben Gleichungen wie beim

Aufstellen von (11). Wenden wir zusätzlich $1 - \sin^2 x = \cos^2 x$ an, folgt:

$$(18) \quad \frac{d}{d\Delta} \left(\frac{df}{d\Delta} |\overline{p'q'}| \right) \Big|_{\Delta=0} = \\ \left(m^2 \frac{\cos^2 \alpha}{|\overline{ap}|} + n^2 \frac{\cos^2 \beta}{|\overline{qb}|} \right) |\overline{pq}|^2 \\ + (m \sin \alpha + n \sin \beta) (m(s - t \cos \gamma) + n(t - s \cos \gamma)) \\ - |\pi(p, q)| (m^2 + n^2 - 2mn \cos \gamma)$$

Mit (18) wird Forderung (16) zu:

$$(19) \quad m^2 \left(\frac{\cos^2 \alpha}{|\overline{ap}|} |\overline{pq}|^2 + \sin \alpha (s - t \cos \gamma) - |\pi(p, q)| \right) \\ + n^2 \left(\frac{\cos^2 \beta}{|\overline{qb}|} |\overline{pq}|^2 + \sin \beta (t - s \cos \gamma) - |\pi(p, q)| \right) \\ + mn (\sin \alpha (t - s \cos \gamma) + \sin \beta (s - t \cos \gamma) + 2 |\pi(p, q)| \cos \gamma) \leq 0$$

Dies soll für alle $m, n \in \mathbb{R}$ gelten. Nun wenden wir folgendes Lemma an, das man leicht direkt aber auch über die negative Semi-Definitheit der Matrix $\begin{pmatrix} a & c/2 \\ c/2 & b \end{pmatrix}$ (vgl. Hauptminoren-Kriterium z.B. auf S. 301 in [Fis95]) beweisen kann:

Lemma 5.1 Für $a, b, c \in \mathbb{R}$ sind folgende Aussagen äquivalent:

(1) Für alle $x, y \in \mathbb{R}$ gilt $ax^2 + by^2 + cxy \leq 0$.

(2) $a \leq 0$, $b \leq 0$ und $4ab \geq c^2$

Dabei kann man in (2) die erste oder zweite Ungleichung auch weglassen.

Also folgen aus (19) folgende 3 Bedingungen:

$$(20) \quad \frac{\cos^2 \alpha}{|\overline{ap}|} |\overline{pq}|^2 + \sin \alpha (s - t \cos \gamma) - |\pi(p, q)| \stackrel{!}{\leq} 0$$

$$(21) \quad \frac{\cos^2 \beta}{|\overline{qb}|} |\overline{pq}|^2 + \sin \beta (t - s \cos \gamma) - |\pi(p, q)| \stackrel{!}{\leq} 0$$

$$(22) \quad 4 \left(\frac{\cos^2 \alpha}{|\overline{ap}|} |\overline{pq}|^2 + \sin \alpha (s - t \cos \gamma) - |\pi(p, q)| \right) \\ \cdot \left(\frac{\cos^2 \beta}{|\overline{qb}|} |\overline{pq}|^2 + \sin \beta (t - s \cos \gamma) - |\pi(p, q)| \right) \\ \stackrel{!}{\geq} (\sin \alpha (t - s \cos \gamma) + \sin \beta (s - t \cos \gamma) + 2 |\pi(p, q)| \cos \gamma)^2$$

Durch Anwendung von (14) und (15) werden diese Bedingungen zu:

$$(23) \quad 0 \leq \frac{\cos^2 \alpha}{|\overline{ap}|} |\overline{pq}|^2 \stackrel{(20),(14)}{\leq} t (\sin \beta + \sin \alpha \cos \gamma)$$

$$(24) \quad 0 \leq \frac{\cos^2 \beta}{|\overline{qb}|} |\overline{pq}|^2 \stackrel{(21),(14)}{\leq} s (\sin \alpha + \sin \beta \cos \gamma)$$

$$\begin{aligned}
(25) \quad & 4 \left(\frac{\cos^2 \alpha}{|ap|} |\overline{pq}|^2 - t(\sin \beta + \sin \alpha \cos \gamma) \right) \\
& \cdot \left(\frac{\cos^2 \beta}{|qb|} |\overline{pq}|^2 - s(\sin \alpha + \sin \beta \cos \gamma) \right) \\
& \stackrel{(22),(14),(15)}{\geq} 4s^2 (\sin \beta + \sin \alpha \cos \gamma)^2
\end{aligned}$$

Aus (25) wird unter Beachtung von $t(\sin \beta + \sin \alpha \cos \gamma)s(\sin \alpha + \sin \beta \cos \gamma) = s^2(\sin \beta + \sin \alpha \cos \gamma)^2$ (siehe (15)):

$$\begin{aligned}
(26) \quad & \frac{\cos^2 \alpha}{|ap|} |\overline{pq}|^2 \frac{\cos^2 \beta}{|qb|} |\overline{pq}|^2 \\
& \geq t(\sin \beta + \sin \alpha \cos \gamma) \frac{\cos^2 \beta}{|qb|} |\overline{pq}|^2 + s(\sin \alpha + \sin \beta \cos \gamma) \frac{\cos^2 \alpha}{|ap|} |\overline{pq}|^2
\end{aligned}$$

Wenden wir (23) bzw. (24) auf die linke Seite dieser Ungleichung an, so erhalten wir:

$$\begin{aligned}
(27) \quad & t(\sin \beta + \sin \alpha \cos \gamma) \frac{\cos^2 \beta}{|qb|} |\overline{pq}|^2 \leq 0 \\
& \text{und} \quad s(\sin \alpha + \sin \beta \cos \gamma) \frac{\cos^2 \alpha}{|ap|} |\overline{pq}|^2 \leq 0
\end{aligned}$$

Also:

$$\begin{aligned}
(28) \quad & \text{falls } \cos \beta \neq 0 : \quad t(\sin \beta + \sin \alpha \cos \gamma) \leq 0 \\
& \text{und falls } \cos \alpha \neq 0 : \quad s(\sin \alpha + \sin \beta \cos \gamma) \leq 0
\end{aligned}$$

Zusammen mit (23), (24) und $s, t > 0$ folgt:

$$\begin{aligned}
(29) \quad & \text{falls } \cos \beta \neq 0 : \quad \sin \beta + \sin \alpha \cos \gamma = 0 \\
& \text{und falls } \cos \alpha \neq 0 : \quad \sin \alpha + \sin \beta \cos \gamma = 0
\end{aligned}$$

Also folgt im ersten Fall ($\cos \beta \neq 0$):

$$\begin{aligned}
(30) \quad 0 &= \sin \beta(s - t \cos \gamma) + \sin \alpha \cos \gamma(s - t \cos \gamma) \\
&\stackrel{(15)}{=} \sin \alpha(t - s \cos \gamma) + \sin \alpha \cos \gamma(s - t \cos \gamma) \\
&= \sin \alpha t(1 - \cos^2 \gamma) \\
&= t \sin \alpha \sin^2 \gamma
\end{aligned}$$

Was wegen $\sin^2 \gamma \neq 0$ und $t \neq 0$ das Verschwinden von $\sin \alpha$ impliziert. Analog können wir im zweiten Fall vorgehen und erhalten insgesamt:

$$\begin{aligned}
(31) \quad & \text{falls } \cos \beta \neq 0 : \quad \sin \alpha = 0 \\
& \text{und falls } \cos \alpha \neq 0 : \quad \sin \beta = 0
\end{aligned}$$

Da aber $\sin \alpha = 0 \Rightarrow \cos \alpha \neq 0$ und $\sin \beta = 0 \Rightarrow \cos \beta \neq 0$ gelten, folgt in beiden Fällen $\sin \alpha = \sin \beta = 0$, was mit (14) einen Widerspruch zu $|\pi(p, q)| > 0$ darstellt.

Also kann ein lokales Maximum in einem echten Randpunktpaar bei nicht parallelen Kanten nur vorliegen, wenn $\cos \alpha = \cos \beta = 0$ ist. D.h. a muss auf der Kante d liegen und b auf der Kante e . Also führt der kürzeste Weg $\pi(p, q)$ über beide Kanten. Diesen Fall betrachten wir in Abschnitt 5.4.

5.2 Paralleler Fall mit eindeutigen a, b

Wenn die Kanten d und e parallel zueinander sind, gehen wir ähnlich vor, wie im nicht parallelen Fall. Um wieder p' und q' über s' und t' parametrisieren zu können, verlängern wir die Kanten ins Unendliche und wählen auf einer Seite der eigentlichen Kanten ein sich direkt gegenüber liegendes Punktpaar (p_0, q_0) als Bezugspunkt. Wir definieren also $s := |\overline{pp_0}|$ und $t := |\overline{qq_0}|$. Außerdem nennen wir den Abstand der beiden Geraden δ , also $\delta := |\overline{p_0q_0}|$.

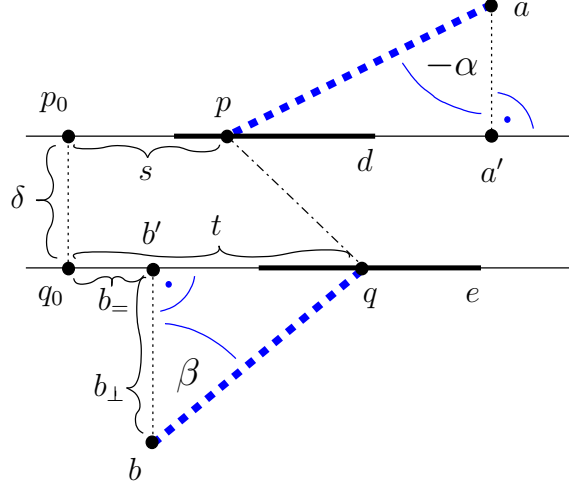


Abbildung 12: Randpunktpaar bei parallelen Kanten

Die restlichen Bezeichnungen $(a, b, a_-, b_-, a_\perp, b_\perp, \alpha, \beta)$ seien wie im nicht-parallelen Fall. Siehe dazu auch Abb. 12.

Liegt nun in (p, q) ein lokales Maximum, dann ist dort insbesondere ein lokales Maximum im Vergleich zu Punktpaaren (p', q') , bei denen die Differenz $\Delta := t' - s'$ gleich ist. Definiere also:

$$\begin{aligned}
 (32) \quad f(s') &:= u_P(p'(s'), q'(s' + \Delta)) \\
 &= \frac{|p'(s')a| + |\pi(a, b)| + |bq'(s' + \Delta)|}{|p'(s')q'(s' + \Delta)|} \\
 &= \frac{\sqrt{(s' - a_-)^2 + a_\perp^2} + |\pi(a, b)| + \sqrt{(s' + \Delta - b_-)^2 + b_\perp^2}}{\sqrt{\Delta^2 + \delta^2}}
 \end{aligned}$$

Wieder berechnen wir die erste Ableitung:

$$(33) \quad \frac{df}{ds'} = \frac{\frac{s' - a_-}{\sqrt{(s' - a_-)^2 + a_\perp^2}} + \frac{s' + \Delta - b_-}{\sqrt{(s' + \Delta - b_-)^2 + b_\perp^2}}}{\sqrt{\Delta^2 + \delta^2}}$$

Die zweite Ableitung ist dann:

$$\begin{aligned}
 (34) \quad \frac{d^2 f}{ds'^2} &= \frac{1}{\sqrt{\Delta^2 + \delta^2}} \cdot \left(\frac{\sqrt{(s' - a_-)^2 + a_\perp^2} - (s' - a_-) \frac{s' - a_-}{\sqrt{(s' - a_-)^2 + a_\perp^2}}}{(s' - a_-)^2 + a_\perp^2} \right. \\
 &\quad \left. + \frac{\sqrt{(s' + \Delta - b_-)^2 + b_\perp^2} - (s' + \Delta - b_-) \frac{s' + \Delta - b_-}{\sqrt{(s' + \Delta - b_-)^2 + b_\perp^2}}}{(s' + \Delta - b_-)^2 + b_\perp^2} \right)
 \end{aligned}$$

Durch ähnliche Umformungen wie beim Übergang zu Gleichung (18) im nicht-parallelen Fall, wird dies an der Stelle $s' = s$ zu:

$$(35) \quad f''(s) = \frac{1}{\sqrt{\Delta^2 + \delta^2}} \left(\frac{\cos^2 \alpha}{|\overline{ap}|} + \frac{\cos^2 \beta}{|\overline{qb}|} \right)$$

Damit in (p, q) ein lokales Maximum vorliegen kann, muss, wie schon im nicht-parallelen Fall benutzt, $f'' \leq 0$ sein. Dies ist offensichtlich nur dann möglich, wenn wieder $\cos \alpha = \cos \beta = 0$ ist, wenn also a und b auf den Kanten d bzw. e liegen. Wieder ist also auf Abschnitt 5.4 zu verweisen.

5.3 Sprung von $\pi(p', q')$ zwischen Ecken

In den unmittelbar vorangehenden Abschnitten sind wir davon ausgegangen, dass $\pi(p', q')$

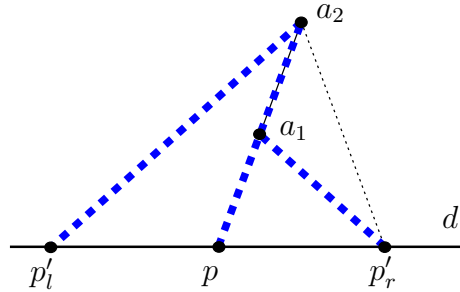


Abbildung 13: Sprung von $\pi(p', q')$ zwischen Ecken

in einer Umgebung von (p, q) über die gleichen Sanduhrecke a bzw. b verläuft. In der Einleitung des Beweises auf S. 18 hatten wir schon den Ausnahmefall erwähnt, in dem $\pi(p', q')$ zu beiden Seiten von p an unterschiedlichen aber benachbarten Sanduhrecke a_1 bzw. a_2 hängt. p liegt dann auf der Geraden durch a_1 und a_2 . Abgebildet ist dieser Fall auch in Abb. 10 auf S.18 für den Punkt p^* .

Wir werden hier nur die Ausnahme im Punkt p betrachten. Liegt sie hingegen in q vor oder gleichzeitig in p und q , dann kann man leicht die folgende Argumentation übertragen.

Sei a_2 , wie in Abb. 13 dargestellt, diejenige Sanduhrecke, die weiter von der Kante d entfernt liegt. Mit den Ausdrücken „links“ und „rechts“ beziehen wir uns ebenfalls auf diese Abbildung. Ohne Einschränkung verlaufe $\pi(p', q')$, wenn p' links von p liegt, über a_2 und sonst über a_1 .

Wir wollen jetzt den tatsächlichen Umweg $u_P(p', q')$ vergleichen mit dem Umweg, der entstehen würde, wenn $\pi(p', q')$ zu beiden Seiten von p über a_2 liefe. Diesen Umweg nennen wir $u_P^{a_2}(p', q')$ und den zugehörigen kürzesten Weg $\pi^{a_2}(p', q') := \overline{p'a_2} \oplus \pi(a_2, q')$.

Falls p' links von p liegt oder gleich p ist, gilt $u_P^{a_2}(p', q') = u_P(p', q')$. Falls p' rechts von p liegt, gilt wegen der Dreiecksungleichung $|\overline{p'a_2}| \leq |\overline{p'a_1}| + |\overline{a_1a_2}|$ dass $|\pi(p', q')| \geq |\pi^{a_2}(p', q')|$ ist, also auch $u_P(p', q') \geq u_P^{a_2}(p', q')$.

Ist nun (p, q) ein lokales Maximum von $u_P(\cdot, \cdot)$ auf dem Kantenpaar (d, e) , ist also für alle $(p', q') \in U \subseteq d \times e$ $u_P(p, q) \geq u_P(p', q')$, dann ist also auch $u_P^{a_2}(p, q) = u_P(p, q) \geq u_P(p', q') \geq u_P^{a_2}(p', q')$. D.h. (p, q) ist genauso ein lokales Maximum bzgl. $u_P^{a_2}(\cdot, \cdot)$. Daher können wir wieder die Überlegungen des normalen parallelen oder unparallelen Falles anwenden, und erhalten, dass $\pi(p', q')$ über beide Kanten laufen muss.

5.4 $\pi(p', q')$ läuft über beide Kanten

Nach dem bisher gezeigten, finden wir entweder ein Eckpunkt-Randpunkt-Paar mit maxi-

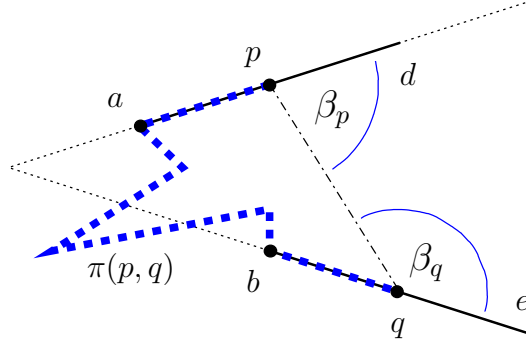


Abbildung 14: $\pi(p, q)$ läuft über beide Kanten

malem Umweg, oder aber ein maximales echtes Randpunktpaar (p, q) von dem aus $\pi(p, q)$ über beide Kanten d und e läuft.

Letzteres ist offensichtlich nur dann möglich, wenn $\pi(p, q)$ von p und von q aus an Kantenendpunkten (a von d bzw. b von e) hängt, was wiederum nur möglich ist (siehe Def. eines Trichters auf S. 5), wenn die beiden Trichter entartet sind.

In einem solchen Fall hängen für alle Punktpaare $(p', q') \in d \times e$ die kürzesten Wege $\pi(p', q')$ an den gleichen Kantenendpunkten a bzw. b . Dann ist aber die Situation genau wie bei polygonalen Ketten. Denn, wenn man die Situation auf der polygonalen Kette $C := d \oplus \pi(a, b) \oplus e$ betrachtet, ergibt sich für alle Punktpaare $(p', q') \in d \times e$ derselbe kürzeste Weg $\pi(p', q')$. Und somit entsprechen sich auch alle Umwegüberlegungen. Wir können direkt die Lemmata 1 und 2 aus [EBKLL01] benutzen und erhalten insgesamt:

Lemma 5.2 *Führt für einen echten Randpunkt p der kürzeste Weg $\pi(p, q)$ über die Kante d , auf der p liegt, so gilt dies auch für alle anderen Randpunkte $p' \in d$ und die zugehörigen kürzesten Wege $\pi(p', q)$. $u_P(p, q)$ ist genau dann maximal verglichen mit den anderen $u_P(p', q)$, wenn $\cos \beta_p = -|\overline{pq}| / |\pi(p, q)|$ ist.*

Ist zusätzlich (p, q) ein echtes Randpunktpaar und führt $\pi(p, q)$ auch über die Kante e von q , dann ist $u_P(p, q)$ genau dann maximal gegenüber $u_P(p', q')$ für alle anderen $(p', q') \in d \times e$, wenn $\cos \beta_q = \cos \beta_p = -|\overline{pq}| / |\pi(p, q)|$ ist. In letzterem Fall kann ein Eckpunkt-Randpunkt-Paar gewählt werden.

Die eigentliche Aussage des Lemmas wird in [EBKLL01] bewiesen. Mit seiner Anwendung erhalten wir nun in jedem Fall ein Eckpunkt-Randpunkt-Paar mit maximalem Umweg. Damit ist endlich der Beweis des Satzes 5.1 von S. 16 abgeschlossen. \square

5.5 Existenz eines in P^C gegenseitig sichtbaren Eckpunkt-Randpunkt-Paares mit maximalem Umweg

Satz 5.1 sichert die Existenz eines Eckpunkt-Randpunkt-Paares mit maximalem Umweg. Nach Lemma 3.1 können wir aber immer ein in P^C gegenseitig sichtbares Randpunktpaar

mit mindestens gleich großem Umweg finden. Dies legt die Vermutung nahe, dass auch ein in P^C gegenseitig sichtbares Eckpunkt-Randpunkt-Paar mit maximalem Umweg existiert. Tatsächlich ist folgende Aussage eine leichte Folgerung:

Korollar 5.1 *In jedem nicht konvexen Polygon gibt es ein in P^C gegenseitig sichtbares Eckpunkt-Randpunkt-Paar mit maximalem Umweg.*

Beweis. Nach Satz 5.1 gibt es ein Eckpunkt-Randpunkt-Paar mit maximalem Umweg. Da P nicht konvex ist, ist (p, q) nicht gegenseitig sichtbar. Wäre nun (p, q) auch in P^C nicht gegenseitig sichtbar, dann gäbe es nach Lemma 3.1 ein Randpunktpaar (p', q') mit echt größerem Umweg, was wegen der Maximalität von $u_P(p, q)$ nicht möglich ist. Also ist (p, q) in P^C gegenseitig sichtbar und das Korollar ist bewiesen. \square

6 Maximale Anzahl von Umwegmaxima

In [EBKLL01] wird in Lemma 5 gezeigt, dass es auf polygonalen Ketten höchstens $O(n)$ viele Punktpaare mit dem gleichen maximalen Umwegwert geben kann. Dies folgt leicht mit der Euler'schen Formel für planare Graphen aus der Tatsache, dass sich die Verbindungsstrecken von Punktpaaren maximalen Umwegs nicht schneiden können.

Im Polygonfall ergibt sich dieselbe Eigenschaft. Sie ist nur, weil die Verläufe der kürzesten Wege im Polygon nicht mehr auf eine vorgegebene Kette eingeschränkt sind, wesentlich schwieriger herzuleiten.

6.1 Beschreibung der Ausgangssituation

Im folgenden werden wir vereinfacht davon sprechen, dass (Umweg-) Maxima sich nicht schneiden. Exakt formuliert ist die Behauptung gemeint, dass sich die direkten Verbindungen umwegmaximaler Punktpaare nicht echt schneiden können.

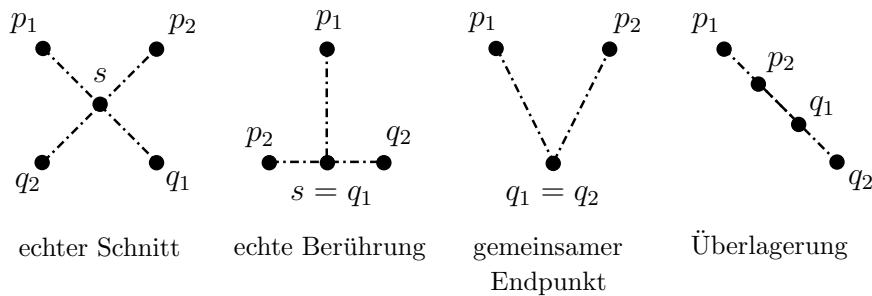


Abbildung 15: verschiedene Schnittarten zweier Liniensegmente

Ein *echter Schnitt* zweier Liniensegmente $\overline{p_1q_1}$ und $\overline{p_2q_2}$ liegt dabei vor, wenn $\overline{p_1q_1} \cap \overline{p_2q_2} = \{s\} \not\subseteq \{p_1, q_1, p_2, q_2\}$ ist. D.h. es gibt genau einen Schnittpunkt, und dieser Schnittpunkt ist nicht einer der vier Endpunkte. Ein solcher echter Schnitt ist mitsamt anderer möglicher Schnittbildungen in Abbildung 15 dargestellt.

Eine *echte Berührung* liegt vor, wenn $\overline{p_1q_1} \cap \overline{p_2q_2} \subset \{p_1, q_1, p_2, q_2\}$ gilt, und p_1, q_1, p_2 und q_2 paarweise verschieden sind. Außerdem können zwei Liniensegmente einen gemeinsamen Endpunkt haben, oder es kann eine *Überlagerung* entstehen. Letzteres ist der Fall, wenn $|\overline{p_1q_1} \cap \overline{p_2q_2}| > 1$ ist.

Wir werden zeigen, dass sich in nicht-konvexen Polygonen Umwegmaxima nicht echt schneiden können. Dies gilt selbstverständlich nur für nicht-konvexe Polygone. In konvexen Polygonen haben schließlich alle Punktpaare maximalen Umweg, und natürlich können sich auch die direkten Verbindungen solcher Punktpaare echt schneiden.

Auch in nicht-konvexen Polygonen ist es durchaus möglich, dass umwegmaximale Punktpaare einen Punkt gemeinsam haben, wie in Abbildung 16 zu sehen ist.

Die restlichen Schnittfälle aus Abbildung 15 können bei Umwegmaxima nicht-konvexer Polygone nicht auftauchen. Denn sowohl im Berührungsfall, als auch bei einer Überlagerung, liegt auf einer der direkten Verbindungen $\overline{p_1q_1}$ oder $\overline{p_2q_2}$ ein weiterer Polygonpunkt. Dann ist das betreffende Punktpaar nicht in P^C gegenseitig sichtbar. Also kann es nach Lemma 3.1 in einem nicht-konvexen Polygon keinen maximalen Umweg haben.

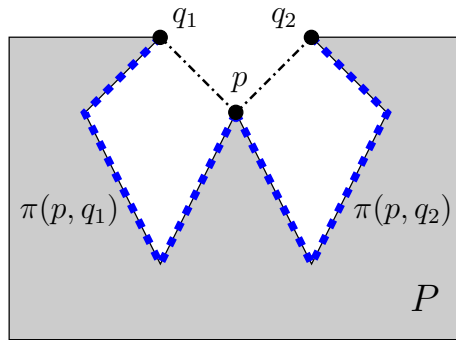


Abbildung 16: Umwegmaxima mit einem gemeinsamen Punkt

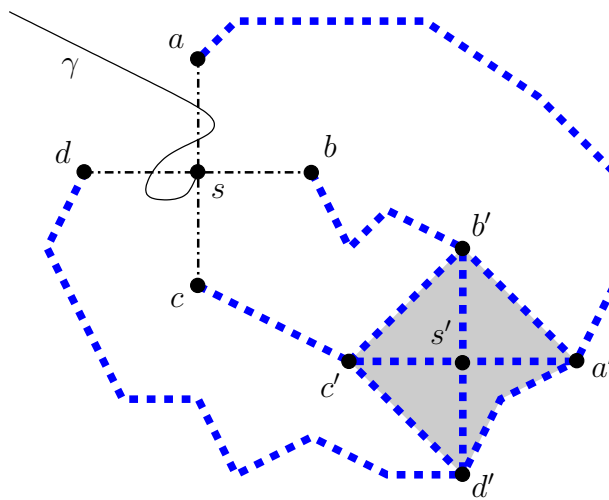


Abbildung 17: sich schneidende Umwegmaxima

Wenden wir uns nun der Situation zu, in der sich in einem nicht-konvexen Polygon zwei Umwegmaxima echt schneiden. Wir wollen diese Annahme zum Widerspruch führen.

Es würde sich eine Situation wie in Abbildung 17 ergeben. Die beiden Umwegmaxima (a, c) und (b, d) schneiden sich in einem Punkt s . Dabei seien o.B.d.A die Maximumpunkte a, b, c und d im Uhrzeigersinn um s angeordnet.

Man beachte, dass im jetzt zu untersuchenden Fall des echten Schnittes $s \notin \{a, b, c, d\}$ ist. Der Schnittpunkt s liegt dann in P^C , da sonst wieder die Maximumpunkte im Widerspruch zu Lemma 3.1 nicht in P^C gegenseitig sichtbar wären. Da P^C zusammenhängend ist (ansonsten hätte P ein Loch und wäre nicht einfach), muss es einen Weg $\gamma \subset P^C$ von s ins Unendliche geben. Dieser Weg γ schneidet genau eine der Strecken \overline{ab} , \overline{bc} , \overline{cd} oder \overline{ad} in einer ungeraden Anzahl. Sei dies o.B.d.A die Strecke \overline{ad} .

Betrachte nun die vier kürzesten Wege $\pi(a, b)$, $\pi(b, c)$, $\pi(c, d)$ und $\pi(a, d)$. Zwei Nachbarwege aus dieser Menge, d.h. Wege, die einen gemeinsamen Endpunkt haben (wie z.B. $\pi(a, b)$ und $\pi(a, d)$), laufen von ihrem gemeinsamen Endpunkt aus evtl. erst ein Teilstück gemeinsam, trennen sich dann, und treffen sich nicht wieder. Wir bezeichnen den Punkt, an dem sich zwei Nachbarwege trennen mit dem Namen des gemeinsamen Endpunktes und einem angehängten Apostroph (z.B. a'). Würden sich die Wege nach der Trennung in einem

weiteren Punkt (z.B. a'') wiedertreffen, dann wären die beiden Teilstücke, die die beiden Punkte (hier a' mit a'') verbinden, als Teile von kürzesten Wegen auch selbst kürzeste Wege. Es gäbe also zwei verschiedene kürzeste Wege zwischen diesen Punkten im Widerspruch zur Eindeutigkeit von kürzesten Wegen (siehe z.B. Proposition 1 in [Mit00]).

Zwei gegenüberliegende Wege aus der betrachteten Menge, d.h. entweder $\pi(a, b)$ und $\pi(b, d)$ oder $\pi(a, d)$ und $\pi(b, c)$, können sich weder schneiden noch berühren, wie wir im folgenden Abschnitt zeigen werden. Also sind die Punkte a', b', c' und d' paarweise verschieden, und die Teilwege $\pi(a', b')$, $\pi(b', c')$, $\pi(c', d')$ und $\pi(a', d')$ bilden ein einfaches Polygon V' . Wir nennen es als Gegenstück zu $abcd$ das *duale Viereck*, und bezeichnen die begrenzenden Teilwege auch als *Seiten des dualen Vierecks*.

Da die geschlossene Randkette dieses Polygons ganz in P liegt, muss auch ganz V' in P liegen. Sonst ergäbe sich wieder ein Widerspruch dazu, dass P^C zusammenhängend ist. Um $V' \subseteq P$ zu verdeutlichen, ist die Fläche von V' in Abbildung 17 grau eingefärbt.

Die Viereckseiten sind nach außen hin konvex. Denn kürzeste Wege im Polygon sind bekanntermaßen (siehe z.B. [GHL⁺87]) polygonale Ketten, deren Knoten Eckpunkte des Polygons sind. Also können die Viereckseiten nur an Polygonecken abknicken. Dort können sie aber nur so abknicken, dass der kleinere Winkel des Knickes den P^C -Teil des Eckpunktes mit einschließt, denn sonst könnte man den Weg verkürzen.

Betrachte nun den Weg $\pi(a, c)$. Nach der schon bekannten Argumentationsweise ist klar, dass er zunächst ein Teilstück auf dem Weg $\pi(a, b)$ (bzw. $\pi(a, d)$) verlaufen kann, ihn nach der Trennung aber nicht wieder trifft. Weil $\pi(a, c)$ außerdem γ , \overline{ac} und \overline{bd} nicht schneiden kann, da sie in P^C liegen, ist wegen der Anordnung klar, dass $\pi(a, c)$ von a aus über $\pi(a, a')$ läuft. Die analoge Betrachtung von c aus zeigt, dass $\pi(a, c)$ über $\pi(c, c')$ läuft. Da auch alle Teile kürzester Wege, kürzeste Wege sind, muss also gelten: $\pi(a, c) = \pi(a, a') \oplus \pi(a', c') \oplus \pi(c', c)$.

Analog beweist man, dass $\pi(b, d) = \pi(b, b') \oplus \pi(b', d') \oplus \pi(d', d)$ ist. Wegen der Lage im dualen Viereck müssen sich daher auch $\pi(a, c)$ und $\pi(b, d)$ in einem Punkt des dualen Vierecks, den wir s' nennen, schneiden.

6.2 Keine Berührung gegenüberliegender Wege

In diesem Abschnitt werden wir beweisen, dass sich in der betrachteten Situation die Wege $\pi(a, b)$ und $\pi(c, d)$ (bzw. $\pi(a, d)$ und $\pi(b, c)$) weder schneiden noch berühren. Der Beweis ist eine Verallgemeinerung des Gesamtbeweises aus [EBKLL01].

Wir beschränken uns o.B.d.A. auf die Betrachtung von $\pi(a, b)$ und $\pi(c, d)$. Und wir verkürzen die Formulierungen dadurch, dass wir nur noch von Berührungen sprechen. Die beiden Wege berühren sich genau dann nicht, wenn $\pi(a, b) \cap \pi(c, d) = \emptyset$ ist.

Wir führen einen Widerspruchsbeweis und nehmen deshalb an, dass sich $\pi(a, b)$ und $\pi(c, d)$ in einem Berührungspunkt \tilde{s} treffen.

Wegen der Dreiecksungleichung, die sogar streng wird, weil s weder auf \overline{ab} noch auf \overline{cd} liegt, gilt:

$$(36) \quad |\overline{ad}| + |\overline{bc}| < |\overline{as}| + |\overline{sd}| + |\overline{bs}| + |\overline{sc}| = |\overline{ac}| + |\overline{bd}|$$

Andererseits folgt wegen der Dreiecksungleichung für $|\pi(\cdot, \cdot)|$:

$$(37) \quad \begin{aligned} |\pi(a, c)| + |\pi(b, d)| &\leq |\pi(a, \tilde{s})| + |\pi(\tilde{s}, c)| + |\pi(b, \tilde{s})| + |\pi(\tilde{s}, d)| \\ &= |\pi(a, d)| + |\pi(b, c)| \end{aligned}$$

Analog zur Argumentation bei der Begründung von (6) auf Seite 11 folgt dann:

$$\begin{aligned} \min(u_P(a, c), u_P(b, d)) &\leq \frac{|\pi(a, c)| + |\pi(b, d)|}{|\overline{ac}| + |\overline{bd}|} \\ &\stackrel{(36), (37)}{<} \frac{|\pi(a, d)| + |\pi(b, c)|}{|\overline{ad}| + |\overline{bc}|} \\ &\leq \max(u_P(a, d), u_P(b, c)) \end{aligned}$$

Diese Ungleichung widerspricht der Maximalität von $u_P(a, c)$ und $u_P(b, d)$. Also ist bewiesen, dass sich $\pi(a, b)$ und $\pi(c, d)$ nicht berühren. Der Beweis, dass sich $\pi(a, d)$ und $\pi(b, c)$ nicht berühren, verläuft vollkommen analog.

6.3 Der Stifte-Ansatz

Ähnliche Argumentationen, die auf Dreiecksungleichungen basieren, scheitern leider an der Lösung des verbliebenen Falles. Zur Vereinfachung der Situation sei deshalb zunächst angenommen, dass die Viereckseiten $\pi(a', b')$, $\pi(b', c')$, $\pi(c', d')$ und $\pi(a', d')$ geradlinig verlaufen. Außerdem sollen im ersten Schritt die Entfernungen der „Zubringerwege“ $|\pi(a, a')|$, $|\pi(b, b')|$, $|\pi(c, c')|$ und $|\pi(d, d')|$ vernachlässigt werden, was allerdings eher eine Erschwerung darstellt (vgl. Abschnitt 6.5). Zur Erzeugung eines Widerspruches wollen wir dann prinzipiell zeigen:

$$(38) \quad u_P(a, c) = u_P(b, d) < \max(u_P(a, b), u_P(b, c), u_P(c, d), u_P(a, d))$$

Im Kontext der einleitend beschriebenen Vereinfachungen bedeutet diese Ungleichung:

$$\frac{|\overline{a'c'}|}{|\overline{ac}|} = \frac{|\overline{b'd'}|}{|\overline{bd}|} < \max\left(\frac{|\overline{a'b'}|}{|\overline{ab}|}, \frac{|\overline{b'c'}|}{|\overline{bc}|}, \frac{|\overline{c'd'}|}{|\overline{cd}|}, \frac{|\overline{a'd'}|}{|\overline{ad}|}\right)$$

Da eine Skalierung der geometrischen Situation der Punkte a' , b' , c' und d' offenbar alle an der Aussage beteiligten Brüche um den gleichen Faktor verändert, können wir o.B.d.A. annehmen, dass $|\overline{a'c'}| = |\overline{ac}|$ und $|\overline{b'd'}| = |\overline{bd}|$ gelten. Es reicht also aus, das folgende Lemma 6.1 zu zeigen. Der zweite Teil der Aussage (39) ist dabei notwendig, weil man unter den gegebenen Voraussetzungen nicht ausschließen kann, dass die geometrische Situation der Punkte a' , b' , c' und d' genau derjenigen von a , b , c und d entspricht.

Lemma 6.1 *Seien acht Punkte $a, b, c, d, a', b', c', d' \in \mathbb{R}^2$ in der Ebene gegeben, so dass sich die Strecken \overline{ac} und \overline{bd} echt schneiden, d.h. $|\overline{ac} \cap \overline{bd}| = 1 \wedge \overline{ac} \cap \overline{bd} \not\subset \{a, b, c, d\}$, und sich die Strecken $\overline{a'c'}$ und $\overline{b'd'}$ schneiden oder berühren, d.h. $(\overline{a'c'} \cap \overline{b'd'}) \neq \emptyset$. Gilt außerdem $|\overline{ac}| = |\overline{a'c'}|$ und $|\overline{bd}| = |\overline{b'd'}|$, dann folgt:*

$$(39) \quad \begin{aligned} &\max\left(\frac{|\overline{a'b'}|}{|\overline{ab}|}, \frac{|\overline{b'c'}|}{|\overline{bc}|}, \frac{|\overline{c'd'}|}{|\overline{cd}|}, \frac{|\overline{a'd'}|}{|\overline{ad}|}\right) > 1 \\ &\text{oder } \frac{|\overline{a'b'}|}{|\overline{ab}|} = \frac{|\overline{b'c'}|}{|\overline{bc}|} = \frac{|\overline{c'd'}|}{|\overline{cd}|} = \frac{|\overline{a'd'}|}{|\overline{ad}|} = 1 \end{aligned}$$

Beweis. Da nun also $|\overline{a'c'}| = |\overline{ac}|$ und $|\overline{b'd'}| = |\overline{bd}|$ gelten, kann man sich die beiden durch die Punkte a, b, c, d bzw. a', b', c', d' gebildeten geometrischen Situationen als zwei unterschiedliche Lagen zweier Stifte vorstellen. Zur Verdeutlichung dient Abbildung 18.

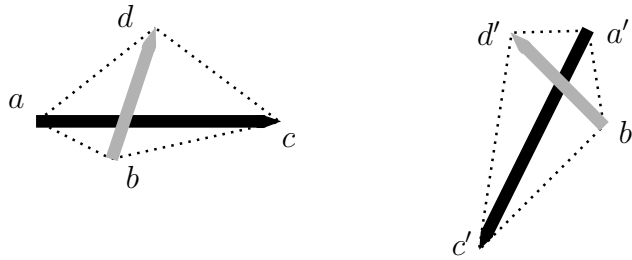


Abbildung 18: unterschiedliche Lagen zweier sich überkreuzender Stifte

In der ersten Situation liegt das stumpfe Ende des schwarzen Stiftes bei a und die Spitze bei c . Der graue Stift verbindet genauso b mit d . In der zweiten Situation bestimmen auf die gleiche Art die mit einem Apostroph bezeichneten Positionen die Lage der Stifte. Wichtig ist, dass die Länge der Stifte in beiden Situationen gleich ist, und dass sich die Stifte in beiden Situationen überschneiden müssen. Denn wir gehen ja gerade davon aus, dass sich \overline{ac} und \overline{bd} in s und $\pi(a', c')$ und $\pi(b', d')$ in s' schneiden. Bei \overline{ac} und \overline{bd} wissen wir sogar, dass der Schnittpunkt keiner der Endpunkte sein darf (siehe Beschreibung der Ausgangssituation auf S. 28 und Annahme in Lemma 6.1).

Die zu zeigende Aussage (39) wäre also bewiesen, wenn man zeigen könnte, dass zwei sich kreuzende Stifte sich nicht auf irgendeine andere Weise so überkreuz legen lassen, dass alle ihre Endpunktabstände ($|ab|$, $|bc|$, $|cd|$ und $|ad|$) höchstens gleich groß bleiben und mindestens einer echt kleiner wird.

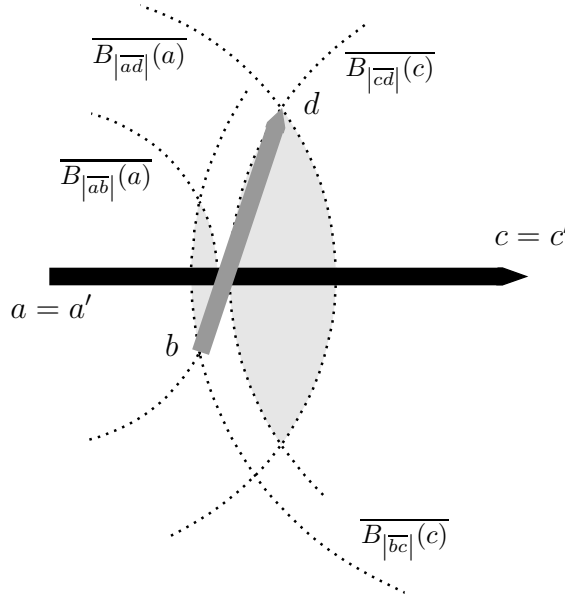


Abbildung 19: für b' und d' erlaubte, linsenförmige Gebiete

Dies ist tatsächlich nicht möglich. Sei dazu o.B.d.A. $|\overline{ac}| \geq |\overline{bd}|$, $a = a' = (0, 0)$ und $c = c' = (|\overline{ac}|, 0)$. Sei weiterhin o.B.d.A. b der Punkt des Punktpaares (b, d) mit der kleineren X-Koordinate ($b_x \leq d_x$). Wegen $|\overline{ac}| \geq |\overline{bd}|$, und weil die Stifte sich überkreuzen, muss b oder d mit seinen x-Koordinaten zwischen a und c liegen. Das heißt $0 < b_x < |\overline{ac}|$ oder $0 < d_x < |\overline{ac}|$.

Betrachte dann die kreisförmigen Gebiete $\overline{B_{|\overline{ab}|}(a)} = \{x \in \mathbb{R}^2 \mid |\overline{ax}| \leq |\overline{ab}|\}$, $\overline{B_{|\overline{ad}|}(a)}$, $\overline{B_{|\overline{bc}|}(c)}$ und $\overline{B_{|\overline{cd}|}(c)}$. Liegt auch der zweite Punkt des Punktpaares (b, d) mit seinen x -Koordinaten zwischen a und c , d.h. gilt $0 < b_x \leq d_x < |\overline{ac}|$, so ergibt sich eine Situation wie in Abbildung 19, sonst sieht sie aus wie in Abbildung 20.

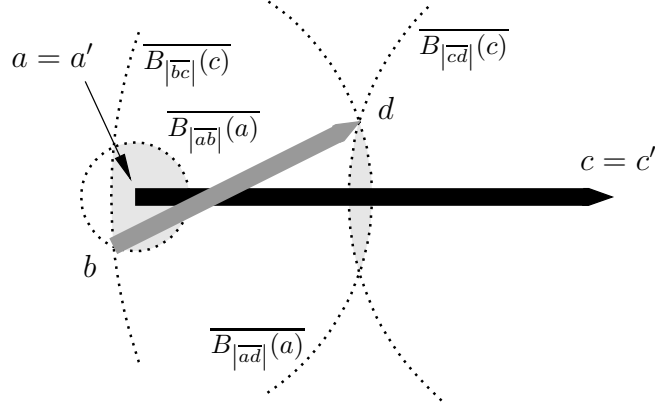


Abbildung 20: linsenförmige Gebiete bei $b_x \notin [a_x, c_x]$

Damit alle beteiligten Endpunktabstände nicht größer werden, müssen offensichtlich b' in $\overline{B_{|\overline{ab}|}(a)} \cap \overline{B_{|\overline{bc}|}(c)}$ und d' in $\overline{B_{|\overline{ad}|}(a)} \cap \overline{B_{|\overline{cd}|}(c)}$ liegen. Diese linsenförmigen Gebiete sind in den Abbildungen 19 und 20 hellgrau eingefärbt. In diesen Gebieten hat aber nur (b, d) und das an der X -Achse dazu gespiegelte Punktpaar den geforderten Abstand $|\overline{bd}|$. Formal ausgedrückt:

$$(40) \quad \forall (b', d') \in \left(\overline{B_{|\overline{ab}|}(a)} \cap \overline{B_{|\overline{bc}|}(c)} \right) \times \left(\overline{B_{|\overline{ad}|}(a)} \cap \overline{B_{|\overline{cd}|}(c)} \right) : \\ |\overline{b'd'}| < |\overline{bd}| \quad \vee \quad (b', d') = (b, d) \quad \vee \quad (b', d') = ((b_x, -b_y), (d_x, -d_y))$$

Dies scheint anschaulich klar, wird aber auch im Anschluss noch formal bewiesen. Damit sind dann auch die Ungleichung (39) und Lemma 6.1 bewiesen.

Wenden wir uns dem formalen Beweis der Aussage (40) zu. Wir wollen dazu den maximalen Punkt Abstand der Ränder der linsenförmigen Gebiete untersuchen. Es ist klar, dass man für Punktpaare (b', d') aus dem Innern der linsenförmigen Gebiete durch Verlängerung der Strecke $\overline{b'd'}$ auf die Ränder immer zwei Punkte mit echt größerem Abstand findet. Also gilt:

$$(41) \quad \forall (b', d') \in \left(B_{|\overline{ab}|}(a) \cap B_{|\overline{bc}|}(c) \right) \times \left(B_{|\overline{ad}|}(a) \cap B_{|\overline{cd}|}(c) \right) : |\overline{b'd'}| < \\ \max \left\{ |\overline{b''d''}| \mid b'' \in \partial \left(B_{|\overline{ab}|}(a) \cap B_{|\overline{bc}|}(c) \right), d'' \in \partial \left(B_{|\overline{ad}|}(a) \cap B_{|\overline{cd}|}(c) \right) \right\}$$

Das Randmaximum kann aber nur von Punktpaaren angenommen werden, dessen Punkte in den Knicken der linsenförmigen Gebiete liegen, d.h. in Punkten, in denen sich die an dem Gebiet beteiligten Kreise schneiden. Ansonsten würde nämlich folgende Argumentation greifen.

Sei (b'', d'') ein Randpunktpaar der linsenförmigen Gebiete mit maximalem Abstand, bei dem d'' nicht in einem Knick liegt. Dann liegt d'' auf einem Rand, der in einer Umgebung um d'' einem Kreis K um a oder c entspricht. Sei o.B.d.A. a der Kreismittelpunkt. Betrachte außerdem den Kreis $K_{|\overline{b''d''}|}(b'')$ um b'' mit dem Radius $|\overline{b''d''}|$. Die beiden Kreise schneiden

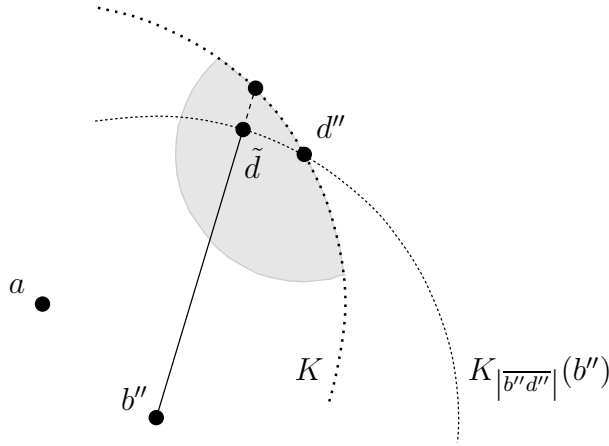


Abbildung 21: b'' und d'' haben keinen maximalen Abstand

oder berühren sich in d'' . Liegt $K_{|b''d''|}(b'')$ auf einer Seite von d'' innerhalb des Kreises K (siehe Abb. 21), dann könnte man dort auf $K_{|b''d''|}(b'')$ einen Vergleichspunkt \tilde{d} setzen, der innerhalb des zugehörigen linsenförmigen Gebietes liegt und den gleichen Abstand zu b'' hat wie d'' . Weiterhin könnte man die Strecke $b''\tilde{d}$ über \tilde{d} hinaus bis zum Rand des Gebietes, d.h. bis zu dem Kreis K verlängern. Der dortige Punkt würde also auf dem Rand des linsenförmigen Gebietes liegen und hätte einen größeren Abstand zu b'' als d'' . Das ist ein Widerspruch zur geforderten Maximalität.

Also muss $K_{|b''d''|}(b'')$ zu beiden Seiten von b'' außerhalb von K liegen. Dies ist nach dem folgenden Lemma 6.2 nur möglich, wenn die Kreismittelpunkte, d.h. b'' und a , mit dem Berührungspunkt d'' auf einer Linie liegen. Dann liegen aber entweder b'' und d'' auf der X-Achse, oder ihre Y-Koordinaten haben das gleiche nicht-verschwindende Vorzeichen, d.h. $b''_y \cdot d''_y > 0$. In letzterem Fall lässt sich durch Spiegelung eines der Punkte an der X-Achse ein Punktpaar mit größerem Abstand finden, das ebenfalls auf den Rändern der linsenförmigen Gebiete liegt, im Widerspruch zur Maximalität von (b'', d'') .

Auch ein Punktpaar (b'', d'') mit verschwindenden Y-Koordinaten kann keinen maximalen Abstand haben. Dies erkennt man, wenn man den grauen Stift \overline{bd} um seinen Schnittpunkt s mit \overline{ac} dreht, bis er auf der X-Achse liegt. In jedem Fall haben alle Punktpaare der linsenförmigen Gebiete mit verschwindenden Y-Koordinaten einen kleineren Abstand voneinander als $|\overline{bd}|$.

Insgesamt ist also mit Hilfe des folgenden Lemmas 6.2 gezeigt, dass ein Punktpaar (b'', d'') auf den Rändern der linsenförmigen Gebiete nur dann einen maximalen Abstand haben kann, wenn beide Punkte in einem der beiden Knicke ihres Gebietes liegen. Dann ist aber ihr Abstand höchstens gleich demjenigen von b und d . Und dies ist nur der Fall, wenn $(b'', d'') \in \{(b, d), ((b_x, -b_y), (d_x, -d_y))\}$ ist. Zusammen mit Aussage (41) folgt also (40).

Liegt nun (b', d') in $(\overline{B_{|ab|}}(a) \cap \overline{B_{|bc|}}(c)) \times (\overline{B_{|ad|}}(a) \cap \overline{B_{|cd|}}(c))$, dann muss wegen der Forderung $|\overline{b'd'}| = |\overline{bd}|$ gelten, dass $(b', d') = (b, d)$ oder $(b', d') = ((b_x, -b_y), (d_x, -d_y))$ ist. Dann folgt unmittelbar:

$$\frac{|\overline{a'b'}|}{|\overline{ab}|} = \frac{|\overline{b'c'}|}{|\overline{bc}|} = \frac{|\overline{c'd'}|}{|\overline{cd}|} = \frac{|\overline{a'd'}|}{|\overline{ad}|} = 1$$

Liegen andererseits b' oder d' nicht im zugehörigen linsenförmigen Gebiet, dann folgt:

$$\max \left(\frac{|\overline{a'b'}|}{|\overline{ab}|}, \frac{|\overline{b'c'}|}{|\overline{bc}|}, \frac{|\overline{c'd'}|}{|\overline{cd}|}, \frac{|\overline{a'd'}|}{|\overline{ad}|} \right) > 1$$

Damit ist (39) bewiesen und somit der Beweis von Lemma 6.1 abgeschlossen. \square

Lemma 6.2 *Berühren oder schneiden sich zwei Kreise $K_1 = K_{r_1}(m_1)$ und $K_2 = K_{r_2}(m_2)$ in einem Punkt b , dann liegt K_2 genau dann (zu beiden Seiten von b) innerhalb von K_1 , wenn die Kreismittelpunkte m_1 und m_2 mit dem Berührungspunkt b auf einer Linie liegen und $r_2 \leq r_1$ ist.*

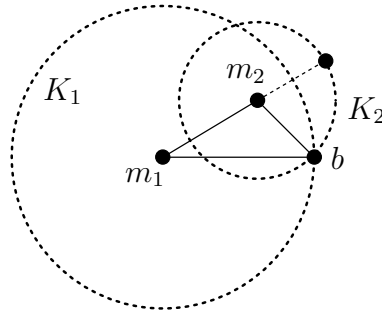


Abbildung 22: sich schneidende Kreise

Beweis. Die Rückrichtung ist klar. Auch ist klar, dass $r_2 \leq r_1$ sein muss, damit K_2 innerhalb von K_1 liegen kann. Liegen dann die drei Punkte nicht auf einer Linie, werden die zu ihnen gehörigen Dreiecksungleichungen strikt, d.h. unter anderem gilt:

$$|\overline{m_1 b}| < |\overline{m_1 m_2}| + |\overline{m_2 b}|$$

Also gilt, weil b der Berührungspunkt ist:

$$r_1 < |\overline{m_1 m_2}| + r_2$$

Wenn man aber die Strecke $\overline{m_1 m_2}$ ausdehnt bis man den Kreis K_2 erreicht, so hat man einen Punkt dieses zweiten Kreises gefunden, der einen Abstand von $|\overline{m_1 m_2}| + r_2$ zu m_1 hat (vgl. Abb. 22). Folglich liegt dieser Punkt außerhalb von K_1 . Da sich zwei Kreise, wenn sie nicht identisch sind, in höchstens zwei Punkten schneiden oder berühren, muss also K_2 schon in jeder Umgebung des einen Schnittpunktes b außerhalb von K_1 liegen. \square

6.4 Geradebiegen des dualen Vierecks

Im folgenden Abschnitt werden wir eine der Einschränkungen aufheben, die zum Stifte-Ansatz geführt hat. Wir setzen nicht mehr die Geradlinigkeit der Seiten des dualen Vierecks voraus. Vielmehr dürfen diese jetzt beliebige nach außen hin konvexe polygonale Ketten sein.

Sind die kürzesten Wege $\pi_{V'}(a', c')$ und $\pi_{V'}(b', d')$ im dualen Viereck V' geradlinig, d.h. (a', c') und (b', d') sind in V' gegenseitig sichtbar, dann kann man die Aussage von Lemma

6.1 problemlos auf den neuen Fall übertragen, da im Vergleich zum durch $\overline{a'b'}$, $\overline{b'c'}$, $\overline{c'd'}$ und $\overline{a'd'}$ gebildeten Polygon die Viereckseiten $\pi(a', b')$, $\pi(b', c')$, $\pi(c', d')$ und $\pi(a', d')$ mindestens gleich lang sind wie die entsprechenden direkten Verbindungen.

Komplizierter wird die Argumentation, wenn $\pi_{V'}(a', c')$ oder $\pi_{V'}(b', d')$ nicht geradlinig verlaufen. Dann kann man aber das duale Viereck V' wie in Abbildung 23 gerade biegen. Es gilt das folgende Lemma.

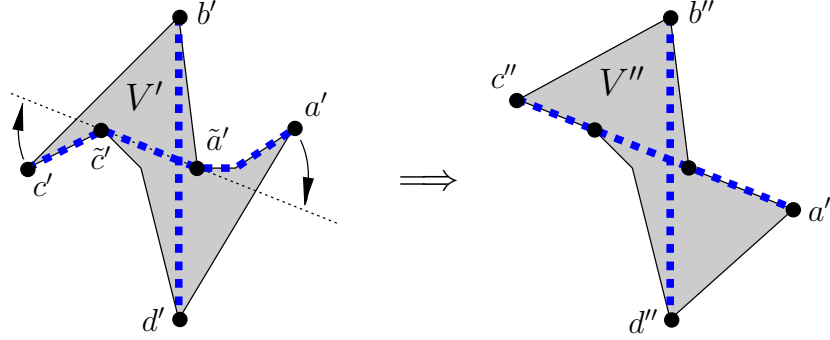


Abbildung 23: Geradebiegen des dualen Vierecks V'

Lemma 6.3 Seien vier Punkte $a', b', c', d' \in \mathbb{R}^2$ in der Ebene gegeben. Seien außerdem vier polygonale Ketten $\pi(a', b')$, $\pi(b', c')$, $\pi(c', d')$ und $\pi(a', d')$ gegeben, die jeweils die im Namen auftauchenden Endpunkte verbinden und ein einfaches Polygon V' bilden. Seien diese Ketten bezüglich V' nach außen hin konvex. Seien $\pi_{V'}(a', c')$ und $\pi_{V'}(b', d')$ kürzeste Wege in V' .

Dann können wir ein einfaches Polygon V'' mit den vier Randpunkten $a'', b'', c'', d'' \in \mathbb{R}^2$ und den vier nach außen konvexen Randketten $\pi(a'', b'')$, $\pi(b'', c'')$, $\pi(c'', d'')$ und $\pi(a'', d'')$ bilden, in dem (a'', c'') und (b'', d'') gegenseitig sichtbar sind, und das folgende Aussagen erfüllt:

$$\begin{aligned} |\pi(a'', b'')| &\leq |\pi(a', b')|, & |\pi(b'', c'')| &\leq |\pi(b', c')|, & |\pi(c'', d'')| &\leq |\pi(c', d')|, \\ |\pi(a'', d'')| &\leq |\pi(a', d')|, & |\overline{a''c''}| &= |\pi_{V'}(a', c')|, & |\overline{b''d''}| &= |\pi_{V'}(b', d')| \end{aligned}$$

Beweis. Wir werden zeigen, dass man in jedem Fall durch Geradebiegen der sich kreuzenden Wege $\pi(a', c')$ und $\pi(b', d')$ ein solches Polygon V'' erhalten kann.

Wie biegt man nun die sich kreuzenden Wege in V' gerade? Wir möchten dies anhand von $\pi(a', c')$ erläutern. Dieser Weg verläuft entweder erst ein Stück mit $\pi(a', b')$ oder erst ein Stück mit $\pi(a', d')$, oder er trennt sich schon in a' von diesen beiden Wegen. Anschließend kann er wegen der Eindeutigkeit der kürzesten Wege im Polygon keinen der beiden Wege wiedertreffen. Nennen wir den Punkt, in dem sich $\pi(a', c')$ von dem letzten der beiden anderen Wege trennt, \tilde{a}' . Hinter \tilde{a}' verläuft $\pi(a', c')$ eventuell ein Stück weit frei und geradlinig im Innern von V' . Dann schließt sich, wegen der analogen Betrachtung von c' aus, ein Wegstück entweder auf $\pi(b', c')$ oder $\pi(d', c')$ bis hin zu c' an. Nenne den Zusammentreffpunkt \tilde{c}' . Dabei kann $\tilde{c}' = c'$ sein. Es sind aber auch sogar Fälle mit $\tilde{a}' = \tilde{c}' = b'$ denkbar.

Ist nun $a' = \tilde{a}'$, dann ist an dem Punkt a' nichts gerade zu biegen. Deshalb sei im folgenden $a' \neq \tilde{a}'$. Und zwar betrachten wir nur den Fall, in dem $\pi(a', c')$ bis zu \tilde{a}' auf $\pi(a', b')$ verläuft. Alle anderen Fälle können analog behandelt werden.

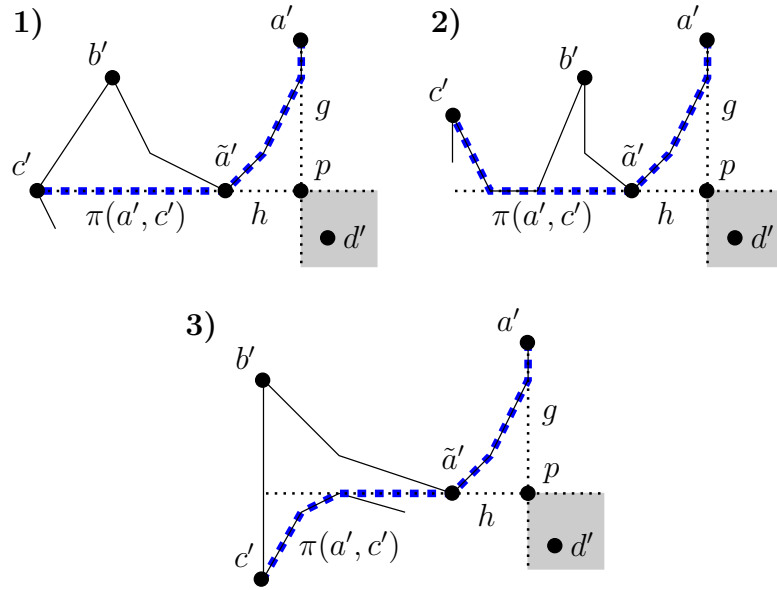


Abbildung 24: möglicher Bereich für d' und \tilde{d}'

Das weitere Vorgehen verdeutlicht Abbildung 24. Verlängere die erste Kante von $\pi(a', c')$ zu einer Geraden g . Und verlängere die erste Kante von $\pi(a', c')$ hinter \tilde{a}' zu einer Geraden h . Ist $g = h$, dann ist im Punkt a' nichts gerade zu biegen. Anderenfalls können g und h nicht parallel sein, da der Weg $\pi(a', c')$ in jedem Fall bis zur ersten Kante hinter a' in eine Richtung hin konvex ist. Also schneiden sich g und h in einem Punkt. Wir nennen ihn p .

Die Geraden g und h teilen die Ebene in vier Teile. Der Punkt d' muss nun in dem Teil liegen, der $\pi(a', \tilde{a}')$ gegenüber liegt. Auf welcher Seite von g der Punkt d' liegt, ist deshalb festgelegt, weil sonst a' und d' nicht mehr durch eine nach außen konvexe Kette verbunden sein könnten. Auf welcher Seite von h der Punkt d' liegt, ist genauso festgelegt, weil sonst c' und d' nicht mehr durch eine nach außen konvexe Kette verbunden sein könnten (siehe dazu die drei Fälle in Abb. 24).

Betrachte nun den Weg $\pi(d', b')$. Er kann von d' aus ein Teilstück lang auf $\pi(d', a')$ verlaufen. Sei \tilde{d}' der Punkt, an dem sich $\pi(d', b')$ und $\pi(d', a')$ trennen. Es ist gut möglich, dass $\tilde{d}' = d'$ ist. Anderenfalls müssen wir aber beim Geradebiegen von $\pi(a', \tilde{a}')$ darauf achten, dass $\pi(d', \tilde{d}')$ nicht verändert wird. Denn der Weg $\pi(b', d')$ soll bei dem lokalen Geradebiegen von $\pi(a', \tilde{a}')$ unangetastet bleiben.

Weil \tilde{d}' auf der nach außen konvexen Kette $\pi(a', d')$ liegt, muss der Punkt auf der gleichen Seite der Geraden g liegen wie d' . Da ansonsten $\pi(b', d')$ verkürzbar wäre, liegt \tilde{d}' auch auf der gleichen Seite der Geraden h wie d' . D.h. auch \tilde{d}' liegt in dem grau eingefärbten Bereich von Abb. 24.

Das Geradebiegen am Punkt a' erfolgt nun, indem man den Punkt a' durch einen Punkt a'' auf der Geraden h ersetzt, der auf der c' gegenüberliegenden Seite von g liegt, und dessen Abstand zu \tilde{a}' die Gleichung $|\overline{a''\tilde{a}'}| = |\pi(a', \tilde{a}')|$ erfüllt. Die Kette $\pi(a', d')$ wird bei diesem Übergang durch die Kette $\pi(a'', d') := \overline{a''\tilde{d}'} \oplus \pi(\tilde{d}', d')$ ersetzt.

Beim Übergang von a' zu a'' ändert sich von den betrachteten 6 Weglängen nur $|\pi(a', d')|$. Dass sich $|\pi(b', d')|$ nicht ändert wird später noch kurz begründet. Bei den anderen Weglängen ist es offensichtlich, weil diese Seiten des dualen Vierecks unverändert bleiben.

Lemma 6.4 Sei K eine polygonale Kette, die die Punkte a und b verbindet. Sei weiterhin ein Punkt p gegeben, so dass K , \overline{ap} und \overline{bp} ein einfaches Polygon bilden, bezüglich dessen K nach außen hin konvex ist. Dann gilt:

$$|\overline{ab}| \leq |K| \leq |\overline{ap}| + |\overline{bp}|$$

Beweis. Die erste Ungleichung folgt leicht durch iterierte Anwendung der Dreiecksungleichung. Zum Beweis der zweiten Ungleichung benutzen wir eine Induktion über die *Kantenzahl* $\#K$ von K . Ist $\#K = 1$, dann ist die zu zeigende Ungleichung die Dreiecksungleichung.

Sei also nun $\#K = n + 1$, und sei die Aussage für Ketten mit Kantenzahl $l \leq n$ schon bewiesen. Sei dann v_1 der erste Knoten von K hinter a . Sei also $\overline{av_1}$ die erste Kante von K . Verlängere diese bis sie auf \overline{bp} trifft. Nenne den dortigen Punkt q (siehe auch Abb. 27). Nun gilt nach Dreiecksungleichung:

$$(43) \quad |\overline{av_1}| + |\overline{v_1q}| = |\overline{aq}| \leq |\overline{ap}| + |\overline{pq}|$$

Entfernt man andererseits die erste Kante aus der Kette, so erhält man eine Kette $K' := K \setminus \{\overline{av_1}\}$ mit n Kanten, die mit dem Punkt q selbst die Voraussetzungen des zu beweisenden Lemmas erfüllt. Da die Aussage für Ketten dieser Kantenzahl als Induktionsvoraussetzung schon bewiesen ist, folgt:

$$(44) \quad |K'| \leq |\overline{v_1q}| + |\overline{bq}|$$

Insgesamt ergibt sich also:

$$\begin{aligned} |K| &= |\overline{av_1}| + |K'| \\ &\stackrel{(44)}{\leq} |\overline{av_1}| + |\overline{v_1q}| + |\overline{bq}| \\ &\stackrel{(43)}{\leq} |\overline{ap}| + |\overline{pq}| + |\overline{bq}| \\ &= |\overline{ap}| + |\overline{bp}| \end{aligned}$$

Damit ist der Induktionsbeweis abgeschlossen. \square

6.5 Schlussfolgerung

Zum endgültigen Beweis dafür, dass sich Maxima nicht schneiden können, fehlt nur noch das Einbeziehen der Zubringerwege $\pi(a, a')$, $\pi(b, b')$, $\pi(c, c')$ und $\pi(d, d')$ in unsere Betrachtungen. Es ergibt sich:

Satz 6.1 In einem einfachen, nicht-konvexen Polygon P können sich zwei Umwegmaxima nicht echt schneiden, nicht echt berühren und nicht überlagern. Es gilt also:

$$\forall a, b, c, d \in P : u_P(a, c) = u_P(b, d) = u_P^{\max} \Rightarrow \overline{ac} \cap \overline{bd} = \emptyset \vee |\{a, b, c, d\}| < 4$$

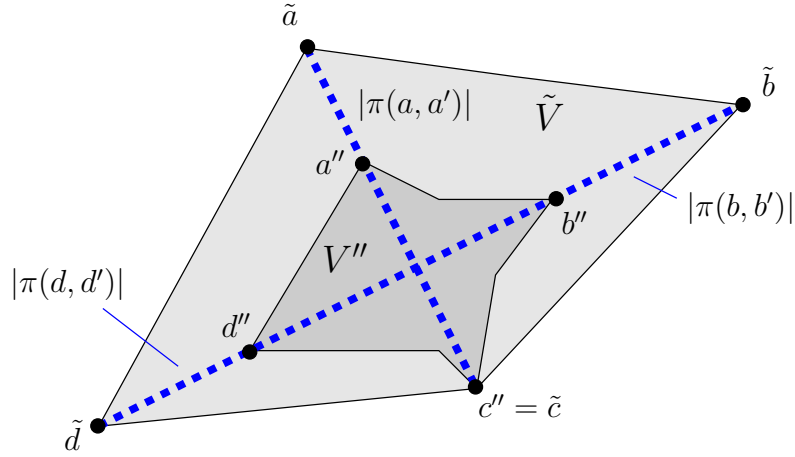


Abbildung 28: Anhängen der Längen der Zubringerwege

Beweis. Dass sich in einem nicht-konvexen Polygon Umwegmaxima nicht echt berühren oder überlagern können, hatten wir schon in Abschnitt 6.1 gezeigt. Würden sich nun zwei Punkt-paare $(a, c), (b, d) \in P \times P$ maximalen Umwegs echt schneiden, dann würde die ebenfalls schon in Abschnitt 6.1 analysierte Situation eintreten. Wir wenden dann Lemma 6.3 an, um das duale Viereck V' zu einem Polygon V'' gerade zu biegen.

Anschließend setzen wir, wie in Abbildung 28 zu sehen ist, die Weglängen der Zubringerwege als gerade Fortsetzung an die Strecken $\overline{a''c''}$ und $\overline{b''d''}$ an die jeweils vom Namen her passenden Endpunkte an, und nennen die so gefundenen neuen Endpunkte $\tilde{a}, \tilde{b}, \tilde{c}$ und \tilde{d} . Nach Skalierung ist auf das so gebildete Polygon \tilde{V} Lemma 6.1 anwendbar. Es folgt also (nach Einbeziehung des Skalierungsfaktors in die erhaltene Ungleichung):

$$\max \left(\frac{|\tilde{a}\tilde{b}|}{|\tilde{a}\tilde{b}|}, \frac{|\tilde{b}\tilde{c}|}{|\tilde{b}\tilde{c}|}, \frac{|\tilde{c}\tilde{d}|}{|\tilde{c}\tilde{d}|}, \frac{|\tilde{a}\tilde{d}|}{|\tilde{a}\tilde{d}|} \right) > \frac{|\tilde{a}\tilde{c}|}{|\tilde{a}\tilde{c}|} = \frac{|\tilde{b}\tilde{d}|}{|\tilde{b}\tilde{d}|}$$

$$\text{oder } \frac{|\tilde{a}\tilde{b}|}{|\tilde{a}\tilde{b}|} = \frac{|\tilde{b}\tilde{c}|}{|\tilde{b}\tilde{c}|} = \frac{|\tilde{c}\tilde{d}|}{|\tilde{c}\tilde{d}|} = \frac{|\tilde{a}\tilde{d}|}{|\tilde{a}\tilde{d}|} = \frac{|\tilde{a}\tilde{c}|}{|\tilde{a}\tilde{c}|} = \frac{|\tilde{b}\tilde{d}|}{|\tilde{b}\tilde{d}|}$$

Aufgrund der Konstruktion von \tilde{V} und Lemma 6.3 gilt aber:

$$(45) \quad \begin{aligned} |\tilde{a}\tilde{c}| &= |\pi(a, c)|, \quad |\tilde{b}\tilde{d}| = |\pi(b, d)|, \\ |\tilde{a}\tilde{b}| &\leq |\tilde{a}a'' \oplus \pi(a'', b'') \oplus b''\tilde{b}| \leq |\pi(a, b)|, \\ |\tilde{b}\tilde{c}| &\leq |\tilde{b}b'' \oplus \pi(b'', c'') \oplus c''\tilde{c}| \leq |\pi(b, c)|, \\ |\tilde{c}\tilde{d}| &\leq |\tilde{c}c'' \oplus \pi(c'', d'') \oplus d''\tilde{d}| \leq |\pi(c, d)|, \\ |\tilde{a}\tilde{d}| &\leq |\tilde{a}a'' \oplus \pi(a'', d'') \oplus d''\tilde{d}| \leq |\pi(a, d)| \end{aligned}$$

Also folgt:

$$\begin{aligned} \max(u_P(a, b), u_P(b, c), u_P(c, d), u_P(a, d)) &> u_P(a, c) = u_P(b, d) \\ \text{oder } u_P(a, b) &= u_P(b, c) = u_P(c, d) = u_P(a, d) = u_P(a, c) = u_P(b, d) \end{aligned}$$

Der zweite Fall ist ausgeschlossen, weil mindestens einer der Zubringerwege nicht verschwindet und deshalb eine Ungleichung in (45) strikt wird. Also ergibt sich ein Widerspruch zur Maximalität von $u_P(a, c)$ und $u_P(b, d)$, und Satz 6.1 ist bewiesen. \square

Wie in [EBKLL01] ergibt sich dann eine lineare obere Schranke für die Anzahl von Punktpaaren maximalen Umwegs.

Korollar 6.1 *In einem einfachen, nicht-konvexen Polygon P mit n Ecken gibt es höchstens $O(n)$ viele Punktpaare mit maximalem Umweg.*

Beweis. Wir bilden einen geometrischen Graphen G (siehe Abb. 29), in dem wir zunächst die Eckpunkte von P als Knoten deuten und die Randkanten als Kanten des Graphen nehmen. Zusätzlich fügen wir für jedes Punktpaar (p, q) maximalen Umwegs die Punkte p und q als Knoten und die Kante \overline{pq} hinzu. Wegen Lemma 3.1 sind die Punkte p und q Randpunkte des Polygons. Also bleibt der gebildete Graph zusammenhängend. Wegen Satz 6.1, und weil die Maximumpaare nach Lemma 3.1 in P^C gegenseitig sichtbar sind, also keine Polygonrandkante kreuzen, ist er außerdem kreuzungsfrei.

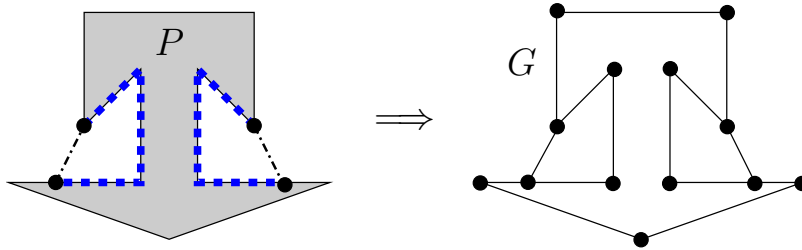


Abbildung 29: Konstruktion des kreuzungsfreien Graphen G

Nun kann man noch je alle Kantenendpunkte in diesem Graphen, die auf der gleichen Randkante des Polygons aber nicht auf einem Eckpunkt liegen, so zusammenschieben, dass schließlich die beteiligten Kanten in einem gemeinsamen Punkt enden, der z.B. in der Mitte der betrachteten Randkante liegt. Die beteiligten Kanten kann man dabei so verbiegen, dass der Graph weiterhin kreuzungsfrei bleibt.

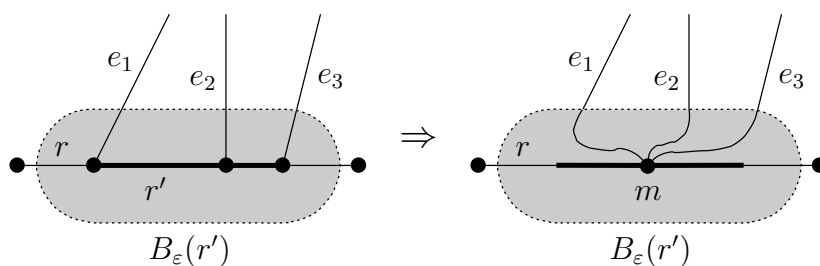


Abbildung 30: Verbiegen aller auf r endenden Kanten zum Mittelpunkt m

Ein mögliches Vorgehen ist in Abbildung 30 dargestellt. Zuerst wählt man ein $\varepsilon > 0$ so klein, dass in dem ε -Schlauch $B_\varepsilon(r')$ um den betroffenen Teil r' der Polygonrandkante r keine weitere Kante des Graphen bis auf die beteiligten Kanten e_1, \dots, e_k , die einen Endpunkt auf r haben, auftauchen. Anschließend verbiegt man die Kanten e_1, \dots, e_k in $B_\varepsilon(r')$ so, dass sie alle auf dem Randkantenmittelpunkt m enden und sich weiterhin untereinander nicht schneiden. Da die Änderungen am Graphen nur im Schlauch $B_\varepsilon(r')$ erfolgen, bleibt der Graph kreuzungsfrei.

Lasse in dem derart gebildeten Graphen alle Knoten mit Grad 2 wegfallen. Die Knotenzahl des entstandenen Graphen ist durch $v \leq 2n$ beschränkt, denn als mögliche Knoten kommen nur die n Polygoneckpunkte und die n Randkantenmittelpunkte in Frage. Außerdem folgt aus der Eulerschen Formel, dass in so einem Graphen die Kantenzahl durch $e < 3v$ beschränkt ist (siehe z.B. [Kle97]). Insgesamt ergibt sich also $e < 3v \leq 6n$. \square

7 Ein exakter Algorithmus

Wir wissen jetzt, dass man sich bei der Suche nach einem Punktpaar mit maximalem Umweg auf Eckpunkt-Randpunkt-Paare, die in P^C gegenseitig sichtbar sind, konzentrieren kann. Durch diese Erkenntnis bietet sich direkt eine naive Herangehensweise an. Man muss einfach nur alle möglichen Eckpunkt-Kanten-Paare des Polygons untersuchen.

Um diese Idee anwenden zu können, benötigen wir ein Verfahren, zu einem gegebenen Eckpunkt p und einer gegebenen Kante e einen Randpunkt $q \in e$ zu finden, welcher auf der Kante e den Umweg zu p maximiert. Es soll also gelten: $u_P(p, q) = \max_{q' \in e} u_P(p, q')$.

Im nächsten Unterabschnitt verallgemeinern wir dazu zunächst einmal die schon als (8) in Abschnitt 4(S. 13) und in Lemma 5.2(S. 26) zitierte Gleichung aus [EBKLL01]. Es ergibt sich ein notwendiges Kriterium für ein einseitiges lokales Maximum des Umweges auf einer Randkante. Der Zusatz „einseitig“ soll dabei darauf hinweisen, dass es sich um ein Maximum gegenüber benachbarten Randpunkten q' von q handelt. Der zweite Punkt p bleibt hingegen in der Betrachtung fest.

7.1 Notwendiges Kriterium für ein einseitiges Maximum

Wir suchen also jetzt ein notwendiges Kriterium dafür, dass ein echter Randpunkt $q \in e$ zusammen mit dem Punkt $p \in P$ ein einseitiges lokales Umwegmaximum realisiert. D.h. für eine Umgebung $U \subseteq e$ von q soll gelten: $u_P(p, q) = \max_{q' \in U} u_P(p, q')$.

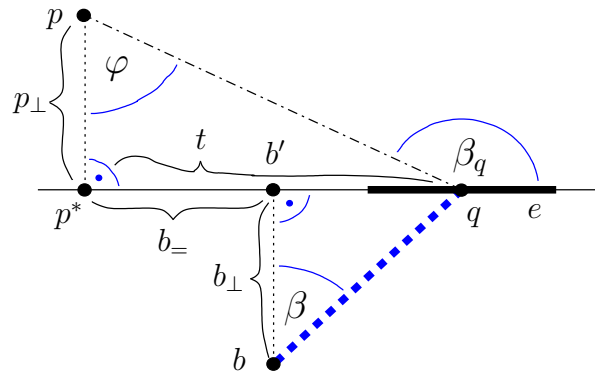


Abbildung 31: einseitige Maximumsuche durch Verschieben von q

Analog zu den Überlegungen in Abschnitt 5.3 (S. 25f) können wir uns auf eine Situation beschränken, in der $\pi(p, q')$ in ganz U von q' aus über den gleichen Eckpunkt b verläuft.

Man könnte das gesuchte notwendige Kriterium auch mit den Parametrisierungen aus den Abschnitten 5.1 und 5.2 erhalten. Allerdings ist, da wir diesmal nur einen der beiden Punkte, nämlich den Randpunkt, verschieben müssen, die Unterteilung in parallelen und nicht-parallelen Fall unnötig. Die im folgenden angewendete direkte Herleitung ist trotz der Einführung einer weiteren Parametrisierung weniger aufwendig als ein Rückgriff auf die genannten Abschnitte.

Wieder parametrisieren wir über t . Wobei diesmal bei $t = 0$ der Punkt $q(t)$ genau auf p^* liegen soll, der senkrechten Projektion von p auf e . Es gelte $|t| := |\overline{p^*q}|$. Das Vorzeichen von t sei so festgelegt, dass q für positive t -Werte auf derjenigen Seite von p^* liegt, auf der

auch b' liegt, die senkrechte Projektion von b auf e . Falls b' direkt auf p^* liegt, ist die Wahl der Orientierung wegen der Symmetrie der Situation sowieso bedeutungslos (vgl. (46) für $b_{=} = 0$).

Die weiteren Bezeichnungen sind in Abb. 31 zu sehen. Wir definieren $b_{=} := \overline{p^*b'}$, also immer positiv. Außerdem seien die Winkel festgelegt durch $\sin \beta = (t - b_{=})/|\overline{bq}|$, $\cos \beta = b_{\perp}/|\overline{bq}|$, $\sin \varphi = t/|\overline{pq}|$, $\cos \varphi = p_{\perp}/|\overline{pq}|$ und $\beta_q = \varphi + \pi/2$.

Betrachte nun Punkte $q'(t')$ auf e in der Nähe von q . Zur Unterscheidung von dem durch q festgelegten t nennen wir die Parametrisierungsvariable wieder t' . Der Umweg zwischen p und $q'(t')$ wird eine Funktion, die nur von t' abhängt:

$$f(t') := u_P(p, q'(t')) = \frac{\sqrt{(t' - b_{=})^2 + b_{\perp}^2} + |\pi(p, b)|}{\sqrt{t'^2 + p_{\perp}^2}}$$

Die erste Ableitung berechnet sich zu:

$$f'(t') = \frac{\frac{t' - b_{=}}{\sqrt{(t' - b_{=})^2 + b_{\perp}^2}} \sqrt{t'^2 + p_{\perp}^2} - \left(\sqrt{(t' - b_{=})^2 + b_{\perp}^2} + |\pi(p, b)| \right) \frac{t'}{\sqrt{t'^2 + p_{\perp}^2}}}{t'^2 + p_{\perp}^2}$$

Damit in t (also bei q) ein einseitiges lokales Maximum vorliegen kann, muss $f'(t) = 0$ sein.

$$(46) \quad \frac{t - b_{=}}{\sqrt{(t - b_{=})^2 + b_{\perp}^2}} \sqrt{t^2 + p_{\perp}^2} - \left(\sqrt{(t - b_{=})^2 + b_{\perp}^2} + |\pi(p, b)| \right) \frac{t}{\sqrt{t^2 + p_{\perp}^2}} \stackrel{!}{=} 0$$

Unter Benutzung der Winkel in den rechtwinkligen Dreiecken ergibt sich daraus:

$$(47) \quad \begin{aligned} 0 &= \sin \beta |\overline{pq}| - |\pi(p, q)| \sin \varphi \\ \Rightarrow \frac{\sin \varphi}{\sin \beta} &= \frac{|\overline{pq}|}{|\pi(p, q)|} \vee \sin \beta = \sin \varphi = 0 \\ \Rightarrow \frac{\cos \beta_q}{\sin \beta} &= -\frac{|\overline{pq}|}{|\pi(p, q)|} \vee \sin \beta = \cos \beta_q = 0 \end{aligned}$$

Dies ist die angekündigte Verallgemeinerung von Gleichung (8). Auf das gleiche Ergebnis kommt man übrigens mit Gleichung (13) aus dem nicht-parallelen Fall in Abschnitt 5.1, wenn man sich klar macht, dass $\sin \varphi = (t - s \cos \gamma)/|\overline{pq}|$ ist.

An der Form von Gleichung (47) ist unschön, dass sich die benutzten Winkel β_q und β nicht entsprechen. In diesem Sinne wäre entweder die Verwendung der beiden Winkel φ und β schöner gewesen, oder die Benutzung von β_q zusammen mit dem Winkel zwischen \overline{pq} und e . Wir haben uns trotzdem für die vorliegende Version entschieden, da die Verwendung von β konsistent zu den Bezeichnungen in Abschnitt 5 ist, und die Verwendung von β_q den Zusammenhang zu der aus [EBKLL01] stammenden Gleichung (8) auf Seite 13 (siehe auch Lemma 5.2 auf S. 26) verdeutlicht. Denn wenn $\pi(p, q)$ auf der Kante e verläuft, dann ist $\beta = \pi/2$, also $\sin \beta = 1$. Dann stimmen die Gleichungen (8) und (47) überein. Gleichung (8) ist folglich ein Spezialfall von (47).

Ein anderes Manko der kompakten Formel (47) ist, dass sie nicht direkt dazu benutzt werden kann, die tatsächliche Position eines Maximums zu berechnen. Denn alle vier auftretenden Terme hängen von der Position t ab. Um eine Formel für t zu bestimmen, gehen wir noch einmal von (46) aus. Da q ein echter Randpunkt sein soll, ist der Eckpunkt b ungleich q . Außerdem verlangen wir, damit $p_{\perp}^2 > 0$ ist, dass p nicht auf der *Verlängerung* der Kante e liegt, die entsteht, wenn man e zu einer Geraden ausdehnt. Bei der Suche nach dem globalen

Maximum wird uns diese Einschränkung nicht stören (siehe Schritt 2 in Algorithmus 1 auf S.49). Nun können wir (46) mit den Nennern multiplizieren und erhalten:

$$\begin{aligned}
& (t - b_{=})(t^2 + p_{\perp}^2) - ((t - b_{=})^2 + b_{\perp}^2) t - |\pi(p, b)| t \sqrt{(t - b_{=})^2 + b_{\perp}^2} = 0 \\
\stackrel{2}{\Rightarrow} & (t - b_{=})^2 (t^2 + p_{\perp}^2)^2 + ((t - b_{=})^2 + b_{\perp}^2)^2 t^2 \\
& - 2(t - b_{=})(t^2 + p_{\perp}^2) ((t - b_{=})^2 + b_{\perp}^2) t = |\pi(p, b)|^2 ((t - b_{=})^2 + b_{\perp}^2) t^2 \\
\Rightarrow & (t^2 - 2b_{=}t + b_{=}^2)(t^4 + 2t^2 p_{\perp}^2 + p_{\perp}^4) + (t^2 - 2b_{=}t + b_{=}^2 + b_{\perp}^2)^2 t^2 \\
& - 2(t^4 - b_{=}t^3 + p_{\perp}^2 t^2 - b_{=}p_{\perp}^2 t) (t^2 - 2b_{=}t + b_{=}^2 + b_{\perp}^2) \\
& = |\pi(p, b)|^2 (t^2 - 2b_{=}t + b_{=}^2 + b_{\perp}^2) t^2 \\
\Rightarrow & t^6 + 2p_{\perp}^2 t^4 + p_{\perp}^4 t^2 - 2b_{=}t^5 - 4b_{=}p_{\perp}^2 t^3 - 2b_{=}p_{\perp}^4 t + b_{=}^2 t^4 + 2b_{=}^2 p_{\perp}^2 t^2 \\
& + b_{=}^2 p_{\perp}^4 \\
& + t^6 - 4b_{=}t^5 + 2b_{=}^2 t^4 + 2b_{\perp}^2 t^4 + 4b_{=}^2 t^4 \\
& - 4b_{=}^3 t^3 - 4b_{=}b_{\perp}^2 t^3 + b_{=}^4 t^2 + 2b_{=}^2 b_{\perp}^2 t^2 + b_{\perp}^4 t^2 \\
& - 2t^6 + 2b_{=}t^5 - 2p_{\perp}^2 t^4 + 2b_{=}p_{\perp}^2 t^3 + 4b_{=}t^5 - 4b_{=}^2 t^4 + 4p_{\perp}^2 b_{=}t^3 - 4b_{=}^2 p_{\perp}^2 t^2 \\
& - 2b_{=}^2 t^4 + 2b_{=}^3 t^3 - 2b_{=}^2 p_{\perp}^2 t^2 + 2b_{=}^3 p_{\perp}^2 t - 2b_{\perp}^2 t^4 + 2b_{\perp}^2 b_{=}t^3 - 2b_{\perp}^2 p_{\perp}^2 t^2 \\
& + 2b_{\perp}^2 b_{=}p_{\perp}^2 t \\
& = |\pi(p, b)|^2 t^4 - 2|\pi(p, b)|^2 b_{=}t^3 + |\pi(p, b)|^2 b_{=}^2 t^2 + |\pi(p, b)|^2 b_{\perp}^2 t^2 \\
\Rightarrow & t^4 \left[b_{=}^2 - |\pi(p, b)|^2 \right] + t^3 \left[2b_{=}(p_{\perp}^2 - b_{=}^2 - b_{\perp}^2 + |\pi(p, b)|^2) \right] \\
& + t^2 \left[p_{\perp}^2 (p_{\perp}^2 - 2b_{=}^2) + (b_{=}^2 + b_{\perp}^2)(b_{=}^2 + b_{\perp}^2 - |\pi(p, b)|^2 - 2p_{\perp}^2) \right] \\
& + t \left[2b_{=}p_{\perp}^2 (-p_{\perp}^2 + b_{=}^2 + b_{\perp}^2) \right] + [b_{=}^2 p_{\perp}^4] = 0
\end{aligned}$$

Damit also zu einem Eckpunkt p an einem echten Randpunkt q ein einseitiges lokales Umwegmaximum auftreten kann, muss der Positionsparameter t eine Nullstelle eines bestimmten Polynoms vierten Grades sein. Wir fassen die erhaltenen Ergebnisse in einem Lemma zusammen:

Lemma 7.1 *Sei q ein echter Randpunkt auf der Kante e . Sei $p \in P$ ein beliebiger Polygonpunkt, der nicht auf der Verlängerung der Kante e liegt. Der kürzeste Weg $\pi(p, q')$ hänge in einer Umgebung $U \subseteq e$ von q von $q' \in U$ aus immer an dem Eckpunkt b . Wenn nun in (p, q) ein einseitiges lokales Maximum von $u_P(\cdot, \cdot)$ liegt, wenn also $u_P(p, q) = \max_{q' \in U} u_P(p, q')$ erfüllt ist, muss mit den Bezeichnungen von Abb. 31 gelten:*

$$(48) \quad \left(\beta = 0 \text{ und } \beta_q = \frac{\pi}{2} \right) \text{ oder } \frac{\cos \beta_q}{\sin \beta} = -\frac{|\overline{pq}|}{|\pi(p, q)|}$$

Äquivalent dazu ist, dass der Positionsparameter t eine Nullstelle des folgenden Polynoms vierten Grades annimmt:

$$\begin{aligned}
(49) \quad & a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 = 0 \\
\text{mit} \quad & a_4 := b_{=}^2 - |\pi(p, b)|^2 \\
& a_3 := 2b_{=}(p_{\perp}^2 - b_{=}^2 - b_{\perp}^2 + |\pi(p, b)|^2) \\
& a_2 := p_{\perp}^2 (p_{\perp}^2 - 2b_{=}^2) + (b_{=}^2 + b_{\perp}^2)(b_{=}^2 + b_{\perp}^2 - |\pi(p, b)|^2 - 2p_{\perp}^2) \\
& a_1 := 2b_{=}p_{\perp}^2 (-p_{\perp}^2 + b_{=}^2 + b_{\perp}^2) \\
& a_0 := b_{=}^2 p_{\perp}^4
\end{aligned}$$

Für die Anwendung wäre eine geschlossene Formel für die Nullstellen des Polynoms (49) hilfreich. Eine solche existiert natürlich. Jedoch füllt ein mit Maple berechnetes Ergebnis schon ganze 6 DIN-A4-Seiten. Eine weitere Vereinfachung war unter Maple nicht zu erreichen. Daher werden wir in den folgenden Algorithmen zuerst die aktuellen Werte für die Variablen $\pi(p, b)$, p_{\perp} , $b_{=}$ und b_{\perp} einsetzen, und erst anschließend Standardverfahren zur Berechnung der Nullstellen verwenden.

Allerdings gibt es einen schon wohl bekannten Spezialfall, in dem sich die Maximumsuche wesentlich vereinfacht.

7.2 Spezialfall des notwendigen Kriteriums

Ist der zu p und e gehörige Trichter entartet, dann hängen für alle $q \in e$ die kürzesten Wege $\pi(p, q)$ an einem bestimmten Kantenendpunkt b , der gleichzeitig der Trichterfußpunkt ist.

In diesem Spezialfall ist die Suche nach einem einseitigen lokalen Maximum von $u_P(p, \cdot)$ auf e nach mehreren Ansätzen möglich. Zum Beispiel kann man, da $\pi(p, \cdot)$ immer über die Kante e verläuft, mit den Ergebnissen von [EBKLL01] nach einem entsprechenden Maximum des Umweges auf der polygonalen Kette $C := \pi(p, b) \oplus e$ suchen. Diese Idee hatten wir schon in Lemma 5.2 und auch in (8) benutzt. Die erhaltene Formel $\cos \beta_q = -|pq| / |\pi(p, q)|$, die man genauso gut aus (48) in Lemma 7.1 gewinnen kann, muss dann nur noch mit dem Parameter t umformuliert werden.

Wir setzen hier hingegen einfach im Polynom (49) $b_{\perp} = 0$ und erhalten:

Korollar 7.1 *Seien die Voraussetzungen von Lemma 7.1 erfüllt. Sei außerdem der zu p und e gehörige Trichter entartet, d.h. es gelte $b_{\perp} = 0$ (vgl. Abb. 31). Dann kann in dem echten Eckpunkt-Randpunktpaar (p, q) nur ein einseitiges lokales Maximum vorliegen, wenn der Positionsparameter t einen der beiden folgenden Werte annimmt:*

$$(50) \quad t \in \left\{ -\frac{p_{\perp}^2}{b_{=} + |\pi(p, b)|}, -\frac{p_{\perp}^2}{b_{=} - |\pi(p, b)|} \right\}$$

Beweis. Für $b_{\perp} = 0$ wird (49) zu:

$$\begin{aligned} & t^4 \left[b_{=}^2 - |\pi(p, b)|^2 \right] + t^3 \left[2b_{=} (p_{\perp}^2 - b_{=}^2 + |\pi(p, b)|^2) \right] \\ & + t^2 \left[p_{\perp}^2 (p_{\perp}^2 - 2b_{=}^2) + (b_{=}^2) (b_{=}^2 - |\pi(p, b)|^2 - 2p_{\perp}^2) \right] \\ & + t \left[2b_{=} p_{\perp}^2 (-p_{\perp}^2 + b_{=}^2) \right] + [b_{=}^2 p_{\perp}^4] = 0 \end{aligned}$$

Dieses Polynom lässt sich in Linearfaktoren zerlegen. Es ergibt sich:

$$((b_{=} - |\pi(p, b)|) t + p_{\perp}^2) ((b_{=} + |\pi(p, b)|) t + p_{\perp}^2) (t - b_{=})^2 = 0$$

Da für $t = b_{=}$ der Punkt q auf dem Polygoneckpunkt b liegen würde, also (p, q) kein echtes Eckpunkt-Randpunkt-Paar wäre, kommen für die Lösungen nur noch die beiden anderen Faktoren in Frage. \square

7.3 Maximumsuche in einem Trichter

Durch Lemma 7.1 bzw. Korollar 7.1 wissen wir, wie man eine konstant beschränkte Anzahl von Randpunktkandidaten q für ein lokales Umwegmaximum findet, wenn der zugehörige

Eckpunkt p , die Kante e auf der q liegt, und der Eckpunkt b , an dem $\pi(p, q)$ hängt, vorgegeben sind. In diesem Unterabschnitt werden wir die Erkenntnisse nutzen, um einen Algorithmus zu entwickeln, der das Umwegmaximum zwischen einem vorgegebenen Eckpunkt p und einer Randkante e ermittelt.

Am Anfang von Abschnitt 5 hatten wir schon Trichter als Teile von Sanduhren vorgestellt (siehe S. 16). Lee und Preparata zeigen in [LP84], dass zu einem Polygonpunkt p und einer Polygonkante e alle kürzesten Wege $\pi(p, q')$ für $q' \in e$ durch so einen Trichter laufen.

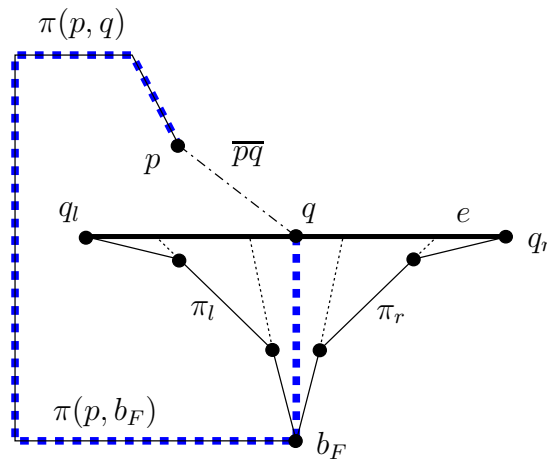


Abbildung 32: Ein kürzester Weg $\pi(p, q)$ im Trichter von p und e

Dargestellt ist die Situation beispielhaft in Abbildung 32. Die Kante e bildet dort die Trichteroberseite. Des weiteren wird der Trichter durch zwei nach außen konvexe Randketten π_l und π_r , die je einen Endpunkt (q_l bzw. q_r) von e mit dem gemeinsamen Trichterfußpunkt b_F verbinden, begrenzt. Dieser eigentliche Trichter ist über die Kette $\pi(p, b_F)$ mit dem Punkt p verbunden. In Anlehnung an die Funktion eines Trichters werden wir $\pi(p, b_F)$ *Schlauch* nennen. Mit dem Begriff Trichter sei nur der von den Randketten π_l und π_r und von der Trichteroberkante e begrenzte Bereich ohne Hinzunahme des Schlauches bezeichnet. Wie Guibas et al. in [GHL⁺87] werden wir das Gebiet dieses Trichters auch mit $\Phi(e)$ bezeichnen.

Eine wichtige Eigenschaft der Randketten ist, dass sie selbst kürzeste Wege im Polygon sind. Es gilt $\pi(p, q_l) = \pi(p, b_F) \oplus \pi_l$ und $\pi(p, q_r) = \pi(p, b_F) \oplus \pi_r$.

Weiterhin sind in Abbildung 32 gepunktete Trennlinien zwischen verschiedenen Trichterbereichen eingezeichnet. Jeder solche Bereich ist dadurch charakterisiert, dass die kürzesten Wege $\pi(p, q')$ von einem Punkt q' dieses Bereiches aus an einem bestimmten dem Bereich zugeordneten Trichtereckpunkt b hängen. So liegt der dargestellte Punkt q eindeutig im zu b_F gehörigen Bereich.

Hat man zu einem Punkt p und einer Kante e den zugehörigen Trichter gegeben, dann lässt sich leicht Lemma 7.1 anwenden, um das Umwegmaximum des Punkt-Kantenpaares zu berechnen. Dazu dient folgender Algorithmus:

Algorithmus 1

Input: ein Punkt $p \in P$, eine Randkante e , die Randketten π_l und π_r des zugehörigen Trichters, die Länge des zugehörigen Schlauches $|\pi(p, b_F)|$

Output: ein Punkt $q \in e$ mit maximalem Umweg, $u_P(p, q) = \max_{q' \in e} u_P(p, q')$

1. TESTE, OB p AUF e LIEGT. WENN JA, DANN GEBE p AUS. WENN NEIN, FAHRE FORT.
2. TESTE, OB p AUF DER VERLÄNGERUNG VON e LIEGT. WENN JA, GEBE VON DEN ENDPUNKTEN q_l UND q_r DER KANTE e DENJENIGEN AUS, DER NÄHER AN p LIEGT. WENN NEIN, FAHRE FORT.
3. DURCHLAUFE VON b_F AUS ERST π_l UND DANN π_r . BERECHNE DABEI FÜR JEDEN TRICHTERECKPUNKT b DIE LÄNGE $|\pi(p, b)| := |\pi(p, b_F)| + |\pi(b_F, b)|$.
4. DURCHLAUFE ALLE TRICHTERECKPUNKTE b INKLUSIVE b_F
 - a) BERECHNE ALLE REELLEN NULLSTELLEN DES MIT DEN WERTEN VON p UND b GEBILDETEN POLYNOMS (49) AUS LEMMA 7.1. ⁴ IST DAS POLYNOM IDENTISCH NULL, DANN BERÜCKSICHTIGE GAR KEINE NULLSTELLE.
 - b) TESTE FÜR JEDE NULLSTELLE t , OB $q(t)$ IM ZU b GEHÖRIGEN BEREICH LIEGT. FINDE EIN UMWEGMAXIMUM DER RICHTIG LIEGENDEN NULLSTELLENPUNKTE $q(t)$ UND DER BEIDEN ZU b GEHÖRIGEN BEREICHSGRENZEN.
5. FINDE EIN MAXIMUM DER MAXIMA AUS 4. UND DEN ZUGEHÖRIGEN RANDPUNKT q . GEBE q AUS.

Wir zeigen zuerst die Korrektheit dieses Algorithmus. Liegt p selbst auf der Randkante e , dann kann p ganz e sehen, also ist auf ganz e der Umweg $u_P(p, \cdot) \equiv 1$. Damit ist auch p selbst ein korrekter Rückgabewert.

Liegt p nicht auf e aber auf der Verlängerung von e , dann wird das Umwegmaximum auf jeden Fall bei dem Endpunkt von e angenommen, der näher an p liegt. Dies sieht man analog zum Beweis von Lemma 3.1 auf S.10ff. Die gesonderte Betrachtung dieses Falles ist notwendig, weil er in Lemma 7.1 (siehe S. 46) ausgeschlossen werden musste, um im Beweis das Auftreten eines verschwindenden Nenners zu verhindern.

Interessant wird der Algorithmus eigentlich erst, wenn die beiden Spezialfälle nicht eintreten. Dann liegt p nicht auf der Verlängerung von e und Schritt 3 wird ausgeführt. Die hier berechneten Distanzen $|\pi(p, q)|$ werden für die Berechnung der Umwegwerte und für die Anwendung des Polynoms (49) benötigt.

Liegt nun die Stelle q eines Punkt-Kanten-Maximums in einem Bereich, in dem $\pi(p, q')$ von q' aus an einem bestimmten Trichtereckpunkt b hängt, dann muss der zugehörige Positionsparameter t nach Lemma 7.1 eine Nullstelle dieses Polynoms (49) sein. Gehen wir zunächst davon aus, dass das Polynom nicht identisch null ist. q wird also im Schritt 4a) entdeckt, besteht dann den Bereichstest aus Schritt 4b) und fließt in die Maximumberechnung ein. Der Algorithmus gibt dann tatsächlich q aus oder einen anderen Randpunkt, dessen Umweg aber den gleichen Wert hat. Auf jeden Fall ist die Ausgabe des Algorithmus korrekt.

Liegt die Stelle q eines Punkt-Kanten-Maximums auf einer Bereichsgrenze, wozu auch die Kantenendpunkte q_l und q_r zählen, dann fließt der Wert in die Maximumberechnung in Schritt 4b) ein. Der Algorithmus gibt entweder q aus, oder einen Randpunkt mit gleichem Umweg. Wieder ist die Ausgabe korrekt.

⁴Verfahren zur exakten Nullstellenbestimmung von Polynomen vierten Grades in konstanter Zeit sind allgemein bekannt. Siehe z.B. [BSMM95] S.31 oder [Bos93] S.88-91 .

Betrachten wir nun den Ausnahmefall, in dem das Polynom (49) identisch null ist. Dann verschwinden alle Koeffizienten a_0 bis a_4 . Kommt der Algorithmus bis zu Schritt 4., können wir schon ausschließen, dass p auf der Verlängerung von e liegt. Also ist $p_{\perp} \neq 0$. Damit $a_0 = b_{\perp}^2 p_{\perp}^4$ trotzdem verschwindet, muss $b_{\perp} = 0$ sein. Dann muss aber, damit auch $a_4 = b_{\perp}^2 - |\pi(p, b)|^2$ verschwindet, $|\pi(p, b)| = 0$ sein. Es ist also $p = b$. Dann ist aber der Umweg im gesamten zu $b = p$ gehörigen Bereich 1. Deshalb reicht es bei weitem aus, nur die Randpunkte des zu b gehörigen Bereiches zu betrachten, um ein Punkt-Kanten-Maximum zu finden. Also arbeitet der Algorithmus auch bei Auftreten dieser Anomalie korrekt und wir haben die Korrektheit von Algorithmus 1 in allen Fällen gezeigt.

Wir sollten allerdings noch einmal genau beschreiben, welche Trichtereckpunkte im Schritt 4 durchlaufen werden. Im Algorithmus selbst haben wir klar gemacht, dass b_F immer dazugehört. Weiterhin müssen alle Eckpunkte der Trichterrandketten π_l und π_r untersucht werden mit Ausnahme der Endpunkte q_l und q_r . Es gibt aber noch eine Ausnahme von der Ausnahme. Ist nämlich der Trichter entartet, hängen also alle kürzesten Wege an dem Kantenendpunkt q_l bzw. q_r , dann muss (als einziges b) $b_F = q_l$ bzw. $b_F = q_r$ untersucht werden. Dies kann auch mit der einfacheren Formel von Korollar 7.1 geschehen.

Betrachten wir nun die Laufzeit des Algorithmus. Sei dazu k die Anzahl der Trichtereckpunkte oder Trichterkanten. Die Schritte 1. und 2. benötigen nur konstante Zeit. Schritt 3. braucht lineare Zeit $O(k)$. Die Schleife in Schritt 4. wird ebenfalls $O(k)$ mal durchlaufen. Das Verfahren zur Nullstellenbestimmung in Schritt 4a) benötigt konstante Zeit und liefert höchstens 4 Nullstellen. Der Bereichstest in Schritt 4b) benötigt für jede Nullstelle konstante Zeit, also auch insgesamt konstante Zeit. Dann wird noch das Maximum aus den Umwegen der höchstens 4 Nullstellen und der zwei Bereichsgrenzen gebildet. Das geht offenbar auch in konstanter Zeit. Da also die Schleife in 4. $O(k)$ mal durchlaufen wird und jeder Durchgang nur konstante Zeit beansprucht, liegt die Gesamtlaufzeit des Algorithmus in der Größenordnung $O(k)$. Wir haben bewiesen:

Lemma 7.2 *Sind zu einem Eckpunkt p und einer Randkante e eines Polygons P die Randketten π_l und π_r des zugehörigen Trichters, sowie die Länge $|\pi(p, b_F)|$ des zugehörigen Schlauches bekannt, dann kann in einer Laufzeit von $O(k)$, wobei k die Anzahl der Kanten im Trichter bezeichnet, ein Eckpunkt-Kanten-Maximum des Umweges berechnet werden. D.h. wir finden einen Punkt $q \in e$ mit $u_P(p, q) = \max_{q' \in e} u_P(p, q')$.*

Dabei ist wichtig, dass als Eingabeparameter $|\pi(p, b_F)|$ und nicht $\pi(p, b_F)$ selbst übergeben wird. Denn sonst müsste man noch die ganze Kette $\pi(p, b_F)$ durchlaufen und die Laufzeit des Algorithmus wäre zusätzlich abhängig von der Anzahl der Schlauchkanten.

7.4 Optimalität der Suche im Trichter

Beim Betrachten von Lemma 7.2 stellt sich die Frage, ob es wirklich nötig ist, zur Ermittlung eines Eckpunkt-Randkanten-Maximums jeden Eckpunkt des zugehörigen Trichters zu untersuchen. Es könnte ja sein, dass bedingt durch die geometrische Anordnung eines Trichters Maxima nur in bestimmten Bereichen auftauchen können, z.B. im zum Fußpunkt b_F gehörigen Bereich.

Um diese Vermutung zu widerlegen, wäre es schön, ein Gegenbeispiel zu konstruieren. Dazu ist am besten ein Trichter geeignet, der in jedem Bereich ein Umwegmaximum hat, wobei die Werte dieser Maxima alle gleich gross sind. Die Idee ist in Abb. 33 dargestellt. Man könnte dann das Eckpunkt-Randkanten-Maximum in jeden gewünschten Bereich verlegen,

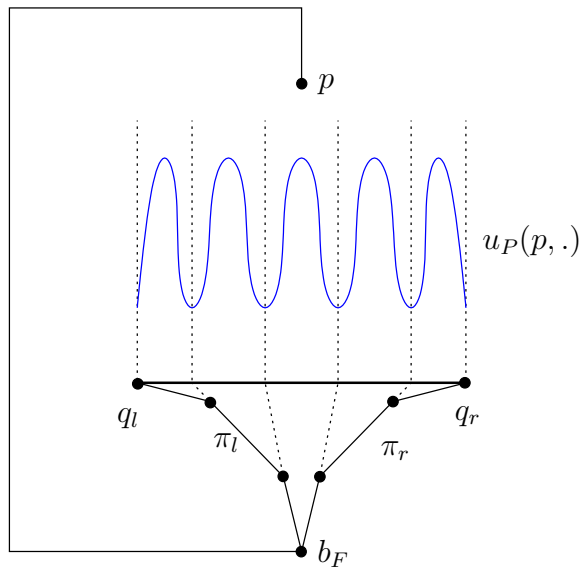


Abbildung 33: Idee zur Konstruktion eines Trichters mit k Maxima

indem man die Trichtereckpunkte nur geringfügig verschiebt. Dies wäre ein erstes Indiz dafür, dass bei der Bestimmung des Eckpunkt-Randkanten-Maximums tatsächlich jeder einzelne Trichterrandekpunkt untersucht werden muss.

Leider ist eine solche Konstruktion nicht leicht. Zur Berechnung des Beispiels würde man eine Formel für den Zusammenhang zwischen dem Wert des maximalen Umweges und der Position des Trichtereckpunktes b benötigen. Dazu würde eine geschlossene Formel für die Nullstellen des Polynoms (49) beitragen. Wie schon im Abschnitt 7.1 im Anschluss an Lemma 7.1 auf S.47 erwähnt, scheint eine solche äußerst komplex zu sein. Ein Versuch mit der erhaltenen Formel unter Maple weiterzuarbeiten ist gescheitert.

Einen Hinweis darauf, dass die beschriebene Beispielkonstruktion zumindest theoretisch

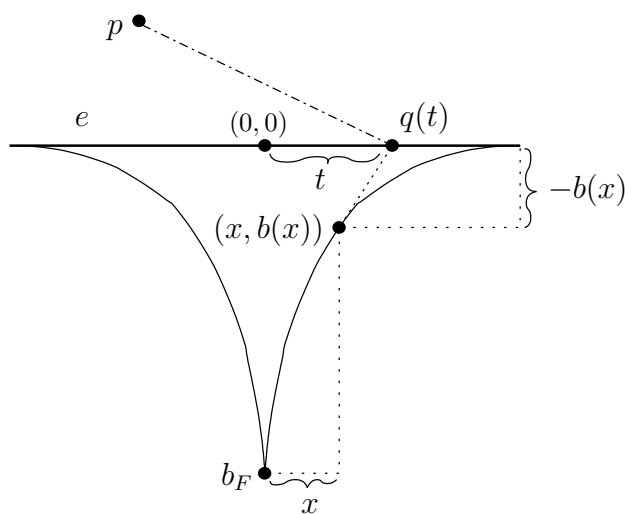


Abbildung 34: Konstruktion eines kontinuierlichen Trichters mit konstantem Umweg

möglich sein müsste, liefert folgende Lösung des entsprechenden kontinuierlichen Problems. Dabei werden als Trichterrand kontinuierliche Funktionen zugelassen und auf der Randkante e soll der Umweg $u_P(p, \cdot) \equiv U$ konstant sein. Die Ausgangssituation sieht aus wie in Abb. 34.

Stellt $b(x)$ den Verlauf des kontinuierlichen Trichterrandes dar, dann lässt sich die Länge des kürzesten Weges von p nach $q(t)$ berechnen als:

$$(51) \quad |\pi(p, q(t))| = |\pi(p, b_F)| + \int_0^x \sqrt{1 + b'^2(s)} ds + \left| \begin{pmatrix} t \\ 0 \end{pmatrix} - \begin{pmatrix} x \\ b(x) \end{pmatrix} \right|$$

Für den direkten Abstand der Punkte p und $q(t)$ gilt:

$$(52) \quad |\overline{pq(t)}| = \left| \begin{pmatrix} t \\ 0 \end{pmatrix} - \begin{pmatrix} p_x \\ p_y \end{pmatrix} \right|$$

Um damit eine Gleichung für die Funktion $b(x)$ aufstellen zu können, benötigen wir nur noch den Zusammenhang zwischen t und x . Dazu muss man sich klar machen, dass die Verbindung zwischen $(x, b(x))$ und $q(t)$ tangential am kontinuierlichen Trichterrand liegt. D.h. die Steigung $-b(x)/(t-x)$ ist gleich $b'(x)$. Nach t umgeformt ergibt sich:

$$(53) \quad t = -\frac{b(x)}{b'(x)} + x$$

Mit Hilfe von (51), (52) und (53) können wir nun eine Integral-Differentialgleichung für b aufstellen.

$$(54) \quad U = \frac{|\pi(p, b_F)| + \int_0^x \sqrt{1 + b'^2(s)} ds + \left| \begin{pmatrix} t \\ 0 \end{pmatrix} - \begin{pmatrix} x \\ b(x) \end{pmatrix} \right|}{\left| \begin{pmatrix} t \\ 0 \end{pmatrix} - \begin{pmatrix} p_x \\ p_y \end{pmatrix} \right|} \\ = \frac{|\pi(p, b_F)| + \int_0^x \sqrt{1 + b'^2(s)} ds - b(x) \sqrt{1 + \frac{1}{b'^2(x)}}}{\sqrt{\left(x - \frac{b(x)}{b'(x)} - p_x\right)^2 + p_y^2}}$$

durch Multiplikation mit dem Nenner und anschließende einmalige Differentiation nach x ergibt sich:

$$\sqrt{1 + b'^2(x)} - b'(x) \sqrt{1 + \frac{1}{b'^2(x)}} - b(x) \frac{-\frac{2b'(x)b''(x)}{b'^4(x)}}{2\sqrt{1 + \frac{1}{b'^2(x)}}} \\ = U \frac{2\left(x - \frac{b(x)}{b'(x)} - p_x\right) \left(1 - \frac{b'^2(x) - b(x)b''(x)}{b'^2(x)}\right)}{2\sqrt{\left(x - \frac{b(x)}{b'(x)} - p_x\right)^2 + p_y^2}}$$

Schließlich lässt sich diese Gleichung vereinfachen zu:

$$\sqrt{1 + b'^2(x)} - \sqrt{b'^2(x) + 1} + \frac{b(x)b''(x)}{b'^2(x)\sqrt{b'^2(x) + 1}} \\ = U \frac{(b'(x)(x - p_x) - b(x))b(x)b''(x)}{b'^2(x)\sqrt{(b'(x)(x - p_x) - b(x))^2 + b'^2(x)p_y^2}}$$

Die ersten beiden Summanden heben sich auf. Anschließend können wir einige Faktoren auf beiden Seiten streichen. Unter anderem fallen so die zweiten Ableitungen weg, und es ergibt

sich eine Differentialgleichung erster Ordnung.

$$\begin{aligned} & \sqrt{(b'(x)(x - p_x) - b(x))^2 + b'^2(x)p_y^2} \\ &= U\sqrt{b'^2(x) + 1} (b'(x)(x - p_x) - b(x)) \end{aligned}$$

Quadrieren auf beiden Seiten ergibt:

$$\begin{aligned} & (b'(x)(x - p_x) - b(x))^2 + b'^2(x)p_y^2 \\ &= U^2 (b'^2(x) + 1) (b'(x)(x - p_x) - b(x))^2 \end{aligned}$$

Was sich noch zu folgender Differentialgleichung zusammenfassen lässt:

$$(55) \quad (U^2 (b'^2(x) + 1) - 1) (b'(x)(x - p_x) - b(x))^2 - p_y^2 b'^2(x) = 0$$

Diese Differentialgleichung hat folgende Lösung:

$$(56) \quad b(x) = -\frac{\left(p_y^{\frac{2}{3}} - (U^2 - 1)^{\frac{1}{3}}(x - p_x)^{\frac{2}{3}}\right)^{\frac{3}{2}}}{U}$$

Die erhaltene Funktion ist, wenn man $(x - p_x)^{\frac{2}{3}}$ als $\sqrt[3]{(x - p_x)^2}$ deutet, spiegelsymmetrisch zur Vertikalen durch p_x . Die x-Koordinate des Trichterfußpunktes ist dann durch die Form der Funktion (siehe Abb. 35) automatisch auf $b_{F x} = p_x$ festgelegt. Zur Vereinfachung können wir dann noch das Koordinatensystem so anpassen, dass $p_x = 0$ ist.

• p

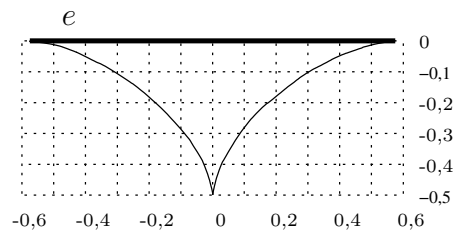


Abbildung 35: Ein kontinuierlicher Trichter mit konstantem Umweg $U = 2$

Versucht man hingegen $(x - p_x)^{\frac{2}{3}}$ als $\text{sign}(x - p_x) \sqrt[3]{(x - p_x)^2}$ zu interpretieren, dann ergibt sich ein Trichterrand, der für $x < p_x$ nicht mehr nach außen hin konvex ist.

Weiterhin lässt sich feststellen, dass die Lösung $b(x)$ (wie auch schon die Differentialgleichung (55)) nicht von $|\pi(p, b_F)|$ abhängt. Vielmehr ist $|\pi(p, b_F)|$ über die geometrische Anordnung durch die Lösung festgelegt. Denn auch in $q(0)$ muss der vorgegebene Umwegwert U erreicht werden, und es folgt $U = u_P(p, q(0)) = \frac{|\pi(p, b_F)| - b(0)}{p_y}$. Unter Benutzung von (56) ergibt sich:

$$|\pi(p, b_F)| = p_y U + \frac{p_y}{U}$$

In Abbildung 35 ist ein kontinuierlicher Trichter mit konstantem Umweg $U := 2$ bei $p_y := 1$ dargestellt. Für die Länge des kürzesten Weges von p nach b_F ergibt sich in diesem Fall $|\pi(p, b_F)| = 2.5$. Also ist die gewünschte Situation eines kontinuierlichen Trichters mit konstantem Umweg tatsächlich generierbar.

Leider ist diese Konstruktion nur ein Hinweis darauf, dass man zur Ermittlung des Eckpunkt-Randkanten-Maximums im schlimmsten Fall alle Trichtereckpunkte untersuchen muss. Allerdings ist die Frage für die Laufzeit des exakten Algorithmus auch nicht entscheidend. Selbst wenn wir ein Verfahren finden könnten, dass für jedes Eckpunkt-Randkantenpaar in konstanter Zeit das Umwegmaximum ermittelt, kämen wir durch das Ausprobieren all solcher Paare immer noch auf eine Laufzeit von $O(n^2)$ für den gesamten Algorithmus. Dieselbe Laufzeitschranke werden wir aber auch in Satz 7.1 erhalten, obwohl wir alle Trichtereckpunkte betrachten. Ein Verfahren zur Ermittlung eines Eckpunkt-Randkanten-Maximums mit konstanter Laufzeit, dessen Existenz nach den Ergebnissen dieses Abschnittes sehr fraglich erscheint, könnte also höchstens für andere Algorithmen interessant sein.

7.5 Bereitstellung der Trichter

Es bleibt zu klären, wie wir die in Algorithmus 1 benötigten Trichter (genauer: die Randketten π_l und π_r) berechnen, ohne die Laufzeit des gesamten exakten Algorithmus höher als die ohnehin durch das Testen aller n^2 Eckpunkt-Randkanten-Paare benötigten $O(n^2)$ Schritte werden zu lassen. Schon auf Seite 48 hatten wir festgestellt, dass die Trichterrandketten Teile der kürzesten Wege $\pi(p, q_l)$ bzw. $\pi(p, q_r)$ sind. Zum Auffinden des Trichters genügt es also, diese kürzesten Wege zur Verfügung zu haben, und sie von q_l bzw. q_r aus nur zurück bis zu dem Punkt ihres Zusammentreffens zu betrachten, der den Trichterfußpunkt b_F darstellt.

Leider können wir zum Auffinden der gesuchten kürzesten Wege $\pi(p, q_l)$ und $\pi(p, q_r)$ nicht einfach den Algorithmus von Guibas und Hershberger aus [GH87] verwenden, denn dieser benötigt nach einer unproblematischen Vorbereitungszeit von $O(n)$ für jede Anfrage eine Laufzeit von $O(\log n + l)$, wenn l die Kantenzahl des gesuchten kürzesten Pfades ist. Schon der kleine logarithmische Anteil würde, da wir vorhaben alle n^2 Eckpunkt-Randkanten-Paare zu durchlaufen, die Gesamtlaufzeit unseres exakten Algorithmus auf $O(n^2 \log n)$ verschlechtern. Und auch der in l lineare Teil ist problematisch.

Eine Lösung dieses Problems bietet der Ansatz, zu einem festen Eckpunkt p die kürzesten Wege zu allen anderen Eckpunkten gleichzeitig zu berechnen. Dabei entsteht der *Shortest*

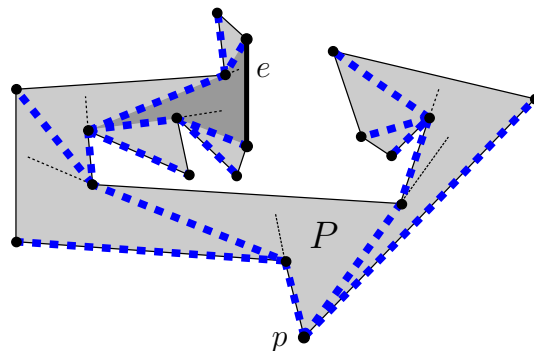


Abbildung 36: Shortest Path Tree mit einem Trichter

Path Tree des Eckpunktes p . Wie z.B. in [LP84] gezeigt, handelt es sich dabei um einen Baum, der alle kürzesten Wege von p zu anderen Eckpunkten des Polygons enthält. Als

Knoten des Baumes tauchen genau alle Eckpunkte von P auf. Die Kanten sind als Teile von kürzesten Wegen direkte, gerade Verbindungsstrecken zwischen gegenseitig sichtbaren Eckpunkten. Ein solcher Shortest Path Tree ist in Abb. 36 dargestellt.

Um den Shortest Path Tree zu erhalten, sind verschiedene Verfahren denkbar. Guibas et al. haben in [GHL⁺87] einen Algorithmus vorgestellt, der in linearer Zeit $O(n)$ zu einem Eckpunkt p eines schon triangulierten Polygons den Shortest Path Tree berechnet. Da nach Chazelle [Cha91] die Triangulierung eines Polygons ebenfalls in linearer Zeit möglich ist, können wir auf diese Weise in Zeit $O(n)$ einen Shortest Path Tree zum Eckpunkt p erhalten.

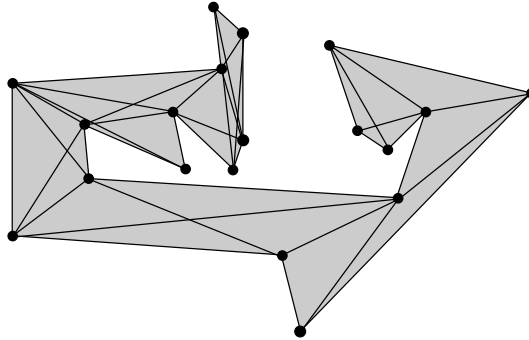


Abbildung 37: der Sichtbarkeitsgraph eines Polygons

Ein andere Möglichkeit besteht darin, zunächst in einer Vorbereitungsphase den *Sichtbarkeitsgraphen* des Polygons zu berechnen (siehe Abb.37). Die Knoten dieses Graphen sind wieder die Eckpunkte des Polygons. Die Kanten sind alle direkten geraden Verbindungen zwischen gegenseitig sichtbaren Eckpunktpaaren. Dieser Sichtbarkeitsgraph eines Polygons lässt sich z.B. in $O(n^2)$ durch n Rundläufe durch den Polygonrand analog zum Vorgehen in [Lee83] bestimmen. Dabei kann man erreichen, dass in der berechneten Datenstruktur, die zu einem Eckpunkt inzidenten Kanten des Sichtbarkeitsgraphen im Uhrzeigersinn sortiert gespeichert sind.

Anschließend kann man durch folgendes leichte Verfahren in linearer Zeit den Shortest Path Tree berechnen. Man beginnt mit den Kanten zu allen von p aus sichtbaren Punkten. Dann setzt man für jede Kante des bisher konstruierten Teiles des Shortest Path Tree, die zu einem bestimmten Eckpunkt q führt, den Baum mit den Kanten des Sichtbarkeitsgraphen in q fort, die bildlich gesprochen hinter die von P^C in q gebildete Ecke biegen (siehe dazu die dünn gepunktet gezeichneten Verlängerungen der Kanten des Shortest Path Tree in Abb. 36). Die zum Auffinden dieser Kanten benötigten Informationen können durch einen linearen Durchlauf durch alle Sichtbarkeitskanten für jeden Eckpunkt vorab berechnet werden.

Wie auch immer man den Shortest Path Tree berechnet, danach können wir durch einen von p aus gestarteten Durchlauf durch die $n - 1$ Kanten jedem Eckpunkt q die Länge des kürzesten Pfades $\pi(p, q)$ zuordnen. Fassen wir das bisher erreichte in einem Lemma zusammen:

Lemma 7.3 *Zu einem einfachen Polygon P und einem der n Eckpunkte p des Polygons kann man in einer Laufzeit von $O(n)$ den zugehörigen Shortest Path Tree berechnen, bei dem zu jedem Knoten q die Distanz $|\pi(p, q)|$ gespeichert ist.*

Um nun zu einer Randkante e mit den Endpunkten q_l und q_r den Trichter bezüglich p

zu berechnen, müssen wir nur von q_l und q_r aus nach oben, d.h. in Richtung der Wurzel p , steigen, bis sich die dabei beschrifteten Teile von $\pi(q_l, p)$ bzw. $\pi(q_r, p)$ treffen. Dabei muss noch bedacht werden, wie man das Aufeinandertreffen der beiden Pfade feststellt, falls $\pi(q_l, b_F)$ und $\pi(q_r, b_F)$ eine unterschiedliche Anzahl von Kanten haben.

Es sind verschiedene Verfahren denkbar. Hat man die Kanten in jedem Knoten im Uhrzeigersinn sortiert, dann kann man z.B. testen, ob im aktuellen Knoten im Shortest Path Tree eine Kante von der richtigen Seite her einmündet. Da q_l und q_r benachbarte Eckpunkte sind, muss die erste Einmündung von der richtigen Seite her der jeweils andere Pfad $\pi(q_l, p)$ bzw. $\pi(q_r, p)$ sein, und man ist im Trichterfußpunkt b_F angekommen.

Eine Variante, die immer funktioniert, ist das Markieren der schon beschrifteten Knoten. Dabei steigt man abwechselnd von q_l und von q_r aus immer eine Kante im Shortest Path Tree nach oben und markiert die besuchten Knoten. Stößt man z.B. bei dem Nach-Oben-Steigen von q_l aus auf einen schon markierten Knoten, dann weiß man, dass man hier auf den Pfad $\pi(q_r, p)$ und somit auf den Trichterfußpunkt b_F gestoßen ist. Man hat dann zwar unter Umständen mehr als die gewünschten $k := \#\pi(q_l, b_F) + \#\pi(q_r, b_F)$ Kanten besucht, wobei $\#K$, wie schon auf S. 40 eingeführt, die Kantenzahl der Kette K bezeichne. Diese Überschreitung ist aber unproblematisch. Die Anzahl der betrachteten Kanten ist dann nämlich $2 \max(\#\pi(q_l, p), \#\pi(q_r, p)) \leq 2(\#\pi(q_l, p) + \#\pi(q_r, p)) = 2k$. Die Laufzeit des Verfahrens bleibt also linear in k . Der Algorithmus und seine Laufzeit seien noch einmal formaler dargestellt.

Algorithmus 2

Input: ein Punkt $p \in P$, der zugehörige Shortest Path Tree mit den Längen $|\pi(p, q)|$ für jeden Eckpunkt q , eine Randkante e mit den Endpunkten q_l und q_r

Output: der zu p und e gehörige Trichter mit Fußpunkt b_F anhand der Randketten π_l und π_r , die Länge $|\pi(p, b_F)|$

1. $\pi_l := (q_l)$, $\pi_r := (q_r)$, $v := q_l$, $w := q_r$, MARKIERE q_l UND q_r
2. SOLANGE b_F NOCH NICHT GEFUNDEN IST, DURCHLAUFE FOLGENDE SCHLEIFE.
 - a) $v := \text{VATER VON } v \text{ IM SHORTEST PATH TREE}$. $\pi_l := \pi_l \oplus v$.
IST v SCHON MARKIERT, DANN SETZE $b_F := v$ UND BRECHE DIE SCHLEIFE AB.
ANSONSTEN MARKIERE v .
 - b) $w := \text{VATER VON } w \text{ IM SHORTEST PATH TREE}$. $\pi_r := \pi_r \oplus w$.
IST w SCHON MARKIERT, DANN SETZE $b_F := w$ UND BRECHE DIE SCHLEIFE AB.
ANSONSTEN MARKIERE w .
3. VERKÜRZE DIE ERHALTENEN PFADE π_l UND π_r SO WEIT, DASS BEIDE IN b_F ENDEN.
4. GEBE π_l , π_r UND $|\pi(p, b_F)|$ AUS. LETZTERER WERT KANN DIREKT AUS DER EINGABE ÜBERNOMMEN WERDEN.

Lemma 7.4 *Algorithmus 2 hat eine Laufzeit von $O(k)$, wenn $k := \#\pi_l + \#\pi_r$ die Anzahl der Trichterrandkanten bezeichnet.*

7.6 Der exakte Algorithmus

In den vorangegangenen Abschnitten 7.1-5 haben wir alle für einen exakten Algorithmus benötigten Elemente zusammengetragen. Er funktioniert dann z.B. wie folgt:

Algorithmus 3

Input: ein einfaches Polygon P

Output: ein Eckpunkt-Randpunkt-Paar (p, q) und dessen maximaler Umweg $u_P(p, q) = u_P^{\max}$

1. TRIANGULIERE DAS POLYGON P NACH CHAZELLE [Cha91].
2. FÜR JEDEN ECKPUNKT p DES POLYGONS P DURCHLAUFE FOLGENDE SCHLEIFE.
 - a) BERECHNE DEN SHORTEST PATH TREE VON p NACH GUIBAS ET AL. [GHL⁺87].
 - b) BERECHNE IN EINEM DURCHLAUF DURCH DEN SHORTEST PATH TREE FÜR JEDEN ECKPUNKT q DIE DISTANZ $|\pi(p, q)|$.
 - c) DURCHLAUFE FÜR JEDE RANDKANTE e DES POLYGONS P DIE FOLGENDE SCHLEIFE.
 - I. BERECHNE DEN TRICHTER VON p UND e MIT ALGORITHMUS 2.
 - II. BERECHNE MIT ALGORITHMUS 1 EINEN PUNKT $q_e \in e$, DER DIE GLEICHUNG $u_P(p, q_e) = \max_{q' \in e} u_P(p, q')$ ERFÜLLT, SOWIE DEN WERT $u_P(p, q_e)$.
 - d) FINDE AUS DEN BERECHNETEN q_e EINEN PUNKT q_p^{\max} MIT MAXIMALEM UMWEG, D.H. $u_P(p, q_p^{\max}) = \max_{e \text{ Kante von } P} u_P(p, q_e)$.
3. FINDE EIN PAAR (p^*, q_p^{\max}) MIT MAXIMALEM UMWEG, D.H. $u_P(p^*, q_p^{\max}) = \max_p \text{ Ecke von } P u_P(p, q_p^{\max})$. GEBE DIESES UND DEN ZUGEHÖRIGEN UMWEG $u_P(p, q_p^{\max})$ AUS.

Satz 7.1 *Zu einem einfachen Polygon P mit n Eckpunkten kann man in einer Laufzeit von $O(n^2)$ ein Punktpaar (p, q) mit maximalem Umweg finden.*

Beweis. Zeigen wir zuerst die Korrektheit des Algorithmus. Nach Satz 5.1 wird das Umwegmaximum von einem Eckpunkt-Randpunkt-Paar angenommen. Algorithmus 3 durchläuft alle Eckpunkt-Randkantenpaare. Für alle solchen Paare wird in Schritt 2(c)ii das Umwegmaximum gebildet. Dazu wird Algorithmus 1 herangezogen, dessen Korrektheit in Lemma 7.2 bewiesen ist. Mindestens einmal muss im Schritt 2(c)ii also auch das globale Umwegmaximum ermittelt werden. Da dieses Punktpaar in die Maximumsuche des Algorithmus einfließt, gibt er das Punktpaar aus oder ein anderes, welches aber den gleichen maximalen Umweg hat. In beiden Fällen ist der Algorithmus korrekt.

Nicht viel schwerer ist die Laufzeit des Algorithmus zu bestimmen. Schritt 1 benötigt nach [Cha91] eine Laufzeit von $O(n)$. Die Maximumbildung in Schritt 3 kann mit in die Schleife gezogen werden und benötigt auch nur lineare Zeit. Entscheidend für das Laufzeitverhalten ist also die Schleife in Schritt 2, die n mal durchlaufen wird.

Nach Lemma 7.3 benötigen die Schritte 2a und 2b pro Aufruf eine Laufzeit von $O(n)$. Schritt 2d kann mit in der inneren Schleife erledigt werden und benötigt dann auch nur lineare Zeit. Insgesamt verursachen die Schritte 2a, 2b und 2d für den Algorithmus also eine Laufzeit von $O(n^2)$.

Es bleibt der Einfluss der inneren Schleife in Schritt 2c zu prüfen. Nach Lemma 7.4 und Lemma 7.2 benötigen die beiden inneren Schritte jeweils eine Laufzeit von $O(k)$, wobei k die Anzahl der beteiligten Trichterrandkanten ist.

Da nun jede Trichterrandkante Teil eines kürzesten Weges zu p ist, ist sie auch eine Kante des Shortest Path Tree. Innerhalb der Schleife in 2c werden also zu jedem Eckpunkt p höchstens $n - 1$ Kanten betrachtet. Weil wir weiterhin zeigen werden, dass jede Kante des Shortest Path Tree zu höchstens zwei Trichtern gehören kann, wird jede dieser $n - 1$ Kanten zu einem Punkt p innerhalb der inneren Schleife also höchstens zweimal betrachtet. Damit ist gezeigt, dass Schritt 2c pro Eckpunkt nur zu einer Laufzeit von $O(n)$ führt. Insgesamt verursacht der Schritt für den Algorithmus also höchstens eine Laufzeit von $O(n^2)$.

Es bleibt nur noch zu zeigen, dass jede Kante d des Shortest Path Tree zu höchstens zwei Trichtern gehören kann. Dazu stellen wir fest, dass jeweils die inneren Gebiete von zwei verschiedenen Trichtern $\Phi(e)$ und $\Phi(f)$ zu einem Eckpunkt p disjunkt sind. Dies ist anschaulich klar (siehe Abb. 36) und Guibas et al. beweisen es in Lemma 2.1.(b) in [GHL⁺87]. Außerdem berührt, da sich die Kanten des Shortest Path Tree (als kürzeste Wege) nicht schneiden, jeder Trichter nur die Kanten des Shortest Path Tree, die zu seinen Randketten gehören.

Falls die betrachtete Kante d des Shortest Path Tree also zu keinem entarteten Trichter gehört (siehe Randkante h in Abb. 8 auf S.17), dann kann sie zu höchstens zwei Trichtern gehören. Denn d kann zu jeder Seite nur einen nicht-entarteten Trichter begrenzen. Ansonsten würden sich deren Flächen überschneiden.

Gehört d andererseits zu einem entarteten Trichter, dann ist d selbst die Oberseite dieses Trichters. Zu einer Seite von d liegt also P^C . d kann folglich nur noch zur anderen Seite hin einen Trichter begrenzen. Dort kann sich tatsächlich höchstens ein Trichter anschließen, weil sich sonst wieder die Flächen der Trichter überschneiden würden. Ein weiterer entarteter Trichter kommt auch nicht in Frage. Denn ein entarteter Trichter besteht nur aus einer Kante, der zugehörigen Polygonrandkante (=Trichteroberseite). d kann also nur Teil eines, nämlich des eigenen, entarteten Trichters sein.

Lässt man übrigens in die Argumentation noch das Resultat aus Lemma 2.1.(b) in [GHL⁺87] einfließen, dass die Trichter $\Phi(\cdot)$ zu einem Eckpunkt p das gesamte Polygon überdecken, dann wird klar, dass jede Kante d des Shortest Path Tree zu genau zwei Trichtern gehört.

Also benötigt Schritt 2c pro Aufruf lineare Zeit, da jede Kante des Shortest Path Tree genau zweimal betrachtet wird, und dieser Baum $n - 1$ Kanten hat. Für den gesamten Algorithmus 3 resultiert damit eine Laufzeit von $O(n^2)$. \square

8 Ein schneller Approximationsalgorithmus

Durch die umfangreichen Vorarbeiten der vorangehenden Abschnitte können wir jetzt einen Approximationsalgorithmus angeben, der im wesentlichen dem Approximationsalgorithmus des Kettenfalles aus Abschnitt 4 von [EBKLL01] entspricht. Wir geben aus Vollständigkeitsgründen die gesamte Argumentation wieder und heben die nötigen Änderungen hervor.

Für den Anfang beschränken wir uns auf Eckpunkt-Randpunkt-Paare, die in einem bestimmten Winkel zueinander liegen. Im folgenden sei bei dem Eckpunkt-Randpunkt-Paar (p, q) immer p ein Eckpunkt. Weiter nennen wir (p, q) ein *Punktpaar in Richtung ρ* , wenn $q - p = |q - p|(\cos \rho, \sin \rho)$ gilt. Über solche Paare können wir folgende Aussage treffen:

Lemma 8.1 *Zu einem gegebenen Winkel ρ kann man in Zeit $O(n \log n)$ ein in P oder P^C gegenseitig sichtbares Eckpunkt-Randpunkt-Paar in Richtung ρ finden, welches einen maximalen Umweg unter allen solchen Eckpunkt-Randpunkt-Paaren realisiert.*

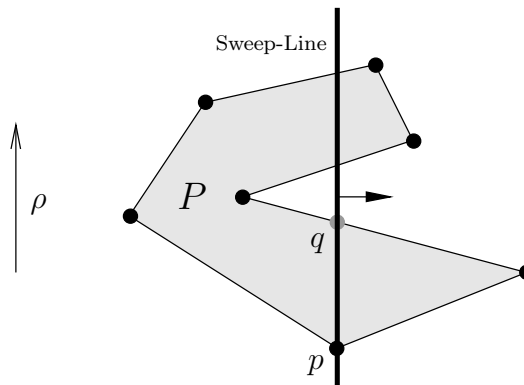


Abbildung 38: Sweep-Algorithmus zum Finden von Punktpaaren in Richtung ρ

Beweis. Mit einem einfachen Sweep-Algorithmus lassen sich, wie in Abbildung 38 gezeigt, in $O(n \log n)$ alle in P oder P^C gegenseitig sichtbaren Eckpunkt-Randpunkt-Paare in Richtung ρ finden. Drehe dazu das Polygon so, dass ρ zu $\pi/2$ wird und lasse dann eine vertikale Sweep-Line von links nach rechts laufen. Die Polygoneckpunkte werden dazu in $O(n \log n)$ nach x-Koordinate sortiert.

Erreicht die Sweep-Line einen Polygoneckpunkt p , so tritt ein Sweep-Event auf. Falls es ein solches gibt, wird das zugehörige Eckpunkt-Randpunkt-Paar (p, q) in Richtung ρ zur Ergebnisliste hinzugefügt. Die Ermittlung von q erfolgt in konstanter Zeit, da der Algorithmus die gerade von der Sweep-Line berührten Polygonrandkanten nach ihrer Höhe sortiert speichert. Während eines Events werden also die durch die gefundenen Polygoneckpunkte beendeten Kanten aus dem Speicher gelöscht und die neu begonnenen werden neu aufgenommen. Benutzt man zur Speicherung der aktuellen Kanten einen balancierten Baum, so funktioniert dies in einer Laufzeit von $O(\log n)$ pro Zugriff. Insgesamt benötigt der Sweep-Algorithmus also eine Laufzeit von $O(n \log n)$ und findet natürlich höchstens $O(n)$ viele Eckpunkt-Randpunkt-Paare in Richtung ρ .

Etwas mehr Gedanken als im Streckenzugfall muss man sich über die Laufzeit der an-

schließenden Maximumsuche machen, da im Polygonfall die Ermittlung des kürzesten Pfades komplizierter ist. Nach Guibas und Hershberger [GH87] lässt sich nach einer allgemeinen Vorbereitungszeit von $O(n)$ zu jedem beliebigem Punktpaar $(p, q) \in P \times P$ in $O(\log n)$ die Länge des kürzesten Pfades $|\pi(p, q)|$ und damit auch $u_P(p, q)$ bestimmen. Insgesamt finden wir also in $O(n \log n)$ ein Eckpunkt-Randpunkt-Paar, das die Suchkriterien erfüllt. \square

Um den maximalen Umweg u_P^{\max} anzunähern, müssen wir auf jeden Fall dem Maximum der Eckpunktpaare nahe kommen. Dazu definieren wir den *maximalen Umweg der Eckpunktpaare* als:

$$u_P^E \max := \max_{p, q \text{ Eckpunkte von } P} u_P(p, q)$$

Diesen maximalen Umweg der Eckpunktpaare können wir nun in Zeit $O(n \log n)$ approximieren:

Lemma 8.2 *Zu einem gegebenen einfachen Polygon P mit n Ecken und einer reellen Zahl $\eta > 0$ lässt sich in Zeit $O(n \log n)$ ein Eckpunktpaar (p, q) finden, für das gilt:*

$$u_P(p, q) \geq \frac{1}{1 + \eta} u_P^E \max$$

Beweis. Sei E die Menge aller Eckpunkte von P . Wir werden mit einem *sparse $(1 + \eta)$ -Spanner* („spärlicher $(1 + \eta)$ -Spanngraph“) von E arbeiten. Dies ist ein Graph mit der Knotenmenge E und $O(n)$ vielen Kanten, in dem es für jedes Paar $p, q \in E$ einen verbindenden Pfad gibt, dessen Länge höchstens $(1 + \eta) |\overline{pq}|$ ist.

Sei nun $(p, q) \in E \times E$ ein Eckpunktpaar mit maximalem Umweg: $u_P(p, q) = u_P^E \max$. Sei $p = p_0, p_1, \dots, p_k = q$ der approximierende Pfad im Spanner. Dann gilt mit einer ähnlichen Argumentation wie im Beweis zu Lemma 3.1:

$$\begin{aligned} u_P^E \max = u_P(p, q) &= \frac{|\pi(p, q)|}{|\overline{pq}|} \leq \frac{\sum_{i=0}^{k-1} |\pi(p_i, p_{i+1})|}{|\overline{pq}|} \\ &\leq (1 + \eta) \frac{\sum_{i=0}^{k-1} |\pi(p_i, p_{i+1})|}{\sum_{i=0}^{k-1} |\overline{p_i p_{i+1}}|} \\ &\leq (1 + \eta) \max_{0 \leq i \leq k-1} \frac{|\pi(p_i, p_{i+1})|}{|\overline{p_i p_{i+1}}|} \\ &= (1 + \eta) \max_{0 \leq i \leq k-1} u_P(p_i, p_{i+1}) \end{aligned}$$

Da $\overline{p_i p_{i+1}}$ nach Konstruktion für jedes i eine Kante des sparse Spanner ist, genügt es zum Beweis des Satzes einen $O(n \log n)$ -Algorithmus anzugeben, der eine Kante eines $(1 + \eta)$ -Spanner mit maximalem Umweg ausgibt.

Algorithmus 4

Input: ein einfaches Polygon P mit n Ecken und eine reelle Zahl $\eta > 0$

Output: ein Eckpunktpaar (p, q) mit $(1 + \eta)u_P(p, q) \geq u_P^E \max$

1. BAUE EINEN SPARSE $(1 + \eta)$ -SPANNER AUS DEN ECKPUNKTEN VON P .
2. BERECHNE FÜR JEDE KANTE DES SPANNER DEN UMWEG.
3. GEBE DAS PUNKTPAAR EINER KANTE MIT MAXIMALEM UMWEG AUS.

Die Korrektheit des Algorithmus ist schon gezeigt. Nach Gutwin und Keil [KG92] hat Schritt 1 eine Laufzeit von $O(n \log n)$. Wegen der Spärlichkeit des Spanners hat er nur $O(n)$ viele Kanten. Wie im Beweis von Lemma 8.1 müssen wir wieder im Vergleich zum Streckenzugfall eine zusätzliche Überlegung für die Laufzeit aufwenden. Wir benutzen wieder das Resultat von Guibas und Hershberger [GH87], nach dem man nach Vorbereitungszeit $O(n)$ in $O(\log n)$ für ein Punktpaar (p, q) den Umweg $u_P(p, q)$ berechnen kann. Deshalb brauchen Schritt 2 und 3 Zeit $O(n \log n)$. Also liegt auch die Gesamtlaufzeit des Algorithmus in $O(n \log n)$. \square

Damit kommen wir zum eigentlichen Approximationsalgorithmus.

Satz 8.1 Sei $\varepsilon > 0$ eine gegebene reelle Zahl, und sei P ein einfaches Polygon mit n Ecken. Mit einer Laufzeit von $O(n \log n)$ kann man dann ein Eckpunkt-Randpunkt-Paar (p, q) bestimmen, das den maximalen Umweg approximiert, d.h. es gilt:

$$u_P(p, q) \geq \frac{1}{1 + \varepsilon} u_P^{\max}$$

Beweis. Am Anfang müssen wir einige Zahlenwerte wählen, die im weiteren Verlauf des Beweises benötigt werden. Zuerst bestimmen wir ein kleines η das erfüllt:

$$(57) \quad 0 < \eta < \varepsilon \text{ und } \frac{1 - \eta}{(1 + \eta)^2} \geq \frac{1}{1 + \varepsilon}$$

Dann wählen wir einen kleinen Winkel $\bar{\beta}$ mit:

$$(58) \quad 0 < \bar{\beta} < \frac{\pi}{2} \text{ und für alle } \beta \in (0, \bar{\beta}]: \cos \beta \geq \frac{1}{1 + \varepsilon}$$

Schließlich benötigen wir noch einen kleinen Winkel ρ , so dass:

$$(59) \quad 0 < \rho < \frac{\pi}{2} \text{ und } \frac{\sin \rho}{\sin \bar{\beta}} \leq \frac{\eta}{2} \text{ und } \cos \rho \geq \max\left(1 - \frac{\eta}{2}, \frac{1}{1 + \eta}\right)$$

Der aufmerksame Leser wird beim Vergleich mit [EBKLL01] feststellen, dass die Bedingungen an ρ und η stärker sind. Wir werden diesen Unterschied später genauer begründen. Zunächst starten wir mit den gewählten Werten folgenden Algorithmus:

Algorithmus 5

Input: ein einfaches Polygon P mit n Ecken und eine reelle Zahl $\varepsilon > 0$

Output: ein Eckpunkt-Randpunkt-Paar (p, q) mit $(1 + \varepsilon)u_P(p, q) \geq u_P^{\max}$

1. FÜR JEDE GANZE ZAHL m ZWISCHEN 0 UND $2\pi/\rho$ FINDE EIN IN P ODER P^C GEGENSEITIG SICHTBARES ECKPUNKT-RANDPUNKT-PAAR IN RICHTUNG $m\rho$ MIT MAXIMALEM UMWEG UNTER SOLCHEN PAAREN.
2. WÄHLE UNTER DEN GEFUNDENEN PAAREN EIN MAXIMALES. WIR NENNEN ES (p_1, q_1) .
3. FINDE MIT ALGORITHMUS 4 EIN ECKPUNKTPAAR (p_2, q_2) MIT $(1 + \eta)u_P(p_2, q_2) \geq u_P^{\max}$.
4. GEBE DAS PUNKTPAAR (p_1, q_1) ODER (p_2, q_2) MIT MAXIMALEM UMWEG AUS.

Die Schritte 1 und 2 laufen nach Lemma 8.1 in $O(n \log n)$, da der Faktor $2\pi/\rho$ nicht von n abhängt. Schritt 3 läuft nach Lemma 8.2 ebenfalls in $O(n \log n)$. Und Schritt 4 läuft natürlich in $O(1)$. Damit ist die Laufzeit des Approximationsalgorithmus $O(n \log n)$.

Im Gegensatz zum Algorithmus aus [EBKLL01] maximieren wir im Schritt 1 den Umweg nicht auf der Kante des Randpunktes. Dies würde, da wir bei der Benutzung von Lemma 7.1 im schlimmsten Fall alle $O(n)$ Trichtereckpunkte betrachten müssten, zu einer zu hohen Laufzeit führen. Der Verzicht auf diese Maximierung führt aber leider zu komplizierteren Überlegungen zur Korrektheit des Algorithmus und damit auch zu den härteren Bedingungen an η und ρ , was am Ende dieses Beweises deutlich wird.

Überhaupt ist die Korrektheit etwas schwieriger zu zeigen. Ist P konvex, dann ist jedes Eckpunkt-Randpunkt-Paar maximal, und deshalb ist auch die Ausgabe des Algorithmus korrekt. Sei also im folgenden P nicht konvex. Dann gibt es nach Korollar 5.1 ein in P^C gegenseitig sichtbares Eckpunkt-Randpunkt-Paar (p, q) mit maximalem Umweg. Handelt es sich um ein Eckpunktpaar, dann gilt

$$u_P^{\max} = u_P(p, q) = u_P^E \max \leq (1 + \eta)u_P(p_2, q_2) \stackrel{(57)}{\leq} (1 + \varepsilon)u_P(p_2, q_2)$$

und die Ausgabe ist korrekt.

Ist hingegen (p, q) ein echtes Eckpunkt-Randpunkt-Paar, bei dem p den Eckpunkt bezeichne, dann gilt nach Lemma 7.1 für den Winkel β_q zwischen \overline{pq} und der Kante e , auf der q liegt:

$$(60) \quad \left(\beta_q = \frac{\pi}{2}\right) \text{ oder } \frac{\cos \beta_q}{\sin \beta} = -\frac{|\overline{pq}|}{|\pi(p, q)|}$$

Wir bezeichnen im folgenden mit β_q^{neu} nicht mehr den im Abschnitt 7.1 durch die Orientierung festgelegten Winkel β_q sondern den kleineren der beiden möglichen Winkel zwischen e und \overline{pq} . Wegen $\beta_q \in (0, \pi)$ heißt dies formal: $\beta_q^{\text{neu}} := \min(\beta_q, \pi - \beta_q)$. Dann gilt $0 < \beta_q^{\text{neu}} \leq \pi/2$, und:

$$(61) \quad \text{falls } \beta_q^{\text{neu}} = \pi - \beta_q : \left(\beta_q^{\text{neu}} = \frac{\pi}{2}\right) \text{ oder } \frac{\cos \beta_q^{\text{neu}}}{\sin \beta} = \frac{|\overline{pq}|}{|\pi(p, q)|}$$

Wir wissen, dass β_q^{neu} in $(0, \pi/2]$ liegt. Ist zusätzlich $\beta_q^{\text{neu}} \leq \overline{\beta}$, so können wir wegen der Wahl von $\overline{\beta}$ (siehe (58)) folgendes folgern:

$$\begin{aligned} \text{falls } \beta_q^{\text{neu}} = \beta_q : \quad u_P^{\max} = u_P(p, q) &= \frac{|\pi(p, q)|}{|\overline{pq}|} \stackrel{(60), \overline{\beta} < \pi/2}{\leq} \frac{\sin \beta}{\cos \beta_q^{\text{neu}}} \\ &\leq \frac{1}{\cos \beta_q^{\text{neu}}} \stackrel{\beta_q^{\text{neu}} \leq \overline{\beta}, (58)}{\leq} (1 + \varepsilon) \\ \text{falls } \beta_q^{\text{neu}} = \pi - \beta_q : \quad u_P^{\max} = u_P(p, q) &= \frac{|\pi(p, q)|}{|\overline{pq}|} \stackrel{(61), \overline{\beta} < \pi/2}{\leq} \frac{\sin \beta}{\cos \beta_q^{\text{neu}}} \\ &\leq \frac{1}{\cos \beta_q^{\text{neu}}} \stackrel{\beta_q^{\text{neu}} \leq \overline{\beta}, (58)}{\leq} (1 + \varepsilon) \end{aligned}$$

Auch in diesen Fällen ist also die Ausgabe des Algorithmus korrekt.

Im folgenden werden wir nur noch mit dem kleineren Winkel β_q^{neu} argumentieren, jedoch das hochgestellte „neu“ weglassen. Es bleibt der Fall zu betrachten, in dem (p, q) ein echtes Eckpunkt-Randpunkt-Paar und $\pi/2 \geq \beta_q > \overline{\beta} > 0$ ist. Dabei benötigen wir, dass für $0 < \nu < \rho$ gilt:

$$(62) \quad \frac{\sin(\beta_q + \nu)}{\sin \beta_q} \geq 1 - \frac{\eta}{2}$$

Dies ist wie folgt einzusehen: Betrachte die Funktion $f(\beta) := \sin(\beta + \nu) / \sin \beta$. Die Ableitung ist $f'(\beta) = (\cos(\beta + \nu) \sin \beta - \sin(\beta + \nu) \cos \beta) / \sin^2 \beta$. Mit Benutzung eines Additionstheorems ergibt sich $f'(\beta) = -\sin(\beta - (\nu + \beta)) / \sin^2 \beta = -\sin \nu / \sin^2 \beta$. Wegen der Annahme über ν und der Beschränktheit von ρ nach Gleichung (59) ist also $f'(\beta)$ nirgends positiv. Insbesondere ist f also in $[0, \pi/2)$ monoton fallend. Also gilt

$$\frac{\sin(\beta_q + \nu)}{\sin \beta_q} \geq \frac{\sin(\frac{\pi}{2} + \nu)}{\sin \frac{\pi}{2}} = \cos \nu \geq \cos \rho \stackrel{(59)}{\geq} 1 - \frac{\eta}{2}$$

und (62) ist bewiesen. Außerdem gilt wegen der Grenzen der betrachteten Werte noch:

$$(63) \quad 0 < \sin \nu < \sin \rho$$

$$(64) \quad \sin \beta_q > \sin \bar{\beta} > 0$$

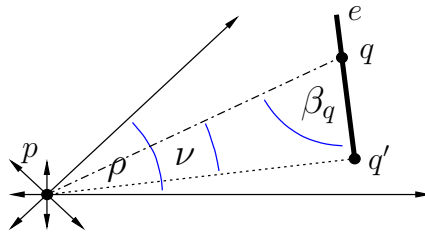


Abbildung 39: Eckpunkt q' liegt noch im Winkelstück ρ

Damit sind die Vorbereitungen abgeschlossen. (p, q) ist weiterhin ein in P^C gegenseitig sichtbares echtes Eckpunkt-Randpunkt-Paar mit maximalem Umweg, und β_q ist der kleinere Winkel zwischen \overline{pq} und der Kante e von q . Für β_q gilt im verbliebenen Fall $\pi/2 \geq \beta_q > \bar{\beta} > 0$.

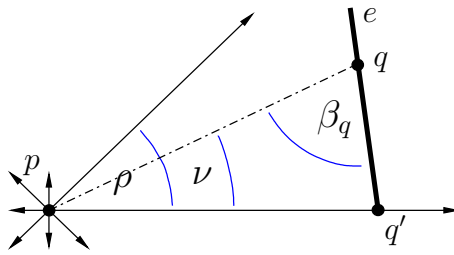


Abbildung 40: Schnittpunkt q' von Kante und Strahl liegt vor Eckpunkt

Wir wandern nun von q aus auf e in die Richtung, in der β_q liegt. Es können zwei verschiedene Fälle auftreten. Entweder wir treffen zuerst auf den Eckpunkt von e (siehe Abb. 39) oder wir treffen zuerst auf einen Punkt, der von p aus in einer Richtung $m\rho$ liegt (siehe Abb. 40).

In beiden Fällen nennen wir den ersten so gefundenen Punkt q' und betrachten das Dreieck pqq' . In q liegt nach Konstruktion der Winkel β_q , und den Winkel in p nennen wir ν . Da nach Konstruktion $\nu \leq \rho$ ist, passt diese Benennung zu unseren Vorüberlegungen ((62), (63)). Wenden wir den Sinussatz auf das Dreieck pqq' an, so ergibt sich:

$$(65) \quad \frac{|\overline{pq'}|}{\sin \beta_q} = \frac{|\overline{qq'}|}{\sin \nu} = \frac{|\overline{pq}|}{\sin(\beta_q + \nu)}$$

Und da sonst $\pi(p, q') \oplus \overline{q'q}$ ein kürzerer Weg wäre als $\pi(p, q)$ gilt:

$$(66) \quad |\pi(p, q)| \leq |\pi(p, q')| + |\overline{q'q}|$$

Insgesamt ergibt sich:

$$(67) \quad \begin{aligned} u_P(p, q') &= \frac{|\pi(p, q')|}{|\overline{pq'}|} \\ &\stackrel{(66)}{\geq} \frac{|\pi(p, q)| - |\overline{q'q}|}{|\overline{pq'}|} \\ &\stackrel{(65)}{=} \frac{\sin(\beta_q + \nu)}{\sin \beta_q} \frac{|\pi(p, q)|}{|\overline{pq}|} - \frac{\sin \nu}{\sin \beta_q} \\ &\stackrel{(62), (63), (64)}{\geq} \left(1 - \frac{\eta}{2}\right) u_P(p, q) - \frac{\sin \rho}{\sin \beta} \\ &\stackrel{(59)}{\geq} \left(1 - \frac{\eta}{2}\right) u_P^{\max} - \frac{\eta}{2} \\ &\stackrel{u_P^{\max} \geq 1}{\geq} (1 - \eta) u_P^{\max} \end{aligned}$$

Ist nun q' ein Eckpunkt, so gilt für das Eckpunktpaar (p_2, q_2) aus Schritt 3 von Algorithmus 5:

$$u_P(p_2, q_2) \stackrel{\text{L. 8.2}}{\geq} \frac{1}{1 + \eta} u_P^E \stackrel{\max}{\geq} \frac{1}{1 + \eta} u_P(p, q') \stackrel{(67)}{\geq} \frac{1 - \eta}{1 + \eta} u_P^{\max} \stackrel{(57)}{\geq} \frac{1}{1 + \varepsilon} u_P^{\max}$$

Und die Ausgabe des Algorithmus ist korrekt.

Ist hingegen q' ein Punkt, der von p aus in Richtung $m\rho$ liegt, dann müssen wir wieder zwei Fälle unterscheiden. Sind p, q' gegenseitig sichtbar oder gegenseitig sichtbar in P^C , dann wird das Paar in Schritt 1 unseres Approximationsalgorithmus untersucht, und es gilt für das in Schritt 2 gefundene Punktpaar (p_1, q_1) :

$$u_P(p_1, q_1) \geq u_P(p, q') \stackrel{(67)}{\geq} (1 - \eta) u_P^{\max} \geq \frac{1 - \eta}{(1 + \eta)^2} u_P^{\max} \stackrel{(57)}{\geq} \frac{1}{1 + \varepsilon} u_P^{\max}$$

Auch dann ist also die Ausgabe des Algorithmus korrekt.

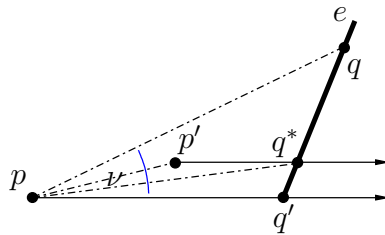


Abbildung 41: ein weiterer Eckpunkt p' im Dreieck pqq'

Leider kann auch der Fall eintreten, dass, obwohl p, q ja in P^C gegenseitig sichtbar sind, sich p und q' nicht in P^C sehen können. Die folgende, im Vergleich zu [EBKLL01] kompliziertere Argumentation und die strikteren Bedingungen an η und ρ sind notwendig, weil wir im Algorithmus 5 bei Schritt 1 nicht auf der Kante maximieren.

Ist (p, q') nicht in P^C gegenseitig sichtbar, dann gibt es einen weiteren Polygon-Eckpunkt p' im Dreieck pqq' . Und man kann p' durch iterierte Anwendung dieser Argumentation

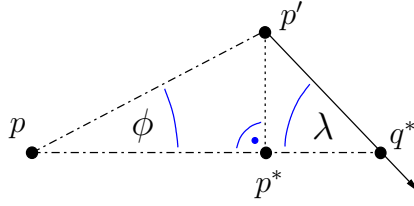


Abbildung 42: die Winkel im Dreieck $pp'q^*$

schließlich so wählen, dass er in Richtung $m\rho$ in P^C bis zur Kante e sehen kann. Den dort gesehenen Punkt nennen wir q^* (siehe Abb. 41).

Betrachte nun das Dreieck $pp'q^*$ (vgl. Abb. 42). Es fällt auf, dass sowohl der Winkel ϕ in p als auch, weil $\overline{p'q^*}$ nach Konstruktion parallel zu $\overline{pq'}$ verläuft, der Winkel λ in q^* nicht größer als ν , also auch nicht größer als ρ , sein können. Daher folgt, wenn man die senkrechte Projektion von p' auf $\overline{pq^*}$ mit p^* bezeichnet:

$$(68) \quad |\overline{pp'}| + |\overline{p'q^*}| = \frac{|\overline{pp^*}|}{\cos \phi} + \frac{|\overline{p^*q^*}|}{\cos \lambda} \leq \frac{1}{\cos \rho} |\overline{pq^*}| \stackrel{(59)}{\leq} (1 + \eta) |\overline{pq^*}|$$

Die Überlegungen im Dreieck pqq' , die zu (67) führen, lassen sich problemlos auf das Dreieck ppq^* übertragen, so dass (67) auch für q^* statt q' gilt. Insgesamt folgt:

$$(69) \quad \begin{aligned} (1 - \eta)u_P^{\max} &\stackrel{(67) \text{ für } q^*}{\leq} u_P(p, q^*) = \frac{|\pi(p, q^*)|}{|\overline{pq^*}|} \\ &\stackrel{\Delta\text{-Ungl. für } |\pi|, (68)}{\leq} (1 + \eta) \frac{|\pi(p, p')| + |\pi(p', q^*)|}{|\overline{pp'}| + |\overline{p'q^*}|} \\ &\leq (1 + \eta) \max(u_P(p, p'), u_P(p', q^*)) \end{aligned}$$

Ist dabei das rechte Maximum gleich $u_P(p, p')$, so können wir, weil (p, p') ein Eckpunkt-paar ist, folgern:

$$u_P(p_2, q_2) \stackrel{\text{L. 8.2}}{\geq} \frac{1}{1 + \eta} u_P^E \max \geq \frac{1}{1 + \eta} u_P(p, p') \stackrel{(69)}{\geq} \frac{1 - \eta}{(1 + \eta)^2} u_P^{\max} \stackrel{(57)}{\geq} \frac{1}{1 + \varepsilon} u_P^{\max}$$

Und die Ausgabe des Algorithmus ist korrekt. Ist andererseits das rechte Maximum in (69) gleich $u_P(p', q^*)$, so können wir, weil (p', q^*) als ein in P^C gegenseitig sichtbares Eckpunkt-Randpunkt-Paar in Richtung $m\rho$ in Schritt 1 von Algorithmus 5 untersucht wird, schließen:

$$u_P(p_1, q_1) \geq u_P(p', q^*) \stackrel{(69)}{\geq} \frac{1 - \eta}{1 + \eta} u_P^{\max} \geq \frac{1 - \eta}{(1 + \eta)^2} u_P^{\max} \stackrel{(57)}{\geq} \frac{1}{1 + \varepsilon} u_P^{\max}$$

Auch dann ist also die Ausgabe des Algorithmus korrekt. Endlich haben wir alle Fälle behandelt, und der Beweis der Korrektheit von Algorithmus 5 und damit auch der Beweis von Satz 8.1 sind abgeschlossen. \square

9 Untere Laufzeitschranken

Im Abschnitt 8 wurde ein Approximationsalgorithmus vorgestellt, der zur Ermittlung eines Punktpaares mit annähernd maximalem Umweg eine Laufzeit von $O(n \log n)$ benötigt. Das Verfahren aus Abschnitt 7 ein Paar zu finden, das exakt den maximalen Umweg realisiert, benötigt nach Satz 7.1 hingegen $O(n^2)$. In diesem Abschnitt soll nun untersucht werden, welcher Zeitaufwand bei der Lösung der Probleme unumgänglich ist.

Wir beweisen Laufzeitschranken für das auf Eckpunktpaare eingeschränkte Problem. Dabei behandeln wir in Unterabschnitt 9.1 zunächst die Suche nach einem Eckpunktpaar im Polygon, das exakt den maximalen Umweg unter solchen Paaren realisiert. Anschließend übertragen wir in Unterabschnitt 9.2 das Verfahren auf den Kettenfall und finden auch hier eine untere Laufzeitschranke für die Suche nach einem exakten Eckpunktmaximum. Die untere Laufzeitschranke lässt sich sowohl im Kettenfall (siehe 9.3) als auch im Polygonfall (siehe 9.4) auf bestimmte Approximationsalgorithmen ausdehnen. Am Ende (siehe 9.5) ordnen wir die untersuchten Probleme in einen größeren Zusammenhang ein, indem wir sie mit ähnlichen Problemen vergleichen.

9.1 Exaktes Eckpunktmaximum in Polygonen

Schon bei der Aufstellung des Approximationsalgorithmus hatten wir in Lemma 8.2 das eingeschränkte Problem betrachtet, bei dem nur Eckpunktpaare untersucht werden. Für die Suche nach einem Eckpunktpaar maximalen Umwegs lässt sich über die Rückführung auf das Problem ELEMENT-UNIQUENESS (siehe [PS90]) eine untere Laufzeitschranke beweisen. Diese bezieht sich auf das für untere Laufzeitschranken wohl am häufigsten benutzte Modell, das *algebraische Entscheidungsbaummodell* (algebraic decision tree model) (entwickelt in [Rei72], [Rab72], [DL79], erklärt z.B. in [PS90] oder [Kle97])

Lemma 9.1 *In einem Polygon P benötigt man zum Ausrechnen des maximalen Eckpunktumweges $u_P^{E \max}$ im schlimmsten Fall eine Laufzeit von $\Omega(n \log n)$ im algebraischen Entscheidungsbaummodell. Dies gilt sogar, wenn man sich auf monotone Polygone beschränkt.*

Natürlich folgt aus diesem Lemma sofort, dass auch für das Auffinden eines Eckpunktpaares (p, q) mit $u_P(p, q) = u_P^{E \max}$ die gleiche Laufzeitschranke gilt. Denn zu gegebenem Punktpaar (p, q) kann man mit dem Algorithmus von Guibas und Hershberger aus [GH87] in Zeit $O(n)$ die Länge des kürzesten Pfades durch das Polygon und damit schließlich auch $u_P(p, q)$ ausrechnen.

Beweis. Die Idee für den folgenden Beweis stammt aus [AERT92]. Dort wird ein ähnliches Vorgehen benutzt, um zu zeigen, dass selbst für n Punkte im zweidimensionalen Raum, die ein monotones oder sternförmiges Polygon bilden, das CLOSEST-PAIR-Problem (siehe [PS90]) eine untere Laufzeitschranke von $\Omega(n \log n)$ hat.

In unserem Fall sei jedoch ein n -Tupel von ganzen Zahlen gegeben. Nach [Yao89] gibt es im algebraischen Entscheidungsbaummodell eine untere Laufzeitschranke von $\Omega(n \log n)$ für das ELEMENT-UNIQUENESS-Problem dieser Zahlen. D.h. jeder Algorithmus, der bestimmt, ob es in dem Tupel mindestens zwei gleiche Zahlen gibt, benötigt im schlimmsten Fall eine Laufzeit von $\Omega(n \log n)$ im algebraischen Entscheidungsbaummodell.

Wir zeigen nun folgendes: Gäbe es einen Algorithmus der für ein beliebiges Polygon im schlimmsten Fall schneller als in $\Omega(n \log n)$ den maximalen Eckpunktumweg berechnen

könnte, dann könnten wir auch das oben beschriebene ELEMENT-UNIQUENESS-Problem in einer schnelleren Laufzeit lösen.

Sei also ein Algorithmus A gegeben, der schneller als in $\Omega(n \log n)$ den maximalen Eckpunktumweg eines Polygons berechnet. Und seien y_1, \dots, y_n die ganzen Zahlen, für die wir das ELEMENT-UNIQUENESS-Problem lösen sollen. Dann bestimmen wir in einer Laufzeit von $O(n)$ das Maximum $\bar{y} := \max_{1 \leq i \leq n} y_i$ und das Minimum $\underline{y} := \min_{1 \leq i \leq n} y_i$.

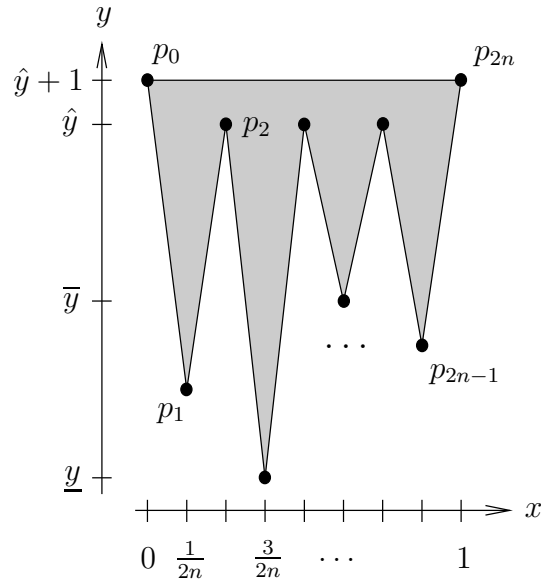


Abbildung 43: aus den y_i konstruiertes kammförmiges Polygon

Anschließend wählen wir eine Zahl \hat{y} , die so groß ist, dass gilt:

$$(70) \quad \hat{y} \geq \frac{\bar{y} \sqrt{1 + \frac{1}{n^2}} - \underline{y} + \frac{1}{2}}{\sqrt{1 + \frac{1}{n^2}} - 1}$$

Mit dieser Zahl konstruieren wir in Zeit $O(n)$ ein kammförmiges, sogar monotones Polygon wie in Abb. 43. Dabei gehen wir wie folgt vor. Aus den gegebenen ganzen Zahlen y_1, \dots, y_n konstruieren wir Eckpunkte durch:

$$(71) \quad p_{2i-1} := \left(\frac{2i-1}{2n}, y_i \right) \text{ für } i \in \{1, \dots, n\}$$

Diese Punkte bilden die Spitzen der Kammzinken. Die Einbuchtungen werden definiert durch:

$$(72) \quad p_{2i} := \left(\frac{2i}{2n}, \hat{y} \right) \text{ für } i \in \{1, \dots, n-1\}$$

Schließlich definieren wir noch, um den Kamm abzuschließen:

$$(73) \quad p_0 := \left(\frac{0}{2n}, \hat{y} + 1 \right) \text{ und } p_{2n} := \left(\frac{2n}{2n}, \hat{y} + 1 \right)$$

Bei der Deutung des maximalen Eckpunktumweges können wir uns auf den Normalfall konzentrieren, bei dem er zwischen zwei Zinkenspitzen, d.h. p_i mit ungeradem i , angenommen

wird. Die genaue Begründung folgt gegen Ende des Beweises. Seien also $a = p_{2j-1}$ und $b = p_{2k-1}$ zwei (verschiedene) Zinkenspitzen. Dann gilt:

$$(74) \quad |\overline{ab}| = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2} = \sqrt{\left(\frac{2j-1}{2n} - \frac{2k-1}{2n}\right)^2 + (y_j - y_k)^2}$$

$$\begin{cases} < \sqrt{1+0} = 1 & \text{falls } y_j = y_k \\ > \sqrt{\frac{1}{n^2} + 1} & \text{falls } y_j \neq y_k \text{ (also } |y_j - y_k| \geq 1) \end{cases}$$

Zum Verständnis der Bemerkung in Klammern muss man sich daran erinnern, dass die y_i ganze Zahlen sind. Für die kürzeste Verbindung durch das Polygon gilt andererseits wegen der Kammform:

$$(75) \quad 2(\hat{y} - \bar{y}) < |\pi(a, b)| < 1 + 2(\hat{y} - \underline{y})$$

Aus (74) und (75) ergibt sich insgesamt:

$$(76) \quad u_P(a, b) \begin{cases} > 2(\hat{y} - \bar{y}) & \text{falls } y_j = y_k \\ < \frac{1+2(\hat{y}-\underline{y})}{\sqrt{\frac{1}{n^2}+1}} & \text{falls } y_j \neq y_k \end{cases}$$

Wegen der Wahl von \hat{y} in (70) liefert dies ein echtes Entscheidungskriterium, denn es gilt:

$$(77) \quad 2(\hat{y} - \bar{y}) \geq \frac{1 + 2(\hat{y} - \underline{y})}{\sqrt{\frac{1}{n^2} + 1}}$$

$$\Leftrightarrow 2 \left(\sqrt{1 + \frac{1}{n^2}} \right) (\hat{y} - \bar{y}) \geq 1 + 2(\hat{y} - \underline{y})$$

$$\Leftrightarrow 2 \left(\sqrt{1 + \frac{1}{n^2}} - 1 \right) \hat{y} \geq 1 + 2 \left(\bar{y} \sqrt{1 + \frac{1}{n^2}} - \underline{y} \right)$$

$$\Leftrightarrow \hat{y} \geq \frac{\bar{y} \sqrt{1 + \frac{1}{n^2}} - \underline{y} + \frac{1}{2}}{\sqrt{1 + \frac{1}{n^2}} - 1} \text{ (siehe (70))}$$

Damit bleibt nur noch zu begründen, warum wir uns auf den Fall, in dem $u_P^{E \max}$ zwischen zwei Zinkenspitzen angenommen wird, beschränken können. Natürlich kann das Maximum nicht zwischen zwei Einbuchtungsecken (p_i mit geradem i) angenommen werden, denn solche Eckpunkte sind (in P) gegenseitig sichtbar.

Um auch gemischte Paare, also Paare von Einbuchtungseckpunkten und Zinkenspitzen, auszuschließen, können wir \hat{y} so groß wählen, dass $\hat{y} - \bar{y} > \sqrt{1 + 1/n^2}$ ist. Schließen wir den trivialen Fall $\bar{y} = \underline{y}$ aus, dann ist das schon durch (70) gewährleistet, was einfache Umformungen zeigen. Ist nun z eine Zinkenspitze und e ein Einbuchtungseckpunkt, dann gilt:

$$(78) \quad u_P(z, e) < \frac{\hat{y} - \underline{y} + 1}{\hat{y} - \bar{y}} \leq \frac{\hat{y} - \underline{y} + 1}{\sqrt{1 + \frac{1}{n^2}}} \leq \frac{1 + 2(\hat{y} - \underline{y})}{\sqrt{\frac{1}{n^2} + 1}} \stackrel{(77)}{\leq} 2(\hat{y} - \bar{y})$$

Daher ändert der Einfluss von gemischten Paaren nichts an unserem durch (76) gegebenen Entscheidungskriterium.

Um nun zu entscheiden, ob unter den gegebenen ganzen Zahlen y_1, \dots, y_n zwei gleiche sind, müssen wir nur den Algorithmus A auf das Kammpolygon anwenden. Er errechnet den

maximalen Eckpunktumweg $u_P^{E \max}$. Ist $u_P^{E \max} > 2(\hat{y} - \bar{y})$, dann gibt es in (y_1, \dots, y_n) zwei gleiche Zahlen, sonst nicht.

Damit ist gezeigt, dass man, falls man den maximalen Eckpunktumweg eines Polygons schneller als in $\Omega(n \log n)$ berechnen könnte, auch das Problem ELEMENT-UNIQUENESS für ganze Zahlen schneller als in $\Omega(n \log n)$ lösen könnte im Widerspruch zu dem einleitend erwähnten Ergebnis aus [Yao89]. Dadurch ist der Beweis von Lemma 9.1 abgeschlossen. \square

Es sei an dieser Stelle noch auf eine mögliche Kritik am benutzten Beweisverfahren eingegangen. Die Ungleichung (70) erzwingt unter Umständen die Wahl eines sehr hohen Wertes für \hat{y} . Denn der Nenner konvergiert für steigende n gegen 0, während der Zähler immer größer als $\bar{y} - \underline{y} + 1/2$ bleibt. Dadurch kann es sein, dass der Algorithmus A , der den maximalen Eckpunktumweg des konstruierten Polygons bestimmt, mit grossen Zahlenwerten umgehen muss.

Im algebraischen Entscheidungsbaummodell wird aber die Größe der auftretenden Zahlen nicht berücksichtigt. Das Ausrechnen von Polynomen einer gewissen Ordnung d wird unabhängig von der Größe der für die Variablen eingesetzten Werte als ein einzelner Rechenschritt angesehen. Man könnte sich nun fragen, ob diese dem Modell inhärente Annahme hier noch gerechtfertigt erscheint. Die Frage ist sicherlich ernst zu nehmen.

Eine ähnliche Problematik entdeckten Kirkpatrick und Reisch in [DK84], als sie es schafften in einem RAM-Modell mit zusätzlichen Instruktionen in linearer Zeit ganze Zahlen zu sortieren. Dieser und darauf folgende andere Ansätze ([FW93], [FW90], [AHNR98]) sind möglich, weil in Zwischenergebnissen sehr hohe Zahlen verwendet werden, ohne dass dies einen Einfluss auf die im Modell nötige Rechenzeit hat.

Um diesen unrealistischen Ergebnissen entgegen zu treten, wurde für Algorithmen, die mit ganzen Zahlen umgehen, das *Word RAM*-Modell entwickelt (siehe [Hag98] und [FW93]), in dem die Bitzahl aller im Algorithmus vorkommender Zahlen auf eine vorgegebene Wortlänge beschränkt ist. Ein entsprechendes Vorgehen bei reellen Zahlen ist uns jedoch nicht bekannt. Dabei würde auch die Schwierigkeit auftreten, dass der zur Speicherung von reellen Zahlen notwendige Speicherplatz nicht nur von der Größe der Zahl, sondern in größerem Ausmaß von der gewünschten Genauigkeit abhängt.

Kommen wir zurück zum hier konstruierten kammförmigen Polygon. \hat{y} steigt proportional zu $\frac{1}{\sqrt{1+\frac{1}{n^2}}-1}$. Durch kanonische Umformungen lässt sich zeigen, dass $\frac{1}{\sqrt{1+\frac{1}{n^2}}-1} \leq 2n^2 + 1$ ist, dass also \hat{y} nur quadratisch in n wächst. Im Abschnitt 9.4 werden wir für diesen Fall sogar einen Reduktionsbeweis finden, bei dem die Zahlengrößen nur linear in n steigen.

Man könnte auch diesen Einfluss mit Sorge betrachten. Allerdings ist es wichtig zu erkennen, dass die Kritik sich nicht auf die Gültigkeit oder den Beweis von Lemma 9.1 bezieht, sondern dass sie eine Schwäche des algebraischen Entscheidungsbaummodells aufdeckt. Eine noch wesentlich praxisferner anmutende Eigenschaft dieses Modells führt Chen in [Che96] anhand eines überraschenden Paradoxons vor. Es beruht im wesentlichen darauf, dass der Test auf Ganzzahligkeit dort nicht durchführbar ist.

Bei aller Kritik sei aber nicht vergessen, dass die Beschränkung auf dieses Modell überhaupt erst die Aufstellung von aussagekräftigen unteren Schranken in der algorithmischen Geometrie ermöglicht hat. Die zusätzliche Berücksichtigung der Größe der Eingabewerte ist ohne Zweifel eine interessante Idee. Sie würde aber den Rahmen der vorliegenden Diplomarbeit sprengen.

9.2 Exaktes Eckpunktmaximum auf polygonalen Ketten

Der Beweis von Lemma 9.1 lässt sich durch eine kleine Modifikation auch auf polygonale Ketten übertragen. Wegen des engen Zusammenhanges zwischen Polygon- und Kettenfall, der sich durch die gesamte Diplomarbeit zieht, sei es gestattet, hier auch kurz auf diese Übertragungsmöglichkeit einzugehen.

Bei einer polygonalen Kette K ist der (relative) Umweg zwischen zwei Punkten p und q , die auf der Kette liegen, analog zum Polygonfall definiert. Man dividiert die Länge des Kettenteiles K_p^q , der p und q verbindet, durch den euklidischen Abstand der beiden Punkte $|\overline{pq}|$. Auch auf polygonalen Ketten kann man das Problem betrachten, den maximalen Umweg zwischen Eckpunktpaaren $u_K^{E \max}$ herauszufinden. In der Literatur werden dabei die Eckpunkte, wie bei allgemeinen Graphen auch, eher Knoten (Vertices) genannt. Für $u_K^{E \max}$ findet man die Begriffe „Stretch Factor“ ([NS00]), „Dilation“ ([PS89]) und „Distortion“ ([LR95]).

Narasimhan und Smid beschreiben in [NS00] ein schnelles Verfahren zur Approximation von $u_K^{E \max}$ im Kettenfall und beweisen eine untere Laufzeitschranke von $\Omega(n \log n)$ für dieses Problem. Allerdings wird in ihrem Reduktionsbeweis eine Kette benutzt, die Selbstschnitte hat, bei der sich sogar viele Kanten teilweise überlagern. Folgendes Lemma beweist die untere Schranke selbst dann, wenn man sich auf monotone polygonale Ketten ohne Selbstschnitte beschränkt. Eine polygonale Kette wird *monoton* genannt, wenn man durch Drehung erreichen kann, dass zu jeder x-Koordinate nur höchstens eine y-Koordinate existiert, mit der (x, y) ein Punkt der Kette ist.

Lemma 9.2 *Auf einer polygonalen Kette K ohne Selbstschnitte benötigt man zum Ausrechnen des maximalen Eckpunktumweges $u_K^{E \max}$ im schlimmsten Fall eine Laufzeit von $\Omega(n \log n)$ im algebraischen Entscheidungsbaummodell. Dies gilt sogar, wenn man sich auf monotone Ketten beschränkt.*

Beweis. Die einfachste Idee zur Übertragung des Beweises von Lemma 9.1 auf den Kettenfall ist, dass man eine Kette konstruiert, indem man einfach im dortigen Kammpolygon (Abb. 43) die oberste Kante $\overline{p_0 p_{2n}}$ wegfällen lässt. So leicht funktioniert die Übertragung des Beweises aber leider nicht.

Denn beim Übergang zur Kette wird bei der Berechnung des Umweges zwischen den Punkten p und q nicht mehr die Länge des kürzesten Weges $\pi(p, q)$ durch das Polygon als Zähler genommen, sondern die Länge des Teilstückes K_p^q der Kette, das die beiden Punkte verbindet. Dadurch würde der maximale Eckpunktumweg bei der so konstruierten Kette in den meisten Fällen zwischen zwei oberen Hilfspunkten angenommen und ein Rückschluss auf die Differenz der y_i wäre nicht mehr möglich.

Eine simple Modifikation jedoch reicht aus, das Problem zu lösen. Wir konstruieren die oberen Hilfspunkte so, dass ihre y-Werte einen Mindestabstand von 1 haben:

$$(79) \quad p_{2i} := \left(\frac{2i}{2n}, \hat{y} + i \right) \text{ für } i \in \{0, \dots, n\}$$

$$(80) \quad p_{2i-1} := \left(\frac{2i-1}{2n}, y_i \right) \text{ für } i \in \{1, \dots, n\}$$

Seien wieder $\bar{y} := \max_{1 \leq i \leq n} y_i$ das Maximum und $\underline{y} := \min_{1 \leq i \leq n} y_i$ das Minimum der ganzen Zahlen y_i .

für obere Eckpunkte:

$$(83) \quad u_K(a, b) < \sqrt{2}n(\hat{y} - \underline{y} + n + \frac{1}{2n})$$

Sind $a = p_{2j-1}$ und $b = p_{2k-1}$ andererseits zwei untere Eckpunkte mit gleichen dazugehörigen ganzen Zahlen $y_j = y_k$, dann ergibt sich mit $m := |j - k|$:

$$(84) \quad u_K(a, b) \geq \frac{2m(\hat{y} - \bar{y})}{m\frac{1}{n}} = 2n(\hat{y} - \bar{y})$$

Um den Beweis analog zu dem von Lemma 9.1 beenden zu können, müssen wir also nur noch zeigen, dass wir \hat{y} so gross wählen können, dass $2n(\hat{y} - \bar{y}) \geq \sqrt{2}n(\hat{y} - \underline{y} + n + \frac{1}{2n})$ gilt. Dies ist tatsächlich möglich:

$$(85) \quad \begin{aligned} 2n(\hat{y} - \bar{y}) &\geq \sqrt{2}n(\hat{y} - \underline{y} + n + \frac{1}{2n}) \\ \Leftrightarrow \hat{y}(\sqrt{2} - 1) &\geq -\underline{y} + n + \frac{1}{2n} + \sqrt{2}\bar{y} \\ \Leftrightarrow \hat{y} &\geq \frac{n + \frac{1}{2n} + \sqrt{2}\bar{y} - \underline{y}}{\sqrt{2} - 1} \end{aligned}$$

Also kann der Beweis wie bei Lemma 9.1 zu Ende geführt werden. \square

\hat{y} wächst hier wegen (85) nur linear in n , was die im Anschluss an den Beweis von Lemma 9.1 dargestellte Sorge wegen der Nichtberücksichtigung der Zahlengrößen im algebraischen Entscheidungsbaummodell etwas entschärft.

9.3 Approximation des Eckpunktmaximums auf Ketten

Während Lemma 9.2 nur eine untere Laufzeitschranke für Algorithmen enthält, die den exakten maximalen Eckpunktumweg $U_K^{E \max}$ berechnen, gilt die von Narasimhan und Smid in [NS00] bewiesene untere Schranke für beliebige Approximationsalgorithmen.

Dies erreichen sie dadurch, dass die benutzte Kette sich selbst schneiden und überdecken darf. Sind nun zwei Zahlen des ELEMENT-UNIQUENESS-Problems, auf dem auch ihr Reduktionsbeweis beruht, gleich, dann liegen zwei verschiedene Eckpunkte der konstruierten Kette übereinander. Der Umweg zwischen diesen Eckpunkten wird deshalb unendlich groß.

Mit unserem Beweisverfahren, dessen Vorteil es gerade ist, auch für polygonale Ketten ohne Selbstschnitte zu gelten, ist daher ein solch allgemeines Ergebnis nicht zu erwarten. Allerdings können wir zeigen, dass sich die untere Laufzeitschranke auf solche Approximationsalgorithmen übertragen lässt, die hinreichend gut sind.

Lemma 9.3 *Sei A ein Algorithmus, der zu einer polygonalen Kette K eine Approximation x des maximalen Eckpunktumweges $u_K^{E \max}$ berechnet, so dass gilt: $\frac{u_K^{E \max}}{1+\varepsilon} \leq x \leq (1+\delta) u_K^{E \max}$ ($\varepsilon, \delta > 0$). Ist die Qualität der Approximation so gut, dass $(1+\varepsilon)(1+\delta) < \sqrt{2}$ ist, dann benötigt A im schlimmsten Fall eine Laufzeit von $\Omega(n \log n)$ im algebraischen Entscheidungsbaummodell. Dies gilt selbst dann, wenn man sich auf polygonale Ketten ohne Selbstschnitte beschränkt.*

Beweis. Der Beweis funktioniert prinzipiell genauso wie derjenige von Lemma 9.2. Bei der Wahl von \hat{y} (vgl. (85)) müssen nur noch ε und δ berücksichtigt werden.

Seien wieder die ganzen Zahlen y_1, \dots, y_n gegeben. Wie bei Lemma 9.2 konstruieren wir aus ihnen eine treppenförmige polygonale Kette nach (79) und (80), d.h. genau wie in Abb. 44. Einzig und allein die Wahl von \hat{y} ist unterschiedlich. Diesmal soll nämlich gelten:

$$(86) \quad \hat{y} \geq \frac{\sqrt{2\bar{y}} + (1 + \varepsilon)(1 + \delta) \left(n + \frac{1}{2n} - \underline{y} \right)}{\sqrt{2} - (1 + \varepsilon)(1 + \delta)}$$

Wegen der Voraussetzung an ε und δ ist dabei der rechte Ausdruck positiv. Und die Voraussetzung an \hat{y} ist hier schärfer als diejenige im Beweis von Lemma 9.2, d.h. aus (86) folgt (85).

Seien $\bar{C} := 2n(\hat{y} - \bar{y})$ und $\underline{C} := \sqrt{2n}(\hat{y} - \underline{y} + n + \frac{1}{2n})$ die im Beweis von Lemma 9.2 auftretenden Schwellenwerte. Dort (vgl. (83) und (84)) hatten wir gezeigt, dass bei Annahme von (85) gilt:

$$(87) \quad u_K^{E \max} \geq \bar{C} \Leftrightarrow \exists j \neq k : y_j = y_k \Leftrightarrow u_K^{E \max} \geq \underline{C}$$

Dies gilt also erst recht im jetzt vorliegenden Fall bei Wahl von \hat{y} nach (86).

Um auch aus dem approximierten Wert x Rückschlüsse ziehen zu können, ist folgende Auswirkung unserer Wahl von \hat{y} entscheidend:

$$(88) \quad \begin{aligned} \hat{y} &\geq \frac{\sqrt{2\bar{y}} + (1 + \varepsilon)(1 + \delta) \left(n + \frac{1}{2n} - \underline{y} \right)}{\sqrt{2} - (1 + \varepsilon)(1 + \delta)} \quad (\text{siehe (86)}) \\ \Leftrightarrow^* \quad \hat{y}(\sqrt{2} - (1 + \varepsilon)(1 + \delta)) &\geq (1 + \varepsilon)(1 + \delta)(-\underline{y} + n + \frac{1}{2n}) + \sqrt{2\bar{y}} \\ \Leftrightarrow \quad \sqrt{2}(\hat{y} - \bar{y}) &\geq (1 + \varepsilon)(1 + \delta)(\hat{y} - \underline{y} + n + \frac{1}{2n}) \\ \Leftrightarrow \quad 2n(\hat{y} - \bar{y}) &\geq (1 + \varepsilon)(1 + \delta)\sqrt{2n}(\hat{y} - \underline{y} + n + \frac{1}{2n}) \\ \Leftrightarrow \quad \bar{C} &\geq (1 + \varepsilon)(1 + \delta)\underline{C} \end{aligned}$$

Dabei ist bei der mit * markierten Äquivalenz entscheidend, dass der Approximationsalgorithmus das Gütekriterium $(1 + \varepsilon)(1 + \delta) < \sqrt{2}$ erfüllt.

Lassen wir nun den Approximationsalgorithmus A auf der in $O(n)$ konstruierten polygonalen Kette laufen, dann erhalten wir nach Voraussetzung einen Ausgabewert x mit:

$$(89) \quad \frac{u_K^{E \max}}{1 + \varepsilon} \leq x \leq (1 + \delta) u_K^{E \max}$$

Mit x können wir dann wegen folgender Implikationen entscheiden, ob es unter den ganzen Zahlen y_1, \dots, y_n ein gleiches Paar gibt:

$$\begin{aligned} x \geq \frac{\bar{C}}{1 + \varepsilon} &\stackrel{(89)}{\Rightarrow} u_K^{E \max} \geq \frac{\bar{C}}{(1 + \varepsilon)(1 + \delta)} \stackrel{(88)}{\Rightarrow} u_K^{E \max} \geq \underline{C} \stackrel{(87)}{\Leftrightarrow} \exists j \neq k : y_j = y_k \\ x < \frac{\bar{C}}{1 + \varepsilon} &\stackrel{(89)}{\Rightarrow} u_K^{E \max} < \bar{C} \stackrel{(87)}{\Leftrightarrow} \forall j \neq k : y_j \neq y_k \end{aligned}$$

Damit ist gezeigt: Würde A schneller als in $\Omega(n \log n)$ laufen, dann könnten wir auch das ELEMENT-UNIQUENESS-Problem von ganzen Zahlen schneller als in $\Omega(n \log n)$ lösen, was wieder dem Ergebnis aus [Yao89] widerspräche. \square

9.4 Approximation des Eckpunktmaximums in Polygonen

Analog zum Vorgehen bei Ketten könnte man versuchen die untere Schranke von Abschnitt 9.1 auf Approximationsalgorithmen für Polygone zu übertragen. Unmittelbar funktioniert das leider nicht. Bei dem Versuch ergibt sich, dass man ε abhängig von n wählen müsste.

Daher liegt die Idee nahe, ein Polygon zu konstruieren, dass der Kette von Abb. 44 ähnlich ist. Dazu muss man im Prinzip nur die Linienbreite als das Innere des Polygons ansehen. Tatsächlich funktioniert dieses Verfahren.

Lemma 9.4 Sei A ein Algorithmus, der zu einem einfachen Polygon P eine Approximation x des maximalen Eckpunktumweges $u_P^{E \max}$ berechnet, so dass gilt: $\frac{u_P^{E \max}}{1+\varepsilon} \leq x \leq (1+\delta) u_P^{E \max}$ ($\varepsilon, \delta > 0$). Ist die Qualität der Approximation so gut, dass $(1+\varepsilon)(1+\delta) < \sqrt{2}$ ist, dann benötigt A im schlimmsten Fall eine Laufzeit von $\Omega(n \log n)$ im algebraischen Entscheidungsbaummodell. Dies gilt selbst dann, wenn man sich auf monotone Polygone beschränkt.

Beweis. Der Beweis verläuft vollkommen analog zu demjenigen von Lemma 9.3. Diesmal müssen wir nur zusätzlich die Ausdehnung des Polygons berücksichtigen. Seien wieder $\bar{y} := \max_{1 \leq i \leq n} y_i$ das Maximum und $\underline{y} := \min_{1 \leq i \leq n} y_i$ das Minimum der gegebenen ganzen Zahlen y_1, \dots, y_n . Dann konstruieren wir das zugehörige Polygon wie folgt:

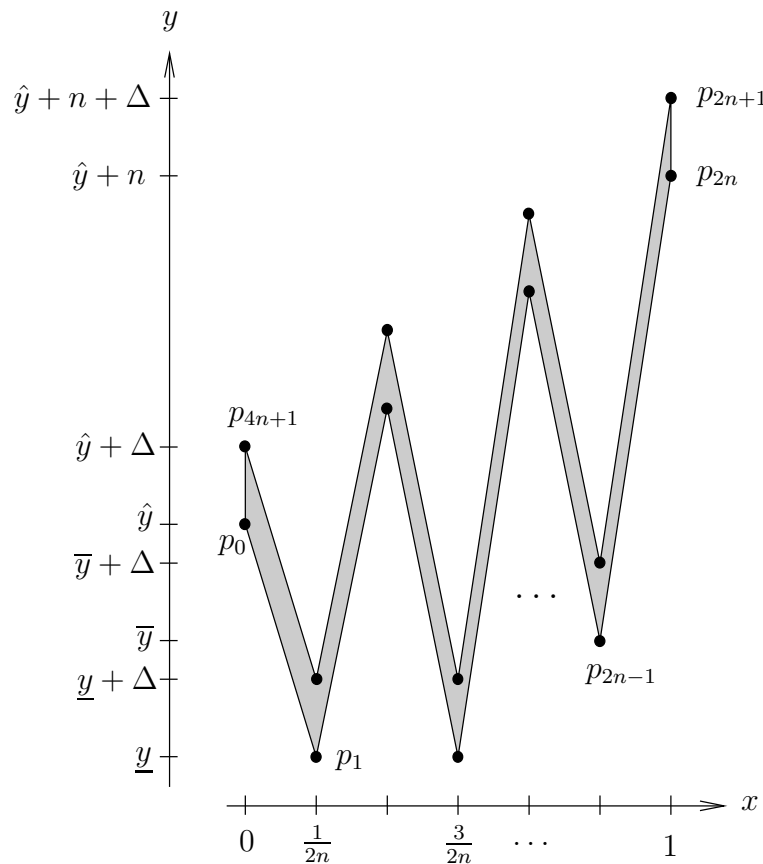


Abbildung 45: treppenförmiges Polygon zum Beweis von Lemma 9.4

$$\begin{aligned}
(90) \quad p_{2i} &:= \left(\frac{2i}{2n}, \hat{y} + i \right) \text{ für } i \in \{0, \dots, n\} \\
p_{2i-1} &:= \left(\frac{2i-1}{2n}, y_i \right) \text{ für } i \in \{1, \dots, n\} \\
p_{4n-2i+1} &:= \left(\frac{2i}{2n}, \hat{y} + i + \Delta \right) \text{ für } i \in \{0, \dots, n\} \\
p_{4n-2i+2} &:= \left(\frac{2i-1}{2n}, y_i + \Delta \right) \text{ für } i \in \{1, \dots, n\}
\end{aligned}$$

Nehmen wir zunächst an, dass $\hat{y} - (\bar{y} + \Delta) \geq \sqrt{1 + (n + \Delta)^2}$ gilt. Diese Annahme folgt, wenn man den trivialen Fall $\bar{y} = y$ ausschließt, wieder direkt aus unserer späteren Wahl von \bar{y} in (96). Dann wird das Eckpunktumwegmaximum wieder entweder zwischen zwei oberen Punkten (d.h. p_{2i} oder $p_{4n-2i+1}$ für $i \in \{0, \dots, n\}$) oder zwischen zwei unteren Punkten (d.h. p_{2i-1} oder $p_{4n-2i+2}$ für $i \in \{1, \dots, n\}$) angenommen.

Denn wenn a ein oberer Punkt ist und b ein unterer Punkt, dann kann man von b aus auf einer an b angrenzenden Polygonkante in die a entgegengesetzte Richtung wandern. Sei \hat{b} der nächste so erreichte obere Punkt, dann ist wegen der von uns angenommenen Ungleichung einerseits $|\overline{ab}| \geq \sqrt{1 + (n + \Delta)^2} \geq |\overline{a\hat{b}}|$ und andererseits ist offensichtlich $\pi(a, b) < \pi(a, \hat{b})$. Also ist insgesamt $u_P(a, b) < u_P(a, \hat{b})$.

Außerdem wird $u_P^{E \max}$ offensichtlich nicht von zueinander gehörigen Punkten angenommen, d.h. von Punktpaaren $(p_{2i}, p_{4n-2i+1})$ oder $(p_{2i-1}, p_{4n-2i+2})$, da diese gegenseitig sichtbar sind.

Ist nun (a, b) ein oberes nicht zueinander gehöriges Punktpaar oder ein unteres nicht zueinander gehöriges Punktpaar mit ungleichen zugehörigen ganzen Zahlen. Dann gilt (analog zu den Beweisen der Lemmata 9.1 bis 9.3):

$$(91) \quad u_P(a, b) < \frac{2m(\hat{y} + n + \Delta - \underline{y} + \frac{1}{n})}{\sqrt{(\frac{m}{n})^2 + (1 - \Delta)^2}} \leq \frac{2n(\hat{y} + n + \Delta - \underline{y} + \frac{1}{n})}{\sqrt{1 + (1 - \Delta)^2}} =: \underline{C}$$

Dabei ist m wieder der Indexabstand der Punkte. Die rechte Ungleichung ist damit zu beweisen, dass für $0 \leq x \leq 1$ und kleine Δ die Ungleichung $\frac{x}{\sqrt{x^2 + (1 - \Delta)^2}} \leq \frac{1}{\sqrt{1 + (1 - \Delta)^2}}$ gilt, was elementare Umformungen zeigen. Damit ist auch schon eine Aussage über das Eckpunktumwegmaximum bewiesen:

$$(92) \quad (\forall j \neq k : y_j \neq y_k) \Rightarrow u_P^{E \max} < \underline{C}$$

Gibt es andererseits ein gleiches Zahlenpaar $y_j = y_k$ für $j \neq k$, dann gilt:

$$(93) \quad u_P^{E \max} \geq u_P(p_{2j-1}, p_{2k-1}) \geq \frac{2m(\hat{y} - (\bar{y} + \Delta))}{m \frac{1}{n}} = 2n(\hat{y} - \bar{y} - \Delta) =: \overline{C}$$

Können wir nun $\overline{C} \geq \underline{C}$ garantieren, dann folgt mit (92) und (93) insgesamt:

$$(94) \quad u_P^{E \max} < \underline{C} \Leftrightarrow (\forall j \neq k : y_j \neq y_k) \Leftrightarrow u_P^{E \max} < \overline{C}$$

Können wir sogar $\overline{C} \geq (1 + \varepsilon)(1 + \delta)\underline{C}$ erreichen, dann gilt für den Approximationswert x mit $\frac{u_P^{E \max}}{1 + \varepsilon} \leq x \leq (1 + \delta)u_P^{E \max}$ (analog zur Herleitung des Entscheidungskriteriums auf

S.73):

$$\begin{aligned} x \geq \frac{\bar{C}}{1+\varepsilon} &\Rightarrow u_P^{E \max} \geq \frac{\bar{C}}{(1+\varepsilon)(1+\delta)} \Rightarrow u_P^{E \max} \geq \underline{C} \stackrel{(94)}{\Leftrightarrow} \exists j \neq k : y_j = y_k \\ x < \frac{\bar{C}}{1+\varepsilon} &\Rightarrow u_P^{E \max} < \bar{C} \stackrel{(94)}{\Leftrightarrow} \forall j \neq k : y_j \neq y_k \end{aligned}$$

Wir könnten also wieder die bekannte Reduktion auf das Problem ELEMENT-UNIQUENESS von ganzen Zahlen erreichen, wenn wir nur $\bar{C} \geq (1+\varepsilon)(1+\delta)\underline{C}$ garantieren können. Das gelingt durch eine passende Wahl von Δ und \hat{y} . Wähle zunächst $\Delta > 0$ so klein, dass gilt:

$$(95) \quad \sqrt{1+(1-\Delta)^2} > (1+\varepsilon)(1+\delta)$$

Dies ist immer möglich, da die linke Seite für kleine Δ gegen $\sqrt{2}$ geht und die rechte Seite nach Voraussetzung kleiner als $\sqrt{2}$ ist. Anschließend gilt dann folgende Umformung:

$$\begin{aligned} &\bar{C} \geq (1+\varepsilon)(1+\delta)\underline{C} \\ \Leftrightarrow &2n(\hat{y} - \bar{y} - \Delta) \geq (1+\varepsilon)(1+\delta) \frac{2n(\hat{y} + n + \Delta - \underline{y} + \frac{1}{n})}{\sqrt{1+(1-\Delta)^2}} \\ \Leftrightarrow &\sqrt{1+(1-\Delta)^2}(\hat{y} - \bar{y} - \Delta) \geq (1+\varepsilon)(1+\delta)(\hat{y} + n + \Delta - \underline{y} + \frac{1}{n}) \\ \Leftrightarrow &\hat{y} \left(\sqrt{1+(1-\Delta)^2} - (1+\varepsilon)(1+\delta) \right) \geq \\ &(1+\varepsilon)(1+\delta)(n + \Delta - \underline{y} + \frac{1}{n}) + \sqrt{1+(1-\Delta)^2}(\bar{y} + \Delta) \\ (96) \quad \stackrel{*}{\Leftrightarrow} &\hat{y} \geq \frac{(1+\varepsilon)(1+\delta)(n + \Delta - \underline{y} + \frac{1}{n}) + \sqrt{1+(1-\Delta)^2}(\bar{y} + \Delta)}{\sqrt{1+(1-\Delta)^2} - (1+\varepsilon)(1+\delta)} \end{aligned}$$

Die Äquivalenz an der mit * markierten Stelle gilt wegen unserer Wahl von Δ in (95). Damit können wir also \hat{y} entsprechend wählen und der Beweis ist abgeschlossen. \square

Das soeben bewiesene Lemma 9.4 zeigt, wenn wir $\delta := 0$ und $\varepsilon := \eta$ betrachten, und wenn $(1+\eta) < \sqrt{2}$ ist, dass der im Beweis von Lemma 8.1 auf S.60 benutzte Algorithmus 4 zum Approximieren von $u_P^{E \max}$ eine optimale Laufzeit hat.

Außerdem beinhaltet Lemma 9.4 natürlich mit $\varepsilon = \delta = 0$ die Aussage von Lemma 9.1. Dabei ist der Beweis sogar insofern verbessert, als dass \hat{y} jetzt nur noch linear von n abhängt (vgl. (96)). Die Kritik, dass bei der Reduktion die auftretenden höheren Zahlenwerte bei der Laufzeitanalyse nicht berücksichtigt werden, hat also etwas weniger Gewicht.

Leider ist eine einfache Verallgemeinerung des vorgestellten Lösungsansatzes auf das Problem, ein allgemeines Punktpaar maximalen Umwegs zu finden, nicht möglich. Versucht man nämlich die Figuren der Eckpunktpaarprobleme (treppenförmige Kette/treppenförmiges Polygon) so zu präparieren, dass das generelle Umwegmaximum bei Eckpunktpaaren angenommen wird, dann muss man die unteren Spitzen ausbeulen und weit voneinander entfernen. Das aber macht die gewohnte Rückführung auf das ELEMENT-UNIQUENESS-Problem zunichte. Deshalb bleibt die Frage nach einer unteren Schranke für das allgemeine Problem der Suche nach einem Umwegmaximum an dieser Stelle unbeantwortet.

Bei der Lösung des allgemeinen Problems lässt sich sowohl im Kettenfall ([EBKLL01]) als auch im Polygonfall (siehe Lemma 3.1 auf S.10) ausnutzen, dass umwegmaximale Punktpaare gegenseitig sichtbar (bzw. in P^C gegenseitig sichtbar) sein müssen. Dies spricht für die Einschätzung, dass das allgemeine Problem eventuell leichter zu lösen ist. Dementsprechend ist es schwerer, untere Laufzeitschranken zu zeigen.

9.5 Klassifikation der Problemstellung

Die vier Probleme, für die wir in den vorangegangenen Abschnitten untere Laufzeitschranken aufstellen konnten, sollen zum Abschluss kurz miteinander und mit anderen verwandten Problemen verglichen werden. Abbildung 46 zeigt im Überblick die Ergebnisse dieses Vergleiches, in dem wiederum nur Eckpunkt-paar-Probleme behandelt werden. Die Pfeile zeigen dabei von leichteren zu schwereren Problemen. Untere Laufzeitschranken pflanzen sich also entlang der Pfeile nach oben fort, während obere Laufzeitschranken ihre Gültigkeit nach unten hin weitergeben.

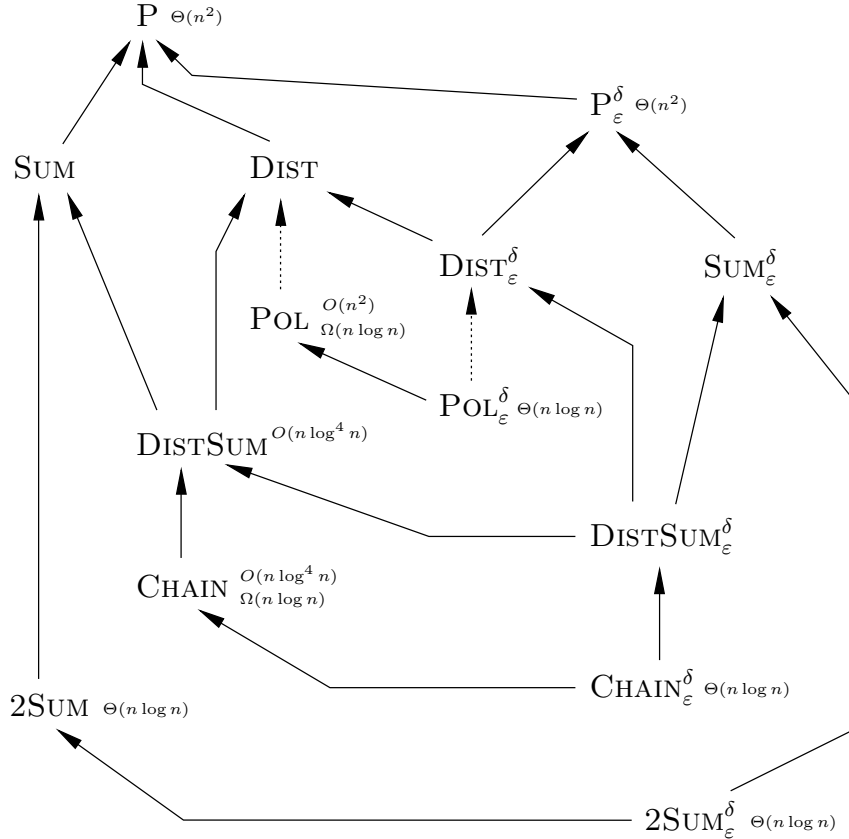


Abbildung 46: verwandte Probleme und deren Beziehungen

Alle betrachteten Probleme lassen sich unter das folgende Grundproblem P einordnen. Dabei sind zwei Matrizen $\Pi, D \in \mathbb{R}^{n \times n}$ gegeben. Gesucht sind nun zwei Indizes $k, l \in \{1, \dots, n\}$ mit:

$$\frac{\Pi_{k,l}}{D_{k,l}} = \max_{k',l'} \frac{\Pi_{k',l'}}{D_{k',l'}}$$

Offenbar handelt es sich bei diesem allgemeinen Problem um eine Maximumsuche in den n^2 Zahlenwerten $\Pi_{k',l'}/D_{k',l'}$. Daher liegt die Laufzeit in $\Theta(n^2)$.

Beim zugehörigen Approximationsproblem P_ϵ^δ wird nicht das direkte Maximum gesucht sondern ein angenäherter Wert. Für die gefundenen Indizes k und l soll gelten:

$$(1 + \delta) \max_{k',l'} \frac{\Pi_{k',l'}}{D_{k',l'}} \geq \frac{\Pi_{k,l}}{D_{k,l}} \geq \frac{1}{1 + \epsilon} \max_{k',l'} \frac{\Pi_{k',l'}}{D_{k',l'}}$$

Analog definieren wir auch zu allen weiteren auftretenden Problemen zugehörige Approximationsprobleme. Natürlich muss man auch bei P_ε^δ alle gegebenen n^2 Zahlenwerte untersuchen, und das naive Vorgehen ergibt die optimale Laufzeit $\Theta(n^2)$.

Die weiteren betrachteten Probleme kann man in Summenprobleme und Distanzprobleme untergliedern. Bei einem Distanzproblem DIST ist die Situation grundsätzlich wie bei P. Die gegebene Matrix D hat aber eine besondere Gestalt. Es sind nämlich Punkte $p_1, \dots, p_n \in \mathbb{R}^d$ gegeben, mit denen $D_{k',l'} = |\overline{p_{k'} p_{l'}}|$ gilt.

Bei einem Summenproblem SUM ist die Situation ebenfalls wie bei P. Zusätzlich sind Zahlen a_1, \dots, a_{n-1} gegeben, mit denen $\Pi_{k',l'} = \sum_{i=k'}^{l'-1} a_i$ gilt.

Ein Summenproblem, das gleichzeitig ein Distanzproblem ist, nennen wir DISTSUM. Das DISTSUM-Problem wollen wir weiterhin dadurch einschränken, dass die Punkte p_i aus der Ebene \mathbb{R}^2 stammen sollen. Wir legen also $d = 2$ fest. Nur durch diese Festlegung funktioniert die Übertragung der oberen Laufzeitschranke von $O(n \log^4 n)$, die wir später erläutern werden.

Das Problem der Suche nach einem Eckpunktpaar mit maximalem Umweg auf polygonalen Ketten, das wir in den Abschnitten 9.2 und 9.3 untersucht hatten, ist offensichtlich ein Spezialfall von DISTSUM, bei dem die Zahlenwerte durch $a_i = |\pi(p_i, p_{i+1})|$ gegeben sind. Wir bezeichnen diesen Fall mit CHAIN.

In Abbildung 46 sind die in den vorangegangenen Abschnitten entwickelten unteren Schranken von $\Omega(n \log n)$ für das Problem CHAIN und das zugehörige Approximationsproblem $\text{CHAIN}_\varepsilon^\delta$ angegeben.

Obere Laufzeitschranken lassen sich aus der Literatur entnehmen. So wird in [EBKLL01] als Teilproblem auch die Suche nach einem approximierten Eckpunktmaximum (siehe dort Theorem 1) in einer Laufzeit von $O(n \log n)$ gelöst. Also ist $O(n \log n)$ eine obere Schranke für das Problem $\text{CHAIN}_\varepsilon^\delta$.

In [LMS02] wird ein randomisierter Algorithmus zur Lösung des allgemeinen Falles auf polygonalen Ketten in $O(n \log^2 n)$ entwickelt. Kurz angedeutet wird auch eine analoge Lösungsmöglichkeit des Eckpunktfalles in einer erwarteten Laufzeit von $O(n \log^2 n)$. Diese ist nicht in Abb. 46 erwähnt, weil wir dort nur obere Laufzeitschranken von deterministischen Algorithmen behandeln.

In [AKKS02] wird in einer Bemerkung ein solcher deterministischer Algorithmus für den allgemeinen Fall auf polygonalen Ketten erwähnt, der durch Anwendung der Parametric Searching-Technik von Megiddo ([Meg83]) konstruiert wird und eine Laufzeit von $O(n \log^4 n)$ hat. Die Lösung des Eckpunktproblems kann analog erfolgen. Das als Teilschritt gelöste Entscheidungsproblem, für gegebenes κ festzustellen, ob $u_K^E \text{max} > \kappa$ ist, wird sogar einfacher. Man muss nur noch untersuchen, ob jeder der dort konstruierten Kegel einen Anteil an deren unteren Kontur hat. So kommen wir auf die in Abb. 46 angegebene obere Laufzeitschranke von $O(n \log^4 n)$ für das Problem CHAIN.

Interessanterweise lassen sich sowohl der randomisierte Algorithmus aus [LMS02] als auch der deterministische Algorithmus aus [AKKS02] auf das Problem DISTSUM verallgemeinern. Bilde dazu für den randomisierten Algorithmus die in [LMS02] vorgestellten Kegel (Cones) mit Kegelspitzen, die aus den Punkten p_i und der zusätzlichen Koordinate $\sum_{j=1}^{i-1} a_j$ gebildet werden. Im deterministischen Algorithmus muss das in [AKKS02] für jeden Kettenpunkt p in Bezug auf einen Knoten p_m definierte Gewicht $\omega(p) = |\pi(p, p_m)| / \kappa$ nur noch allgemein für jede Zahl $a_{k'} \in \{a_1, \dots, a_{n-1}\}$ definiert werden. Setze dazu $\omega(a_{k'}) := \sum_{i=1}^{k'-1} a_i / \kappa$. Außerdem muss man bei den dort benutzten Kegeln ein Vorzeichen ändern. Man definiert $C_{p_{k'}}(x) := |\overline{p_{k'} x}| + \omega(a_{k'})$. Ansonsten können die beiden Algorithmen unverändert übernommen werden.

Auch der Approximationsalgorithmus des Kettenfalles aus [EBKLL01] ist direkt auf den Fall DISTSUM übertragbar. Allerdings funktioniert dies nur, wenn die Zahlen a_i nicht-negativ sind. Deshalb ist die betreffende obere Schranke von $O(n \log n)$ in Abbildung 46 nicht angegeben.

Das analoge Problem auf Polygonen, das wir mit POL abkürzen, ist kein Spezialfall von DISTSUM. Die Länge des kürzesten Weges $\pi(p_{k'}, p_{l'})$ lässt sich nämlich in der Regel nicht als eine so einfache Summe darstellen. Immerhin ist aber auch POL ein Spezialfall von DIST. Zusätzlich ist $\Pi_{k', l'} = |\pi(p_{k'}, p_{l'})|$ festgelegt. Allerdings ist bei dieser Einordnung noch etwas zu beachten. Wir hatten die bisherigen Probleme mit der impliziten Annahme formuliert, dass man die Werte der Matrix Π in konstanter Zeit abfragen kann. Im Fall POL benötigt man aber zur Ermittlung von $\Pi_{k', l'} = |\pi(p_{k'}, p_{l'})|$ nach einer linearen Vorbereitungszeit eine Laufzeit von $O(\log n)$ (siehe [GH87]). Aufgrund dieser Diskrepanz ist in Abbildung 46 der entsprechende Pfeil nur gepunktet gezeichnet.

Das Approximationsproblem $\text{POL}_\epsilon^\delta$ kann nach Lemma 8.2 mit Algorithmus 4 in einer Laufzeit von $O(n \log n)$ gelöst werden. Dass dort noch $\delta = 0$ festgelegt ist, ist eine Erschwerung. Die untere Schranke gilt also auch für Fälle mit größerem δ -Wert. Die Bestimmung des exakten Eckpunkt-paar-Maximums kann analog zu unserem Vorgehen in Kapitel 7 mindestens in $O(n^2)$ gelöst werden. Außerdem sind in Abbildung 46 noch die unteren Schranken aus 9.1 und 9.4 eingetragen.

Schließlich ist in der Auflistung auch das einfachste Problem 2SUM enthalten. Dabei handelt es sich um einen Spezialfall von SUM, in dem auch die Matrix D sich aus einer Summe berechnet. Mit gegebenen Zahlen $b_1, \dots, b_{n-1} \in \mathbb{R}$ gilt:

$$D_{k', l'} = \sum_{i=k'}^{l'-1} b_i$$

Eine $O(n \log n)$ -Lösung dieses Problems ist wie folgt möglich. Man bildet zunächst Punkte p_1, \dots, p_{n+1} durch:

$$(97) \quad p_i := \left(\sum_{j=1}^{i-1} b_j, \sum_{j=1}^{i-1} a_j \right)$$

Für diese Punkte löst man den leichtesten denkbaren Fall von SLOPE SELECTION. D.h. man sucht die beiden Punkte, für die die Steigung der durch diese Punkte verlaufenden Geraden maximal ist. In [KS93] und [CSSS89] werden optimale $O(n \log n)$ -Algorithmen dafür entwickelt, Punktpaare zu finden, deren zugehöriger Steigungswert an k -ter Stelle liegt.

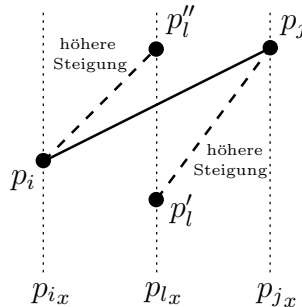


Abbildung 47: Punktpaare maximaler Steigung sind x-Nachbarn

Ist wie hier jedoch $k = 1$, dann funktioniert SLOPE SELECTION noch einfacher. Das Punktpaar mit maximaler Steigung muss nämlich x-kordinatenmäßig benachbart sein, wie

man leicht anhand von Abbildung 47 erkennt. Man kann also einfach in $O(n \log n)$ die Punkte nach ihrer X-Koordinate sortieren, und anschließend die benachbarten Punktpaare in einem linearen Durchlauf nach der maximalen Steigung durchsuchen.

Eine untere Schranke von $\Omega(n \log n)$ lässt sich für den Fall $k = 1$ von SLOPE SELECTION durch folgende Rückführung auf ELEMENT-UNIQUENESS bilden. Man nimmt die gegebenen ganzen Zahlen z_1, \dots, z_n als X-Koordinaten und fügt als Y-Koordinaten jeweils $y_i := i$ hinzu. Zwei ganze Zahlen sind damit genau dann gleich, wenn die zugehörige Steigung unendlich wird. Analog kann man eine untere Schranke für das zugehörige Approximationsproblem beweisen. Dabei führt man das Problem auf ELEMENT-UNIQUENESS für ganze Zahlen zurück. Schließlich gelten die gefundenen unteren Schranken genauso für 2SUM, da die Rückführung aus Gleichung (97) auch in Gegenrichtung funktioniert.

Übrigens entspricht die Dualität von 2SUM und SLOPE SELECTION derjenigen, die Langerman, Morin und Soss in [LMS02] benutzen, um den exakten Algorithmus für polygonale Ketten aufzustellen. Auch dort wird die Suche nach einem maximalen Umweg in eine Suche nach einem Punktpaar maximaler Steigung umgewandelt. Der entscheidende Unterschied ist, dass der in [LMS02] bei der Konstruktion der Kegel betrachtete Fall um eine Dimension komplexer ist. Auch das Verfahren aus [AKKS02] basiert auf einer analogen Idee.

Außerdem sei noch erwähnt, dass 2SUM eng verwandt mit einem weiteren in der Literatur auftauchenden Problem ist. In [EH97] suchen Eppstein und Hirschberg das Maximum von Brüchen von Teilsummen, in denen die durchlaufene Indexmenge nicht zusammenhängend sein muss. Sie erreichen eine lineare Lösung, wenn sie verlangen, dass die betrachteten Zahlen positiv sind, und dass die untersuchten Indexmengen eine vorgegebene Größe haben. Gesucht ist also eine Indexmenge $I \subseteq \{1, \dots, n\}$ mit $|I| = n - k$ und:

$$\frac{\sum_{i \in I} a_i}{\sum_{i \in I} b_i} = \max_{J \subseteq \{1, \dots, n\}, |J|=n-k} \frac{\sum_{j \in J} a_j}{\sum_{j \in J} b_j}$$

Mit der Erwähnung dieses Problems, das wegen der Freiheit der gesuchten Indexmenge leider nicht sinnvoll in Abbildung 46 integriert werden konnte, schließen wir die Betrachtungen zur Klassifikation der Problemstellung und beenden damit auch das Kapitel unterer Laufzeitschranken.

10 Einordnung und Ausblick

In der vorliegenden Diplomarbeit haben wir die Suche nach Punktpaaren maximalen Umwegs innerhalb von Polygonen untersucht. Über die Analyse von Eigenschaften solcher Punktpaare (Eckpunkt-Randpunkt-Paare, keine Schnitte) sind wir zu der Aufstellung eines quadratischen exakten Algorithmus und eines Approximationsalgorithmus mit einer Laufzeit von $O(n \log n)$ gelangt. Abschließend haben wir untere Schranken von $\Omega(n \log n)$ für das Problem des Findens von Eckpunktmaxima und von approximierten Eckpunktmaxima bewiesen.

Bei der Untersuchung der unteren Laufzeitschranken ist vor allem die Frage offen geblieben, ob man das Ergebnis von $\Omega(n \log n)$ auch auf das allgemeine Problem bei Aufhebung der Beschränkung auf Eckpunktpaare ausdehnen kann. Dies gilt sowohl für den Polygon- als auch für den Kettenfall.

Auch scheinen der schnelle Approximationsalgorithmus aus Kapitel 8 und die sub-quadratischen exakten Algorithmen des Kettenfalls aus [LMS02], [AKKS02], [EBKLL01] und [NS00] darauf hinzudeuten, dass möglicherweise auch im Polygonfall ein sub-quadratischer exakter Algorithmus möglich ist. Die direkte Übernahme der Ideen des Kettenfalles ist jedoch deshalb nicht möglich, weil bei den dortigen exakten Algorithmen essentiell die Eigenschaft einfließt, dass bei Ketten die kürzesten Wege immer über einen vorgegebenen Pfad verlaufen. Die Frage nach einem sub-quadratischen exakten Algorithmus für den Polygonfall ist also die zweite grosse offen gebliebene Frage.

Eventuell kann man sich bei der Suche nach einem schnellen exakten Algorithmus zuerst auf

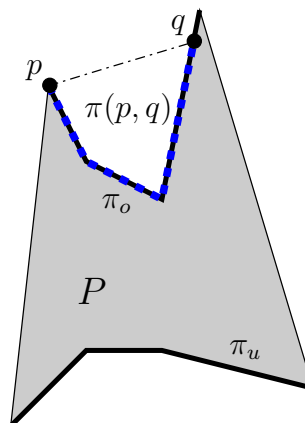


Abbildung 48: monotones Polygon mit nach außen konvexen Randketten

einfache Spezialfälle beschränken. Einen sehr einfachen Spezialfall bilden spezielle monotone Polygone, die aus zwei nach außen konvexen Randketten gebildet werden (siehe Abb. 48). Das Maximum kann dort nur von Punkten, die auf derselben Randkette liegen, angenommen werden.

Weil die kürzesten Wege dann immer ganz über die entsprechende Randkette verlaufen, liegt eine Beschränkung auf den Kettenfall vor. Im Kettenfall können Maxima nur so in Knoten der Kette auftauchen, dass die Verbindung zwischen den beteiligten Punkten im größeren Winkel des Knotens endet. Dies folgt leicht aus Lemma 1 in [EBKLL01].

Für die betrachtete Art von Polygonen hat dies zur Folge, dass Umwegmaxima nur auftreten können, wenn einer der Endpunkte der Randketten beteiligt ist. Da man also für jeden der vier Randkettenendpunkte höchstens alle $O(n)$ zugehörigen Randkettenkanten

untersuchen muss, ist eine Lösung der Maximumsuche in linearer Laufzeit möglich. Vielleicht kann man über derartige Ansätze auch einen sub-quadratischen Algorithmus für monotone Polygone realisieren.

Als letztes grosses Gebiet für weitere Forschung sei die Ausdehnung der Untersuchung auf allgemeine geometrische Graphen genannt.

Noch etwas komplizierter ist der Fall von Polygonen mit Löchern. Daher haben wir uns bei der abschließenden Abbildung für eine etwas unschöne Lösung mit einem einfachen Polygon entschieden. Das Punktpaar mit maximalem Umweg wurde mit Hilfe des schon in der Einleitung erwähnten Java-Programms „Polygonal Detour“ (siehe <http://www.8ung.at/a.gruene/>) berechnet.

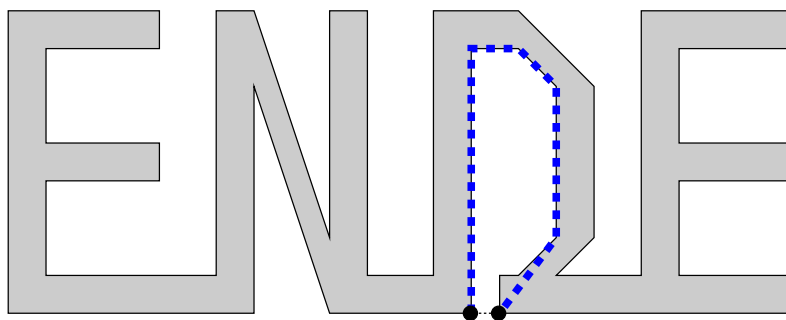


Abbildung 49: spezielles Polygon mit maximalem Umweg

Literatur

- [AERT92] A. Aggarwal, H. Edelsbrunner, P. Raghavan, and P. Tiwari. Optimal time bounds for some proximity problems in the plane. *Inform. Process. Lett.*, 42(1):55–60, 1992.
- [AHNR98] Andersson, Hagerup, Nilsson, and Raman. Sorting in linear time? *J. Comput. Syst. Sci.*, 57:74–93, 1998.
- [AKKS02] Pankaj K. Agarwal, Rolf Klein, Christian Knauer, and Micha Sharir. Computing the detour of polygonal curves. Technical report, Freie Universität Berlin, Fachbereich Mathematik und Informatik, 2002.
- [Bos93] Karl Bosch. *Mathematik-Taschenbuch*. R. Oldenbourg Verlag, München, 4. edition, 1993.
- [BSMM95] I. N. Bronstein, K. A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harry Deutsch, Frankfurt am Main, 2. edition, 1995.
- [Cha91] Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6(5):485–524, 1991.
- [Che96] Jianer Chen. Computational geometry: Methods and applications. Lecture notes, Computer Science Department, Texas A&M University, <http://faculty.cs.tamu.edu/chen/notes/geo.pdf>, 1996.
- [CSSS89] R. Cole, J. Salowe, W. Steiger, and E. Szemerédi. An optimal-time algorithm for slope selection. *SIAM J. Comput.*, 18(4):792–810, 1989.
- [DK84] S. Reisch D. Kirkpatrick. Upper bounds for sorting integers on random access machines. *Theoret. Comput. Sci.*, 28:263–276, 1984.
- [DL79] D. P. Dobkin and R. J. Lipton. On the complexity of computations under varying sets of primitives. *J. Comput. Syst. Sci.*, 18:86–91, 1979.
- [EBKLL01] Annette Ebbers-Baumann, Rolf Klein, Elmar Langetepe, and Andrzej Lingas. A fast algorithm for approximating the detour of a polygonal chain. *ESA 2001 - European Symposium on Algorithms*, 2001.
- [EH97] David Eppstein and Daniel S. Hirschberg. Choosing subsets with maximum weighted average. *J. Algorithms*, 24(1):177–193, 1997.
- [Fis95] Gerd Fischer. *Lineare Algebra*. vieweg, Braunschweig/Wiesbaden, 10. edition, 1995.
- [FW90] F. W. Fredman and D. E. Willard. Trans-dichotomous algorithms for minimum spanning trees and shortest paths. In *Proc. 31st Annu. IEEE Sympos. Found. Comput. Sci.*, pages 719–725, 1990.
- [FW93] F. W. Fredman and D. E. Willard. Surpassing the information theoretic bound with fusion trees. *J. Comput. Syst. Sci.*, 48(3):424–436, 1993.
- [GH87] Leonidas J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 50–63, 1987.
- [GHL⁺87] Leonidas J. Guibas, J. Hershberger, D. Leven, Micha Sharir, and R. E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.

- [Hag98] T. Hagerup. Sorting and searching on the word ram. In *Proceedings of STACS'98*, volume 1373 of *Springer Lecture Notes in Computer Science*, pages 366–398, Paris, 1998.
- [KG92] J. M. Keil and C. A. Gutwin. Classes of graphs which approximate the complete Euclidean graph. *Discrete Comput. Geom.*, 7:13–28, 1992.
- [Kle97] Rolf Klein. *Algorithmische Geometrie*. Addison-Wesley, Bonn, 1997.
- [KS93] M. J. Katz and Micha Sharir. Optimal slope selection via expanders. *Inform. Process. Lett.*, 47:115–122, 1993.
- [Lee83] D. T. Lee. Visibility of a simple polygon. *Comput. Vision Graph. Image Process.*, 22:207–221, 1983.
- [LMS02] Stefan Langerman, Pat Morin, and Michael A. Soss. Computing the maximum detour and spanning ratio of planar paths, trees and cycles. *STACS 2002*, pages 250–261, 2002.
- [LP84] D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14:393–410, 1984.
- [LR95] London Linial and Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15, 1995.
- [Meg83] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM*, 30(4):852–865, 1983.
- [Mit00] Joseph S. B. Mitchell. Geometric shortest paths and network optimization. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.
- [NS00] G. Narasimhan and M. Smid. Approximating the stretch factor of Euclidean graphs. *SIAM J. Comput.*, 30:978–989, 2000.
- [PS89] D. Peleg and A. Schäffer. Graph spanners. *J. Graph Theory*, 13:99–116, 1989.
- [PS90] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 3rd edition, October 1990.
- [Rab72] M. O. Rabin. Proving simultaneous positivity of linear forms. *J. Comput. Syst. Sci.*, 6:639–650, 1972.
- [Rei72] E. M. Reingold. On the optimality of some set algorithms. *J. ACM*, 19:649–659, 1972.
- [Yao89] A. C. C. Yao. Lower bounds for algebraic computation trees with integer inputs. In *Proc. 30th IEEE FOCS*, pages 308–313, 1989.

Index

- $\#K$, *siehe* Kantenzahl
- $\Phi(\cdot)$, *siehe* Trichtergebiet
- P^C , *siehe* Komplement
- $\pi(\cdot, \cdot)$, *siehe* kürzester Weg
- b_F , *siehe* Fußpunkt
- $u_K^{E\max}$, *siehe* maximaler Umweg von Eckpunkt-paaren
- $u_P(\cdot, \cdot)$, *siehe* Umweg
- $u_P^{E\max}$, *siehe* maximaler Umweg von Eckpunkt-paaren
- u_P^{\max} , *siehe* maximaler Umweg
- Überlagerung, 28
- 2SUM, 79

- algebraic decision tree model, *siehe* algebraisches Entscheidungsbaummodell
- algebraisches Entscheidungsbaummodell, 66
 - Kritik, 69
- Algorithmus
 - exakter, 57
 - zum Finden von $\pi(p, q)$, 54
- Applet, 7, 82
- Approximationsalgorithmus, 61
 - für Eckpunktmaximum, 60

- Berührung
 - echte, 28

- CHAIN, 78
- CLOSEST-PAIR, 66
- Cone, *siehe* Kegel
- Cusp, *siehe* Fußpunkt

- Dilation, 70
- DIST, *siehe* Distanzproblem
- Distanzproblem, 78
- Distortion, 70
- DISTSUM, 78
- Dreiecksungleichung
 - für $|\pi(\cdot, \cdot)|$, 10, 30

- Eckpunkt
 - spitzer, 13
- Eckpunkt-Randpunkt-Paar
 - echtes, 16
- Eckpunktmaximum, 60
 - Approximationsalgorithmus, 60
 - untere Laufzeitschranke, 66
- Eindeutigkeit
 - des kürzesten Weges, 11, 30
 - des Umwegmaximums, 15
- ELEMENT-UNIQUENESS, 66, 67, 69, 80

- Entscheidungsbaummodell
 - algebraisches, 66
- Existenz
 - eines Umwegmaximums, 15

- Funnel, *siehe* Trichter
- Fußpunkt, 17, 48

- Graph
 - geometrischer, 5
- Grundproblem, 77

- Hauptminoren-Kriterium, 22
- Hershberger-Algorithmus, 54
- Hourglass, *siehe* Sanduhr

- Java-Programm, 7, 82

- kammförmiges Polygon, 67
- Kantenzahl, 40, 56
- Kegel, 78
- Kette, 5
 - einseitiges Umwegmaximum, 47
 - Kantenzahl, 40, 56
 - monotone, 70
 - polygonale, *siehe* Kette
 - treppenförmige, 70
- Komplement, 9
- Kriterium
 - notwendiges, 46
 - Spezialfall, 47

- Maximum
 - des Umwegs, *siehe* Umwegmaximum
 - lokales, 16

- Oberseite, 17, 48

- P, *siehe* Grundproblem
- Parametric Searching, 78
- POL, 79
- Polygon, 8
 - einfaches, 8
 - kammförmiges, 67
 - konvexes, 9
 - treppenförmiges, 74
- Polygonal Detour, 7, 82
- Programm, 7, 82
- Punkt-paar in Richtung ρ , 59

- Randpunkt
 - echter, 16
- Randpunkt-paar, 16
 - echtes, 16

Randschnittpunkt
 echter, 10

Sanduhr, 16
 geschlossene, 16
 offene, 16

Schlauch, 48

Schnitt
 echter, 28

Semi-Definitheit
 negative, 22

Shortest Path Tree, 54

sichtbar
 gegenseitig, 9
 in P^C gegenseitig, 9

Sichtbarkeitsgraph, 55

SLOPE SELECTION, 79

spärlicher $(1+\eta)$ -Spanngraph, *siehe* sparse
 $(1+\eta)$ -Spanner

sparse $(1+\eta)$ -Spanner, 60

Spezialfall des notwendigen Kriteriums, 47

Stetigkeit
 von $|\pi(\cdot, \cdot)|$, 13
 von $u_P(\cdot, \cdot)$, 13

Stifte-Ansatz, 31

Streckenzug, *siehe* Kette

Stretch Factor, 70

SUM, *siehe* Summenproblem

Summenproblem, 78

 lokales, 16
 notwendiges Kriterium, 46

untere Laufzeitschranke
 Approximation des Eckpunktmaximums,
 74
 Approximation des Eckpunktmaximums
 auf Kette, 72
 Eckpunktmaximum, 66
 Eckpunktmaximum auf Kette, 70

Verlängerung
 einer Kante, 45

Viereck
 duales, 30
 Seiten, 30

Weg
 an Eckpunkt hängender, 18
 kürzester, 8
 kürzester
 Eindeutigkeit, 11, 30

Word RAM, 69

treppenförmige Kette, 70

treppenförmiges Polygon, 74

Triangulierung, 55

Trichter, **16**, 48
 entarteter, 17, 47
 kontinuierlicher, 53

Trichterbereich, 48

Trichtergebiet, 48

Umweg, 5, **8**
 maximaler, 8
 von Eckpunktpaaren, 60, 70
 Maximum, *siehe* Umwegmaximum
 relativer, *siehe* Umweg
 Stetigkeit, 13

Umwegmaxima
 mit gemeinsamem Punkt, 28
 sich schneidende, 28

Umwegmaximum
 der Eckpunktpaare, 60
 Eindeutigkeit, 15
 einseitiges, 46
 auf Kette, 47
 Existenz, 15
 in Eckpunkt-Randpunkt-Paar, 16