Das euklidische TSP Problem: Polynomiale Approximationsschemata

Vortrag von: Houssem Belloum

Blockseminar Baerenthal vom 9. bis 11. Juli 2004:

Approximationsschemata in Ablaufplanung, Graphentheorie und Geometrie

Dozent: Dr. Alexander Wolff

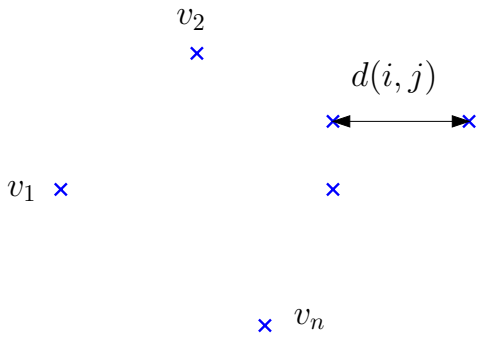
Betreuer: Marc Benkert

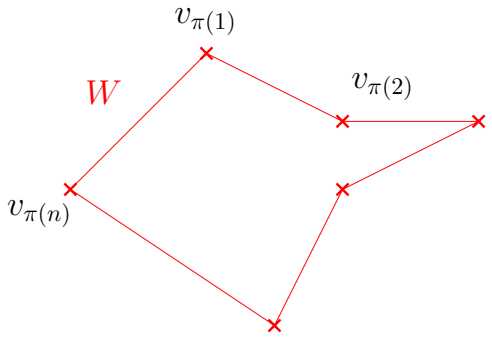
TSP: *Traveling Salesman Problem*

Gegeben: n Punkte $\{v_1, \ldots, v_n\}$ in einem beliebigen Raum $\mathcal{R}$

und eine Metrik $dist : \mathcal{R}^2 \mapsto \mathbb{R}$; $\{v_i, v_j\} \mapsto dist(v_i, v_j) = d(i, j)$.

Suche W mit: $\sum_{i=1}^{n-1} d(\pi(i), \pi(i+1)) + d(\pi(n), \pi(1))$ minimal.





E-TSP: $dist$ wird mit euklidischer Norm berechnet.

Approximationschemata

OPT: Optimaler Lösungswert für geg. Optimierungsproblem Π.

$\mathcal{A}$ mit Wert $OPT(\mathcal{A})$ ist $(1+r)$-Approximationsalgorithmus wenn:

$OPT(\mathcal{A}) - OPT \leq r \cdot OPT$, Π ist Minimierungsproblem

PTAS: Polynomial Time Approximation Scheme

Menge von $(1+\varepsilon)$-Approximationsalgorithmen, mit polynomialem Zeitaufwand in der Grösse der Eingabe für jedes $\varepsilon > 0$.

Komplexität

-TSP ist NP-schwer (Karp 1972)

-TSP-$(1+r)$-Approx. ist NP-schwer für $r > 0$ (Gonzalez 1976)

-E-TSP ist NP-schwer (Papadimitriou und and. 1977)

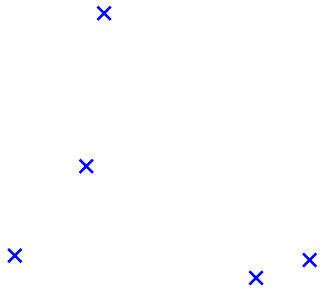
-E-TSP ist MAX-SNP-schwer für $d = O(\log n)$ (Trevisan 1997):

Es existiert $\gamma > 0$, so dass $(1+\gamma)$-Approximation NP-schwer ist.

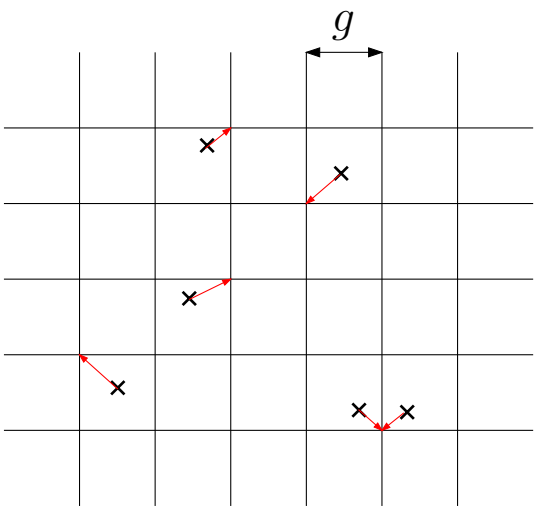
Werkzeuge und Datenstrukturen

- Perturbation
- Randomisierung
- Quadtree
- Portale
- Dynamisches Programmieren

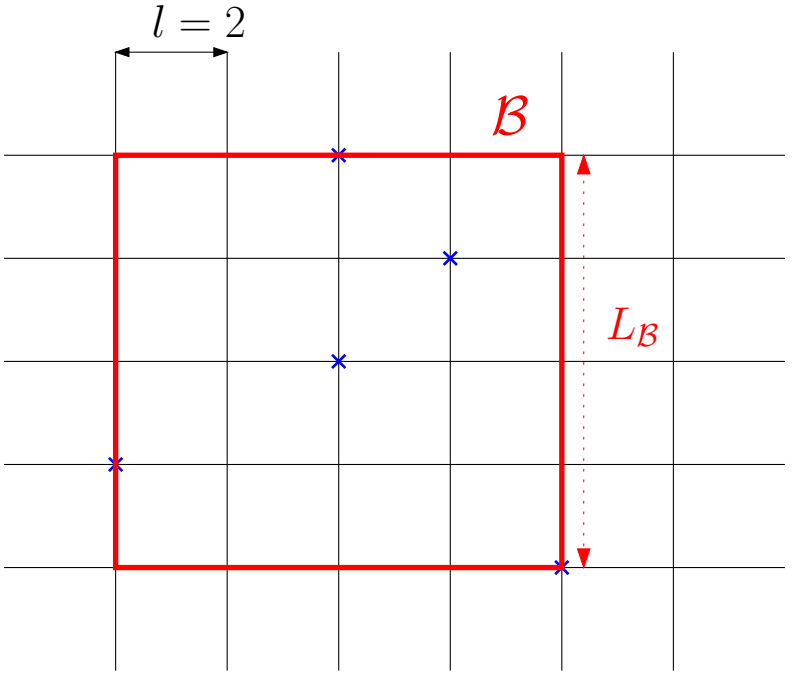
Perturbation: Beschreibung



Punktemenge



Verschiebung



Skalierung

Perturbation: Analyse

-Sei $d = \max d(i,j)$. Es gilt: $2d \leq OPT \leq nd$.

-Annahme: $\varepsilon > 2/n^{1/3}$.

wähle $g = \varepsilon \cdot d/n^{3/2}$, Distanzen durch $2g$ teilen

$\Longrightarrow$Koordinaten ganzzahlig, $l = 2$ und $L_{\mathcal{B}} \leq \frac{d}{2g} = \frac{n^{3/2}}{\varepsilon} \leq n^2/2$.

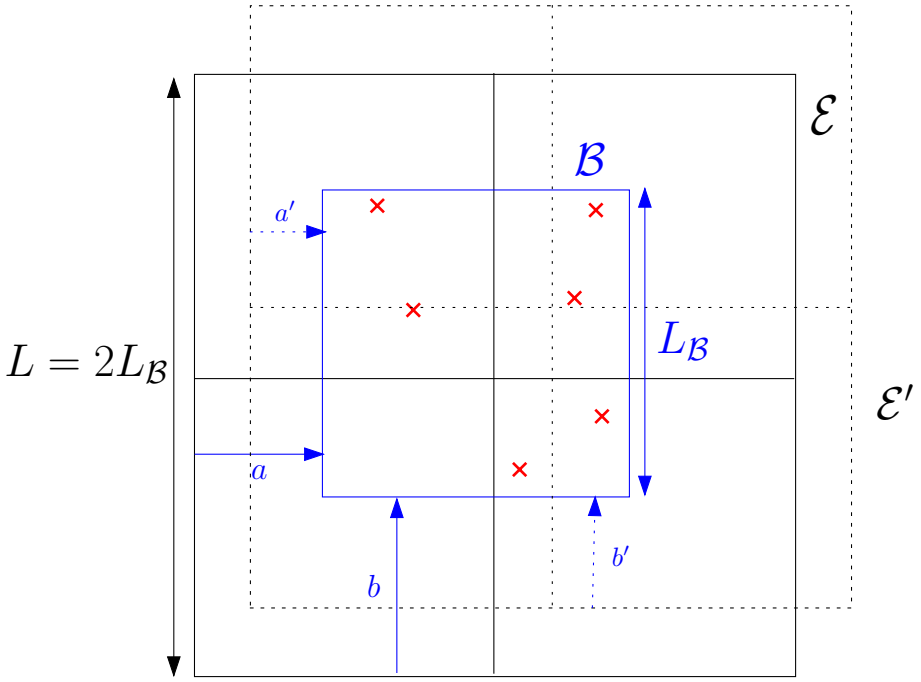
-KOSTEN(pert) $\leq n \cdot g = \varepsilon \cdot d/n^{1/2} \leq \varepsilon \cdot \frac{OPT}{2} \cdot \frac{1}{\sqrt{n}} \ll \varepsilon \cdot OPT$.

-Zeitaufwand: $O(n \log n)$.

Werkzeuge und Datenstrukturen

- Perturbation
- Randomisierung
- Quadtree
- Portale
- Dynamisches Programmieren

Randomisierung



Ziel: Konstuiere Weg der Länge OPT' so dass,

$$\underset{a,b}{E}[OPT' - OPT] \leq \frac{\varepsilon}{2} \cdot OPT$$

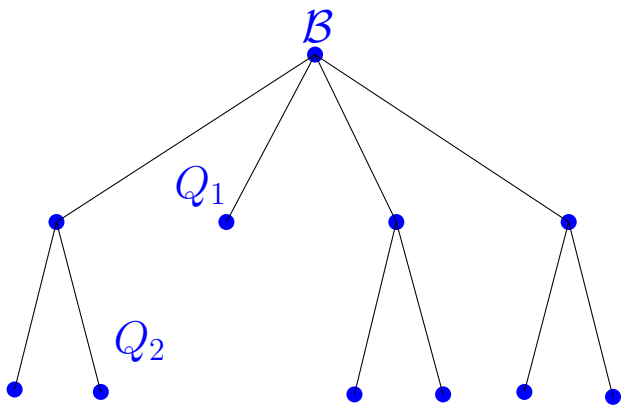
$$\overset{Markov}{\Longrightarrow} \underset{a,b}{P}[OPT' \leq (1+\varepsilon) \cdot OPT] \geq \frac{1}{2}.$$

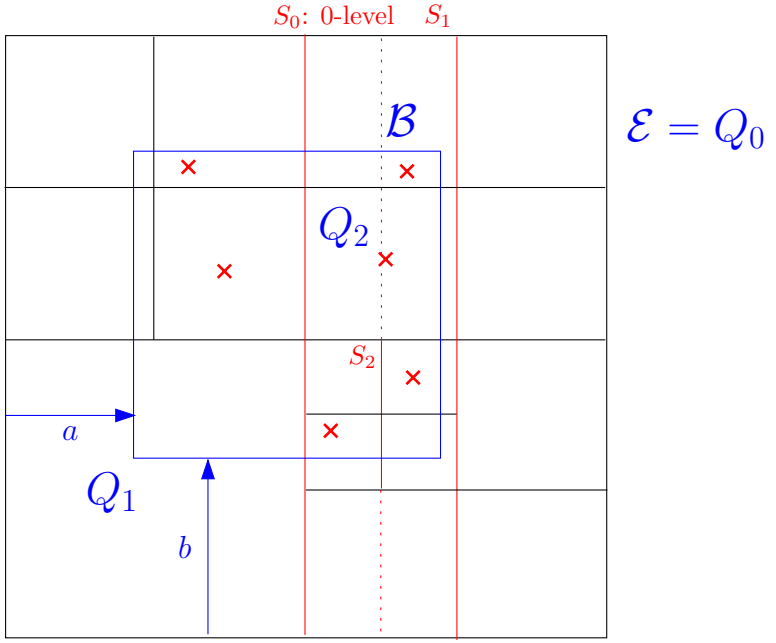
Werkzeuge und Datenstrukturen

- Perturbation
- Randomisierung
- Quadtree
- Portale
- Dynamisches Programmieren

Quadtree: Konstruktion

-Iterative Dissektion von $\mathcal{E}$ in 4 gleichgrosse Quadrate.





-2^i horizontale und 2^i vertikale i-level Linien S_i.

Quadtree: Analyse

- ♯Blätter=$O(n)$
- Tiefe des Baumes = $O(\log n)$
- ♯Quadrate=$O(n \log n)$
- Aufwand=$O(n \log^2 n)$: auf sortieren basierendes Verfahren

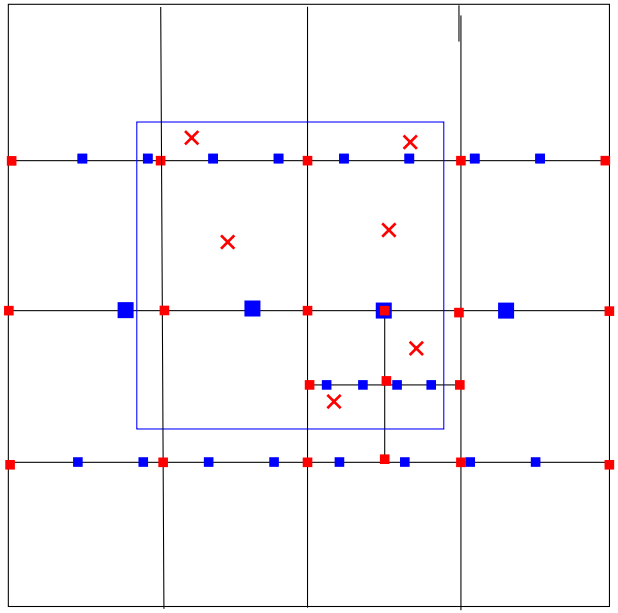
Werkzeuge und Datenstrukturen

- Perturbation
- Randomisierung
- Quadtree
- Portale
- Dynamisches Programmieren

Portale: Definition

-auf den S_i: $2^{i+1}m$ äquidistante Knoten (m: Portalparameter).

- Mit den Ecken der Q_i: Menge der Portale.



$m = 2$

Portale: Eigenschaften

-höchstens $m + 2$ Portale pro Seite von Q_i

-höchstens $4m + 4$ Portale pro Q_i

-portaltreue Tour: Tour kreuzt alle Q_i nur an Portalen

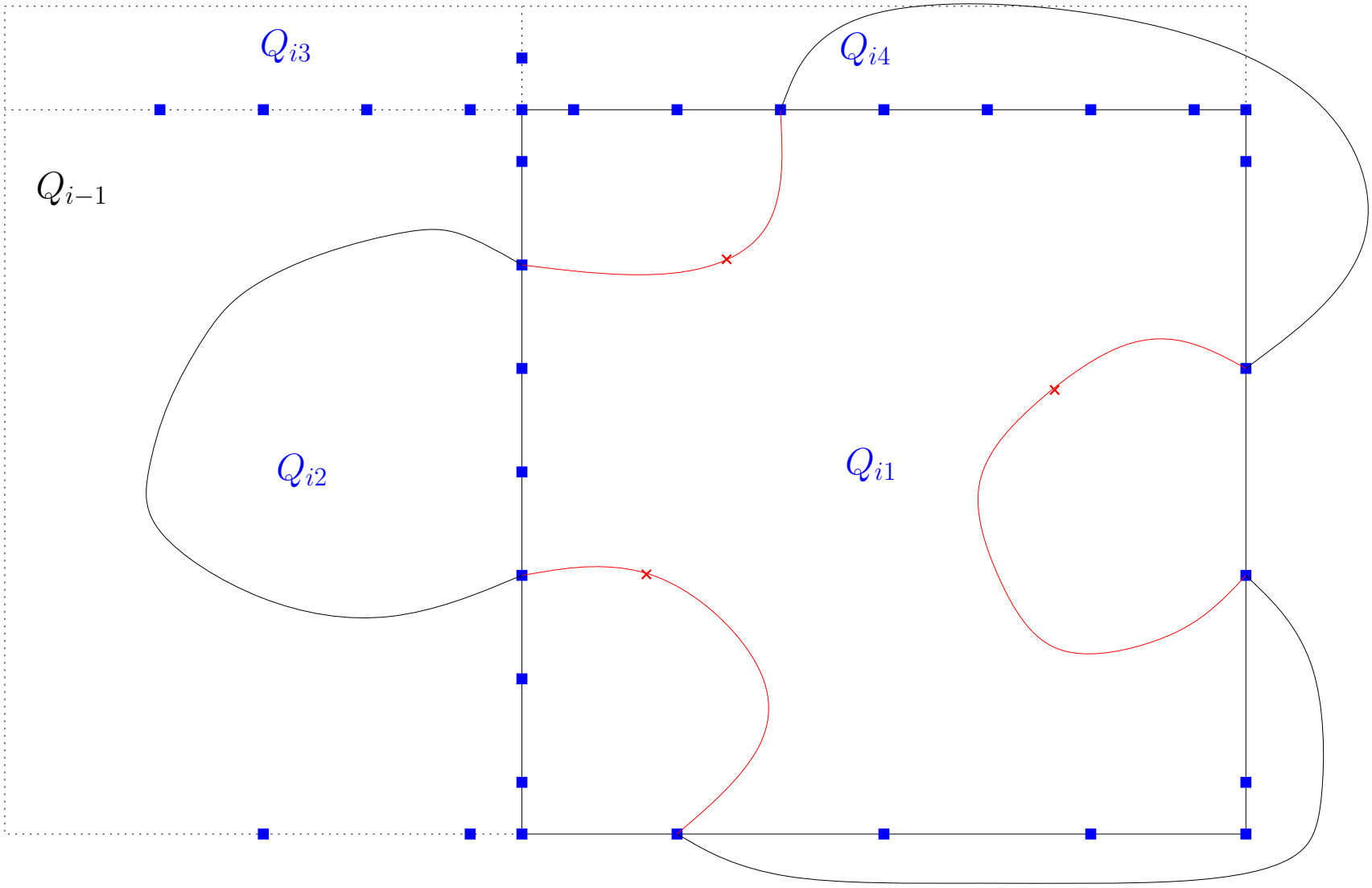
-k-leichte Tour: höchstens k Kreuzungen an jeder Seite der Q_i

-(m, k)-leichte Tour: portaltreue k-leichte Tour mit Parameter m

Werkzeuge und Datenstrukturen

- Perturbation
- Randomisierung
- Quadtree
- Portale
- Dynamisches Programmieren

Dynamisches Programmieren(DP): Vorgehensweise



DP: Grundidee

- S_i: Schnittstelle zwischen Q_i und Q_{i-1}
- Schnittstelleninfo: Portalmenge, Reihenfolge der Portale
- Quadtree, Portale $\longrightarrow$ kleinere Instanzen mit gleicher Struktur
- Für die Blätter: brute-force Algorithmus
- Für größere: Aufzählung aller Möglichkeiten, Lösungen kleinerer Instanzen konsistent zusammenzufügen

DP: Vorhersage-Tabelle

Tabelle mit Einträgen der Kosten der **optimalen** (m, k)-leichte Tour ($\bigcup$ von Pfaden) für Tripel:

1. Quadrat Q_i

2. Portalmenge $\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_4$ mit:

$$|\mathcal{P}_l| \leq k \text{ und } \sum_{l=1}^{4} |\mathcal{P}_l| = 2p \leq 4k$$

3. Reihenfolge von Ein-/Ausgängen: $\{a_1, a_2\} \cup \cdots \cup \{a_{2p-1}, a_{2p}\}$

Menge: $\mathcal{P} = \bigcup_{j=1}^{6} a_j$; **Reihenfolge:** $\{a_1, a_2\} \cup \{a_3, a_4\} \cup \{a_5, a_6\}$

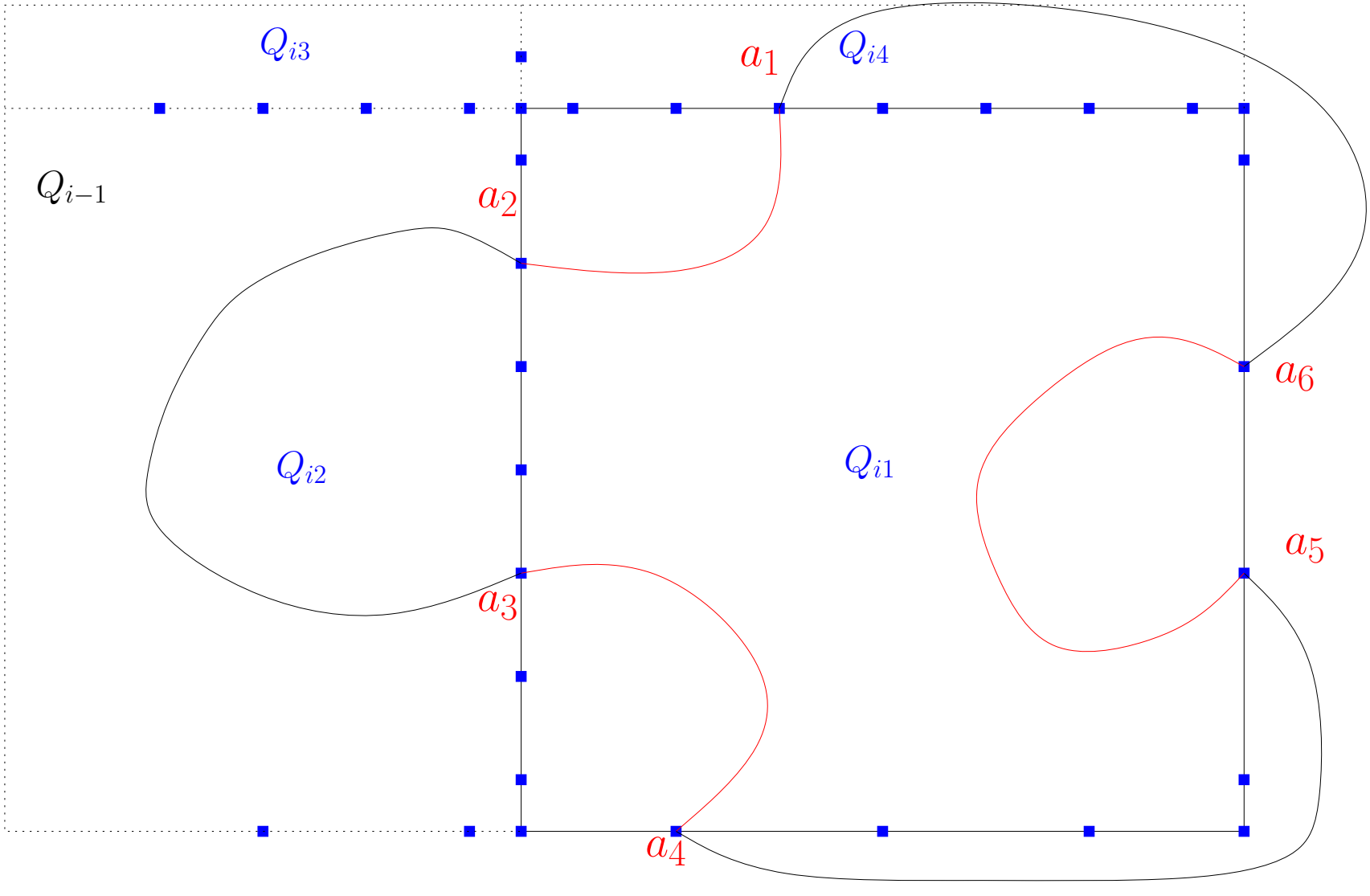


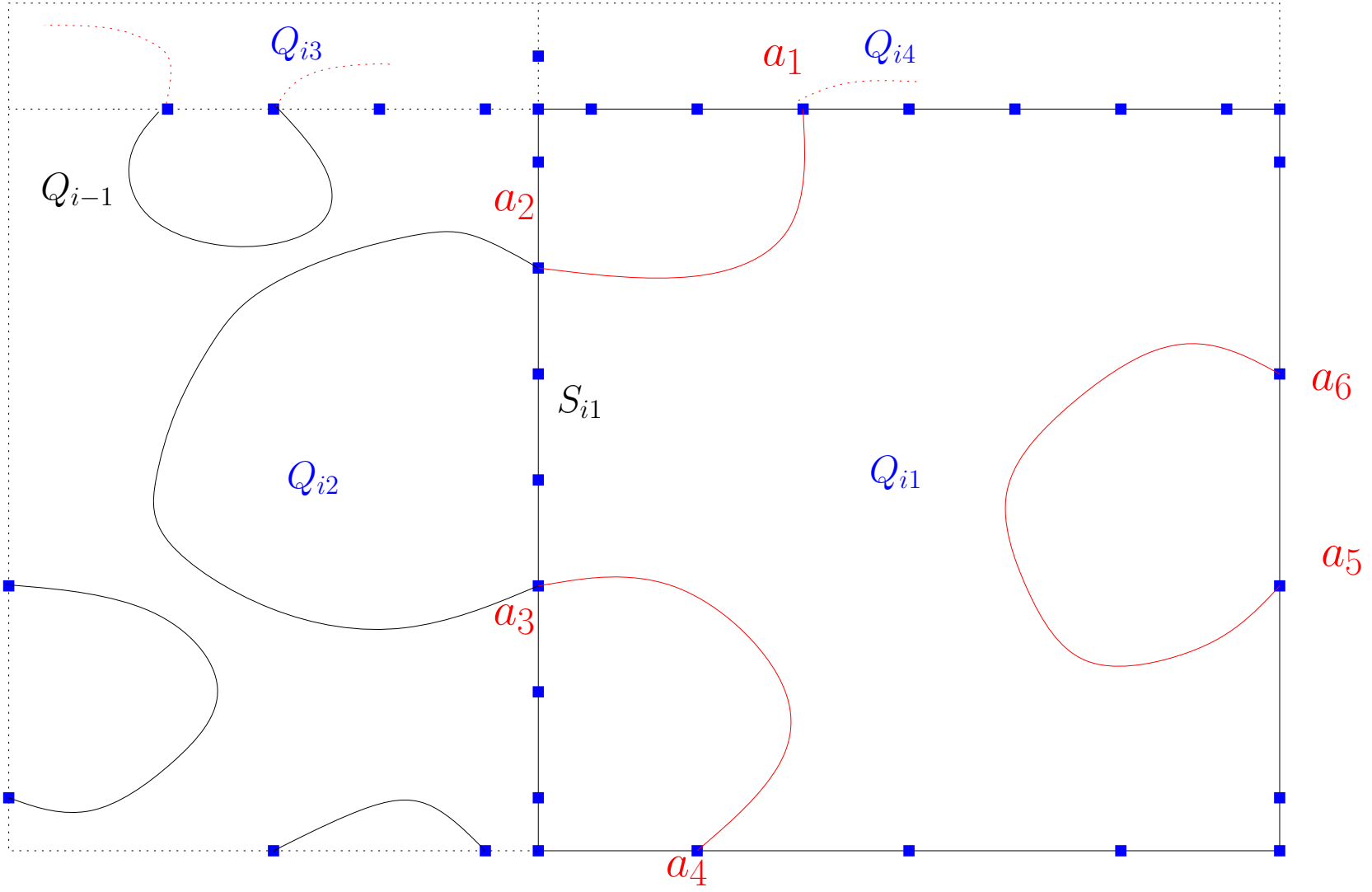
Tabelle: Komplexitätsanalyse

Tabelle enthält Lösung: Eintrag für $Q_i = \mathcal{B}$ und $\mathcal{P} = \emptyset$.

- $\sharp$Quadrate $= O(n \log n)$
- $\sharp\mathcal{P} = O((4m+4)^{4k})$: $4m+4$ Portale/Quadrat, $|\mathcal{P}| \leq 4k$
- $\sharp$Reihenfolgen $\leq (4k)!$: Permutationen der bis zu $4k$ Portale
- Tabellengrösse$=$ $O(n \log n \cdot (\mathbf{m}+\mathbf{4})^{\mathbf{4k}} \cdot (\mathbf{4k})!)$

$\Longrightarrow$ Wahl von m und besonders von k entscheidend.

induktive Vorgehensweise: $Q_i \longrightarrow Q_{i-1}$



Einträge in der Tabelle

- Blätter: Lösung in $O(k)$ (1 Knoten, bis zu $4k$ Möglichkeiten)

- Lösungen für $Q_{i1}, Q_{i2}, Q_{i3}, Q_{i4} \longrightarrow$ Lösung für Q_{i-1}:

  -Aussenseiten der Q_{ij} durch Auswahl von Tripel $\{Q_i, \mathcal{P}, R\}$ fest

  -Auswahl der Portalmenge für die 4 Innenseiten S_{ij}

  -Reihenfolge in der die Tour S_{ij} kreuzt.

Einträge: Komplexität

- ♯Portalmengen $=((m+2)^k)^4$: 4 Seiten, k Portale aus $m+2$/Seite

- ♯Reihenfolgen $= (4k)^{4k} \cdot (4k)!$ (Mögliche Wege/Portal $\times$ Permutationen)

- Zeitaufwand $= O($Tabellegrösse $\times$ Aufwand für Einträge) $= O(n \log n \cdot m^k \cdot k!)$

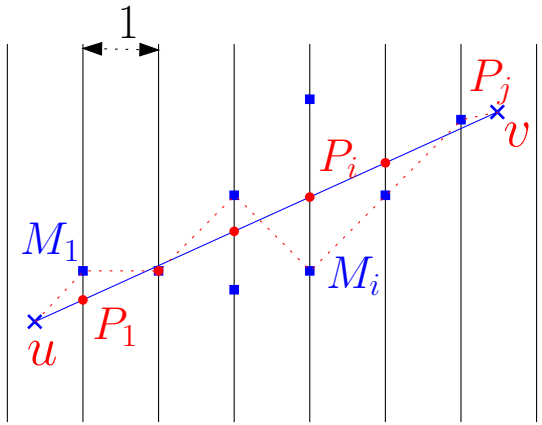
Erste Lösung

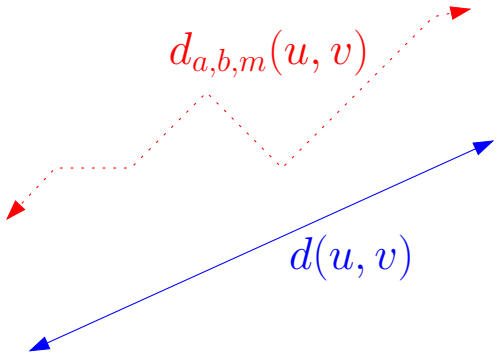
-Idee: opt. Tour $\Pi_{OPT} \longrightarrow$ opt. portaltreue Tour $\Pi_{POR} = \Pi$.

-Π_{OPT} besteht aus Kanten $(u, v) = (u, P_1), (P_1, P_2), \ldots, (P_j, v)$.

-Π_{POR} besteht aus Pfaden $(u, v) = (u, M_1), (M_1, M_2), \ldots, (M_j, v)$.

-Kosten(Π) $:= OPT_{a,b,m} = \sum_{(u,v) \in \Pi_{OPT}} d_{a,b,m}(u, v)$.





Kosten der Verschiebung: Zuschlagsargument

Lemma 1:

$$E := \underset{a,b}{E}[d_{a,b,m}(u,v) - d(u,v)] \leq \frac{2\log L}{m} \cdot d(u,v)$$

Beweis: sei δ der Umweg für die Verschiebung *eines* P_i

$\delta \leq$ Portalabstand auf Linie $S = \frac{L}{2^{i+1} \cdot m}$, falls S i-level Linie ist.

Es gilt: $\underset{a,b}{P}[S \text{ ist } i\text{-level Linie}] = \frac{2^i}{L_{\mathcal{B}}} = \frac{2^{i+1}}{L}$. (1)

$$\Longrightarrow \underset{a,b}{E}(\delta) \leq \sum_{i=0}^{\log L - 1} \frac{2^{i+1}}{L} \cdot \frac{L}{2^{i+1}\, m} = \frac{\log L}{m}.$$

$E = j \cdot E(\delta)$ mit $j \leq 2d(u,v)$ wegen Mindestabstand 2 $\square$.

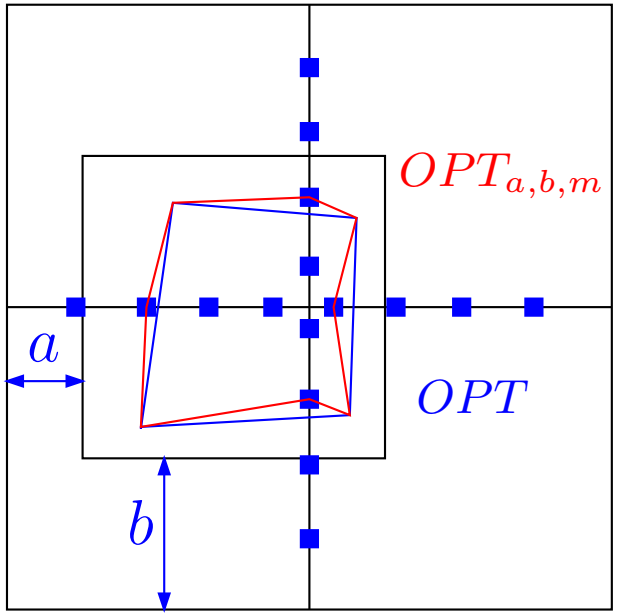
Struktur Theorem

Theorem 1:

$$\underset{a,b}{E}[OPT_{a,b,m} - OPT] \leq \frac{2\log L}{m} \cdot OPT.$$

Beweis: folgt aus Lemma 1 wegen $OPT = \sum d(u,v)$.

Folgerung: $P[OPT_{a,b,m} - OPT \leq 2 \cdot \frac{2\log L}{m} \cdot OPT] \overset{Markov}{\geq} \frac{1}{2}$.



$(1+\epsilon)$-Approximation

Erinnerung: $L \leq n^2$ aus Perturbation

Wähle $m \geq \frac{8 \log n}{\varepsilon}$

$$\Longrightarrow \underset{a,b}{E}[OPT_{a,b,m} - OPT] \leq \frac{2 \cdot \log n^2}{\frac{8 \cdot \log n}{\varepsilon}} = \frac{\varepsilon}{2} \cdot OPT$$

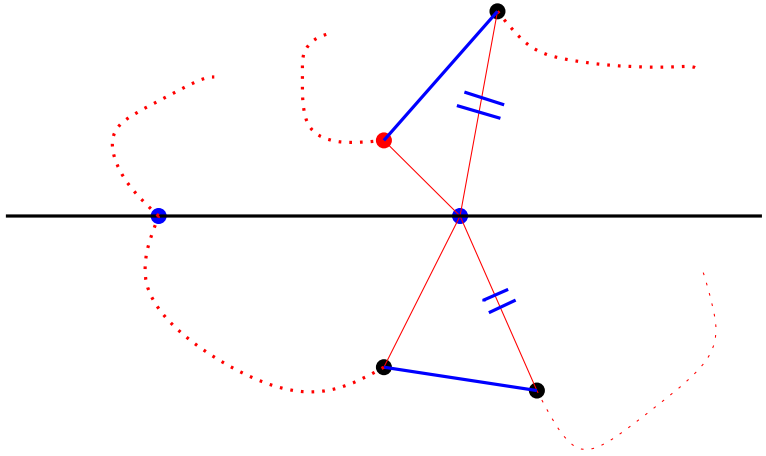
$$\Longrightarrow \underset{\mathbf{a,b}}{\mathbf{P}}\left[\mathbf{OPT_{a,b,m}} \leq (1+\varepsilon) \cdot \mathbf{OPT}\right] \geq \frac{\mathbf{1}}{\mathbf{2}}$$

Komplexitätsanalyse

Dreiecksung. $\longrightarrow$ 1 Besuch pro Portal $\Longrightarrow$ Π ist $(m, m+2)$-leicht.

$\Longrightarrow k = O(m)$.

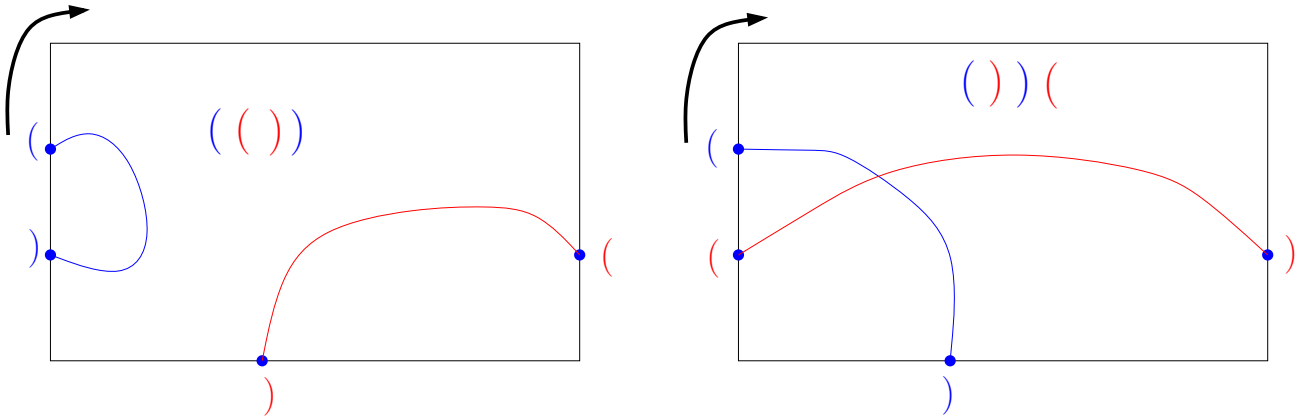
Aufwand$= O(n \log n \cdot m^k \cdot k!) = O(n \log n \cdot m^m \cdot m!)$.



Laufzeit

Π optimal $\Rightarrow$ kreuzungsfrei $\Rightarrow$Reihenfolge $\equiv$ valide Klammerung

$\Longrightarrow O(m^m \cdot m!) \rightarrow 2^{O(m)}$ (*Catalan-Zahlen*: $C_{2p} \leq 2^{2p} \leq 2^{4k}$).



Setze $m = O(\frac{\log n}{\varepsilon})$

$$\Longrightarrow \text{Aufwand} = n \log n \cdot 2^{O(\frac{\log n}{\varepsilon})} = n^{O(\frac{1}{\varepsilon})}.$$

Zum Schluss . . .

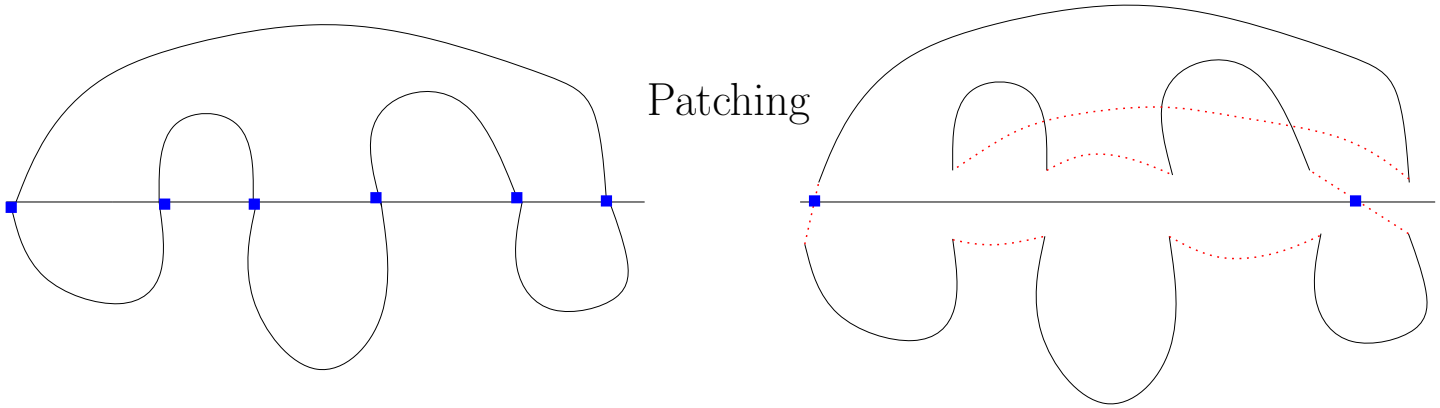
- Weg aus den Entscheidungen in der Tabelle ablesen.
- Rundgang durch verschmolzene Punkte (Perturbation) einfügen.
- Derandomisierung: alle Wahlen von a und b, n^2 viele.

Zusammenfassung

Laufzeit: $n \log n \cdot 2^{O(k)} = n^{O(\frac{1}{\varepsilon})}$.

Kann $k = O(\frac{\log n}{\varepsilon})$ verkleinert werden? $k = O(\frac{1}{\varepsilon})$ funktioniert!

Approximation: das Z.a nutzt Zulässigkeit **aller** $m + 2$ Portale.



$\Longrightarrow$ Zweiter Ansatz.

zweite Lösung: Struktur Theorem

Theorem 2:

$$\underset{a,b}{E}[OPT_{a,b,m,k} - OPT] \leq (\frac{2\log L}{m} + \frac{24}{k-5}) \cdot OPT$$

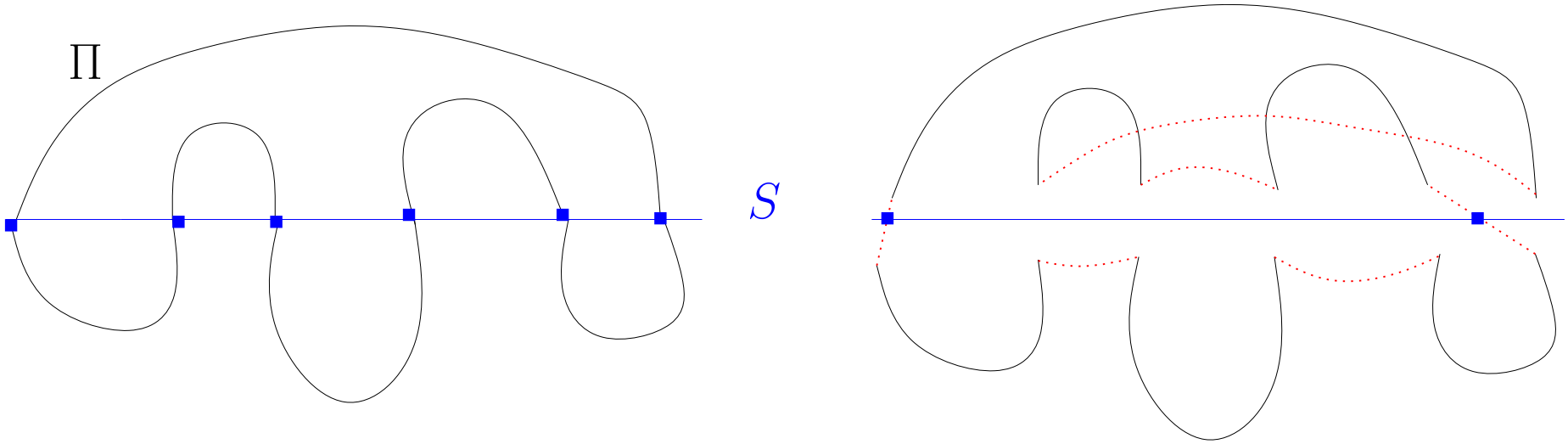
Beweisskizze:

optimale Tour $\overset{Patching}{\Longrightarrow}$ k-leichte Tour $\overset{Z.a}{\Longrightarrow}$ (m,k)-leichte

OPT $\Longrightarrow$ $\frac{24}{k-5} \cdot OPT$ $\Longrightarrow$ $(\frac{2\log L}{m} + \frac{24}{k-5}) \cdot OPT$

Patching-Lemma

Lemma 2: Sei S Segment der Länge s und Π geschlossener Weg der Länge OPT, der S mindestens 3 mal kreuzt. Aus S und Π ensteht Π', der S höchstens 2 mal kreuzt mit Länge $\leq OPT + 6s$.



Beweis Patching-Lemma

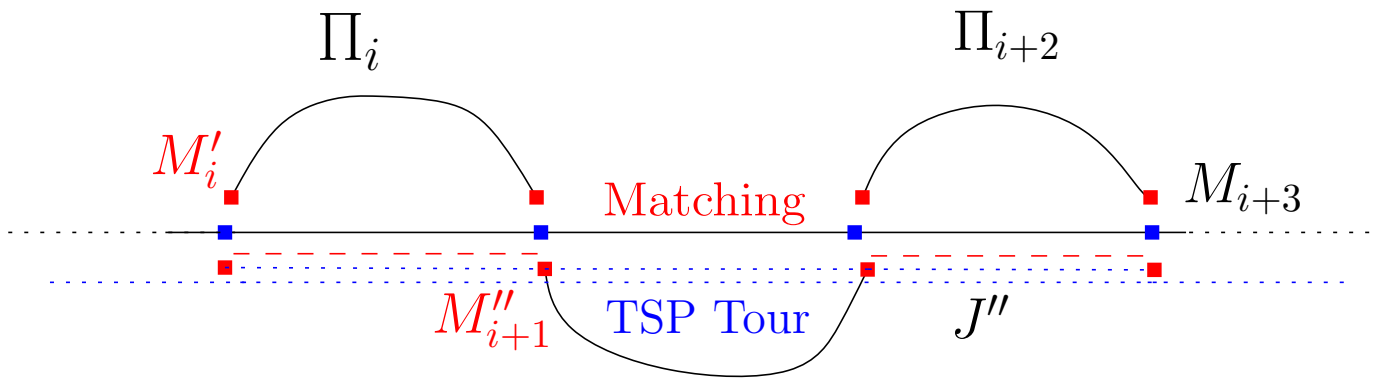
Seien M_i die t Punkte an denen S von Π gekreuzt wird.

Π zerfällt in t Wege $\Pi_1, \ldots, \Pi_t \longrightarrow$ erzeuge Kopien M_i', M_i'' von M_i

Sei $2j$ die grösste gerade Zahl $< t$.

$J = \{\text{TSP Tour von } M_1, \ldots, M_t\} \cup \{\text{min. Matching für } M_1, \ldots, M_{2j}\}$

$\Longrightarrow$ Länge $J \leq 2s + s = 3s \longrightarrow$ erzeuge 2 Kopien J' und J'' von J.

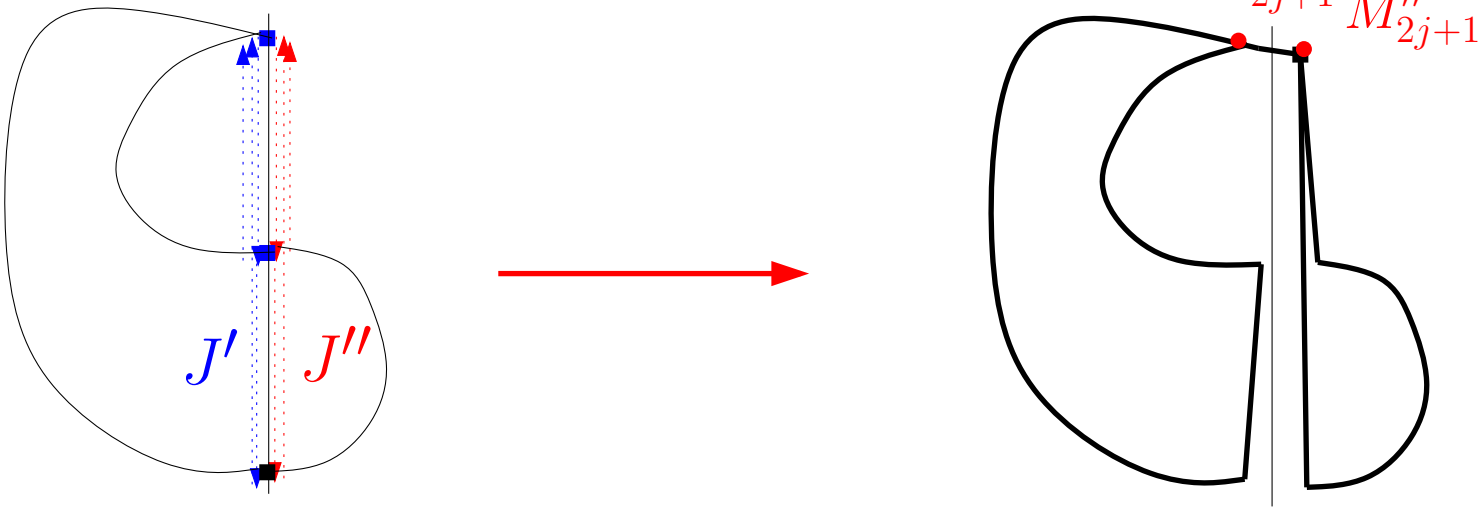


Konstruktion von Π'

$$t = 2j+1 \Rightarrow \Pi' = \bigcup_{i=1}^{t} \Pi_i \cup J' \cup J'' \cup (M'_{2j+1}, M''_{2j+1})$$

$$t = 2j+2 \Rightarrow \Pi' = \bigcup_{i=1}^{t} \Pi_i \cup J' \cup J'' \cup (M'_{2j+1}, M''_{2j+1}) \cup (M'_{2j+2}, M''_{2j+2})$$

$$\Longrightarrow \text{Länge } \Pi' \leq \text{Länge } \bigcup_{i=1}^{t} \Pi_i + 3s + 3s + 0 = OPT + 6s \quad \square$$



Hilfslemma

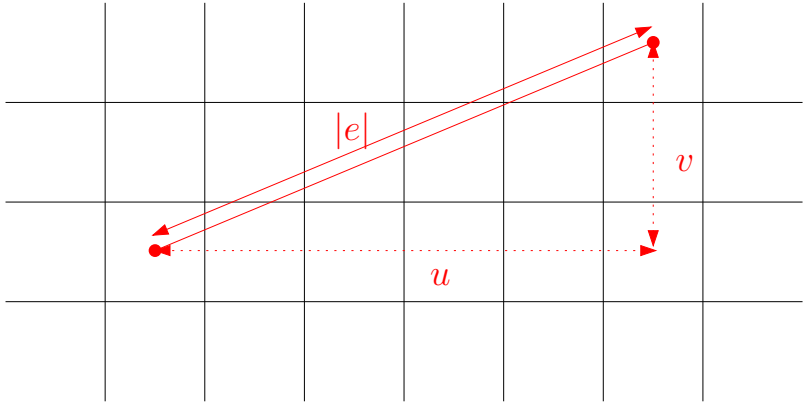
Sei l Linie eines Netzes der Weite 1 und Π ein Weg der Länge OPT.

$t(\Pi, l)$:= Anzahl der Kreuzungen von Π und l.

Lemma 3 : $$\sum_{l\ vertikal} t(\Pi, l) + \sum_{l\ horizontal} t(\Pi, l) \leq 2OPT$$

Beweis: Sei λ der Beitrag einer Kante e zur linken Seite. Es gilt:

$$\lambda \leq (u+1)+(v+1) \leq \sqrt{2(u^2+v^2)}+2 \leq \sqrt{2|e|^2}+2 \leq 2|e| \square$$



Beweis Struktur Theorem

Definition: Ein Segment von l heißt durch Π *belastet* falls es mindestens $s+1$ mal von Π gekreuzt wird, ansonsten *entlastet*.

Idee: optimale Tour $\longrightarrow$ s-leichte Tour (bez. den Vertikalen)

$\longrightarrow$ k-leichte Tour (bez. allen Linien)

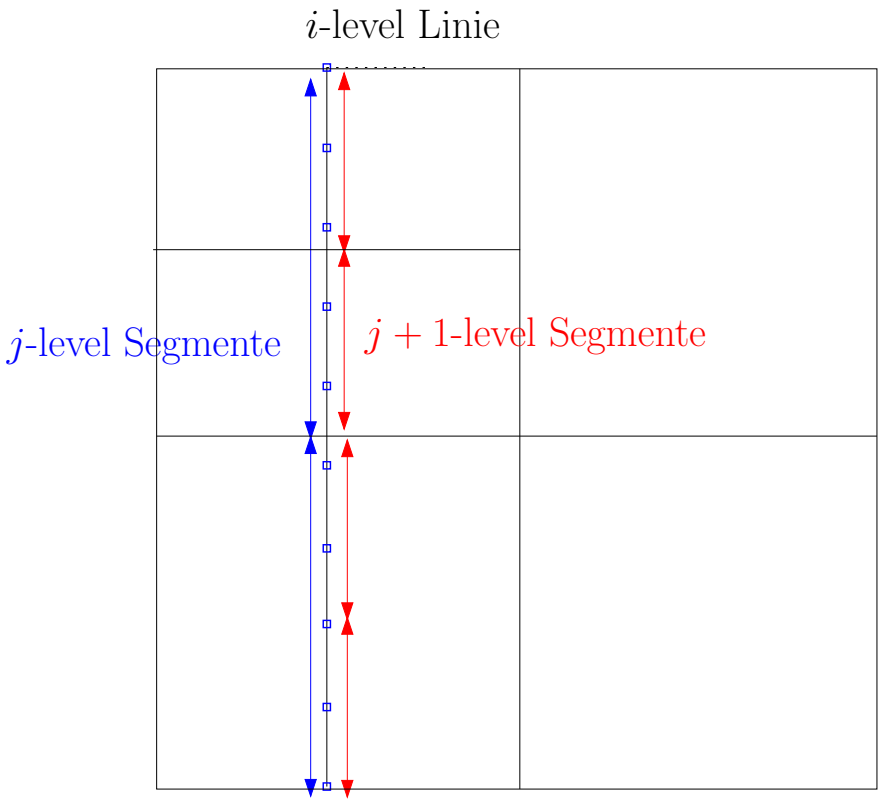
Zu zeigen: -Kosten für eine s-leichte Tour $= \frac{24}{s-1} \cdot OPT$.

-Kosten für eine k-leichte Tour $= \frac{24}{k-5} \cdot OPT$.

s-leichte Tour

Sei l vertikale i-level Linie. (analog für die Horizontalen)

l zerfällt in 2^j j-level Segmente für $j \geq i$.



$proc(i)$

Ziel: höchstens s Kreuzungen auf jedem Segment.

Prozedur $proc(i)$:

($j := \log L - 1$)

solange ein i-level Segment l belastet ist

-Patching auf j-level: $\geq s + 1$ Kreuzungen $\longrightarrow \leq 2$

-$j := j - 1$

Gib Π zurück

Kostenabschätzung

Fiktive Prozedur $proc(0, l)$: Patching auf l bis Level 0.

$X_{l,j}(b) := \sharp$ (belastete j-level Segmente auf l, die $proc(0, l)$ patcht)

Behauptung:

$$\sum_{j \geq 0} X_{l,j}(b) \leq \frac{t(\Pi, l)}{s - 1} \tag{2}$$

Beweis: $t(\Pi, l)$ Kreuzungen von Π und l. Für jedes j ersetzt das

Patching-Lemma mindestens $s + 1$ Kreuzungen mit höchstens 2.

$\Rightarrow s - 1$ Kreuzungen weniger pro Schritt.

Eigentliche Kosten

j-level Segmente haben die Länge $L/2^j$. Patching-Lemma liefert:

$$\text{Kosten}(proc(0,l)) \leq \sum_{j \geq 1} X_{l,j}(b) \cdot \frac{6L}{2^j} \qquad (3)$$

Eigentliche Kosten werden von a bestimmt. Aus (3) erhalten wir:

$$\text{wenn } l \text{ } i\text{-level Linie ist: } \text{Kost}(proc(0,l)) \leq \sum_{j \geq i+1} X_{j,l}(b) \cdot \frac{6L}{2^j} \qquad (4)$$

Sei $Y_{l,a}$:= Kost$(proc(0,l))$, wenn a horizontaler Shift ist.

Erinnerung: $\underset{a,b}{P}[S \text{ ist } i\text{-level Linie}] = \frac{2^i}{L_{\mathcal{B}}} = \frac{2^{i+1}}{L}$ (1)

$$\sum_{j \geq 0} X_{l,j}(b) \leq \frac{t(\Pi,l)}{s-1} \qquad (2)$$

Kosten der s-leichten Tour

für jedes b gilt: $E_a[Y_{l,a}] \leq \frac{12t(\Pi, l)}{s-1}$

daraus folgt: $E_a[\sum_l Y_{l,a}] = \sum_l \frac{12t(\Pi, l)}{s-1}$

$$\leq 24\frac{OPT}{s-1}$$ aus Lemma 3

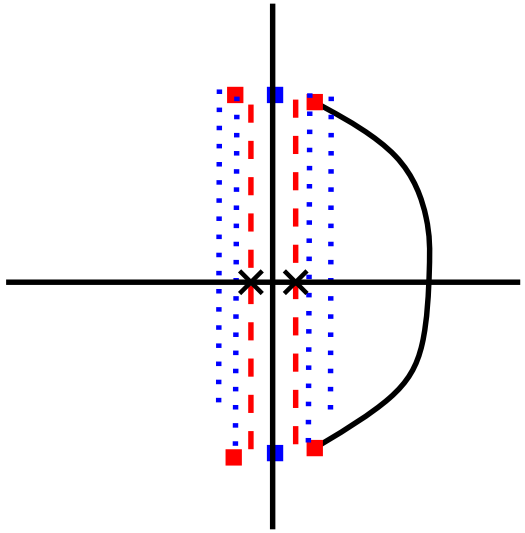
$$= \frac{24}{s-1} \cdot OPT$$

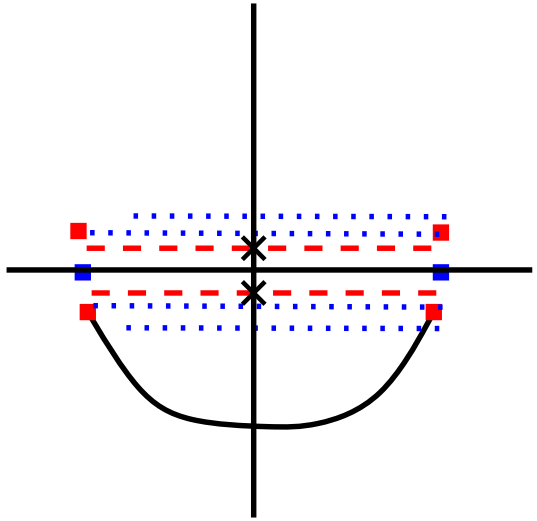
Interferenz der Entlastungen

Behauptung: Entlastung einer Vertikalen $\longrightarrow$ höchstens 2

zusätzliche Kreuzungen auf einer Horizontalen und umgekehrt.

Idee: Patching kostenlos, da kein horiz. (bzw. vert.) Abstand.





k-leichte Tour

Betrachte jedes Paar {Vertikale, Horizontale}.

$\Longrightarrow$Für jede Seite eines Quadrats: bis zu $s+4$ Kreuzungen (s für das Innere, 2 pro Ecke)

Setze $k = s + 4$

$\Longrightarrow$$k$-leichte Tour mit Kosten $= OPT + \frac{24}{k-5} \cdot OPT$ $\square$

Approximation und Laufzeit

k-leichte Tour $\stackrel{Z.a}{\Longrightarrow}$ (m,k)-leichte portaltreue Tour

$$\underset{a,b}{E}[OPT_{a,b,m,k} - OPT] \leq (\frac{2\log L}{m} + \frac{24}{k-5}) \cdot OPT$$

Folgerung: Wähle $m \geq \frac{16\log n}{\varepsilon}$, $k \geq \frac{96}{\varepsilon} + 5$

$$\underset{a,b}{E}[OPT_{a,b,m,k} - OPT] \leq (\frac{2\log n^2}{\frac{16\log n}{\varepsilon}} + \frac{24}{\frac{96}{\varepsilon} + 5 - 5}) \cdot OPT = (\frac{\varepsilon}{4} + \frac{\varepsilon}{4}) \cdot OPT$$

Laufzeit$= O(n\log n \cdot m^k \cdot k!) = O(n \cdot (\log n)^k)$

Ergebnisse

Approximation:

$$\underset{a,b}{P}[OPT_{a,b,m,k} \leq (1+\varepsilon) \cdot OPT] \geq 1/2$$

Laufzeit:

$$n \cdot (\log n)^{O(1/\varepsilon)}$$

Zusammenfassung

Perturbation: gerundete Instanz (Diskretisierung).

Randomisierung: Erwartung $\longrightarrow$ Wahrscheinlichkeit.

Quadtree: kleinere Instanzen gleicher Struktur.

Portale: Schnittstellen zum Zusammenfügen von Teillösungen.

$\Longrightarrow$**Dynamisches Programmieren**: kein redundantes Rechnen.

Parameter m und k: zur Steuerung von Approximation und Laufzeit.

Meilensteine in der TSP-Geschichte

1973: effektive Heuristiken (K-OPT, Lin-Kernighan)

1976: 3/2-Approximation in polynomieller Zeit von Christofides

1977: probabilistische Heuristik von Karp und Held (4/3-Approx.)

1988: optimale Lösung in $2^{O(\sqrt{n})}$ von Smith

1996: PTAS in $n^{O(1/\varepsilon)}$ und später in $n(\log n)^{O(1/\varepsilon)}$ von Arora

1996: PTAS von Mitchell in $n^{O(1/\varepsilon)}$ aber nur für $\mathbb{R}^2$

Wie praktikabel sind Aroras Ansätze?

1996 gar nicht: Aufwand= $n \cdot (\log n)^c$ mit c =30 bis 100

Bessere Resultate mit K-OPT oder Karps Heuristik.

1998: Parallelisierbarkeit. Effiziente NC-Implementierung:

Laufzeit poly($\log n$) mit poly(n) Prozessoren.

Einsatz in strukturierten Instanzen (Verkehr).

metatheoretische Überlegungen

Reduzierung der Übergänge zwischen „Regionen“(Quadrate)

$\Longrightarrow$lokaler Austausch als mächtiges Werkzeug.

Gute Heuristiken (K-OPT) auch, aber kein PTAS.

K-OPT: *Top-down*-Algorithmus mit backtracking $\Rightarrow$Redundanz!

DP-Schlüsselidee: disjunkte Regionen, Lösen von $Q_i \rightarrow$ „abhacken“.

$\Longrightarrow$entspricht Variablenelimination (wie beim Gausschen Vefahren).

IP Formulierung

Definiere $x_{ij} = 1$ falls $v_i, v_j \in \Pi$, ansonsten 0.

IP Formulierung:

Minimiere lin. Funktion: $\sum_{i=1}^{n} \sum_{j=i+1}^{n} d(i,j) \cdot x_{ij}$

Nebenbedingungen:

$$\sum_{(i=k)\vee(j=k)} x_{ij} = 2, \qquad l \leq k \leq n,$$

$$\sum_{i \in S,\, j \in S} x_{ij} \geq 2, \quad \varnothing \neq S \subseteq \{1, 2, \ldots, n\},$$

und (ganzzahlig!) $x_{ij} \in \{0, 1\}, \qquad 1 \leq i < j \leq n.$

Bearbeitung eines Quadrats $\Longleftrightarrow$ Elimination einer Menge von x_{ij}.

Hintergrund zur Randomisierung

PCP: Probabilistically Checkable Proofs

Aroras Ziel:

$$\underset{a,b}{P}[OPT' \leq (1+\epsilon)\cdot OPT] < \frac{1}{2} \Longrightarrow (1+\epsilon)\text{- Approximation ist NP-schwer}$$

Neue Charakterisierung von NP:

Eine Sprache L liegt in NP falls ein Verifizierer V für $x \in L$ $\log n$ zufällige Bits τ einliest und folgendes erfüllt:

$$x \in L \Longrightarrow \underset{\tau}{P}[V \text{ akzeptiert } x] = 1$$

$$x \notin L \Longrightarrow \underset{\tau}{P}[V \text{ akzeptiert } x] < \frac{1}{2}$$

Erwartung von $Y_{l,a}$

Für jedes b gilt:

$$E_a[Y_{l,a}] = \sum_{i\geq 1} \frac{2^{i+1}}{L} \cdot (\text{Kosten von } l), \text{ wenn } l \; i\text{-level Linie ist.} \qquad \text{aus (1)}$$

$$\leq \sum_{i\geq 1} \frac{2^{i+1}}{L} \cdot \sum_{j\geq i+1} X_{l,j}(b) \cdot \frac{6L}{2^j} \qquad \text{aus(4)}$$

$$= 6 \cdot \sum_{j\geq 1} \frac{X_{l,j}(b)}{2^j} \cdot \sum_{i\leq j-1} 2^i$$

$$= 6 \cdot \sum_{j\geq 1} \frac{X_{l,j}(b)}{2^j} \cdot (2^j - 1)$$

$$\leq 6 \cdot \sum_{j\geq 1} 2 \cdot X_{l,j}(b)$$

$$\leq \frac{12t(\Pi, l)}{s-1} \qquad \text{aus (2)}$$