Computing Feed-links

Philipp Schneider

24. November 2009

Inhalt

1 Einführung
- Problemstellung und Definitionen

2 Ein Feed-link
- Heuristische und optimale Platzierung eines Feed-Links

3 Hindernisse
- Definitionen und Auswirkungen

4 Mehrere Feed-links
- Berechnung der Streckung und heuristische Platzierung mehrerer Feed-links

Inhalt

1 Einführung
- Problemstellung und Definitionen

2 Ein Feed-link
- Heuristische und optimale Platzierung eines Feed-Links

3 Hindernisse
- Definitionen und Auswirkungen

4 Mehrere Feed-links
- Berechnung der Streckung und heuristische Platzierung mehrerer Feed-links

Problemstellung

Gegeben:

Einfaches Polygon P.
Punkt p im Inneren von P.

Problemstellung

Gegeben:

Einfaches Polygon P.
Punkt p im Inneren von P.

Gesucht:

Punkt $q \in$ Rand P und Feed-link $l = \overline{pq}$ sodass:
die Streckung (Umweg) minimal wird.

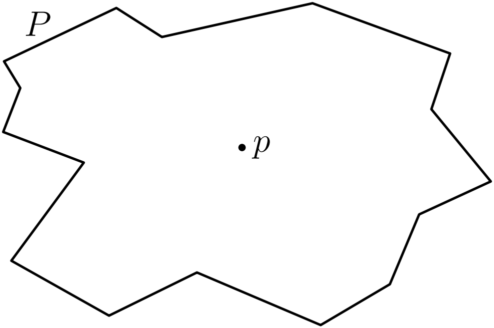
Problemstellung

Gegeben:

Einfaches Polygon P.
Punkt p im Inneren von P.

Gesucht:

Punkt $q \in$ Rand P und Feed-link $l = \overline{pq}$ sodass:
die Streckung (Umweg) minimal wird.



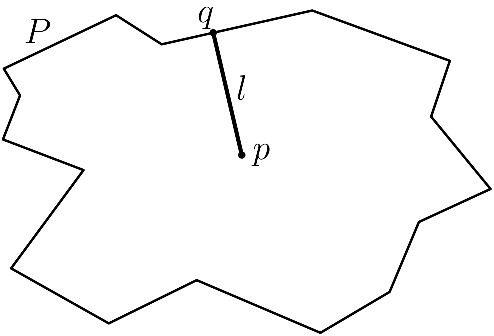
Problemstellung

Gegeben:

Einfaches Polygon P.
Punkt p im Inneren von P.

Gesucht:

Punkt $q \in$ Rand P und Feed-link $l = \overline{pq}$ sodass:
die Streckung (Umweg) minimal wird.



Streckung eines Punktes

- Was ist Umweg?

Streckung eines Punktes

- Was ist Umweg?
- Verhältnis von Distanz im Graph und Euklid-Distanz.

Streckung eines Punktes

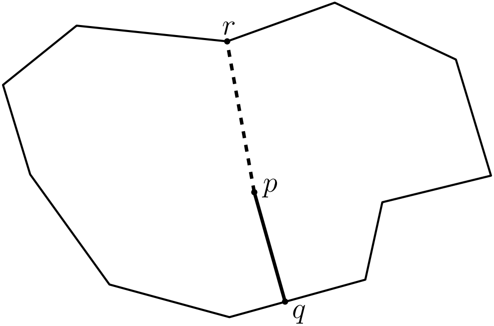
- Was ist Umweg?
- Verhältnis von Distanz im Graph und Euklid-Distanz.
- Umweg oder Streckung (engl. dilation) von $r \in P$ mit Feed-link $\overline{pq}$:

$$\delta_q(r) = \frac{\text{sp-length}(p, q) + \text{bd-length}(q, r)}{\text{sp-length}(r, p)}.$$

Streckung eines Punktes

- Was ist Umweg?
- Verhältnis von Distanz im Graph und Euklid-Distanz.
- Umweg oder Streckung (engl. dilation) von $r \in P$ mit Feed-link $\overline{pq}$:

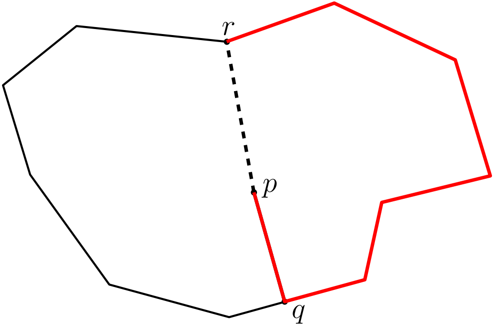
$$\delta_q(r) = \frac{\text{sp-length}(p, q) + \text{bd-length}(q, r)}{\text{sp-length}(r, p)}.$$



Streckung eines Punktes

- Was ist Umweg?
- Verhältnis von Distanz im Graph und Euklid-Distanz.
- Umweg oder Streckung (engl. dilation) von $r \in P$ mit Feed-link $\overline{pq}$:

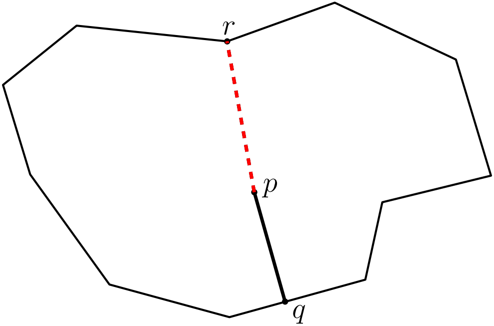
$$\delta_q(r) = \frac{\text{sp-length}(p, q) + \text{bd-length}(q, r)}{\text{sp-length}(r, p)}.$$



Streckung eines Punktes

- Was ist Umweg?
- Verhältnis von Distanz im Graph und Euklid-Distanz.
- Umweg oder Streckung (engl. dilation) von $r \in P$ mit Feed-link $\overline{pq}$:

$$\delta_q(r) = \frac{\text{sp-length}(p, q) + \text{bd-length}(q, r)}{\text{sp-length}(r, p)}.$$



Streckung des Polygons

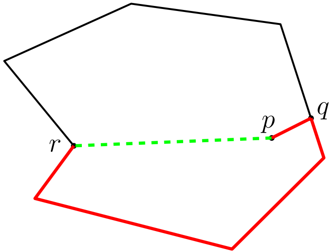
- Die Streckung von P bei Wahl von $\overline{pq}$ ist die größte Streckung aller $r \in P$:

$$\delta_q(P) = \max_{r \in P} \delta_q(r).$$

Streckung des Polygons

- Die Streckung von P bei Wahl von $\overline{pq}$ ist die größte Streckung aller $r \in P$:

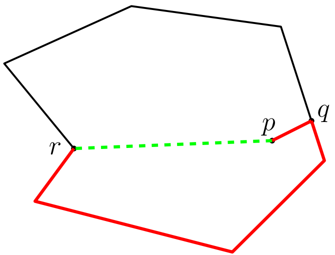
$$\delta_q(P) = \max_{r \in P} \delta_q(r).$$



Streckung des Polygons

- Die Streckung von P bei Wahl von $\overline{pq}$ ist die größte Streckung aller $r \in P$:

$$\delta_q(P) = \max_{r \in P} \delta_q(r).$$

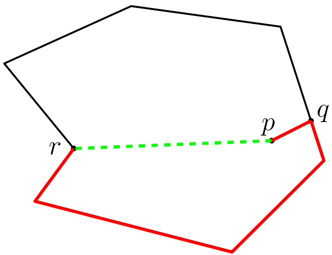


Beispiel für kleinere Streckung

Streckung des Polygons

- Die Streckung von P bei Wahl von $\overline{pq}$ ist die größte Streckung aller $r \in P$:

$$\delta_q(P) = \max_{r \in P} \delta_q(r).$$



Beispiel für größere Streckung

Inhalt

1 Einführung
- Problemstellung und Definitionen

2 Ein Feed-link
- Heuristische und optimale Platzierung eines Feed-Links

3 Hindernisse
- Definitionen und Auswirkungen

4 Mehrere Feed-links
- Berechnung der Streckung und heuristische Platzierung mehrerer Feed-links

Erster Ansatz für die Wahl des Feed-links

- Der optimale Feed-link minimiert die Streckung des Polygons.

Erster Ansatz für die Wahl des Feed-links

- Der optimale Feed-link minimiert die Streckung des Polygons.
- Wie finden wir ihn? ($\rightarrow$ später).

Erster Ansatz für die Wahl des Feed-links

- Der optimale Feed-link minimiert die Streckung des Polygons.
- Wie finden wir ihn? ($\rightarrow$ später).
- Der näheste Punkt auf P ist eine gute Wahl.

Erster Ansatz für die Wahl des Feed-links

- Der optimale Feed-link minimiert die Streckung des Polygons.
- Wie finden wir ihn? ($\rightarrow$ später).
- Der näheste Punkt auf P ist eine gute Wahl.

Lemma

Sei $q \in P$ sodass $|pq|$ minimal ist und $q' \in P$ sodass $\delta_{q'}(P)$ minimal ist. Dann gilt: $\delta_q(P) \leq 2\delta_{q'}(P)$.

Erster Ansatz für die Wahl des Feed-links

- Der optimale Feed-link minimiert die Streckung des Polygons.
- Wie finden wir ihn? ($\rightarrow$ später).
- Der näheste Punkt auf P ist eine gute Wahl.

Lemma

Sei $q \in P$ sodass $|pq|$ minimal ist und $q' \in P$ sodass $\delta_{q'}(P)$ minimal ist. Dann gilt: $\delta_q(P) \leq 2\delta_{q'}(P)$.

Erster Ansatz für die Wahl des Feed-links

- Der optimale Feed-link minimiert die Streckung des Polygons.
- Wie finden wir ihn? ($\rightarrow$ später).
- Der näheste Punkt auf P ist eine gute Wahl.

Lemma

Sei $q \in P$ sodass $|pq|$ minimal ist und $q' \in P$ sodass $\delta_{q'}(P)$ minimal ist. Dann gilt: $\delta_q(P) \leq 2\delta_{q'}(P)$.

Erster Ansatz für die Wahl des Feed-links

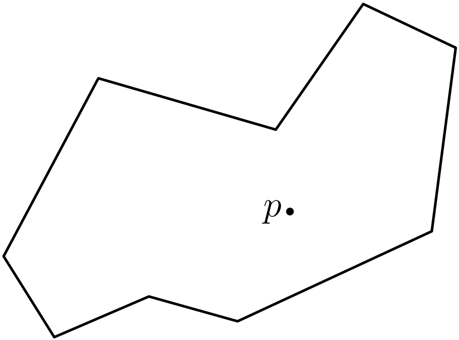
- Der optimale Feed-link minimiert die Streckung des Polygons.
- Wie finden wir ihn? ($\rightarrow$ später).
- Der näheste Punkt auf P ist eine gute Wahl.

Lemma

Sei $q \in P$ sodass $|pq|$ minimal ist und $q' \in P$ sodass $\delta_{q'}(P)$ minimal ist. Dann gilt: $\delta_q(P) \leq 2\delta_{q'}(P)$.

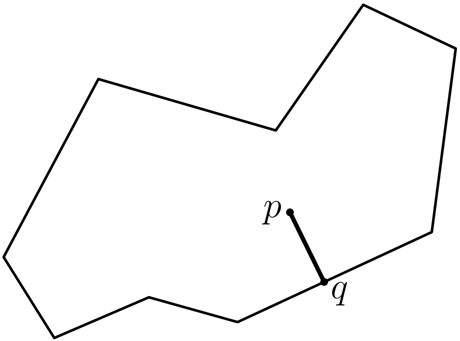
Beweis des Lemmas

Gegeben: einfaches Polygon P.



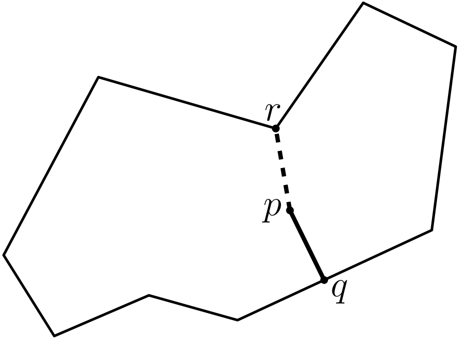
Beweis des Lemmas

Sei $q \in P$ sodass $|pq|$ minimal.



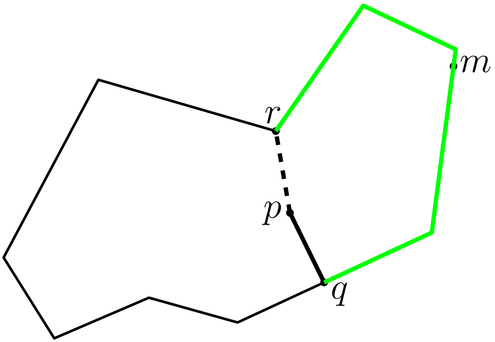
Beweis des Lemmas

Sei $r \in P$ sodass $\delta_q(r)$ maximal.



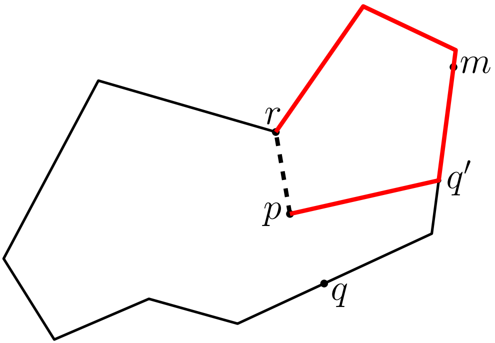
Beweis des Lemmas

Sei $m \in P$ in der Mitte des kürzeren Pfades von q nach r.



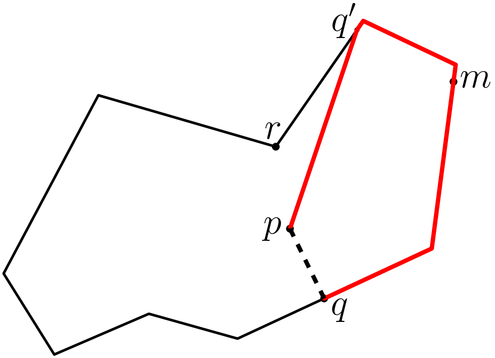
Beweis des Lemmas

Feed-link an q' zwischen q und $m \Rightarrow \delta_{q'}(r) \geq \frac{1}{2}\delta_q(r)$.



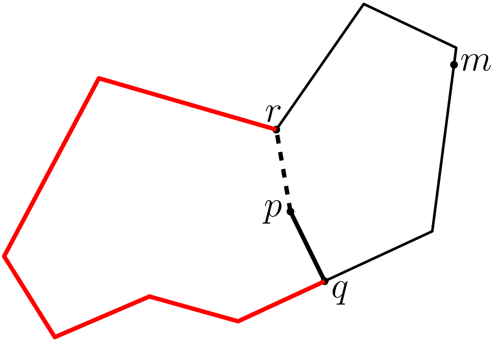
Beweis des Lemmas

Feed-link an q' zwischen m und $r \Rightarrow \delta_{q'}(q) \geq \frac{1}{2}\delta_q(r)$.



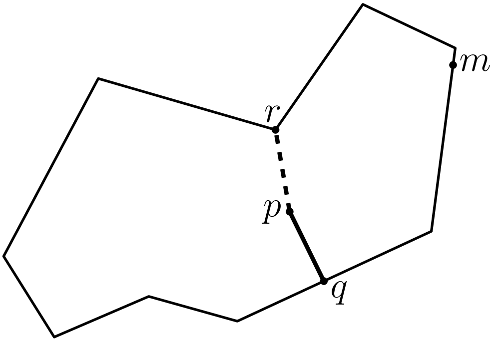
Beweis des Lemmas

Selbes Argument gilt für den längeren Pfad von q nach r.

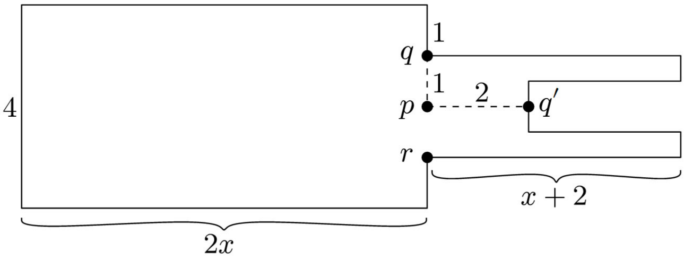


Beweis des Lemmas

Insgesamt folgt also: $\delta_q(P) \leq 2\delta_{q'}(P)$.

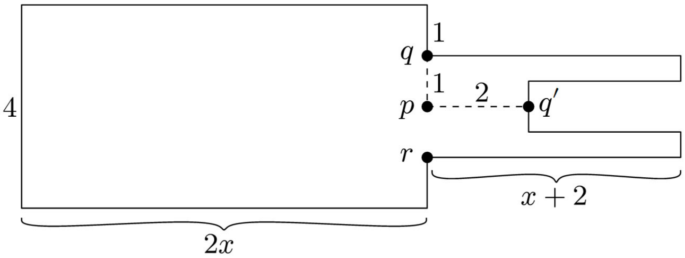


Bewertung der Abschätzung



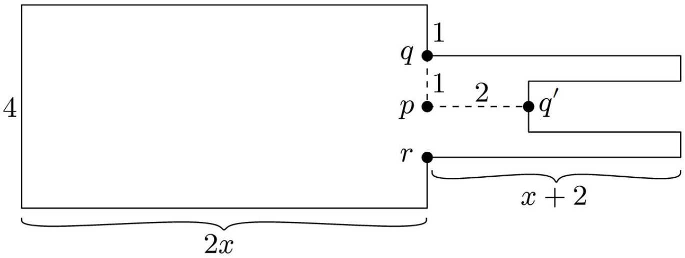
Die obige Abschätzung ist scharf.

Bewertung der Abschätzung



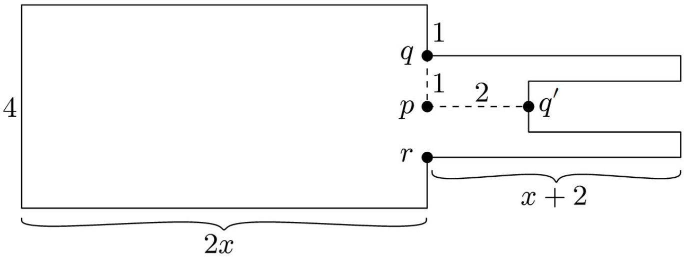
Feed-link an $q \Rightarrow \delta_q(P) = \delta_q(r) = 4x + 7$.

Bewertung der Abschätzung



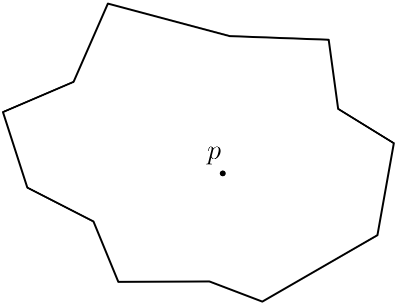
Feed-link an $q' \Rightarrow \delta_{q'}(P) = \delta_{q'}(r) = 2x + 5$.

Bewertung der Abschätzung



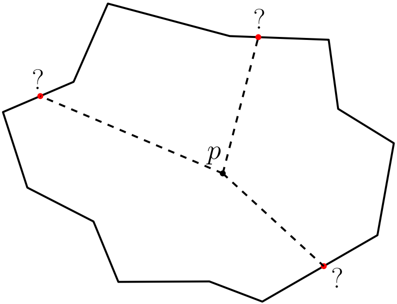
$$\lim_{x\to\infty} \frac{\delta_q(r)}{\delta_{q'}(r)} = \lim_{x\to\infty} \frac{4x+7}{2x+5} = 2.$$

Optimales Platzieren eines Feed-links



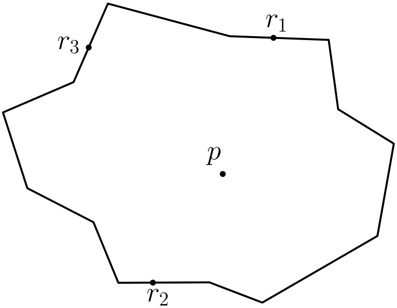
Gegeben: p im Inneren eines einfachen Polygons P, diskrete Menge S von Punkten auf dem Rand von P.

Optimales Platzieren eines Feed-links



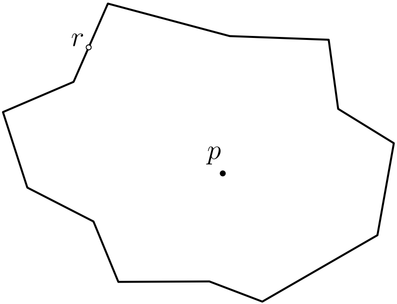
Gesucht: $q \in P$ sodass $\max_{r \in S} \delta_q(r)$ minimal ist.

Optimales Platzieren eines Feed-links



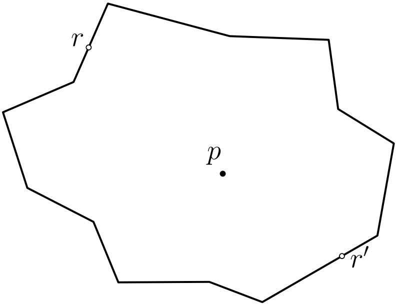
Im Beispiel: $S = \{r_1, r_2, r_3\}$.

Optimales Platzieren eines Feed-links



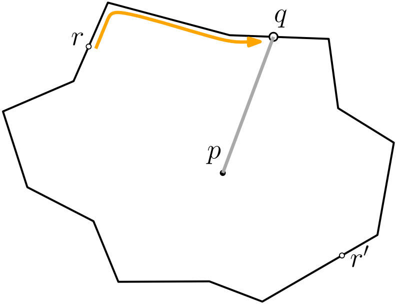
Sei $r \in S$.

Optimales Platzieren eines Feed-links



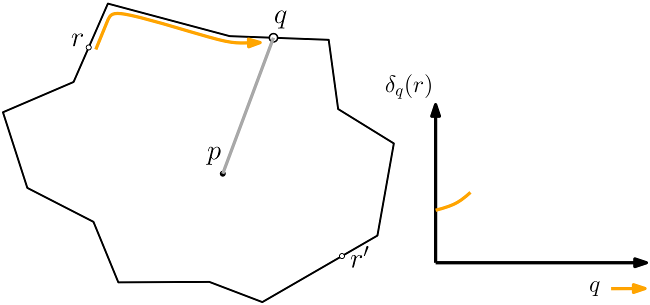
Sei $r' \in P$ sodass $\mu(r, r') = \mu(r', r) = \frac{\mu(P)}{2}$.
Dabei ist $\mu(P)$ der Umfang von P und $\mu(r, r')$ die Länge $\mu(P[r, r'])$ des Teilpolygons $P[r, r']$ von r nach r' im UZS.

Optimales Platzieren eines Feed-links

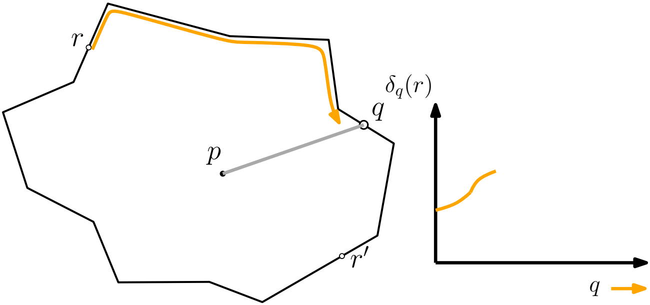


Wir beobachten Streckung von r wenn q im UZS entlang P läuft.

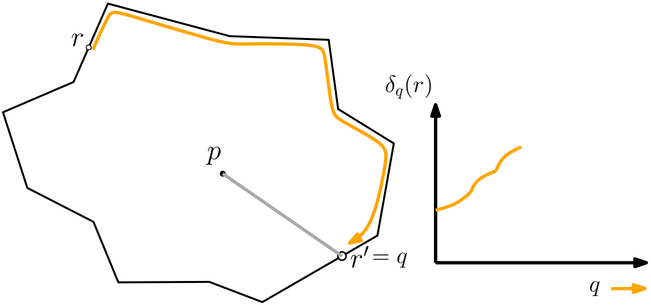
Optimales Platzieren eines Feed-links



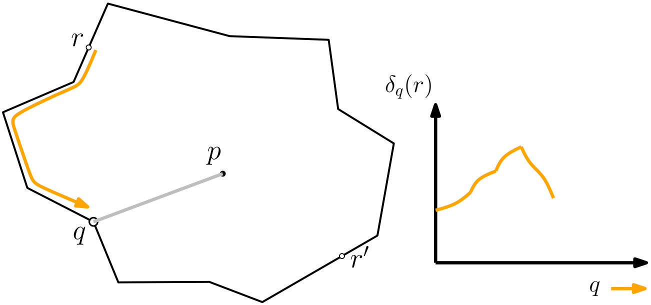
Optimales Platzieren eines Feed-links



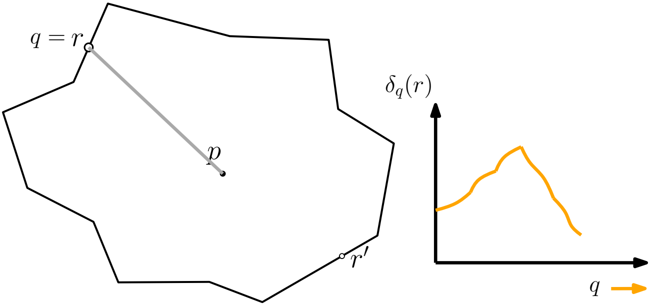
Optimales Platzieren eines Feed-links



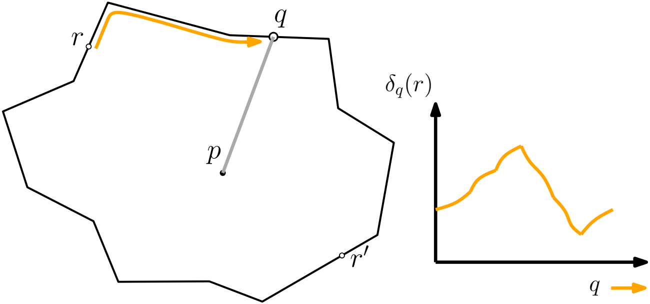
Optimales Platzieren eines Feed-links



Optimales Platzieren eines Feed-links

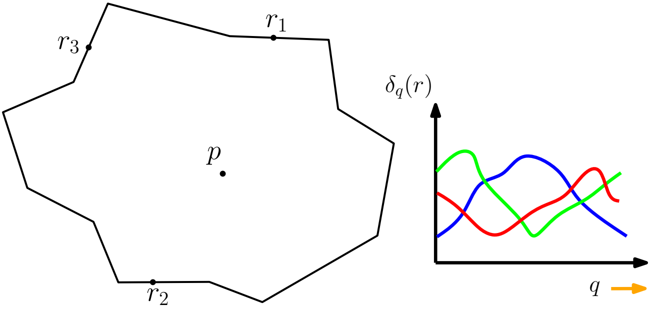


Optimales Platzieren eines Feed-links



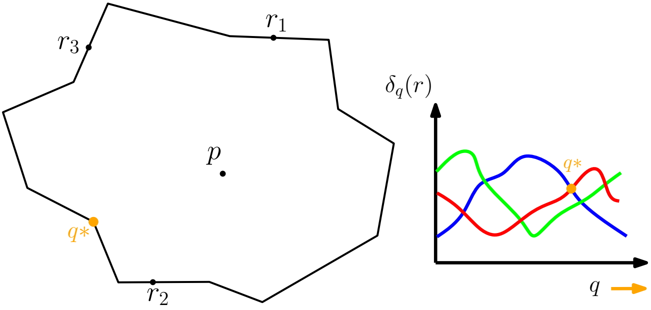
Wir erhalten einen Graph der Streckung in Abhängigkeit der Position von q.

Optimales Platzieren eines Feed-links



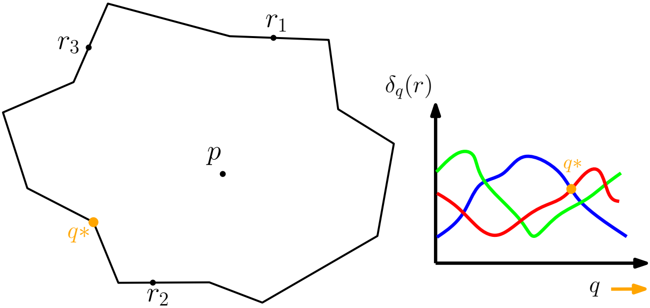
Wir könnten für jedes r_i so einen Graph machen.

Optimales Platzieren eines Feed-links



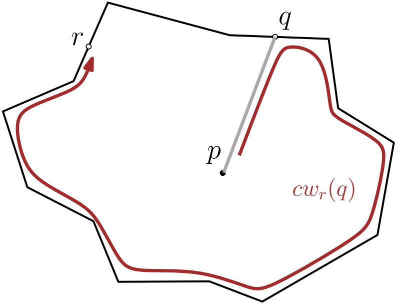
Und dann die Position des optimalen q^* bestimmen.

Optimales Platzieren eines Feed-links



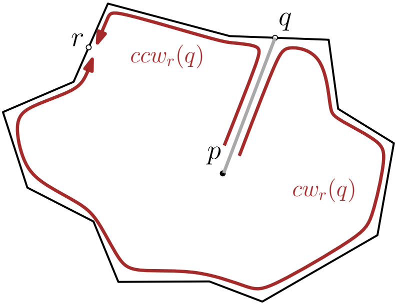
Leider ist das nicht effizient, da die Graphen irgendeine Form haben können.

Optimales Platzieren eines Feed-links



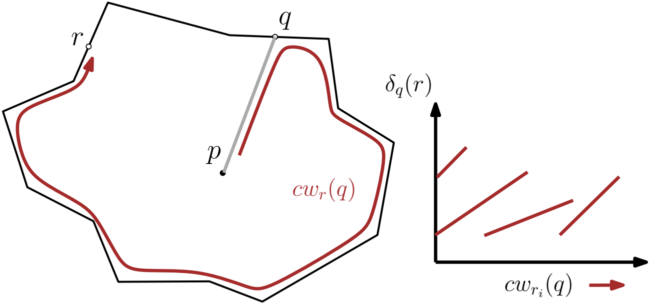
Wir suchen eine bessere Parametrisierung und definieren dazu die Distanz $cw_r(q) = |pq| + \mu(q, r)$ von p über q nach r im UZS.

Optimales Platzieren eines Feed-links



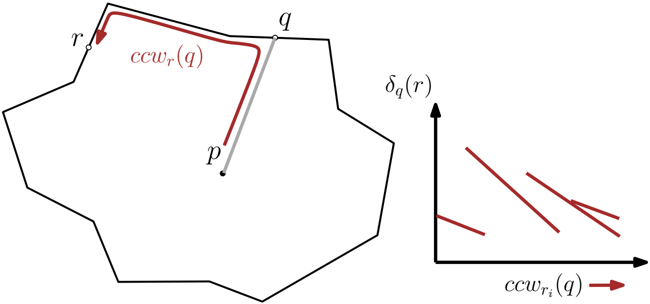
Analog defnieren wir $ccw_r(q) = |pq| + \mu(r, q)$ gegen den UZS.

Optimales Platzieren eines Feed-links



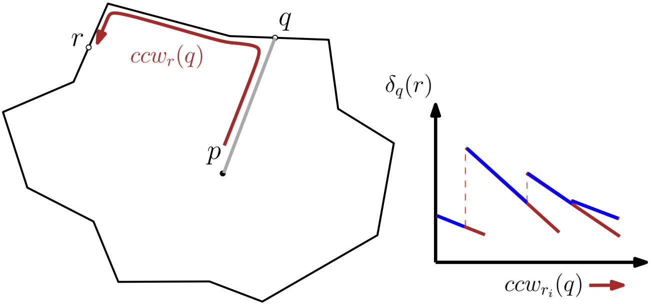
Tragen wir dann $\delta_q(r)$ über $cw_r(q)$ haben wir lineare Funktionen zwischen r und r'!

Optimales Platzieren eines Feed-links



Ebenso für $ccw_r(q)$ zwischen r' und r.

Optimales Platzieren eines Feed-links



Wir bilden die obere Hülle für jede der beiden Geradenscharen.

Optimales Platzieren eines Feed-links

- Wir scannen in beiden Hüllen nach dem Minimum.

Optimales Platzieren eines Feed-links

- Wir scannen in beiden Hüllen nach dem Minimum.
- Liegt q so, dass wir in einem Intervall I landen bestimmt durch die Kanten e_1 und e_2 der Hüllen,

Optimales Platzieren eines Feed-links

- Wir scannen in beiden Hüllen nach dem Minimum.
- Liegt q so, dass wir in einem Intervall I landen bestimmt durch die Kanten e_1 und e_2 der Hüllen,
- dann unifizieren wir die beiden Graphen durch $cw_{r_1}(q) = -ccw_{r_1}(q) + 2|pq| + \mu(P)$.

Optimales Platzieren eines Feed-links

- Wir scannen in beiden Hüllen nach dem Minimum.
- Liegt q so, dass wir in einem Intervall I landen bestimmt durch die Kanten e_1 und e_2 der Hüllen,
- dann unifizieren wir die beiden Graphen durch $cw_{r_1}(q) = -ccw_{r_1}(q) + 2|pq| + \mu(P)$.
- Dadurch wird eine lineare Funktion zu einer quadratischen, wegen $|pq|$.

Optimales Platzieren eines Feed-links

- Wir scannen in beiden Hüllen nach dem Minimum.
- Liegt q so, dass wir in einem Intervall I landen bestimmt durch die Kanten e_1 und e_2 der Hüllen,
- dann unifizieren wir die beiden Graphen durch $cw_{r_1}(q) = -ccw_{r_1}(q) + 2|pq| + \mu(P)$.
- Dadurch wird eine lineare Funktion zu einer quadratischen, wegen $|pq|$.
- Nun suchen wir im Intervall I das Minimum der oberen Hülle einer quadratischen und einer linearen Funktion.

Optimales Platzieren eines Feed-links

- Wir scannen in beiden Hüllen nach dem Minimum.
- Liegt q so, dass wir in einem Intervall I landen bestimmt durch die Kanten e_1 und e_2 der Hüllen,
- dann unifizieren wir die beiden Graphen durch $cw_{r_1}(q) = -ccw_{r_1}(q) + 2|pq| + \mu(P)$.
- Dadurch wird eine lineare Funktion zu einer quadratischen, wegen $|pq|$.
- Nun suchen wir im Intervall I das Minimum der oberen Hülle einer quadratischen und einer linearen Funktion.
- Das geht (Extremwertbestimmung).

Optimales Platzieren eines Feed-links

- Wir scannen in beiden Hüllen nach dem Minimum.
- Liegt q so, dass wir in einem Intervall I landen bestimmt durch die Kanten e_1 und e_2 der Hüllen,
- dann unifizieren wir die beiden Graphen durch $cw_{r_1}(q) = -ccw_{r_1}(q) + 2|pq| + \mu(P)$.
- Dadurch wird eine lineare Funktion zu einer quadratischen, wegen $|pq|$.
- Nun suchen wir im Intervall I das Minimum der oberen Hülle einer quadratischen und einer linearen Funktion.
- Das geht (Extremwertbestimmung).
- Wir wiederholen den Vorgang für jedes Intervall I.

Optimales Platzieren eines Feed-links - Bewertung

- Bestimmen der Liniensegmente: $O(n+k)$ (n Knoten in P).

Optimales Platzieren eines Feed-links - Bewertung

- Bestimmen der Liniensegmente: $O(n + k)$ (n Knoten in P).
- Bestimmen der oberen Hüllen: $O(k \cdot \log k)$ (Algorithmus von Hershberger).

Optimales Platzieren eines Feed-links - Bewertung

- Bestimmen der Liniensegmente: $O(n + k)$ (n Knoten in P).
- Bestimmen der oberen Hüllen: $O(k \cdot \log k)$ (Algorithmus von Hershberger).
- Anzahl der Intervalle: $O(k \cdot \alpha(k)) \subseteq O(k \cdot \log k)$ (Davenport-Schinzel Sequenz).

Optimales Platzieren eines Feed-links - Bewertung

- Bestimmen der Liniensegmente: $O(n+k)$ (n Knoten in P).
- Bestimmen der oberen Hüllen: $O(k \cdot \log k)$ (Algorithmus von Hershberger).
- Anzahl der Intervalle: $O(k \cdot \alpha(k)) \subseteq O(k \cdot \log k)$ (Davenport-Schinzel Sequenz).
- Minimum in jedem Intervall bestimmen: $O(1)$.

Optimales Platzieren eines Feed-links - Bewertung

- Bestimmen der Liniensegmente: $O(n+k)$ (n Knoten in P).
- Bestimmen der oberen Hüllen: $O(k \cdot \log k)$ (Algorithmus von Hershberger).
- Anzahl der Intervalle: $O(k \cdot \alpha(k)) \subseteq O(k \cdot \log k)$ (Davenport-Schinzel Sequenz).
- Minimum in jedem Intervall bestimmen: $O(1)$.
- Gesamt: $O(n + k \cdot \log(k))$.

Optimales Platzieren eines Feed-links - Bewertung

- Bestimmen der Liniensegmente: $O(n+k)$ (n Knoten in P).
- Bestimmen der oberen Hüllen: $O(k \cdot \log k)$ (Algorithmus von Hershberger).
- Anzahl der Intervalle: $O(k \cdot \alpha(k)) \subseteq O(k \cdot \log k)$ (Davenport-Schinzel Sequenz).
- Minimum in jedem Intervall bestimmen: $O(1)$.
- Gesamt: $O(n + k \cdot \log(k))$.
- Der Algorithmus lässt sich erweitern sodass er den optimalen Feed-link bezüglich **allen** Punkten von P findet.

Optimales Platzieren eines Feed-links - Bewertung

- Bestimmen der Liniensegmente: $O(n+k)$ (n Knoten in P).
- Bestimmen der oberen Hüllen: $O(k \cdot \log k)$ (Algorithmus von Hershberger).
- Anzahl der Intervalle: $O(k \cdot \alpha(k)) \subseteq O(k \cdot \log k)$ (Davenport-Schinzel Sequenz).
- Minimum in jedem Intervall bestimmen: $O(1)$.
- Gesamt: $O(n + k \cdot \log(k))$.
- Der Algorithmus lässt sich erweitern sodass er den optimalen Feed-link bezüglich **allen** Punkten von P findet.
- Hat den Aufwand $O(\lambda_7(n) \log(n))$.

Inhalt

1 Einführung
- Problemstellung und Definitionen

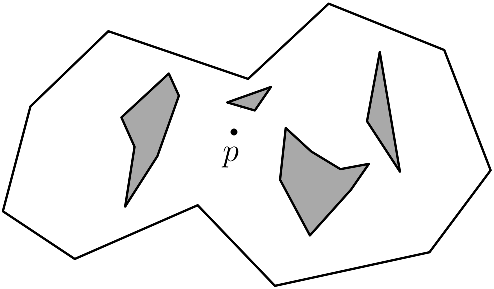
2 Ein Feed-link
- Heuristische und optimale Platzierung eines Feed-Links

3 Hindernisse
- Definitionen und Auswirkungen

4 Mehrere Feed-links
- Berechnung der Streckung und heuristische Platzierung mehrerer Feed-links

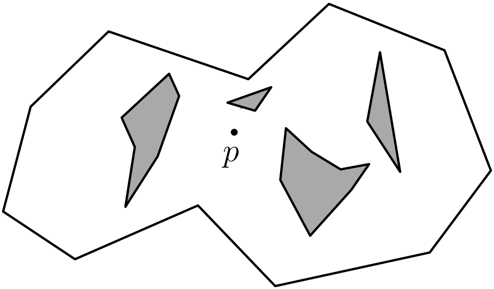
Erweiterung der Problemstellung: Hindernisse

- Wir lassen jetzt eine Menge Hindernissen zu.



Erweiterung der Problemstellung: Hindernisse

- Wir lassen jetzt eine Menge Hindernissen zu.
- Das sind disjunkte Polygone die den Rand von P nicht kreuzen.



Anpassung der Streckung

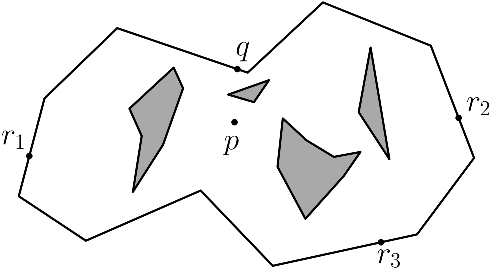
- Wir passen den Begriff der Streckung an.

Anpassung der Streckung

- Wir passen den Begriff der Streckung an.
- sp-length$(p, x) =$ Länge des kürzesten Pfades zwischen p und x der Hindernisse vermeidet.

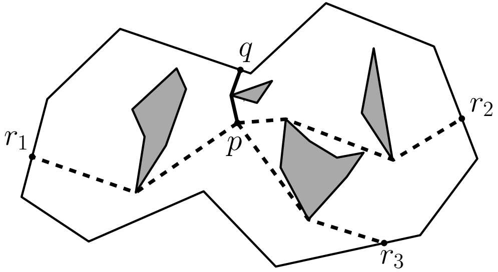
Anpassung der Streckung

- Wir passen den Begriff der Streckung an.
- sp-length(p, x) $=$ Länge des kürzesten Pfades zwischen p und x der Hindernisse vermeidet.



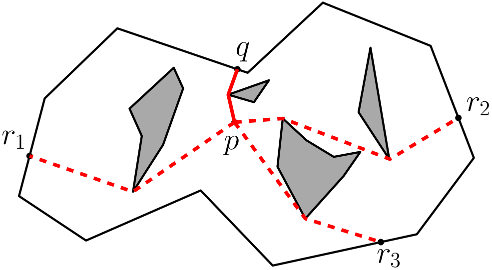
Anpassung der Streckung

- Wir passen den Begriff der Streckung an.
- sp-length(p, x) $=$ Länge des kürzesten Pfades zwischen p und x der Hindernisse vermeidet.



Anpassung der Streckung

- Wir passen den Begriff der Streckung an.
- sp-length(p, x) $=$ Länge des kürzesten Pfades zwischen p und x der Hindernisse vermeidet.



Auswirkungen

- Die Aussage Lemmas über den kürzesten Feed-link bleibt erhalten.

Auswirkungen

- Die Aussage Lemmas über den kürzesten Feed-link bleibt erhalten.
- Wähle im Beweis $q \in P$ sodass sp-lenght(p, q) minimal.

Auswirkungen

- Die Aussage Lemmas über den kürzesten Feed-link bleibt erhalten.
- Wähle im Beweis $q \in P$ sodass sp-lenght(p, q) minimal.
- Auch die Auswirkung auf die Streckung können wir beschränken:

Auswirkungen

- Die Aussage Lemmas über den kürzesten Feed-link bleibt erhalten.
- Wähle im Beweis $q \in P$ sodass sp-lenght(p, q) minimal.
- Auch die Auswirkung auf die Streckung können wir beschränken:

Lemma

Sei $\delta(r)$ die Streckung von r mit einem Feed-link l in einem Polygon P ohne Hindernisse.
Sei $\delta'(r)$ die Streckung in der gleichen Situation, aber jetzt kann P Hindernisse enthalten.
Dann gilt: $\delta'(r) \leq 3\delta(r)$.

Auswirkungen

- Die Aussage Lemmas über den kürzesten Feed-link bleibt erhalten.
- Wähle im Beweis $q \in P$ sodass sp-lenght(p, q) minimal.
- Auch die Auswirkung auf die Streckung können wir beschränken:

Lemma

Sei $\delta(r)$ die Streckung von r mit einem Feed-link l in einem Polygon P ohne Hindernisse.
Sei $\delta'(r)$ die Streckung in der gleichen Situation, aber jetzt kann P Hindernisse enthalten.
Dann gilt: $\delta'(r) \leq 3\delta(r)$.

Auswirkungen

- Die Aussage Lemmas über den kürzesten Feed-link bleibt erhalten.
- Wähle im Beweis $q \in P$ sodass sp-lenght(p, q) minimal.
- Auch die Auswirkung auf die Streckung können wir beschränken:

Lemma

Sei $\delta(r)$ die Streckung von r mit einem Feed-link l in einem Polygon P ohne Hindernisse.
Sei $\delta'(r)$ die Streckung in der gleichen Situation, aber jetzt kann P Hindernisse enthalten.
Dann gilt: $\delta'(r) \leq 3\delta(r)$.

Auswirkungen

- Die Aussage Lemmas über den kürzesten Feed-link bleibt erhalten.
- Wähle im Beweis $q \in P$ sodass sp-lenght(p, q) minimal.
- Auch die Auswirkung auf die Streckung können wir beschränken:

Lemma

Sei $\delta(r)$ die Streckung von r mit einem Feed-link l in einem Polygon P ohne Hindernisse.
Sei $\delta'(r)$ die Streckung in der gleichen Situation, aber jetzt kann P Hindernisse enthalten.
Dann gilt: $\delta'(r) \leq 3\delta(r)$.

Inhalt

1 Einführung
- Problemstellung und Definitionen

2 Ein Feed-link
- Heuristische und optimale Platzierung eines Feed-Links

3 Hindernisse
- Definitionen und Auswirkungen

4 Mehrere Feed-links
- Berechnung der Streckung und heuristische Platzierung mehrerer Feed-links

Mehrere Feed-links

- Wir wollen die Streckung durch eine Konstante beschränken.

Mehrere Feed-links

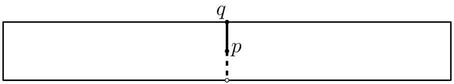
- Wir wollen die Streckung durch eine Konstante beschränken.
- Dafür reicht ein einziger Feed-link nicht immer aus.

Mehrere Feed-links

- Wir wollen die Streckung durch eine Konstante beschränken.
- Dafür reicht ein einziger Feed-link nicht immer aus.

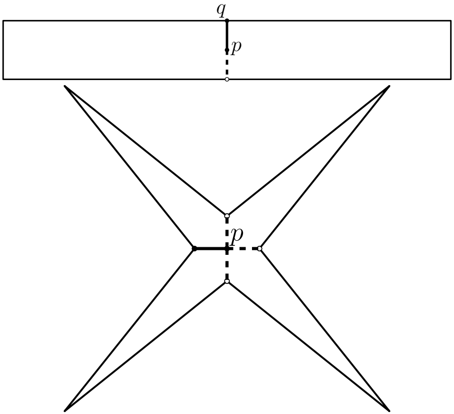
Mehrere Feed-links

- Wir wollen die Streckung durch eine Konstante beschränken.
- Dafür reicht ein einziger Feed-link nicht immer aus.



Mehrere Feed-links

- Wir wollen die Streckung durch eine Konstante beschränken.
- Dafür reicht ein einziger Feed-link nicht immer aus.



Aussagen über die benötigte Anzahl von Feedlinks

- In einem konvexen Polygon reichen 2 Feed-links um eine Streckung $\leq 3 + \sqrt{3}$ zu erreichen.

Aussagen über die benötigte Anzahl von Feedlinks

- In einem konvexen Polygon reichen 2 Feed-links um eine Streckung $\leq 3 + \sqrt{3}$ zu erreichen.
- In der gleichen Situation reichen k Feed-links aus um eine Streckung von $1 + O(\frac{1}{k})$ zu erreichen.

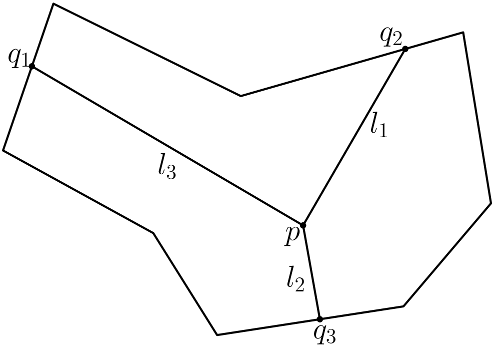
Berechnung der Streckung gegeben mehrere Feed-links

- Gegeben: Feed-links $l_1, ..., l_k$ an den Punkten $q_1, ..., q_k$ an P anliegen.

Berechnung der Streckung gegeben mehrere Feed-links

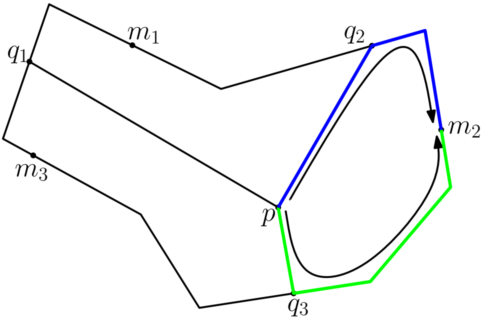
- Gegeben: Feed-links $l_1, ..., l_k$ an den Punkten $q_1, ..., q_k$ an P anliegen.
- Gesucht: Streckung $\delta(P)$ (hier: nur ohne Hindernisse).

Berechnung der Streckung - Vorgehen



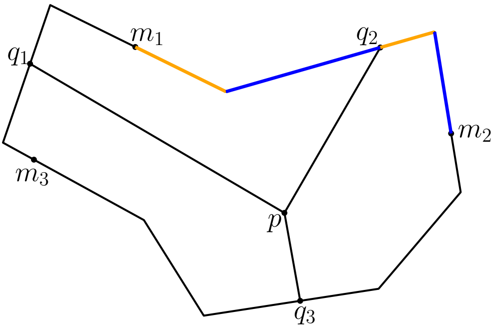
Gegeben: Feed-links $l_1, ..., l_k$ an $q_1, ..., q_k$.

Berechnung der Streckung - Vorgehen



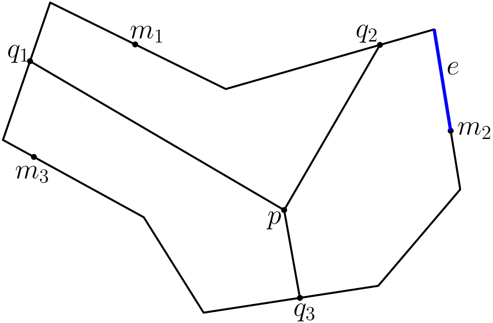
Bestimme m_i so dass: sp-length(p, q_i) + bd-length(q_i, m_i) = sp-length(p, q_{i+1}) + bd-length(q_{i+1}, m_i).

Berechnung der Streckung - Vorgehen



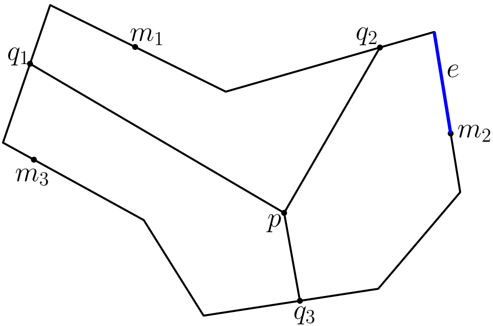
Der kürzeste Pfad aller Punkte auf Kanten zwischen m_i und m_{i+1} führt über q_{i+1}.

Berechnung der Streckung - Vorgehen



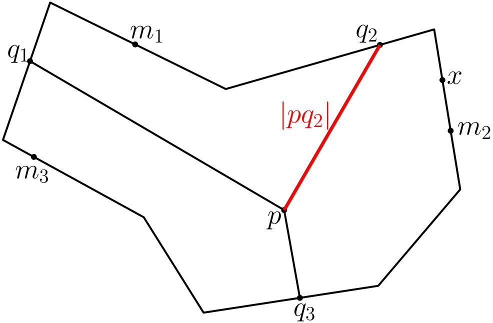
Sei *e* eine Kante zwischen m_i und m_{i+1}.

Berechnung der Streckung - Vorgehen



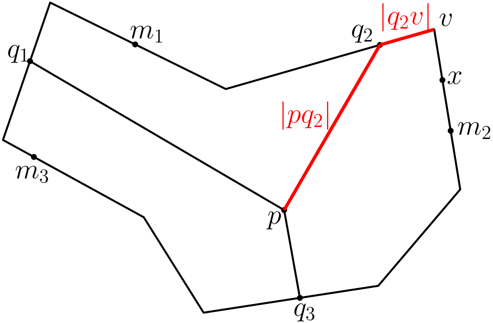
Wir wollen die Streckung $\delta(e) := \max_{x \in e} \delta(x)$ von e bestimmen.

Berechnung der Streckung - Vorgehen



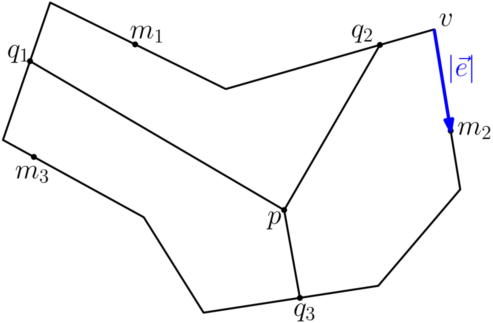
$$\delta(e) = \max_{x \in e} \frac{|pq_{i+1}| + \text{bd-length}(q_{i+1}, x)}{|px|}.$$

Berechnung der Streckung - Vorgehen



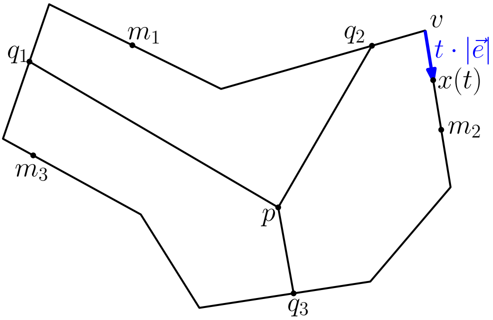
$$\delta(e) = \max_{x \in e} \frac{|pq_{i+1}| + \text{bd-length}(q_{i+1}, v) + \text{bd-length}(v, x)}{|px|}.$$

Berechnung der Streckung - Vorgehen



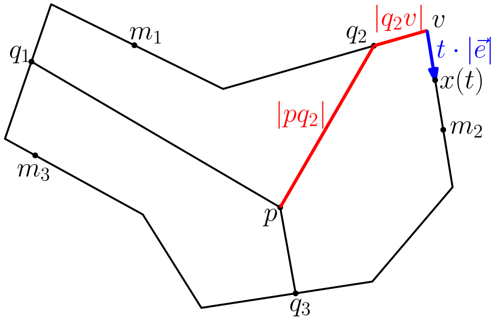
Sei $\vec{e}$ der Vektor entlang e von v in Richtung x.

Berechnung der Streckung - Vorgehen



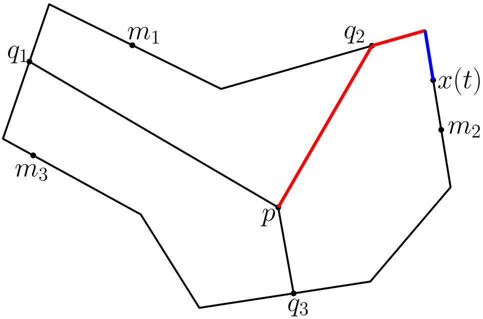
Wir parametrisieren diesen Vektor mit $t \in [0, 1]$.

Berechnung der Streckung - Vorgehen



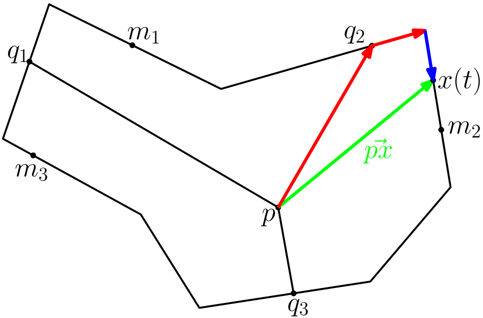
$$\delta(e) = \max_{t \in [0,1]} \frac{|pq_{i+1}| + \text{bd-length}(q_{i+1}, v) + t \cdot |\vec{e}|}{|px(t)|}.$$

Berechnung der Streckung - Vorgehen



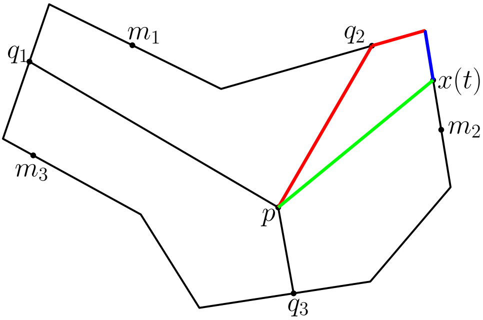
$$\delta(e) = \max_{t\in[0,1]} \frac{at+b}{|px(t)|}.$$

Berechnung der Streckung - Vorgehen



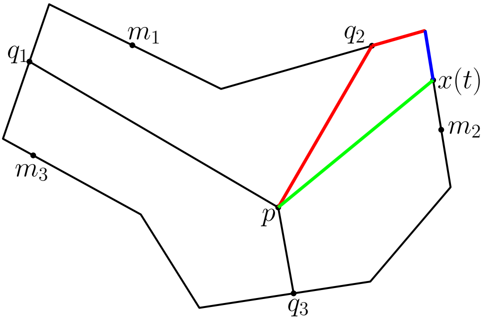
$\vec{px}$ ist Summe aus $\vec{pq_{i+1}}$, den Kantenvektoren zwischen q_{i+1} und e und $t\vec{e} \Rightarrow \vec{px(t)} = \vec{u} + t\vec{e}$.

Berechnung der Streckung - Vorgehen



Deshalb ist $|\vec{px}| = \sqrt{At^2 + Bt + C}$.

Berechnung der Streckung - Vorgehen



$$\delta(e) = \max_{t\in[0,1]} \frac{at+b}{\sqrt{At^2+Bt+C}}.$$

Berechnung der Streckung - Vorgehen

- Wir bestimmen das Maximum von $\frac{at+b}{\sqrt{At^2+Bt+C}}$ für $t \in [0, 1]$.

Berechnung der Streckung - Vorgehen

- Wir bestimmen das Maximum von $\frac{at+b}{\sqrt{At^2+Bt+C}}$ für $t \in [0, 1]$.
- Kandidaten: $t = 0$, $t = 1$, $t = \frac{-bB+2aC}{2Ab-aB}$.

Berechnung der Streckung - Vorgehen

- Wir bestimmen das Maximum von $\frac{at+b}{\sqrt{At^2+Bt+C}}$ für $t \in [0, 1]$.
- Kandidaten: $t = 0$, $t = 1$, $t = \frac{-bB+2aC}{2Ab-aB}$.
- Wir merken uns den Wert des Maximums (also $\delta(e)$).

Berechnung der Streckung - Vorgehen

- Wir bestimmen das Maximum von $\frac{at+b}{\sqrt{At^2+Bt+C}}$ für $t \in [0,1]$.
- Kandidaten: $t = 0$, $t = 1$, $t = \frac{-bB+2aC}{2Ab-aB}$.
- Wir merken uns den Wert des Maximums (also $\delta(e)$).
- Wir bestimmen $\delta(e)$ für jede Kante $e \in P$.

Berechnung der Streckung - Vorgehen

- Wir bestimmen das Maximum von $\frac{at+b}{\sqrt{At^2+Bt+C}}$ für $t \in [0, 1]$.
- Kandidaten: $t = 0$, $t = 1$, $t = \frac{-bB+2aC}{2Ab-aB}$.
- Wir merken uns den Wert des Maximums (also $\delta(e)$).
- Wir bestimmen $\delta(e)$ für jede Kante $e \in P$.
- $\delta(P) = \max_{e \in P} \delta(e)$.

Berechnung der Streckung - Aufwandsbewertung

- Reihenfolge der q_i finden: $O(k \cdot \log(k))$.

Berechnung der Streckung - Aufwandsbewertung

- Reihenfolge der q_i finden: $O(k \cdot \log(k))$.
- Mittelpunkte m_i bestimmen in $O(n + k)$.

Berechnung der Streckung - Aufwandsbewertung

- Reihenfolge der q_i finden: $O(k \cdot \log(k))$.
- Mittelpunkte m_i bestimmen in $O(n+k)$.
- Wir teilen die n Kanten von P an den Punkten $q_1, ..., q_k, m_1, ..., m_k$ auf, haben also $n+2k$ Kanten.

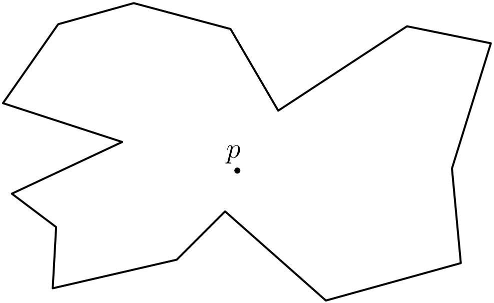
Berechnung der Streckung - Aufwandsbewertung

- Reihenfolge der q_i finden: $O(k \cdot \log(k))$.
- Mittelpunkte m_i bestimmen in $O(n + k)$.
- Wir teilen die n Kanten von P an den Punkten $q_1, ..., q_k, m_1, ..., m_k$ auf, haben also $n + 2k$ Kanten.
- Berchnung maximaler Streckung aller Kanten also $\delta(P)$ in $O(n + k)$ (Bei geschickter Durchführung).

Berechnung der Streckung - Aufwandsbewertung

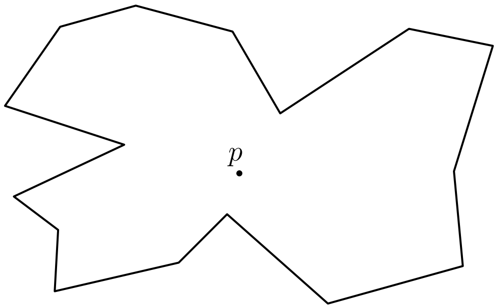
- Reihenfolge der q_i finden: $O(k \cdot \log(k))$.
- Mittelpunkte m_i bestimmen in $O(n + k)$.
- Wir teilen die n Kanten von P an den Punkten $q_1, ..., q_k, m_1, ..., m_k$ auf, haben also $n + 2k$ Kanten.
- Berchnung maximaler Streckung aller Kanten also $\delta(P)$ in $O(n + k)$ (Bei geschickter Durchführung).
- Gesamt: $O(n + k \cdot \log(k))$.

Platzieren mehrerer Feed-links - Heuristik



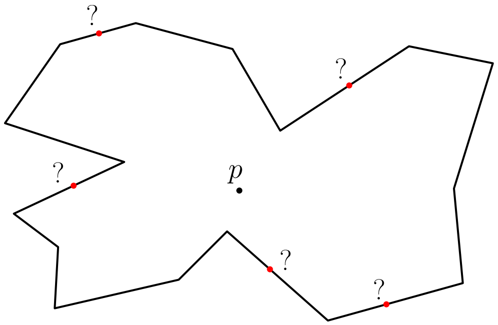
Wir benutzen eine Greedy-Heuristik zur Platzierung von k Feed-links.

Platzieren mehrerer Feed-links - Heuristik



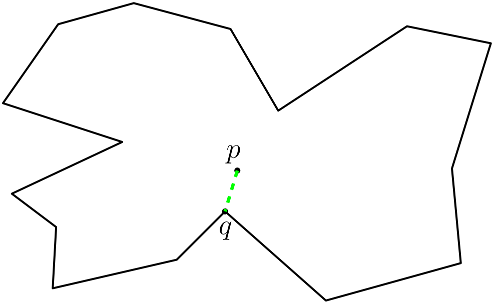
Gegeben: p im Inneren eines einfachen Polygons P.

Platzieren mehrerer Feed-links - Heuristik



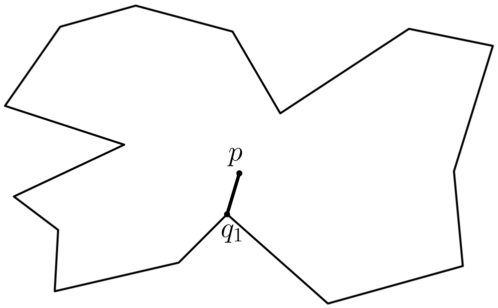
Gesucht: $q_1, ..., q_k$ sodass $\delta(P)$ möglichst klein.

Platzieren mehrerer Feed-links - Heuristik



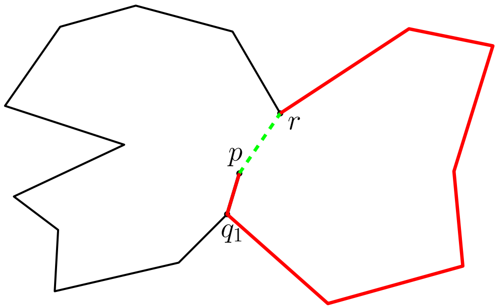
Sei $q \in P$ sodass $|pq|$ minimal.

Platzieren mehrerer Feed-links - Heuristik



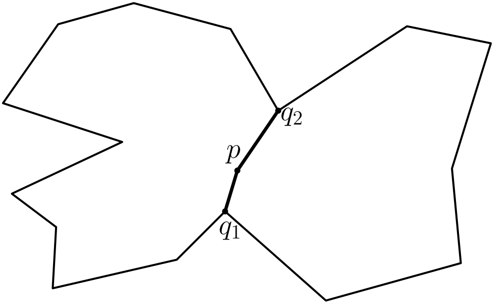
Setze $q_1 = q$.

Platzieren mehrerer Feed-links - Heuristik



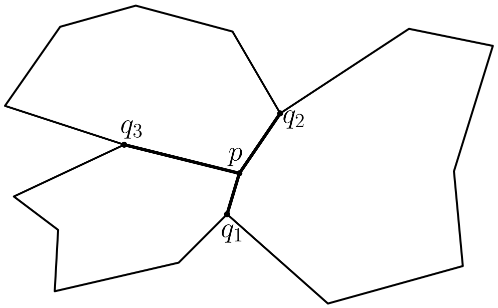
Iterativ: Finde $r \in P$ sodass $\delta(r) = \delta(P)$.

Platzieren mehrerer Feed-links - Heuristik

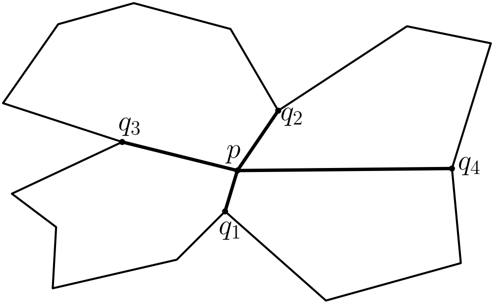


Setze $q_i = r$.

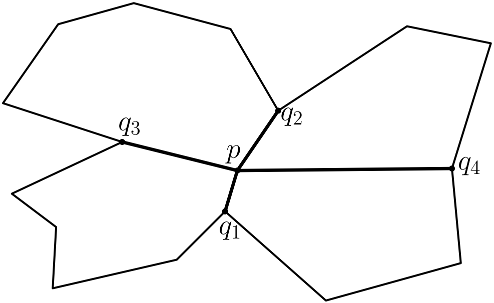
Platzieren mehrerer Feed-links - Heuristik



Platzieren mehrerer Feed-links - Heuristik



Platzieren mehrerer Feed-links - Heuristik



Aufwandsbewertung der Heuristik

- Für $k = O(1)$ und P ohne Hindernisse ist die Greedy-Heuristik in $O(n)$.

Aufwandsbewertung der Heuristik

- Für $k = O(1)$ und P ohne Hindernisse ist die Greedy-Heuristik in $O(n)$.
- Experimente zeigen: Der Greedy-Ansatz liefert gute Ergebnisse.

Aufwandsbewertung der Heuristik

- Für $k = O(1)$ und P ohne Hindernisse ist die Greedy-Heuristik in $O(n)$.
- Experimente zeigen: Der Greedy-Ansatz liefert gute Ergebnisse.
- Streckung etwa 10% bis 20% höher als der optimale Wert.

Aufwandsbewertung der Heuristik

- Für $k = O(1)$ und P ohne Hindernisse ist die Greedy-Heuristik in $O(n)$.
- Experimente zeigen: Der Greedy-Ansatz liefert gute Ergebnisse.
- Streckung etwa 10% bis 20% höher als der optimale Wert.
- Effizientes Finden einer optimalen Lösung ist eine offene Frage!

Minimierung der Anzahl der Feed-links bei vorgegebener Streckung

- Gegeben: *p* im Inneren eines einfachen Polygons *P*, c>1.

Minimierung der Anzahl der Feed-links bei vorgegebener Streckung

- Gegeben: p im Inneren eines einfachen Polygons P, c>1.
- Gesucht: (wenige) Feed-links sodass $\delta(P) \leq c$.

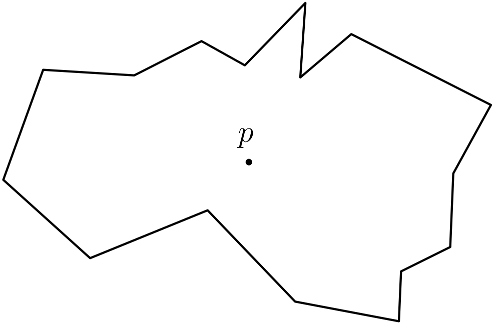
Minimierung der Anzahl der Feed-links bei vorgegebener Streckung

- Gegeben: p im Inneren eines einfachen Polygons P, c>1.
- Gesucht: (wenige) Feed-links sodass $\delta(P) \leq c$.
- Optimale Lösung nicht bekannt.

Minimierung der Anzahl der Feed-links bei vorgegebener Streckung

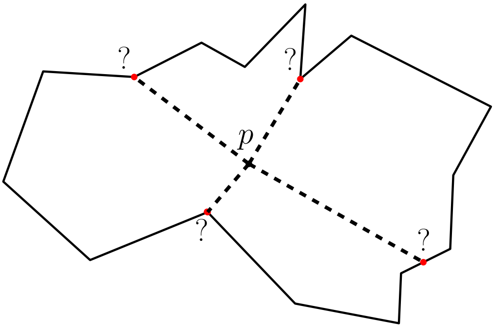
- Gegeben: p im Inneren eines einfachen Polygons P, c>1.
- Gesucht: (wenige) Feed-links sodass $\delta(P) \leq c$.
- Optimale Lösung nicht bekannt.
- Wir machen wieder einen Greedy Ansatz.

Minimierung Anzahl Feed-links - Heuristik



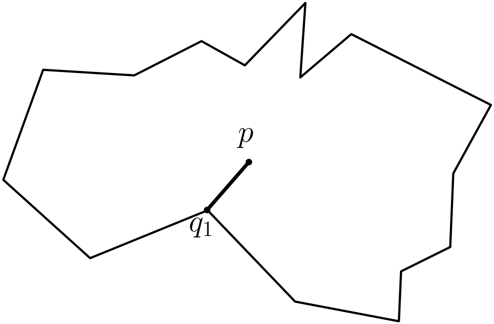
Gegeben: p im Inneren eines einfachen Polygons P, c>1.

Minimierung Anzahl Feed-links - Heuristik



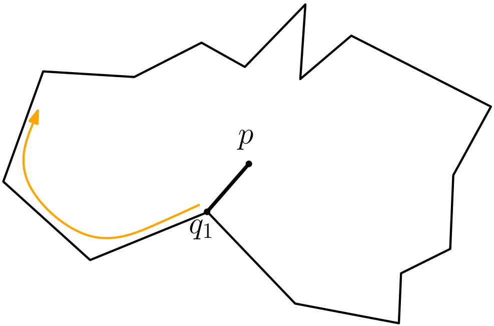
Gesucht: k Feed-links an $q_1, ..., q_k$ sodass $\delta(P) \leq c$.

Minimierung Anzahl Feed-links - Heuristik



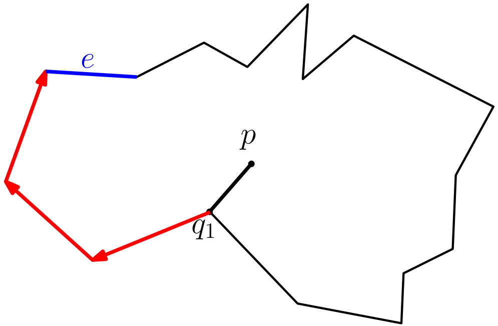
Wähle q_1 beliebig z.B. den nähesten an p.

Minimierung Anzahl Feed-links - Heuristik



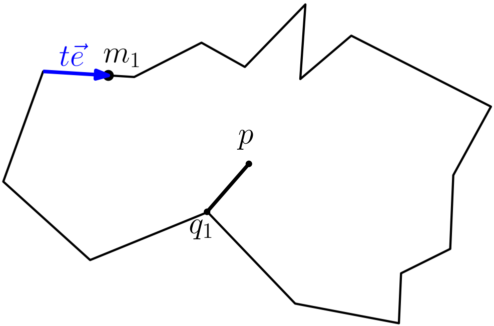
Wir wollen den nächsten Feedlink im Uhrzeigersinn möglichst weit weg setzen.

Minimierung Anzahl Feed-links - Heuristik



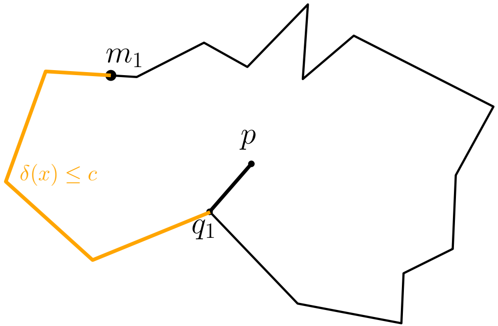
Traversiere Kanten von P im UZS bis zur Kante e mit $\delta(e) > c$.

Minimierung Anzahl Feed-links - Heuristik



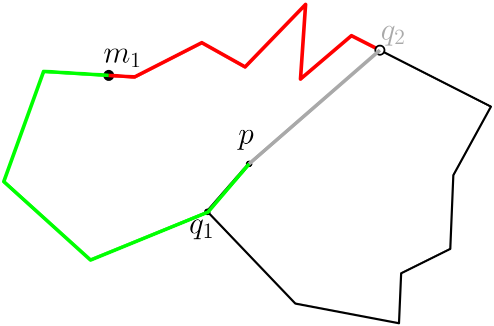
Löse die Gleichung von vorhin mit der durch $t \in [0,1]$ parametrisierten Kante e, sodass $\delta(m_1) = c$ für $m_1 = t\vec{e}$ und $\delta(t^{+}\vec{e}) > c$.

Minimierung Anzahl Feed-links - Heuristik



Alle Punkte zwischen q_1 und m_1 haben Streckung $\leq c$

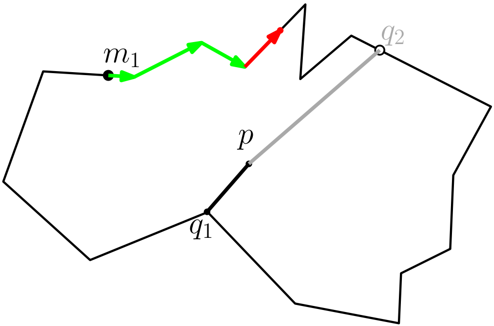
Minimierung Anzahl Feed-links - Heuristik



Setze q_2 *hypothetisch*, sodass

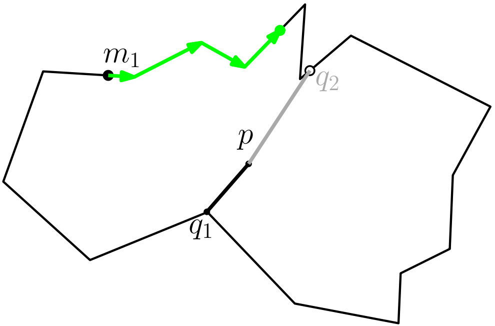
$$\text{bd-length}(p, m_1) = \text{bd-length}(m_1, q_2).$$

Minimierung Anzahl Feed-links - Heuristik



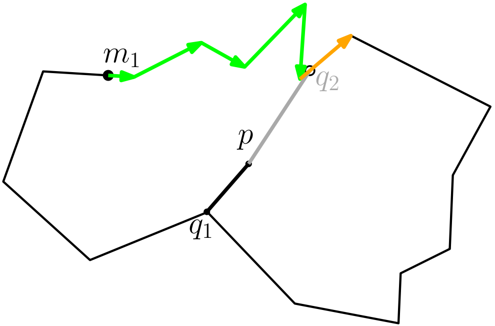
Traversiere Kanten von m_1 bis q_2 und prüfe ob Streckung $>$ c.

Minimierung Anzahl Feed-links - Heuristik



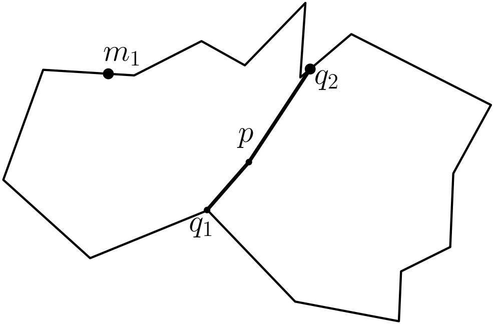
Falls ja für Kante e, Berechne q_2 neu sodass $\delta(e) = c$.

Minimierung Anzahl Feed-links - Heuristik



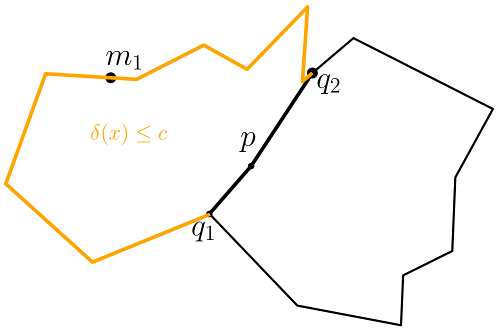
Traversiere solange bis wir am *hypothetischen* q_2 vorbeikommen.

Minimierung Anzahl Feed-links - Heuristik



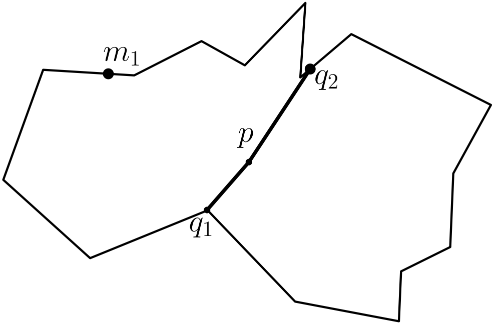
Setzte q_2.

Minimierung Anzahl Feed-links - Heuristik



Alle Punkte zwsichen q_1 und q_2 haben Streckung $\leq c$

Minimierung Anzahl Feed-links - Heuristik



Führe das Verfahren iterativ fort bis alle Kanten besucht wurden.

Aufwandsbewertung der Heuristik

- Der obige Ansatz ist absolut 1-approximativ.

Aufwandsbewertung der Heuristik

- Der obige Ansatz ist absolut 1-approximativ.
- D.h. man braucht höchstens eine Kante mehr als die beste Lösung.

Aufwandsbewertung der Heuristik

- Der obige Ansatz ist absolut 1-approximativ.
- D.h. man braucht höchstens eine Kante mehr als die beste Lösung.
- Ist k die Mindestanzahl benötigter Feed-links und n die Anzahl der Kanten, so ist die Heuristik in $O(n+k)$.

Zusammenfassung

- Wir haben gesehen wie wir günstig einen guten Feed-link finden.

Zusammenfassung

- Wir haben gesehen wie wir günstig einen guten Feed-link finden.
- Wir können einen optimalen Feed-link effizient berechnen.

Zusammenfassung

- Wir haben gesehen wie wir günstig einen guten Feed-link finden.
- Wir können einen optimalen Feed-link effizient berechnen.
- Wir können die Streckung - gegeben k Feed-links - effizient berechnen.

Zusammenfassung

- Wir haben gesehen wie wir günstig einen guten Feed-link finden.
- Wir können einen optimalen Feed-link effizient berechnen.
- Wir können die Streckung - gegeben k Feed-links - effizient berechnen.
- Wir haben effiziente Heuristiken gesehen, die k Feed-links gut platzieren,

Zusammenfassung

- Wir haben gesehen wie wir günstig einen guten Feed-link finden.
- Wir können einen optimalen Feed-link effizient berechnen.
- Wir können die Streckung - gegeben k Feed-links - effizient berechnen.
- Wir haben effiziente Heuristiken gesehen, die k Feed-links gut platzieren,
- oder mit möglichst wenigen Feed-links die Streckung unter einem vorgegebenen $c > 1$ halten.

Zusammenfassung

- Wir haben gesehen wie wir günstig einen guten Feed-link finden.
- Wir können einen optimalen Feed-link effizient berechnen.
- Wir können die Streckung - gegeben k Feed-links - effizient berechnen.
- Wir haben effiziente Heuristiken gesehen, die k Feed-links gut platzieren,
- oder mit möglichst wenigen Feed-links die Streckung unter einem vorgegebenen $c > 1$ halten.
- Offen bleibt die Frage ob wir effizient k Feed-links optimal setzen können?

Vielen Dank!


ICH BIN VOLL AM ENDE! ICH MACH' SCHLUSS...!
GUT. ICH MUSS EH GEHEN.
ild