§4 Constrained shortest paths (CSP)

4.1 Komplexität

$\boxed{\text{4.1 Proposition: Das CSP ist schwach NP-vollständig}}$

Beweis: Reduktion von PARTITION

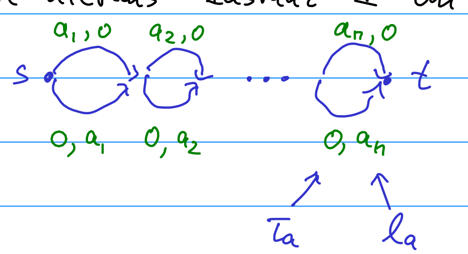
Geg: $a_1, \dots, a_n \in \mathbb{N}$ mit $\sum a_i = 2b \qquad b \in \mathbb{N}$

Frage: $\exists$? Indexmenge $I \subseteq \{1, \dots, n\}$ mit $\sum_{i \in I} a_i = b$

Sei I Instanz von PARTITION

Konstruiere hieraus Instanz I' von CSP

G:



$L = b$

Frage: $\exists$? Weg P von s nach t mit $\tau(P) \le L$, $\ell(P) \le L$

Typischer Weg

Kanten oben $\to \tau(P)$

Kanten unten $\to \ell(P)$

$$\ell(P) + \tau(P) = 2b$$

$\Rightarrow$ Weg hat $\ell(P) \le b$, $\tau(P) \le b$ $\Leftrightarrow$ Weg beschreibt eine Lösung von PARTITION $\square$

$\boxed{\text{4.2 CSP kann in pseudopolynomieller Zeit gelöst werden}}$

4.1 Beweis: $\boxed{\text{Aufgabe 4.1}}$

4.2 Lösungsansätze

1) Beasley & Christofides '89 : Lagrange Relaxation und Branch & Bound
2) Labeling Algorithmus (erweiterter Dijkstra)
   Verwandtschaft zu Pareto-optimalen Wegen

3) Geometrisch (Mehlhorn & Ziegelmann 2000)

4.2.1 Der Ansatz von Beasley & Christofides

Branch & Bound Baum $\hat{=}$ IP - Formulierung

↳ untere Schranken durch Lagrange Relaxation

$$\min \tau(P)$$
$$\text{so dass } \ell(P) \leq L \quad \text{in Zielfkt.}$$
$$P \text{ ist } s,t\text{-Weg}$$

$$(LR_\mu) \quad \min \underbrace{\tau(P) + \mu(\ell(P) - L)}_{\sum_{a\in P}(\tau_a + \mu \ell_a) - \mu L} =: \Lambda(\mu) \qquad \mu \geq 0$$

$-\mu L$ konstant unabh. vom Weg

P s,t-Weg

↓

Kürzeste Wege Problem mit veränderten Kantenkosten $\tau_a + \mu \ell_a$

Sei P^* opt. Lösung des CSP und $\tau^* := \tau(P^*)$

Dann gilt (1) $\Lambda(\mu) \leq \tau^* \quad \forall \mu$

Opt. für LR_μ ist untere Schranke für Opt von CSP

(2) $\Lambda^* := \underbrace{\max_{\mu \geq 0} \Lambda(\mu)}_{} \leq \tau^*$

bestmögliche untere Schranke durch Lagrange Relaxation

mit Subgradienten verfahren

- Ist P_{μ_0} kürzester Weg in LR_{μ_0}, so ist $\ell(P_{\mu_0}) - L$ (relaxierte Bed.) ein Subgradient in μ_0 an $\Lambda(\mu)$
- Gehe Schritt in diese Richtung $\mu^{neu} = \mu^{alt} + \Theta(\ell(P_{\mu^{alt}}) - L)$

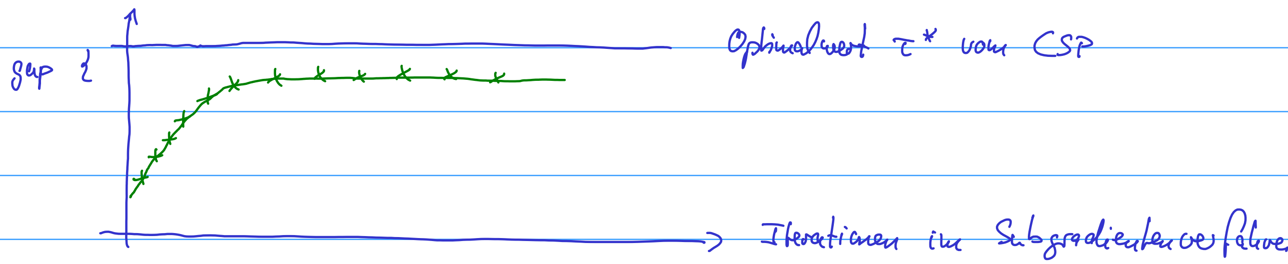
$\mu^{alt} = \mu_0$

- neue Kantenbewertung $\tau_a + \mu^{neu} l_a$

Unterschiede in Kantenbewertung pro Iteration $\underbrace{(\mu^{neu} - \mu^{alt})}\, l_a$

dieselbe Zahl für alle Kanten
ungünstig

typischer Verlauf:



unbefriedigend

4.2 Aufgabe 4.2

$$\left.\begin{array}{l} P \text{ Optimallösung von } (LR_\mu) \\ \text{mit } l(P) \leq L \end{array}\right\} \overset{?}{\Longrightarrow} P \text{ Opt. für CSP}$$

klären

4.3 Aufgabe 4.3

$\tau^* - \Lambda^*$ heißt Dualitätslücke. I.A. ist die Lücke positiv

Opt. Wert von CSP

Opt. Wert der LR_μ für bestes μ

4.2.2 Pareto-optimale Wege

Geg: $G = (V, A)$ $s, t \in V$

Für jede Kante a ein Bewertungsvektor $\lambda(a) = \begin{pmatrix} \lambda_1(a) \\ \vdots \\ \lambda_r(a) \end{pmatrix} \geq 0$

Länge $\lambda(P)$ eines Weges ist $\lambda(P) := \sum_{a \in A} \lambda(a)$ (Vektoraddition)

$$= \begin{pmatrix} \sum_{a \in P} \lambda_1(a) \\ \vdots \\ \sum_{a \in P} \lambda_r(a) \end{pmatrix}$$

hier allgemeinere Operationen möglich

Σ, max, ...

Gesucht: alle Pareto-optimalen Wege

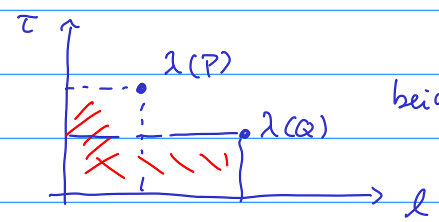
P heißt Pareto-optimal

$:\Longleftrightarrow$ $\nexists$ Weg P' mit $\lambda(P') \leq \lambda(P)$ und $\lambda_i(P') < \lambda_i(P)$ für ein i

komponentenweise

(keine Dominanz)

Bsp.:



beide Pareto-optimal

Der Dijkstra Algorithmus für Pareto-optimale Wege

Wiederholung: Dijkstra für kürzeste Wege $(\lambda_a \geq 0)$

- berechnet für jeden Knoten eine Distanz $d[v]$ $\stackrel{!}{=}$ Länge kürzeste Weg von s zu v + einen Knoten Vorgänger$[v]$, der Vorgänger von v auf kürzesten Weg von s zu v ist
- Werte $d[v]$, Vorgänger$[v]$ zunächst vorläufig
  Im Algo werden Knoten "markiert", für markierte Knoten sind diese Werte endgültig

Initialisierung : $d[v] := \begin{cases} 0 & v = s \\ \lambda_{(s,v)} & \text{falls Kante } (s,v) \text{ exist.} \\ \infty & \text{sonst} \end{cases}$

Vorgänger $[v] := \begin{cases} s & \text{falls } (s,v) \text{ Kante} \\ nil & \text{sonst} \end{cases}$

nur s markiert

Hauptschleife

while $\exists$ unmarkierten Knoten do
    wähle unmarkierten Knoten v mit kleinstem $d[v]$
    markiere v
    for alle Kanten (v,w) mit unmarkiertem w do
        if $d[v] + \lambda_{(v,w)} < d[w]$ then
            $d[w] := d[v] + \lambda_{(v,w)}$
            Vorgänger $[w] := v$
        end if
    end for
end while

Variation des Dijkstra Algo für Pareto-optimale Wege [Theorie 95]

statt $d[v]$ r-dim Vektoren $d[v] = \begin{pmatrix} d_1[v] \\ \vdots \\ d_r[v] \end{pmatrix}$ $d_k[v]$ bzgl. λ_k

In jedem Knoten mehrere $d[v]$ möglich
$\hat{=}$ bisher ermittelten Pareto-optimalen s,v-Wegen

Initialisierung $d[v] := \begin{cases} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} & \text{falls } v = s \\ \lambda_{(s,v)} & (s,v) \text{ Kante} \\ \begin{pmatrix} \infty \\ \vdots \\ \infty \end{pmatrix} & \text{sonst} \end{cases}$

Markierung Knoten $\longrightarrow$ Markierung von Vektoren $d[v]$

Regel: Wähle lexikographisch kleinsten unmarkierten Vektor $d[v]$
als nächsten zu markierenden Vektor

Aktualisierung der $d[w]$

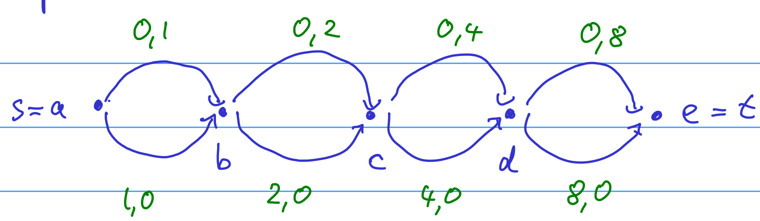
Betrachte zu gewählten $d[v]$ alle Kanten (v,w)
nehme $d[v] + \lambda_{(v,w)}$ zu den bereits in w abgespeicherten Vektoren
hinzu und streiche nicht Pareto-optimale und aktualisiere
ggf Vorgänger$[d[w]]$ $\leftarrow$ jetzt haben Vektoren Vorgänger

Anfangs nur $d[s]$ markiert

Hauptschleife

<u>while</u> $\exists$ unmarkierten Vektor $d[v]$ <u>do</u>
  wähle lexikographisch kleinsten unmarkierten Vektor $d[v]$
  markiere diesen Vektor
  Sei v der zugehörige Knoten
  <u>for</u> alle Kanten (v,w) <u>do</u>
    nehme $d[w] := d[v] + \lambda_{(v,w)}$ zu den Vektoren von w hinzu
    streiche nicht Pareto-optimale bei w
    falls $d[w]$ nicht gestrichen wird, so setze Vorgänger$[d[w]] := d[v]$
  <u>end for</u>
<u>end while</u>

Bsp:



Pareto Optima

a	b	c	d	e
(0,0)*	(0,1)*	(0,3)*	(0,7)*	(0,15)*
	(1,0)*	(2,1)*	(4,3)	(8,7)
		(1,2)*	(1,6)*	(1,14)*
		(3,0)	(5,2)	(9,6)
			(2,5)	(2,13)
			(6,1)	(10,5)
			(3,4)	(3,12)
			(7,0)	(11,4)
				(4,11)
				(12,3)
				(5,10)
				(13,2)
				(6,9)
				(14,1)
				(7,8)
				(15,0)

* $\hat{=}$ markiert

$\longrightarrow$ $\hat{=}$ Vorgänger

exponentielles Wachstum der # der Pareto-Optima

$\Rightarrow$ exponentielle Algorithmus

Korrektheit analog zum normalen Dijkstra Algo per Induktion nach # Markierungen

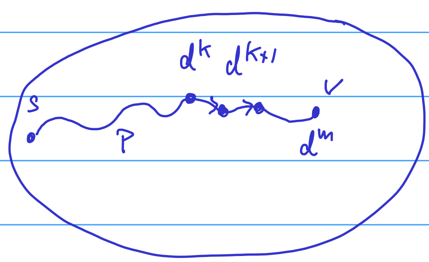
Zeigen folgende Invariante:

> Sobald Vektor markiert, gehört er zu der Menge der Pareto Optima im zugehörigen Knoten

Beweis: Betrachte den Zeitpunkt, in dem $d[v]$ zur Markierung gewählt wird

Ann: $d[v]$ nicht Pareto-optimal in v

$\Rightarrow \exists$ Pareto-opt Weg P von s nach v mit $\lambda(P) \lneq d[v]$



> P Pareto-optimal, $\lambda \geq 0$
> $\Rightarrow$ Anfangsstücke von P ebenfalls Pareto-optimal

(*)

Seien $s = v_1, v_2, \dots, v_m = v$ die Zwischenknoten auf P
und $d^1, d^2, \dots, d^m$ die zugehörigen Pareto-Optima,
also $d^i = d^{i-1} + \lambda_{(v_{i-1}, v_i)}$
Dann ist $\lambda(P) = d^m \lneqq d[v]$

Sei d^k der letzte Vektor auf P, der bereits markiert ist zum Zeitpunkt der Auswahl von $d[v]$ (exist., da $d^1 = d[s]$ bereits markiert)
$\Rightarrow$ $k < m$, da sonst $d^m \lneqq d[v]$ bereits bei v markiert würde und $d[v]$ gestrichen würde

$\Rightarrow$ Zum Zeitpunkt der Markierung von d^k wird $d^{k+1} = d^k + \lambda_{(v_k, v_{k+1})}$ zu v_{k+1} hinzugefügt (und ist Pareto-optimal wegen $(*)$) und bleibt unmarkiert bis zur Wahl von $d[v]$ (nach Wahl des Index k)

$\Rightarrow$ Dann ist $d^{k+1} \leq d^m \lneqq d[v]$

$\Rightarrow$ $d^{k+1} <_{lex} d[v]$ und d^{k+1} unmarkiert
$\Rightarrow$ Widerspruch zur Auswahlregel
$\Rightarrow$ jeder markierte Vektor ist Pareto-optimal

Analoger Beweis: jedes Pareto-Optimum wird im Algo konstruiert $\square$

Frage: Wie groß wird die # der Pareto Optima
Annahme $\lambda_k(a) \in \mathbb{Z}_+$
$L_k(v) :=$ maximale Weglänge (elementarer Weg) von s nach v bzgl. λ_k
$\text{pareto}(v) :=$ # Pareto Optima in v

$\Rightarrow$ pareto(v) ist polynomiell in n, falls $\lambda_k(a)$ polynomiell in n und r fest ist $\Rightarrow$ Pareto Dijkstra polynomiell denn: $\mathrm{pareto}(v) \leq \prod_{k=1}^{r} L_k(v)$ ← polynomial unter den obigen Annahmen

etwa: $\lambda_k(a) \leq n^p \Rightarrow L_k(v) \leq (n-1)\, n^p \leq n^{p+1}$

$\Rightarrow \mathrm{pareto}(v) \leq n^{(p+1)r}$

oder $\underset{\substack{\uparrow \\ \text{erfüllt in Straßennetzen}}}{\lambda_k(a) \leq L_0} \Rightarrow \mathrm{pareto}(v) \leq ((n-1)L_0)^r$

erfüllt in Straßennetzen (≤ 1000 m) $n \sim 10.000$ in Berlin

⇒ lex. kleinstes Pareto-optimum, das die Längenbeschränkung einhält ist Optimallösung des CSP

(Abbrechen wenn zum ersten Mal ein Vektor in t markiert wird)

Praktische Erfahrung

2 Subgradientenschritte $\gtrsim$ Pareto Dijkstra

↑ angenäherte Lösung; ↑ Laufzeit; ↑ optimale

4.3 Approximation Pareto-optimaler Wege

1. Gewichtete Summe der Kriterien (Jaffe et al '84)

Zulässigkeitsproblem: Gegeben: $L_1, \ldots, L_r$

Gesucht: Weg P mit $\lambda(P) \leq \begin{pmatrix} L_1 \\ \vdots \\ L_r \end{pmatrix}$

Transformation

kürzeste Wege Problem bzgl. $\lambda'(a) = \alpha_1 \lambda_1(a) + \ldots + \alpha_r \lambda_r(a)$

Gewichte, die von $L_1, \ldots, L_r$ abhängen i.a. nur schlechte Approximation

Bsp. $\alpha_1 \leq \alpha_2 \Rightarrow (0,15)$ wird im Bsp. berechnet

schlecht für $L_1 = 7,\ L_2 = 8$

allgemein gilt: [Theune 95]

Sei P^* Pareto optimal mit $\lambda(P^*) \leq \begin{pmatrix} L_1 \\ \vdots \\ L_r \end{pmatrix}$

Sei P' der bzgl. λ' mit $\alpha_1 = \dots = \alpha_r = 1$ berechnete kürzeste Weg

$\Rightarrow \max\limits_k \lambda(P') \leq r \max\limits_k L_k$ (i.A. scharf)

im Bsp ist für $\alpha_1 = \alpha_2$ der Weg P' über die oberen Kanten ein kürzester bzgl λ' und $\max\limits_k \lambda(P') = 15 \leq 2 \cdot \max\{7,8\} = 16$

2. <u>Einfache Skalierung der Längen</u> [Theune '95]

Skaliere $\lambda(a) = \begin{pmatrix} \lambda_1(a) \\ \vdots \\ \lambda_r(a) \end{pmatrix}$ auf den weniger wichtigen Kriterien

$\downarrow$

$$\lambda'(a) = \left(\lambda_1(a), \left\lfloor \frac{\lambda_2(a)}{\alpha} \right\rfloor, \dots, \left\lfloor \frac{\lambda_r(a)}{\alpha} \right\rfloor \right)^T \qquad \alpha > 1$$

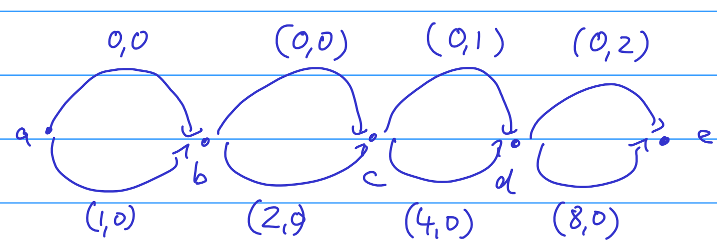
$\Rightarrow$ pareto(v) wird kleiner

Skalierung erhält lexikographische Ordnung

$\Rightarrow$ Berechnete Pareto optimale Wege bzgl λ' sind "nahezu" Pareto-optimale Wege bzgl. λ

Bsp:

$\alpha = 4$



Pareto Optima bzgl λ'				
a	b	c	d	e
(0,0)	(0,0)	(0,0)	(0,1)	(0,3)
			(4,0)	(8,1)
				(4,2)
				(12,0)

zugehörige Pareto Optima bzgl λ				
a	b	c	d	e
(0,0)	(0,1)	(0,3)	(0,7)	(0,15)
			(4,3)	(8,7)
				(4,11)
				(12,3)

ergibt Teilmenge Y der Menge X aller Pareto Optima bzgl λ

Genauer:

zu Pareto-Optimalen Weg P mit Bewertung $d \in X$
exist. Pareto-optimaler Weg P' mit Bewertung $d' \in Y$
mit $|d_k - d_k'| \le \ell \cdot (\alpha - 1) \qquad k = 1, .., r$

ℓ = # Kanten von P'

← pessimistisch aber scharf

Bsp: $d = (9,6) \rightsquigarrow d' = (8,7)$
$d = (13,3) \rightsquigarrow d' = (12,3)$

3. Ein voll-polynomiales Approximationsschema [Hassin 92] für CSP

(CSP) Gegeben: Digraph $G = (V, A)$, $s, t \in V$
Kantenbewertungen τ_a, ℓ_a, Längenschranke L, alle aus $\mathbb{Z}_+$
Gesucht: s,t-Weg P mit $\tau(P)$ minimal unter $\ell(P) \le L$

Annahme: Kennen den Optimalwert τ^* eines besten längenbeschränkten s,t-Wegs

Idee: Legen in jedem Knoten Schubfächer der Länge $\frac{\tau^*}{(n-1)/\varepsilon}$ für Labels (= Vektoren) an, die das Intervall $[0, (1+\varepsilon)\tau^*]$

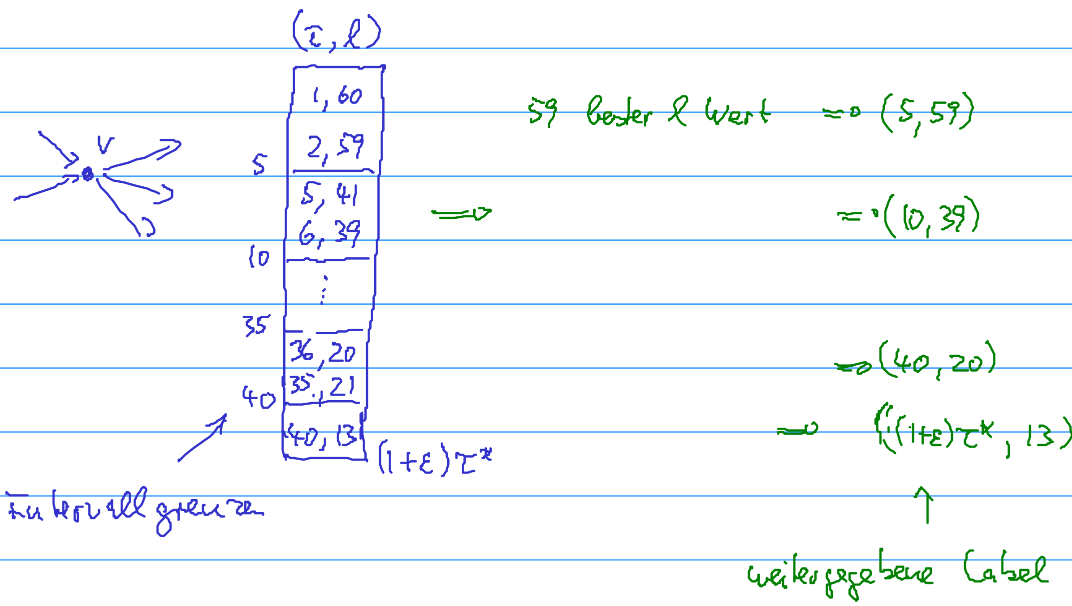
äquidistant unterteilen, bis auf das letzte.

Im Schubfach i: alle Labels von v mit τ-Wert im Intervall

$$\left]\,(i-1)\cdot\frac{\tau^*}{(n-1)/\varepsilon}\ ,\ i\,\frac{\tau^*}{(n-1)/\varepsilon}\,\right]$$

Algorithmus: ε-Schubfach Dijkstra

- Gehe vor wie bei Pareto-Dijkstra für Längenbeschränkung
- Trick: • gebe pro Schubfach nur ein Label weiter, und zwar das bzgl. ℓ-Wert beste Label (sofern $\ell \le L$)
  • runde den τ-Wert des Labels auf die obere Grenze des Schubfachs
- Wähle am Ende den Weg mit kleinstem τ-Wert



$\Downarrow$

Gefundener Weg ist zulässig bzgl. Längenschranke, aber i.A. nicht minimal bzgl τ

Algorithmus hat nur 1 Label pro Schubfach

$$\Rightarrow\ \le\ (1+\varepsilon)\tau^* \Big/ \left(\frac{\tau^*}{(n-1)/\varepsilon}\right) = \frac{(1+\varepsilon)(n-1)}{\varepsilon}$$ viele Labels pro Knoten

$\uparrow$

polynomial in n und $\frac{1}{\varepsilon}$

4.3 Lemma: Der vom ε-Schubfach Dijkstra gefundene Weg ist längenbeschränkt und hat einen maximal um den Faktor $(1+\varepsilon)$ größeren τ-Wert als der kürzeste längenbeschränkte Weg

Beweis: Sei P_ε der vom ε-Schubfach Dijkstra gefundene Weg und P^* ein optimaler Weg

Der ε-Dijkstra macht bei P^* pro Kante einen Rundungsfehler im τ-Wert $\leq$ Intervalllänge $= \frac{\tau^*}{(n-1)/\varepsilon}$

$$\Rightarrow \text{Gesamtfehler} \leq \underbrace{(n-1)}_{\text{elementarer Weg}} \frac{\tau^*}{(n-1)/\varepsilon} = \tau^* \cdot \varepsilon \quad \text{bzgl. } P^*$$

Fehler bei P^*

$$\text{Außerdem} \quad \tau(P_\varepsilon) \underset{\text{da } P_\varepsilon \text{ bester Weg im } \varepsilon\text{-Dijkstra}}{\leq} \tau(P^*) \leq \tau^* + \tau^* \cdot \varepsilon = (1+\varepsilon)\tau^* \quad \square$$

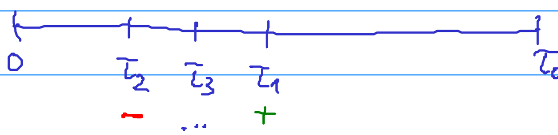
Problem: kennen τ^* nicht!

$\Rightarrow$ binäre Suche verwenden, obere Schranke für τ^* ist $(n-1)\,\tau_{max} =: \tau_0$

kein längenbeschränkter Weg für aktuelles τ gefunden

$\Rightarrow$ rechts weitersuchen

Weg gefunden $\Rightarrow$ links weiter



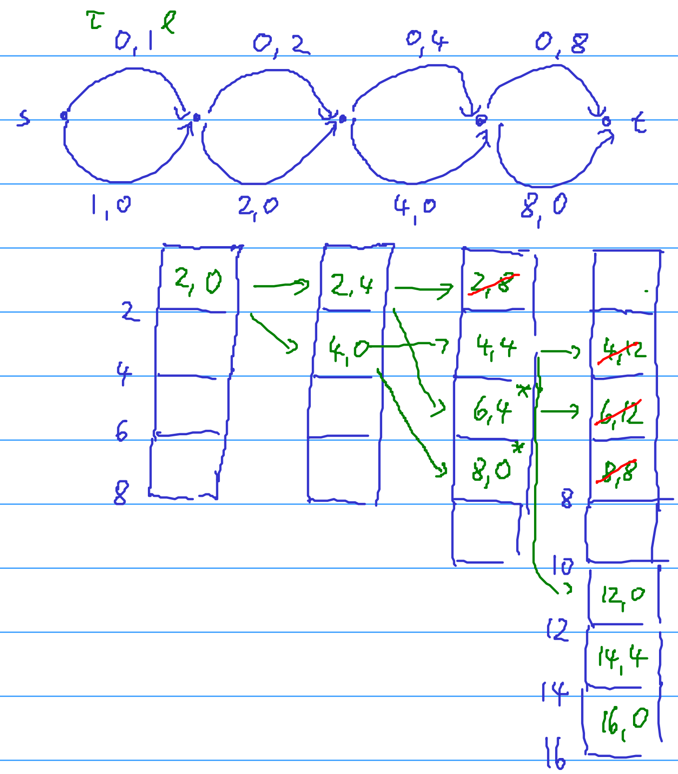
$\Rightarrow \log((n-1)\,\tau_{max})$ viele Aufrufe vom ε-Schubfach Dijkstra

$\Rightarrow$ Laufzeit polynomial in $n, \frac{1}{\varepsilon}$

Beispiel vom Pareto Dijkstra mit $L = 7$, $\tau^* = 8$, $\varepsilon = 1$

⇒ Intervall-Länge $\frac{\tau^*}{(n-1)/\varepsilon} \approx \frac{8}{4} \approx 2$

abzudeckendes Intervall $[0,(1+\varepsilon)\tau^*] = [0,16]$



← bester Weg approximativ

$\tau = 12 \le (1+\varepsilon)\,\tau^* = (1+1)\,8 = 16$

4.4 Die Zwei-Schnitt Methode von Mehlhorn & Ziegelmann [2000]

nutzt Weg-basierte IP Formulierung und LP-Relaxation

↓

geometrisch im $\mathbb{R}^2$ interpretiert

zum Schluss Pareto Dijkstra

Primales LP: 0,1 Variable x_P für jeden s,t Weg P

$$\tau_P = \sum_{a\in P} \tau_a \;, \quad \ell_P = \sum_{a \in P} \ell_a$$

$$\min \sum_P \tau_P \cdot x_P$$ ← τ_P minimieren

$$\text{unter} \sum_P \ell_P \cdot x_P \le L$$ ← Längenschranke einhalten

$$\left.\begin{array}{l}\sum_p x_p = 1 \\ x_p \in \{0,1\}\end{array}\right\} \Rightarrow \text{nur ein Weg in Lösung}$$

LP-Relaxierung

Dualvariable

$$\min \sum_p \tau_p x_p$$

unter $\sum_p \ell_p x_p \le L \quad \rightarrow \quad -\sum_p \ell_p x_p \ge -L$ $\qquad v$

$\sum_p x_p = 1$ $\qquad u$

$x_p \ge 0 \quad \leftarrow \quad x_p \le 1$ wegen

Duales LP

$$\max\ u - Lv$$

unter $u - \ell_p v \le \tau_p$

$v \ge 0$ u nicht vorzeichenbeschränkt

Ersetzung $v \rightarrow -v$ ergibt

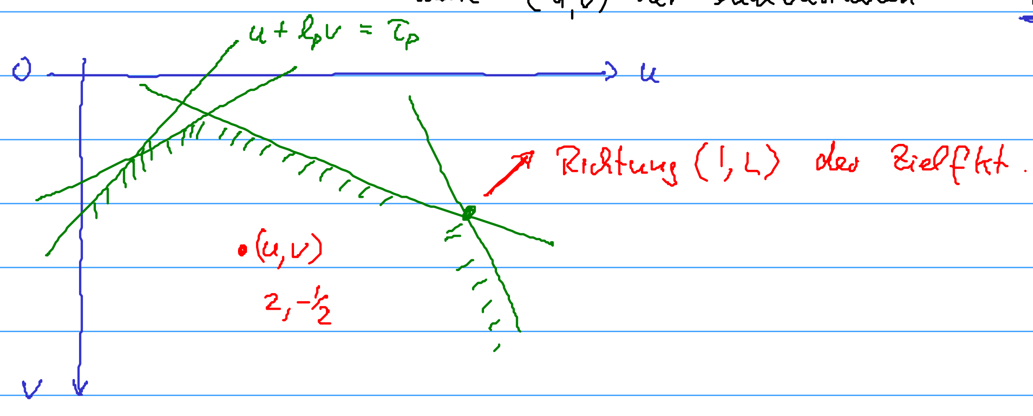
$$\boxed{\begin{array}{ll}\max & u + Lv \\ \text{unter} & u + \ell_p v \le \tau_p \quad \forall P \\ & v \le 0,\ u \text{ beliebig}\end{array}} \quad (D)$$

nur 2 Variable $\Rightarrow$ Standardinterpretation als 2-dim LP

Nebenbedingungen $\hat{=}$ Halbebenen, geb. durch

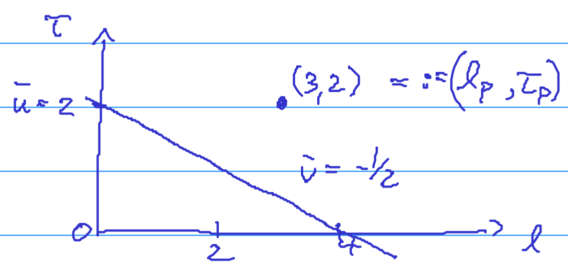
Geraden $u + \ell_p v = \tau_p$

Werte (u,v) der Dualvariablen $\hat{=}$ Punkte



andere, geometrische duale Interpretation

- Paar (u,v) als Gerade $\tau = v \cdot \ell + u$ in der (ℓ,τ)-Ebene
- Nebenbedingungen $u + \ell_p v \leq \tau_p$ $\hat{=}$ Punkt (ℓ_p, τ_p) $\hat{=}$ Pfad
- Ungleichung $\bar{u} + \ell_p \bar{v} \leq \tau_p$ erfüllt für Paar $(\bar{u}, \bar{v})$
  $\Leftrightarrow$ Punkt (ℓ_p, τ_p) liegt oberhalb der Geraden $\tau = \bar{v}\ell + \bar{u}$



$(\bar{u}, \bar{v}) = (2, -\frac{1}{2})$ definiert die Gerade $\tau = -\frac{1}{2}\ell + 2$

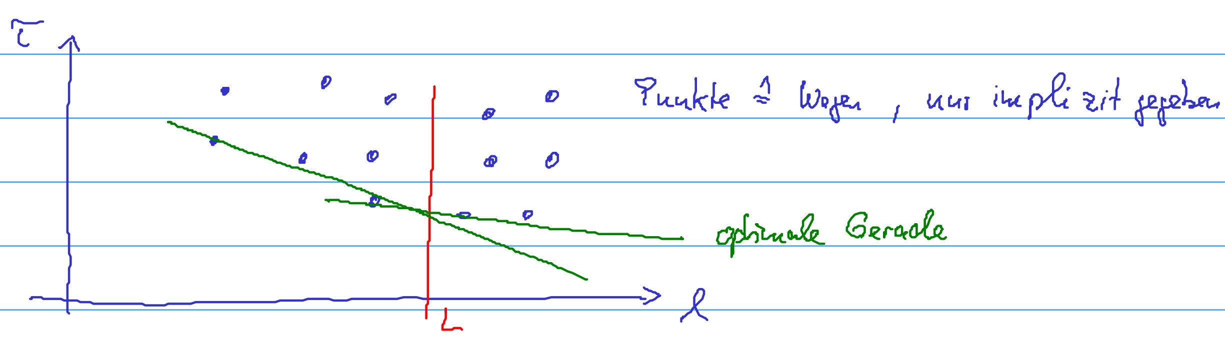
Punkt $(3,2)$ liegt oberhalb der Geraden

$\Leftrightarrow \quad 2 \geq \underbrace{-\frac{1}{2} \cdot 3 + 2}_{0{,}5}$

Optimierung $\hat{=}$ Suche nach einem Paar (u^*, v^*) bzw. einer Geraden $\tau = v^* \ell + u^*$ mit nicht-positiver Steigung $(v^* \leq 0)$ so dass:

1. alle Punkte (ℓ_p, τ_p) P s,t-Weg liegen oberhalb dieser Geraden

2. Der Schnittpunkt der Geraden $\tau = v^* \ell + u^*$ mit der Geraden $\ell = L$ ist so groß wie möglich, denn der Schnittpunkt hat den Wert $v^* L + u^*$ = Zielfkt von (D)

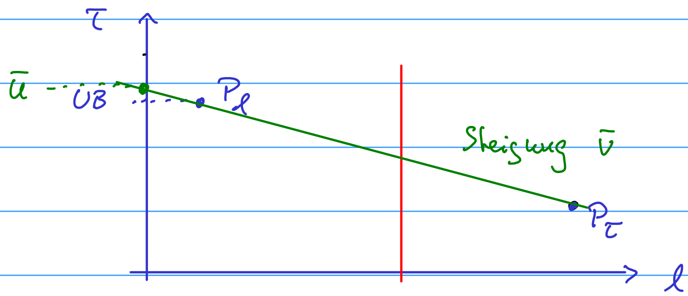


untere Einhüllende der implizit geg. Punktemenge bei $\ell = L$ berechnen

↑ geometrische Aufgabe

Konstruktion erfolgt dem "Hüllenansatz"

1. Berechne den kürzesten Weg $P_{\ell_{min}}$ bzgl ℓ

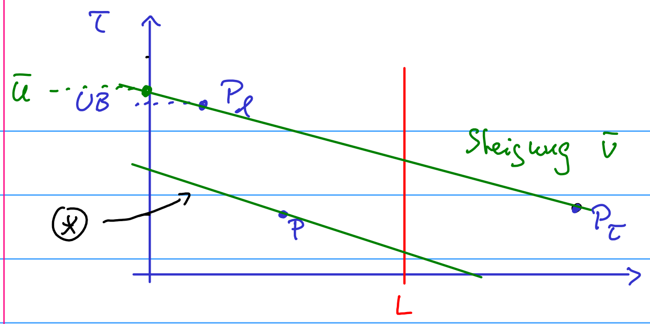


if $\ell(P_{\ell_{min}}) > L$ then return "Keine zulässige Lösung"
else $UB := \tau(P_{\ell_{min}})$ // UB $\hat{=}$ obere Schranke an opt. τ-Wert

Berechne kürzesten Weg $P_{\tau_{min}}$ bzgl τ
if $\ell(P_{\tau_{min}}) \leq L$ then return $P_{\tau_{min}}$ // optimal
else $P_\tau := P_{\tau_{min}}$; $P_\ell := P_{\ell_{min}}$ // Initialisierung P_τ, P_ℓ

// P_τ und P_ℓ erzeugen erste Gerade $\tau = \bar{v}\ell + \bar{u}$
// durch die Punkte P_τ, P_ℓ
// mit Steigung $\bar{v} = \left(\tau(P_\tau) - \tau(P_\ell)\right) / \left(\ell(P_\tau) - \ell(P_\ell)\right) < 0$

2. Prüfe ob es Punkte (ℓ_P, τ_P) von s,t-Wegen P unterhalb der Geraden gibt



(*) Parallelverschiebung der ersten Geraden durch (ℓ_P, v_P)

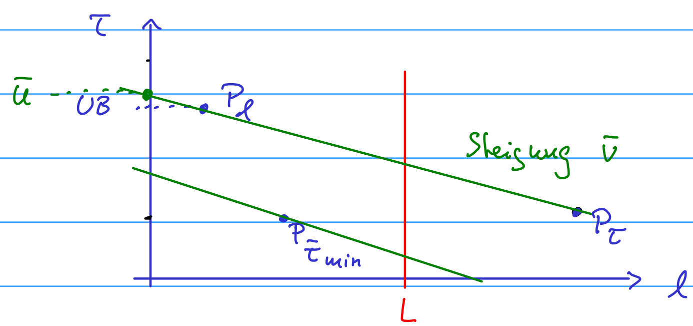
Ein solcher Pkt entspricht einem Weg P mit

$$\bar v\,\ell_P + \bar u > \tau_P \quad \Leftrightarrow \quad \underbrace{\tau_P - \bar v\,\ell_P}_{\substack{\text{Weglänge von P bzgl.}\\ \text{Kantenbewertung } \tau_a - \bar v \ell_a}} < \bar u$$

und kann durch eine kürzeste Wege Berechnung bzgl. $\tau_a - \bar v \ell_a =: \bar\tau_a$ berechnet werden

Graphisch entspricht das einer Parallelverschiebung der Geraden auf einen bzgl. $\bar\tau_a$ extremalen Punkt (dem kürzesten Weg bzgl. $\bar\tau_a$)

Sei $P_{\bar\tau_{min}}$ der neue Pkt bzgl. $\bar\tau_a$



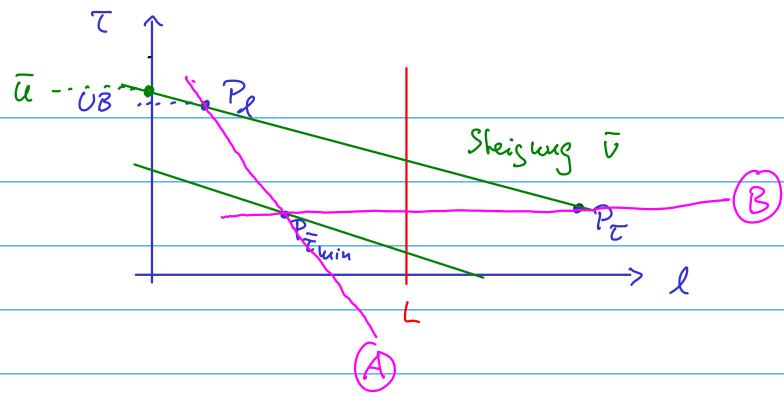
if $\bar\tau(P_{\bar\tau_{min}}) = \bar u \Rightarrow P_{\bar\tau_{min}}$ liegt auf der alten Geraden

$\Rightarrow$ alte Gerade ist optimal

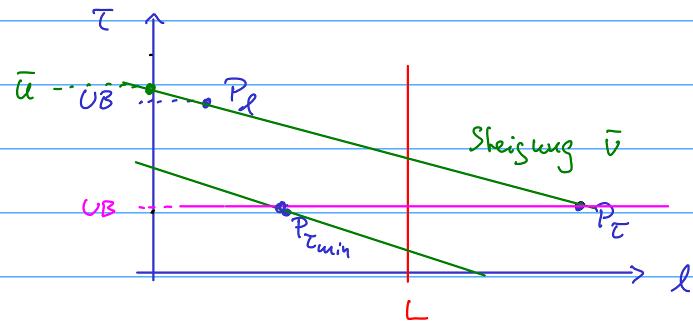
Schnittpunkt mit $\ell = L$ ist Optimum

else weitersuchen in 3

3. Aktualisiere die alte Gerade durch eine der beiden Geraden $\overline{P_\ell P_{\bar\tau_{min}}}$ oder $\overline{P_{\bar\tau_{min}} P_\tau}$



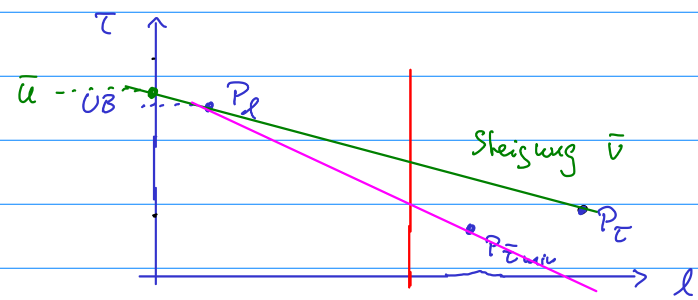
Frage: Welche der beiden Geraden ist besser, d.h. der Schnittpunkt auf L ist höher?



Fall 1

Fall 1: $P_{\bar{\tau}_{min}}$ ist links von $\ell = L$ $\Rightarrow$ $\overline{P_{\bar{\tau}_{min}} P_\tau}$ ist besser

Fall 2: $P_{\bar{\tau}_{min}}$ ist rechts von $\ell = L$ $\Rightarrow$ $\overline{P_\ell P_{\bar{\tau}_{min}}}$ ist besser



Fall 2

Fall 1: setze $P_\ell := P_{\bar{\tau}_{min}}$ $\qquad$ $UB := \tau(P_{\bar{\tau}_{min}})$

Fall 2: setze $P_\tau := P_{\bar{\tau}_{min}}$ $\qquad$ UB kann nicht aktualisiert werden

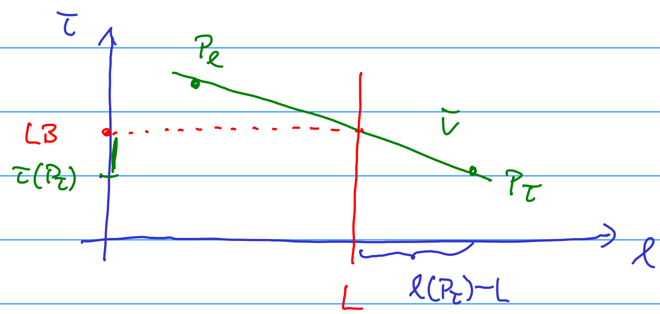
Iteriere mit neuer Geraden $\overline{P_\ell, P_\tau}$

Abbruch erfolgt in dem Fall, wenn keine Parallelverschiebung der aktuellen

Geraden mehr möglich ist

Dann ist die berechnete untere Schranke LP an die optimale Länge des CSP

$$LB = \tilde{v}\,(L - \ell(P_\tau)) + \tau(P_\tau)$$



$LB \leq OPT(CSP)$

$\parallel$

Opt Wert der LP-Relaxation

$\parallel$

Opt Wert der Lagrange Relaxation, da relaxierte Problem in der Lagrange Relaxierung ganzz. Ecken hat (ist kürzeste Wege Problem, in Kantenformulierung vollständig unimodular)

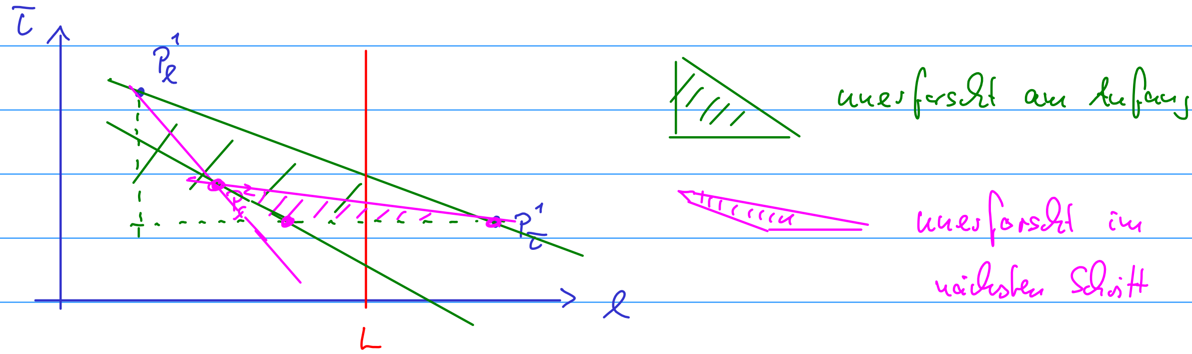
Insgesamt bleibt dieselbe "Dualitätslücke" wie bei der Lagrange Relaxation. Aber die Laufzeit kann klar besser abgeschätzt werden und ist polynomial

4.4 SATZ (Mehlhorn & Ziegelmann 2000)
Der Hüllenansatz arbeitet korrekt und hat eine Laufzeit von

$$O\Big(\underbrace{\log(n \cdot \tau_{max} \cdot \ell_{max})}_{\#\,\text{Iterationen}} \cdot \underbrace{(n \log n + m)}_{\text{kürzeste Wege Berechnung}}\Big)$$

mit $\tau_{max} := \max_a \tau_a$, $\ell_{max} := \max_a \ell_a$

Beweis: Korrektheit klar aus vorhergegangenen Überlegungen zur Laufzeit: geht über die Größe der noch "unerforschten Gebiete" von noch möglichen Punkten (Wegen) unterhalb der momentanen Geraden



Fläche erstes Dreiecks ist maximal $\frac{1}{2}(n \cdot \ell_{max})(n \cdot \tau_{max})$

Fläche eines solchen Dreiecks ist minimal $\frac{1}{2}$ da alle Dreiecke (=> Punkte) ganzzahlig sind

4.4 Für aufeinanderfolgende Iterationen gilt $A_{i+1} \leq \frac{1}{4} A_i$

Fläche der Dreiecke

Aufgabe 4.4

$\Rightarrow$ maximal N Iterationen bis $\frac{1}{2} \geq \left(\frac{1}{4}\right)^N \frac{1}{2} \cdot n^2 \ell_{max} \cdot \tau_{max}$

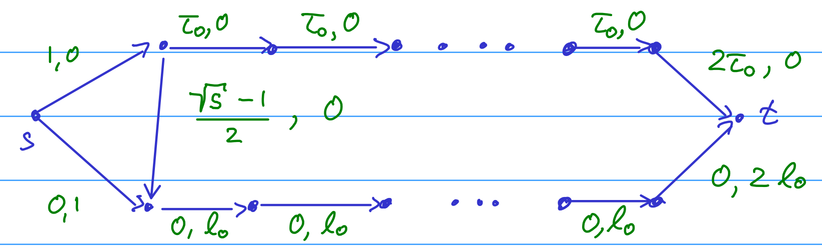
$\Rightarrow O(\log(n\, \ell_{max} \cdot \tau_{max}))$ Iterationen □

Bemerkung: kann zeigen, dass $O(m \log^2 m)$ Iterationen reichen (Jüttner 2003) => streng polynomialer Algorithmus

Insgesamt bestimmt der Hüllenansatz eine obere Schranke UB und eine untere Schranke LB für OPT(CSP). Die Güte der Schranke kann allerdings i.A. beliebig schlecht werden

4.5

4.5 Beispiel



$s := 1 + \frac{n}{2} \cdot \tau_0 \qquad n := \#\text{ Knoten} \geq 4$ gerade

$L := \frac{n \cdot \ell_0}{2} \qquad \ell_0 > \tau_0 > 0$

Aufgabe 4.5 : Zeige

- Mittelansatz findet $UB = s$ und $LB = \frac{s}{T}$, $T := 1 + \frac{n}{2} \ell_0$
- Optimalwert $= \frac{\sqrt{s}-1}{2}$
- Approximationsgüte $\frac{OPT}{LB} \in \Omega\left(\sqrt{n \cdot \tau_0}\right)$

<u>Schließen der Lücke zwischen LB und UB</u>

z.B. durch Pareto Dijkstra, wobei Label (τ, ℓ) verworfen werden, wenn $\tau \geq UB$, $\tau < LB$, oder $\ell \geq L$

$\Rightarrow$ wesentliche Beschleunigung, kleinere Labellisten an Knoten

4.5 Anwendung auf Constrained System Optimum

$$\min \sum_a \tau_a(x_a(f)) \cdot x_a(f)$$

unter $\sum_{P \in \mathcal{P}_k^\varepsilon} f_P = d_k$

$\hookrightarrow \{ (s_k, t_k)\text{-Pfade } P \text{ mit } \sum_{a \in P} \tau_a(UE_a) \leq (1+\varepsilon) L_k \}$

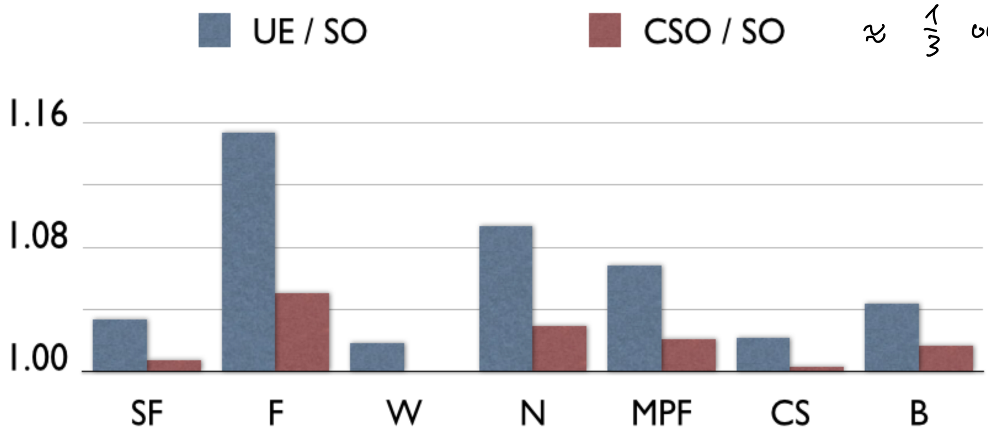
Längenbeschränkung bzgl $\tau_a(UE_a)$

⇓

lineare Problem im Frank-Wolfe Algo zerfällt in # commodity viele CSP Probleme

Paper Jahn, M., Schulz, Stier → www
CSP gelöst mit Pareto Dijkstra

Results for some cities



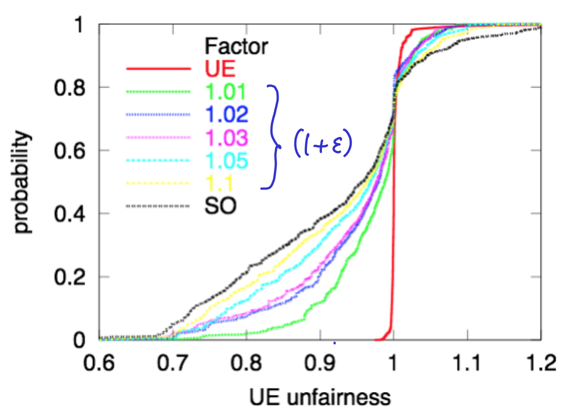
≈ $\frac{1}{3}$ von UE/SO

SO ≙ 1,00

SF	Sioux Fall	N	Neukölln
F	Friedrichshain	MPF	Berlin Mitte
W	Winnipeg	CS	Chicago Sketch

B Berlin

Analysis of fairness



- 75% of the users travel less than in equilibrium
- Only 0.4% of the users travels 10% more than in equilibrium
- For the system optimum, these are more than 5%