

## §4 Constrained shortest paths

## 4.1 Komplexität

4.1 PROPOS. Das CSP ist schwach NP-vollständig

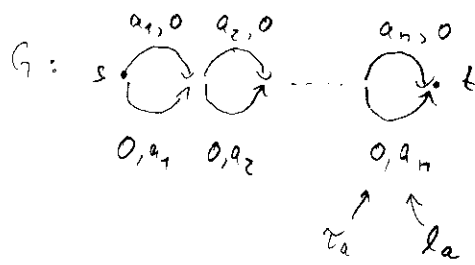
Beweis: Reduktion von PARTITION:

Gegeben Zahlen  $a_1, \dots, a_n$  mit  $\sum a_i = 2b$

Frage: Gibt es Indexmenge  $I$  mit  $\sum_{i \in I} a_i = b$  ?

Sei  $I$  Instanz von PARTITION.

Konstruiere daraus Instanz  $I'$  von CSP:



Frage: Gibt es st Weg  $P$  mit  $\tau(P) \leq b$ ,  $l(P) \leq b$

Offenbar ist die Antwort auf diese Frage ja

$\Leftrightarrow \exists$  Indexmenge  $I$  mit  $\sum_{i \in I} a_i = b$   $\square$

4.2 PROPOSITION: CSP kann in pseudopolynomiales Zeit gelöst werden.

Beweis: analog zu SUBSET SUM (Verallgemeinerung von PARTITION)

aus ADH II  $\square$

## 4.2 Lösungsansätze

- 1) Beasley & Christofides 29 Brand & Bound + Lagrange Relaxation
- 2) Labeling Algorithmus (erweitertes Dijkstra)  
Verwandtschaft zu Pareto-optimalem Weg
- 3) Geometrisch (Kehlman + Ziegelmann 2000)  $\rightarrow$  § 4.4

4.2.1 Der Ansatz von Beasley & Christofides

Brand & Bound Baum  $\hat{=}$  wie bei Telekomm Problem aus ADM II  
 $\hat{=}$  IP-Formulierung  
 Festlegen von Kanten

Lagrange Relaxation:

$$\min \tau(P)$$

$$\text{so dass } l(P) \leq L$$

P ist Weg von s nach t

} in Zielfkt

$$(LR_{\mu}) \quad \min \tau(P) + \mu (l(P) - L) =: \Delta(\mu), \mu \in \mathbb{R}_+^1$$

$$\sum_{a \in P} (\tau_a + \mu l_a) - \mu L$$

Konstant für festes  $\mu$

Kürzester Weg bzgl. neuer Kantenbewertung  $\tau_a + \mu l_a$

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

Dann gilt (1)  $\Delta(\mu) \leq \tau^* \quad \forall \mu \geq 0$

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

(3) Ist  $P$  eine Optimallösung von  $(LR_\mu)$  und  $l(P) \leq L$  }  $\Rightarrow P$  opt für CSP

$\tau^* - \Delta^*$  heißt Dualitätslücke

Subgradientenverfahren analog zu ADMM, § 7.4

Ist  $P_{\mu_0}$  kürzester Weg in  $(LR_{\mu_0})$ , so ist  $l(P_{\mu_0}) - L$  (verletzte Rest.)  
ein Subgradient in  $\mu_0$  von  $\Delta(\mu)$

Gehe Schritt in diese Richtung

$$\mu^{\text{neu}} = \mu^{\text{alt}} + \Theta (l(P_{\mu_0}) - L)$$

↓

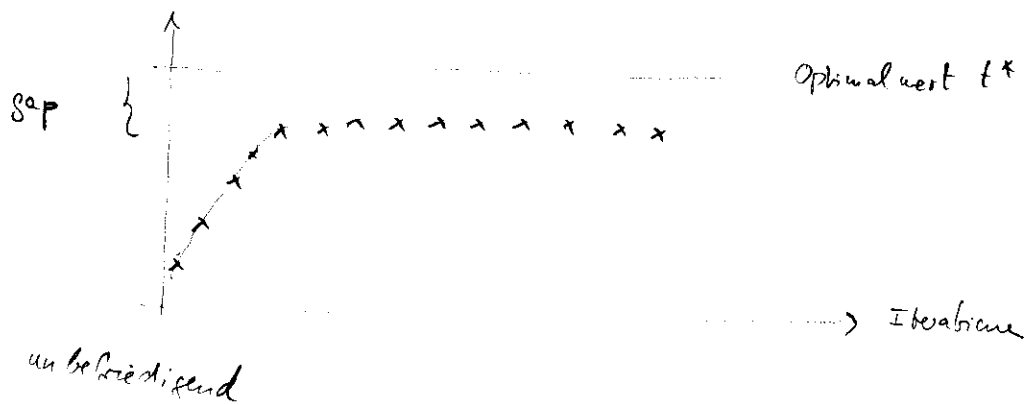
neue Kantenbewertung

$$\tau_a + \mu^{\text{neu}} l_a$$

Unterschiede in Kantenbewertung: gerade  $(\mu^{\text{neu}} - \mu^{\text{alt}}) l_a$

↑  
dieselbe Zahl für alle Kanten

typischer Verlauf



## 4.2.2 Pareto-optimale Wege

Ges.  $G = (V, A)$   $s, t \in V$ Für jede Kante  $a$  ein  $r$ -dim Bewertungsvektor  $\lambda(a) = \begin{pmatrix} \lambda_1(a) \\ \vdots \\ \lambda_r(a) \end{pmatrix} \geq 0$ Länge  $\lambda(P)$  eines Weges  $P$  ist

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

hier unterschiedliche Operationen möglich

 $\Sigma, \max, \min, \dots$ 

[allgemein: Semiringe]

Gesucht: alle Pareto-optimale  $s, t$  Wege $P$  heißt Pareto-optimal $\Leftrightarrow \nexists$  Weg  $P'$  mit  $\lambda(P') \leq \lambda(P)$  und  $\lambda_i(P') < \lambda_i(P)$  für ein  $i$ (keine Dominanz)↑  
Komponentenweise Ordnung

Anwendungen: Autoverkehr: Zeit, Entfernung, verbotene Strecken als Kriterien  
 Bahn: Zeit, Kosten, # Umstiege

Der Dijkstra Algorithmus für Pareto-optimale Wege

unabhängig: Dijkstra für kürzeste Wege ( $\lambda_e \geq 0$ )

- berechnet für jeden Knoten  $v$  eine Distanz  $d[v]$

$\hat{=}$  Länge kürzesten Weg von  $s$  zu  $v$

+ einen Knoten Vorgänger  $[v]$ , der Vorgänger von  $v$  auf kürzestem Weg von  $s$  zu  $v$  ist

- Werte  $d[v]$ , Vorgänger  $[v]$  sind unendlich vererblich

Im Lauf des Algorithmus werden Knoten markiert, für markierte Knoten sind diese Werte endgültig

Initialisierung:

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

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

zu  $s$  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 Dijkstras für Pareto-optimale Wege [Theorie 95]  
 statt  $d[v]$   $r$ -dim Vektoren  $d[v] = \begin{pmatrix} d_r[v] \\ \vdots \\ d_r[v] \end{pmatrix}$   $d_k[v]$  bzgl  $\lambda_k$

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

Initialisierung  

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

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

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

Aktualisierung des  $d[w]$ :

Betrachte zu gewähltem  $d[v]$  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  $d[v]$

Anfangs nur  $d[s]$  markiert

Hauptschleife

while  $\exists$  unmarkierten Vektor do

wähle lexikographisch kleinsten unmarkierten Vektor  $d[v]$

markiere diesen Vektor

sei  $v$  der zugehörige Knoten

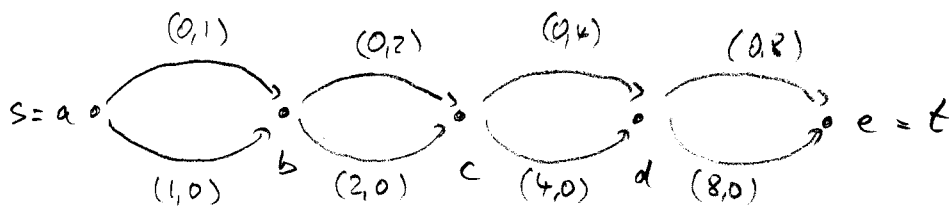
for alle Kanten  $(v, w)$  do

nehme  $d[w] := d[v] + \lambda_{(v,w)}$  den Vektoren von  $w$  hinzu;

streiche nicht Pareto-optimale bei  $w$ ;

falls  $d[w]$  nicht gestrichen wird, so setze Vorgänger  $d[w] := d[v]$

Beispiel:



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)
$z^0$	$z^1$	$z^2$	$z^3$	$z^4$

A parenthelles Wachstum  $\Rightarrow$  exponentielles Algorithmus

Korrektheit analog zu normalen Dijkstra per Induktion:

nach # Markierungen:

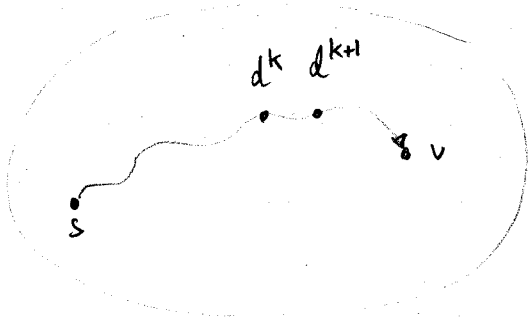
Sobald Vektor markiert, gehört es zur Menge der Pareto-Optima zum zugehörigen Knoten



Beweis: Betrachte Zeitpunkt zu dem  $d[v]$  gewählt wird

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

$\Rightarrow \exists$  <sup>Pareto-opt</sup> Weg  $P$  von  $s$  nach  $v$  mit  $\lambda(P) \stackrel{\neq}{\leq} d[v]$



$P$  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 dem Weg  
und  $d^1, d^2, \dots, d^m$  die zugehörigen Pareto-optimalen  
Bewertungsvektoren, also  $d^i = d^{i-1} + \lambda_{(v_{i-1}, v_i)}$

Sei  $d^k$  der letzte Vektor des Bereichs markiert ist zum Zeitpunkt  
der Auswahl von  $d[v]$  (exist, da  $d^1 = d[s]$  markiert)

$\Rightarrow k < m$ , da sonst  $d^m \stackrel{\neq}{\leq} d[v]$  bereits bei  $v$  markiert wäre und  $d[v]$  gestrichen wäre.

$\Rightarrow$  Zum Zeitpunkt der Markierung von  $d^k$  wird  $d^{k+1} = d^k + \lambda_{(v_k, v_{k+1})}$  in

$v_{k+1}$  hinzugefügt und ist Pareto-optimal <sup>(wegen \*)</sup> und bleibt bis zu

Wahl von  $d[v]$  unmarkiert (nach Wahl des Index  $k$ ).

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

$\Rightarrow d^{k+1} <_{\text{lex}} d[v] \Rightarrow$  Widerspruch zu Auswahlregel

$\Rightarrow$  Jeder markierte Vektor ist Pareto-optimal

Wegen (\*) wird jedes Pareto-Optimum im Algorithmus konstruiert  $\square$

Frage: Wie groß wird die maximale Anzahl der Pareto-Optima

Annahme:  $\lambda_k(a) \in \mathbb{Z}_+$

$L_k(v) :=$  Maximale Weglänge (elementares Weg) von  $s$  nach  $v$  <sup>bzgl.  $\lambda_k$</sup>

$\text{pareto}(v) := \#$  Pareto Optima in  $v$

$$\text{pareto}(v) \leq (L_k(v))^r$$

$\Rightarrow$   $\text{pareto}(v)$  ist polynomial in  $n$ , falls  $\lambda_k(a)$  polynomial in  $n$  und  $r$  fest ist  $\Rightarrow$  algorithmisches polynomial

etwa:  $\lambda_k(a) \leq n \Rightarrow L_k(v) \leq (n-1)n \leq n^2 \Rightarrow \text{pareto}(v) \leq n^{2r}$

$\lambda_k(a) \leq L \Rightarrow \text{pareto}(v) \leq (L \cdot (n-1))^r$

$\uparrow$   
 erfüllt in Straßennetz ( $\leq 1000$  m)

Anwendung bei CSP

[Arora, Agarwal & Nair 1983]

min  $\tau(P)$  unter  $l(P) \leq L$   $P$  s.t. Weg

Kantenbewertung  $\lambda(a) = \begin{pmatrix} \tau_a \\ l_a \end{pmatrix}$

Pareto Dijkstra anwenden

Vektoren  $d(v) = \begin{pmatrix} \tau \\ l \end{pmatrix}$  verwerfen, falls  $l > L$

$\Rightarrow$  liefert alle Pareto optima mit  $l \leq L$

$\Rightarrow$  lex-kleinstes Pareto-Optimum ist Optimallösung von CSPD  
 (Abbruch falls  $t$  zum erstenmal erreicht)

Praktische Erfahrung:

2 Subgradienten-Schritte  $\approx$  Pareto-Diagramm  
 $\downarrow$   $\downarrow$   
 ausgehend optimal

(unvollständig)  
 max. Unterschied  
 Faktor 1-20

für mehr Schritte deutlich schlechter!

## 4.3 Approximation: Pareto-optimales Wege

1. Gewichte Summe der Kriterien (Zaffe 84)

Zulässigkeitsproblem

Gegeben  $\begin{pmatrix} L_1 \\ \vdots \\ L_r \end{pmatrix}$  Finde Weg  $p$  mit  $\lambda(p) \leq \begin{pmatrix} L_1 \\ \vdots \\ L_r \end{pmatrix}$ 

} Transformation

Kürzeste Wege Problem löst  $\lambda'(a) := \alpha_1 \lambda_1(a) + \dots + \alpha_r \lambda_r(a)$   
 $\uparrow \qquad \qquad \qquad \uparrow$   
 Gewichte, die von  $L_1, \dots, L_r$  abhängen

i.a. um schlechte Approximationen

Bsp.  $L_1 < L_2 \Rightarrow \alpha_1 < \alpha_2 \Rightarrow (0,15)$  wird berechnet im Beispiel  
 schlecht für  $L_1 = 7$   $L_2 = 8$

allgemein gilt:  $\max_{k=1, \dots, r} \lambda'_k(p) \leq r \cdot \max_{k=1, \dots, r} L_k$   
 $\uparrow$   
 i.a. scharf

## 2. Einfache Skalierung der Längen [Theorie 95]

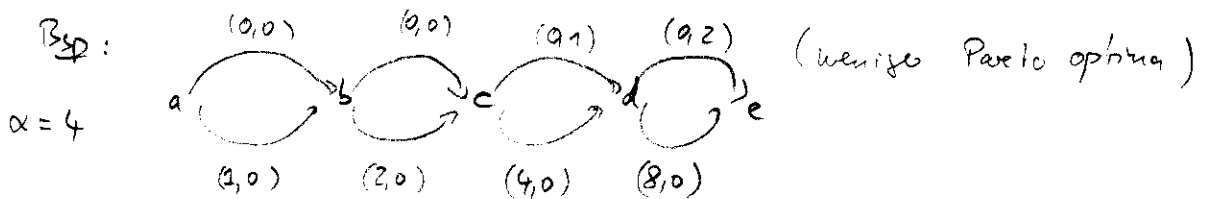
Skalare  $\lambda(a) = \begin{pmatrix} \lambda_1(a) \\ \vdots \\ \lambda_r(a) \end{pmatrix}$  auf den weniger wichtige Kriterien

$$\downarrow$$

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

=> pareto(v) wird kleiner!

Skalierung erhält Ordnung => Berechnete pareto-optimale Wege bzgl  $\lambda'$  sind pareto-optimale Wege bzgl  $\lambda$



Pareto optimal bzgl $\lambda'$					zusätz. Pareto optimal bzgl $\lambda$				
a	b	c	d	e	a	b	c	d	e
(0,0)	(0,0)	(0,0)	(0,1)	(0,3)	(0,0)	(0,1)	(0,3)	(0,7)	(0,15)
			(4,0)	(8,1)				(4,3)	(8,7)
				4,2					(4,11)
				(12,0)					(12,3)

ergibt Teilmenge  $Y$  aller Pareto Optimierung  $X$

zu pareto-opt Weg  $P$  mit Bewertung  $d \in X$   
 exist Pareto opt Weg  $P'$  mit Bewertung  $d' \in Y$   
 mit  $|d_k - d'_k| \leq l(\alpha - 1) \quad k = 1, \dots, r$   
 $\uparrow$   
 # Kanten von  $P'$

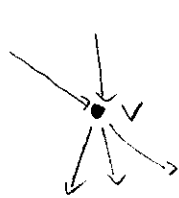
← pessimistisch  
aber scharf

Bsp  $d = \begin{pmatrix} 9,6 \\ 13,2 \end{pmatrix} \rightsquigarrow d' = \begin{pmatrix} 8,7 \\ 12,3 \end{pmatrix}$

## 3. Ein voll-polynomiales Approximationsschema [Hassin 92]

Annahme: kennen den optimalen Wert  $\tau^*$  eines besten Längenbeschränkten  $s, t$  Weges

Idee: Legen in jedem Knoten  $v$  Subfächer der Länge  $\frac{\tau^*}{(n-1)/\epsilon}$  für Labels an,



(1,60)
(1,63)
(5,41)
(5,56)
⋮
(35,67)
(35,90)
(40,13)

=

(1,60)
(1,63)
(5,41)
(5,56)
⋮
(35,67)
(35,90)
(40,13)

die das Intervall

 $[0, (1+\epsilon)\tau^*]$ 

äquidistant unterteilen

(bis auf das letzte)

In Subfächer  $i$ : alle Labels von  $v$  mit  $\tau$ -Wert im Intervall

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

Algorithmus:  $\epsilon$ -Subfächer Dijkstra

- Gehe vor wie beim verallgemeinerten Dijkstra Algorithmus

- Trick: gehe pro Subfächer nur ein Label weiter:

das bzgl.  $L$ -Wert beste Label

runde den  $\tau$ -Wert des Labels auf die obere Grenze des Subfächeres

- wähle am Ende den Weg mit kleinstem  $\tau$ -Wert

⇓

Gefundener Weg ist zulässig bzgl. Längenschränke  $L$

(aber i.a. nicht optimal)

Algorithmus hat nur 1 Label pro Subfächer

$$\Rightarrow \leq (1+\epsilon)\tau^* \left\lceil \frac{\tau^*}{(n-1)/\epsilon} \right\rceil = \frac{(1+\epsilon)(n-1)}{\epsilon} \tau^* \text{ viele Labels pro Knoten } n$$

ist längenbeschränkt und

4.3 LEMMA: Der von  $\epsilon$ -Schubfed Dijkstra gefundene Weg hat maximal einen um den Faktor  $(1+\epsilon)$  höheren  $\tau$ -Wert als der kürzeste längenbeschränkte Weg.

Beweis: Sei  $P_\epsilon$  der von  $\epsilon$ -Dijkstra gefundene Weg

$P_\epsilon$  ist elementar  $\Rightarrow \leq n-1$  Kanten

pro Kante Rundungsfehler im  $\tau$ -Wert  $\leq$  Intervall-Länge  $= \frac{\tau^*}{(n-1)/\epsilon}$

$$\Rightarrow \text{Gesamtfehler} \leq (n-1) \frac{\tau^*}{(n-1)/\epsilon} = \tau^* \cdot \epsilon$$

$$\Rightarrow \tau(P_\epsilon) \leq \tau^* + \tau^* \cdot \epsilon = (1+\epsilon) \tau^* \quad \square$$

$\uparrow$  nicht größer, da optimales Weg im Ausgangsproblem bei Runden  $\leq \tau^* + \tau^* \epsilon$

Problem: kennen  $\tau^*$  nicht

$\Rightarrow$  binäre Suche verwenden, obere Schranke für  $\tau^*$  ist  $(n-1) \cdot \tau_{\max}$

kein längenbeschränkter Weg für aktuelles  $\tau$  gefunden.

$\Rightarrow$  rechts weitersuchen

Weg gefunden  $\Rightarrow$  links weitersuchen

$\Rightarrow$   $\log((n-1)\tau_{\max})$  viele Aufrufe von  $\epsilon$ -Schubfed Dijkstra

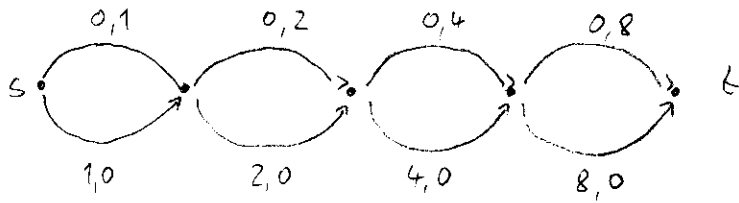
$\Rightarrow$  Laufzeit polynomial in  $n$  und  $\frac{1}{\epsilon}$

Beispiel von Pareto Dijkstra mit  $L=7$   $\tau^*=8$ ,  $\epsilon=1$

$$\Rightarrow \text{Intervall-Länge} \frac{\tau^*}{(n-1)/\epsilon} = \frac{8}{4} = 2$$

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

4-16



$$L = 7$$

$$T^* = 8$$

2	2,0	2	2,2	2	2,6	2	-
4		4	4,0	4	4,4	4	
6		6		6	6,2	6	
8		8		8	8,0	8	
						10	10,6
						12	12,4
						14	14,2
						16	16,0

← beste Weg



4.4 Die Zwei-Step-Methoden von Heldman & Ziegelmann (2000)

nutzt Weg-basierte IP-Formulierung und LP-Relaxation

$$0,1 \text{ Variable } x_p \text{ für jeden st-Weg} \quad \tau_p := \sum_{a \in P} \tau_a, \quad l_p := \sum_{a \in P} l_a$$

=> Primales IP:

$$\min \sum_P \tau_p x_p$$

←  $\tau_p$  minimieren

$$\text{unter } \sum_P l_p x_p \leq L$$

← Längenschränke eingehalten

$$\sum_P x_p = 1$$

$$x_p \in \{0,1\}$$

} ← nur ein zulässiger Weg  
in Lösung

LP Relaxierung:

$$\min \sum_P \tau_p x_p$$

$$v \quad \text{unter } -\sum_P l_p x_p \geq -L$$

$$u: \quad \sum_P x_p = 1$$

$$x_p \geq 0$$

←  $x_p \leq 1$  tritt in Optimum automatisch ein,  
da  $\tau_p \geq 0$

Duales LP:

$$\max u - Lv$$

$$\text{unter } u - l_p v \leq \tau_p \quad \forall P$$

$$v \geq 0, \quad u \text{ nicht vorher beschränkt}$$

Ersetzung  $v \rightarrow -v$  ergibt

$$\max u + Lv$$

$$\text{unter } u + l_p v \leq \tau_p \quad \forall P$$

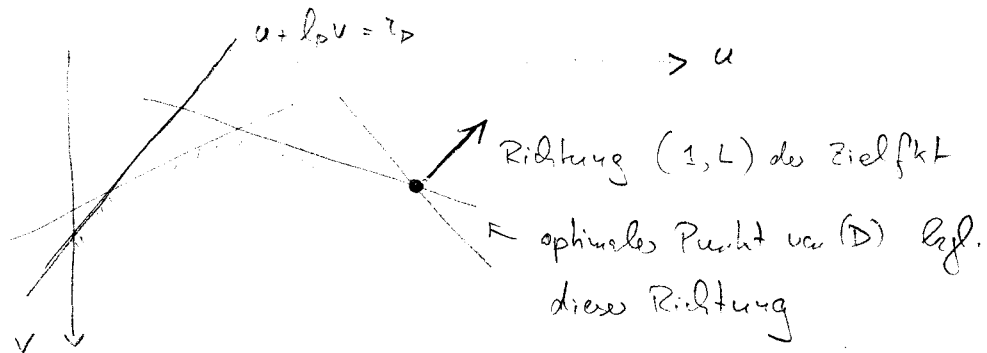
$$v \leq 0 \quad u \text{ nicht vorher beschränkt}$$

(D)

### Standardinterpretation von (D)

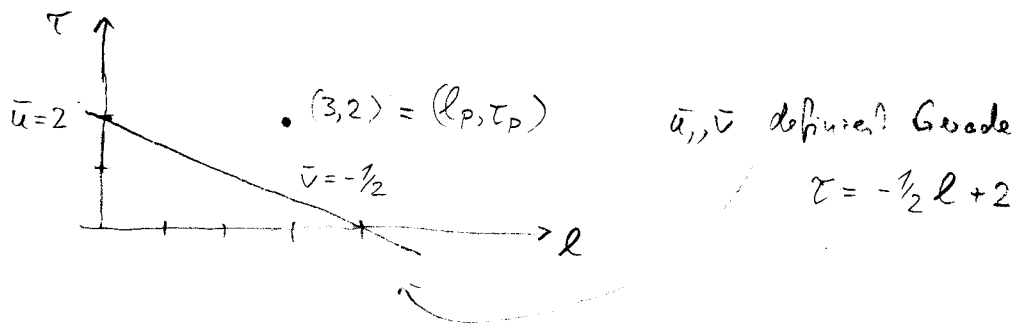
Nebenbedingungen = Halbebenen

Werte der Dualvariablen  $(u, v)$  = Punkte in der Ebene



andere, geometrisch duale Interpretation:

- Paar  $(u, v)$  als Gerade  $\tau = v \cdot l + u$  in der  $(l, \tau)$  Ebene auffassen
- Nebenbedingung  $u + l_p v \leq \tau_p \Leftrightarrow$  Punkt  $(l_p, \tau_p) \in$  Pfad  $P$
- Ungleichung  $\bar{u} + l_p \bar{v} \leq \tau_p$  erfüllt für Werte  $\bar{u}, \bar{v}$  von  $u, v$   
 $\Leftrightarrow (l_p, \tau_p)$  liegt oberhalb der Geraden  $\tau = \bar{v} l + \bar{u}$

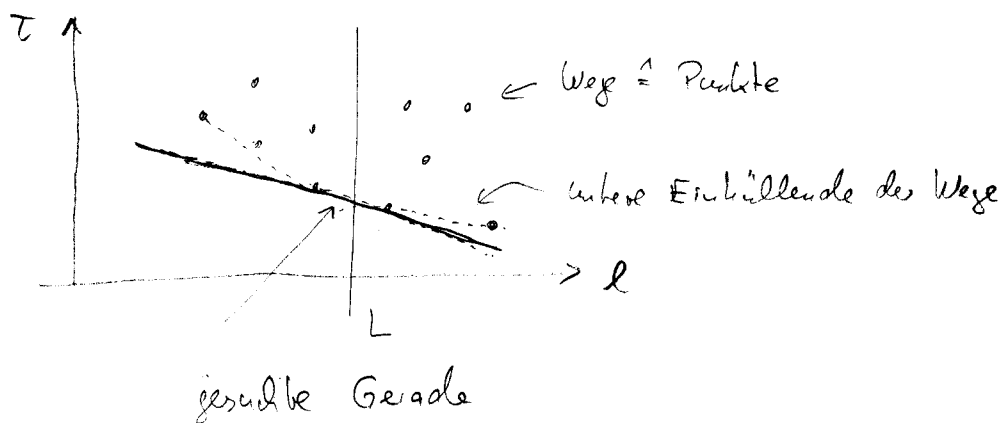


Punkt  $(3, 2)$  liegt oberhalb von  $\tau = -\frac{1}{2} l + 2$

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

- Optimierung  $\hat{=}$  Suche eines Paares  $u^*, v^*$  bzw. einer Geraden  $\tau = v^* \cdot l + u^*$  mit nicht-positiver Steigung ( $v^* \leq 0$ )  
 so dass:

1. alle Punkte  $(l_p, \tau_p)$ ,  $P$ -s-Weg, liegen oberhalb dieser Geraden
2. Der Schnittpunkt der Geraden  $\tau = v^*l + u^*$  mit der Geraden  $l = L$  ist so groß wie möglich, dass der Schnittpunkt hat den Wert  $v^*L + u^* = \text{Zielfkt. von (D)}$

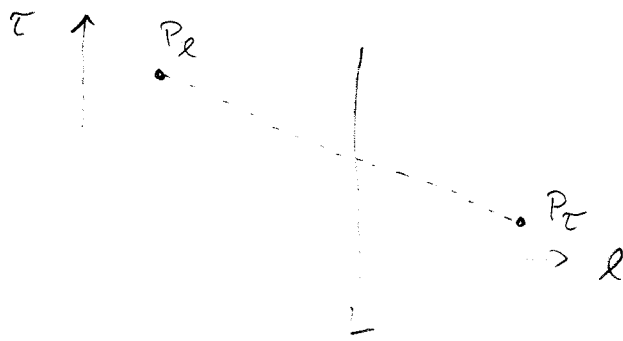


Konstruktion dieser optimalen Geraden erfolgt mit dem Hüllenansatz

Konstruktion der unteren Einhüllenden bei  $l = L$

1. Berechne kürzesten Weg  $P_{l_{\min}}$  bzgl  $l$   
 if  $l(P_{l_{\min}}) > L$  then return "keine zulässige Lösung"  
 else  $UB := \tau(P_{l_{\min}})$ ; // diese Schwanke für optimalen Weg

- Berechne kürzesten Weg  $P_{\tau_{\min}}$  bzgl  $\tau$   
 if  $l(P_{\tau_{\min}}) \leq L$  then return  $P_{\tau_{\min}}$  // optimal  
 else  $P_{\tau} := P_{\tau_{\min}}$ ;  $P_l := P_{l_{\min}}$  // Initialisierung  
 //  $P_{\tau}$  und  $P_l$  erzeugen erste Gerade  $\tau = \bar{v} \cdot l + \bar{u}$  mit  
 // Steigung  $\bar{v} = \frac{\tau(P_{\tau}) - \tau(P_l)}{l(P_{\tau}) - l(P_l)}$



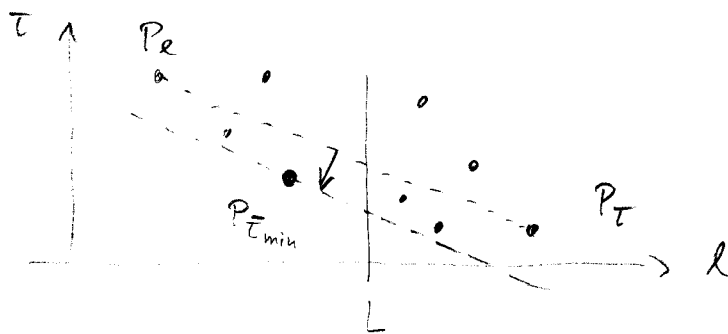
2. Prüfe ob es Punkte unterhalb der Geraden gibt.

Ein solcher Punkt entspricht einem Weg  $P$  mit

$$\bar{v} l_P + \bar{u} > \tau_P \Leftrightarrow \tau_P - \bar{v} \cdot l_P < \bar{u}$$

und kann durch kürzeste Wege Berechnung bzgl.  $\tau_a - \bar{v} l_a =: \bar{\tau}_a$  berechnet werden  
 $\geq 0$  wegen  $\bar{v} \leq 0$

Graphisch entspricht dies einer Parallelverschiebung der Geraden auf einen extremalen Punkt (den kürzesten Weg)



Sei  $P_{\bar{\tau}_{\min}}$  der neue Weg/Punkt bzgl.  $\bar{\tau}_a = \tau_a - \bar{v} l_a$

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

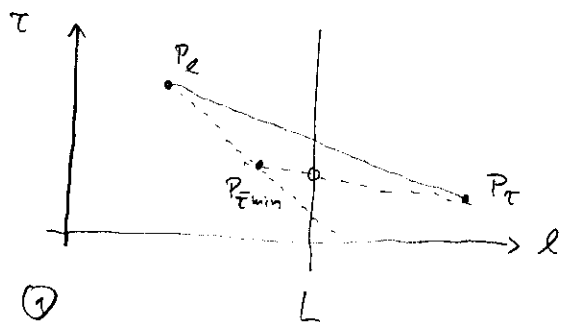
$\Rightarrow$  alte Gerade ist optimal,

Schnittpunkt mit  $l=L$  ist Optimum

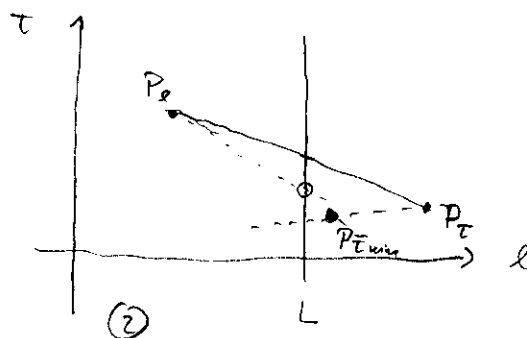
also Weitersuchen in 3

3. Aktualisiere die alte Gerade durch eine der

beiden Geraden durch  $P_L$  und  $P_{\bar{\tau}_{\min}}$  oder  $P_{\bar{\tau}_{\min}}$  und  $P_T$



①



②

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

hängt offenbar von der Lage des Punktes  $P_{\tau_{\min}}$  ab:

links von  $L$  (d.h.  $\tau(P_{\tau_{\min}}) \leq L$ )  $\Rightarrow$  ① ist besser

rechts von  $L$  (d.h.  $\tau(P_{\tau_{\min}}) > L$ )  $\Rightarrow$  ② ist besser

Fall ①: setze  $P_L := P_{\tau_{\min}}$ ;  $UB := \tau(P_{\tau_{\min}})$

und iteriere mit neuer Geraden  $\overline{P_L, P_T}$

Fall ②: setze  $P_T := P_{\tau_{\min}}$ ,  $UB$  kann nicht aktualisiert werden

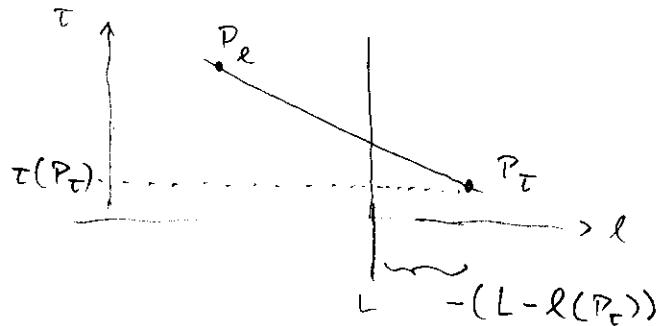
iteriere mit neuer Geraden  $\overline{P_L, P_T}$

Abbruch erfolgt in dem Fall, dass die Parallelverschiebung der aktuellen Gerade  $\tau = \bar{v} \cdot l + \bar{u}$  keine Verbesserung ergibt.

Wert LB des Optimums ergibt sich aus

$$LB = \bar{v} \cdot (L - l(P_T)) + \tau(P_T)$$

mit  $\bar{v} = \text{Steigung der neuen Gerade} = \frac{\tau(P_T) - \tau(P_e)}{l(P_T) - l(P_e)}$



LB ist untere Schranke für den Optimalwert des CSP - Problem

"

Opt.-Wert des LP-Relaxation

"

Opt.-Wert des Lagrange-Relaxation, da relaxiertes Problem

in der Lagrange Relaxation ganzzahlige Ecken hat

(ist kürzeste Wege Problem, in Knotenformulierung vollständig unimodular)

Insgesamt bleibt also dieselbe "Dualitätslücke" wie bei der Lagrange-Relaxation. Allerdings kann die Laufzeit besser abgeschätzt werden

4.4 SATZ (Kochlauer & Ziegelmann 00): Der Hüllensatz arbeitet

korrekt und hat eine Laufzeit von

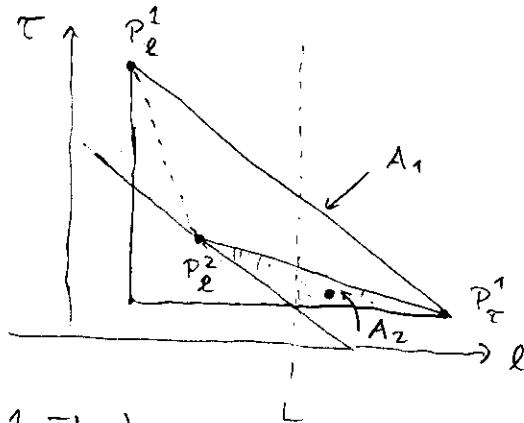
$$O(\underbrace{\log(n \tau_{\max} \cdot l_{\max})}_{\# \text{ Iterationen}} \cdot \underbrace{(n \log n + m)}_{\text{kürzeste Wege Berechnung}})$$

# Iterationen

kürzeste Wege Berechnung

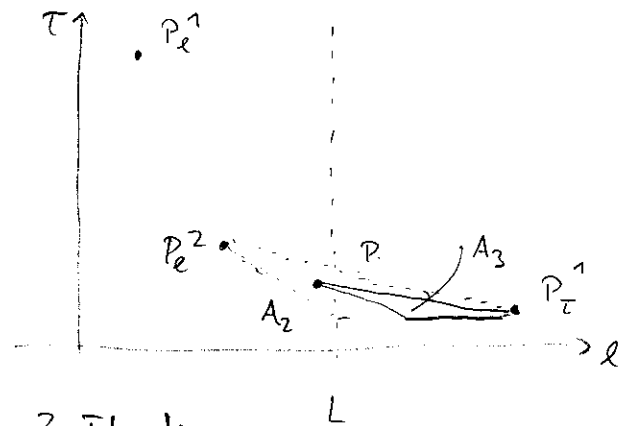
mit  $\tau_{\max} := \max_a \tau_a$ ,  $l_{\max} := \max_a l_a$

Beweis: Korrektheit lässt aus den vorausgesetzten Überlegungen  
 zur Laufzeit: geht über die Größe der "unentdeckten" Gebiete  
 an noch möglichen Punkten unterhalb der  
 momentanen Geraden



1. Iteration

Zu Beginn können  
 unentdeckte Punkte im  
 Dreieck  $A_1$ . Nach Parallel-  
 verschiebung der Geraden  $\overline{P_l^1 P_l^1}$   
 zum Punkt  $P_l^2$  können  
 unentdeckte Punkte nun noch  
 im Dreieck  $A_2$  liegen



2. Iteration

Jetzt wird  $P_l^3$  gefunden  
 und die Fläche verkleinert  
 sich weiter

Fläche eines Dreiecks ist maximal  $\frac{1}{2} \cdot (n \cdot l_{\max}) \cdot (n \cdot \tau_{\max})$   
 und minimal  $\frac{1}{2}$ , da alle Daten (= Punkte)  
 ganzzahlig sind

Zwischen 2 Iterationen  $i$  und  $i+1$  des Hüllensatzes gilt:

$$A_{i+1} \leq \frac{1}{4} A_i$$

**AUFGABE 7**

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

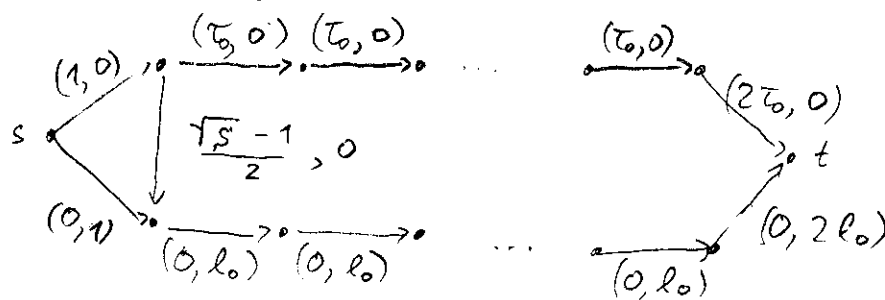
$\Rightarrow O(\log(n l_{\max} \tau_{\max}))$  Iterationen  $\square$

Bemerkung: kann zeigen dass  $O(m \log^2 m)$  Iterationen reichen  
(Jüttner 03)  $\Rightarrow$  streng polynomiales Algorithmus

Insgesamt bestimmt der Hüllensatz eine obere Schranke UB  
und eine untere Schranke LB für das Optimum des CSP

Die Güte der Schranken kann allerdings i. d. beliebig schlecht werden

4.5 BEISPIEL:



$$S := 1 + \frac{n}{2} \cdot \tau_0$$

$$n = \# \text{ Knoten} \geq 4 \\ \text{gerade}$$

$$L = \frac{n \cdot l_0}{2}, \quad l_0 > \tau_0 > 0$$

Hüllensatz findet  $UB = S$  und  $LB = \frac{S}{T}$  mit  $T := 1 + \frac{n}{2} \cdot l_0$

Optimale Kosten liegen bei  $\frac{\sqrt{S} + 1}{2}$

$\Rightarrow$  nur  $\Omega(\sqrt{n \cdot \tau_0})$  Approximationen für

AUFGABE 8

Schließen der Lücke zwischen LB und UB:

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

$\Rightarrow$  wesentliche Beschleunigung, kleinere Labellisten an Knoten  
und Gesamtlaufzeit mit am besten



## 4.5 Anwendung auf dem Constrained System Optimum

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

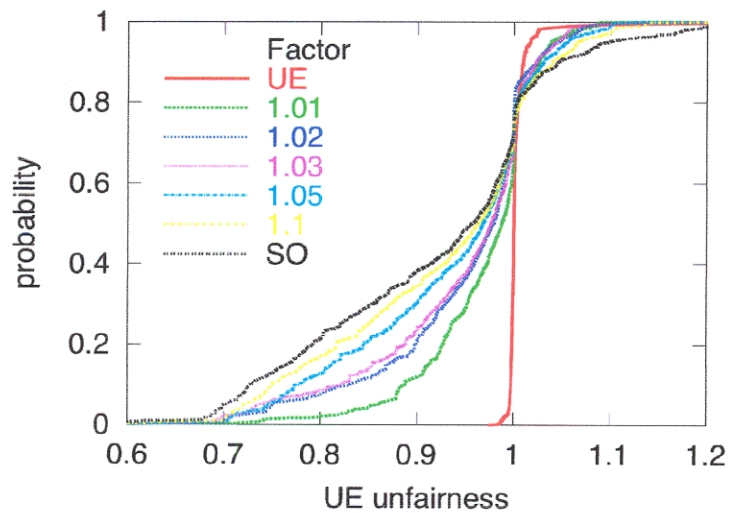
$$\text{unter } \sum_{P \in P_k^E} f_P = d_k$$

$$f_P \geq 0$$

→ lineares Problem im  
 Simplexverfahren zerfällt  
 in  $k$  CSP-Probleme

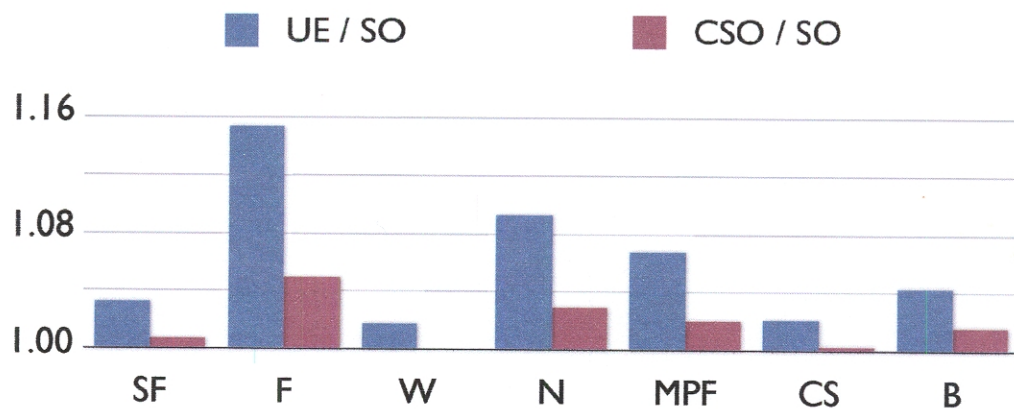
Ergebnisse: Papas, Jahn, H., Schulz, Stier → WWW

## Analysis of fairness



- 75% of the users travel less than in equilibrium
- Only 0.4% of the users travels 10% more than in equilibrium  
in SO users also 5%

## Results



*SF* Sioux Fall  
*F* Friedrichshain  
*W* Winnipeg

*N* Neukölln  
*MPF* Berlin Mitte  
*CS* Chicago Sketch

*B* = Berlin