§ 4 Constrained shortest paths

4.1 Komplexität

4.1 PROPOS. Das CSP ist schwach NP-vollständig

Beweis: Reduktion von PARTITION:

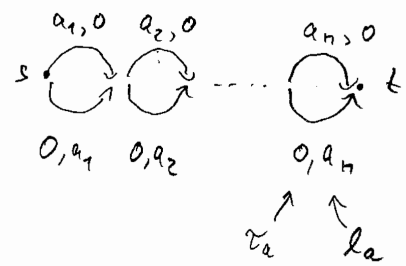
Geg. Zahlen $a_1, \ldots, a_n$ mit $\sum a_i = 2b$

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

Sei I Instanz von PARTITION.

Konstruiere daraus Instanz I' von CSP:

G :



Frage: Gibt es s,t Weg P mit $\tau(P) \leq b$, $\ell(P) \leq b$

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 pseudopolynomialer Zeit gelöst werden.

Beweis: analog zu SUBSET SUM (Verallgemeinerung von PARTITION) aus ADM I $\square$

Aufgabe 4.1

4.2 Lösungsansätze

1) Beasley & Christofides 89 Branch & Bound + Lagrange Relaxation

2) Labeling Algorithmus (erweiterter Dijkstra)

   Verwandschaft zu Pareto-optimalen Wegen

3) Geometrisch (Mehlhorn + Ziegelmann 2000) → § 4.4

4.2.1 Der Ansatz von Beasley & Christofides

Branch & Bound Baum ≙ wie bei Telekom Problem aus ADM II alt

≙ IP-Formulierung Festlegen von Kanten

Lagrange Relaxation:

$$\min \tau(P)$$

so daß $\ell(P) \leq L$] in Zielfkt

P ist Weg von s nach t

$$(LR_\mu) \qquad \min \underbrace{\tau(P) + \mu(\ell(P) - L)}_{\sum_{a \in P}(\tau_a + \mu \ell_a) - \underbrace{\mu L}_{\text{Konstant für festes } \mu}} =: \Lambda(\mu), \ \mu \in \mathbb{R}^1_+$$

↗

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

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

Dann gilt (1) $\Lambda(\mu) \le \tau^* \quad \forall \mu \geq 0$

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

$\boxed{\text{Aufgabe 4.2}}$ Ist P eine Optimallösung von (LR_μ) und $\ell(P) \le L$ $\Big\}$ $\overset{?}{\Rightarrow}$ P opt für CSPP

$\tau^* - \Lambda^*$ heißt <u>Dualitätslücke (gap)</u> $\boxed{\text{Aufgabe 4.3}}$ $\boxed{\text{i.A. ist gap} > 0}$

<u>Subgradientenverfahren</u> analog zu ADM II, § 7.4

Ist P_{μ_0} kürzester Weg in (LR_{μ_0}), so ist $\ell(P_{\mu_0}) - L$ (verletzte Restr.) ein Subgradient in μ_0 von $\Lambda(\mu)$

Gehe Schritt in diese Richtung

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

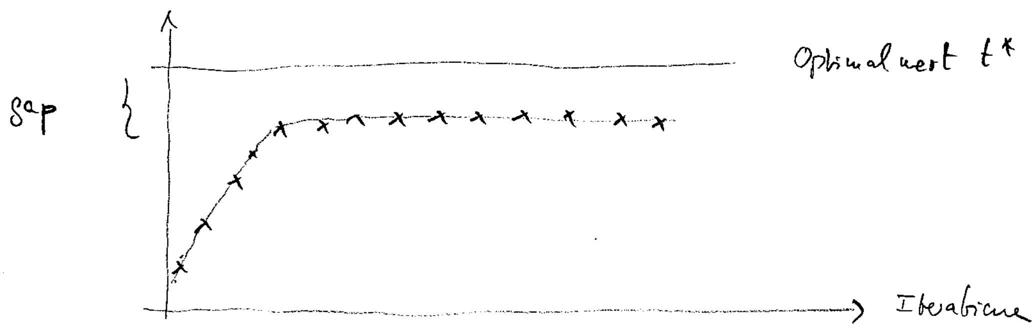
$\downarrow$

neue Kantenbewertung

$$\tau_a + \mu^{neu} \ell_a$$

Unterschiede in Kantenbewertung gerade $\underbrace{(\mu^{neu} - \mu^{alt})}_{\text{dieselbe Zahl für alle Kanten}} \ell_a$

typischer Verlauf



unbefriedigend

4.2.2 Pareto-optimale Wege

Ges. $G = (V, A)$ $\quad s, t \in V$

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

Σ, max, min ...

[allgemein: Semiringe]

Gesucht: Alle Pareto-optimalen 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

komponentenweise Ordnung

(keine Dominanz)

Anwendungen: Autoverkehr: Zeit, Entfernung, vertraute Strecken als Kriterien

Bahn Zeit, Kosten, # Umstiege

Der Dijkstra Algorithmus für Pareto-optimale Wege

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

- berechnet für jeden Knoten v eine Distanz d[v]
  $\hat{=}$ Länge kürzester 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] sind zunächst vorläufig
  Im Lauf des Algorithmus werden Knoten <u>markiert</u>, für markierte Knoten sind diese Werte endgültig

Initialisierung:

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

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

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 für Pareto-optimale Wege [Theune 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 - 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 $\longrightarrow$ Markierung von Vektoren $d[v]$

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

Aktualisierung der $d[w]$:

Betrachte zu 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

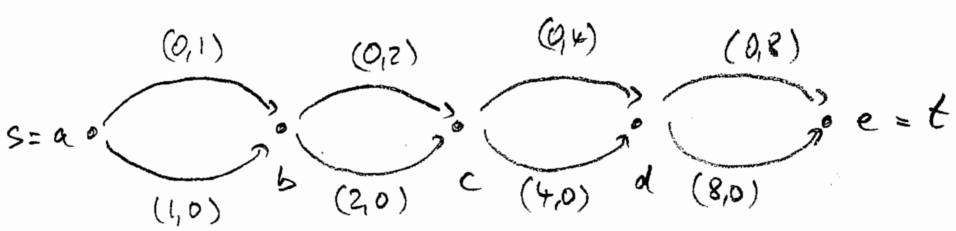
for alle Kanten (v,w) **do**

nehme $d[w] := d[v] + \lambda(v,w)$ zu den Vektoren von w hinzu;

streiche nicht Pareto-optimale bei w;

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

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)
2^0	2^1	2^2	2^3	2^4

exponentielles Wachstum $\Rightarrow$ exponentieller Algorithmus

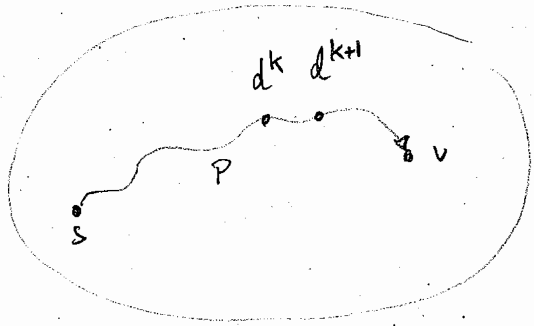
Korrektheit analog zu normalem Dijkstra per Induktion:
nach # Markierungen:

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

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

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

$\Rightarrow \exists$ 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 dem Weg P und $d^1, d^2, \dots, d^m$ die zugehörigen Pareto-optimalen Bewertungsvektoren, also $d^i = d^{i-1} + \lambda_{(v_{i-1}, v_i)}$. Dann ist $d^m \lneq d[v]$.

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

$\Rightarrow k < m$, da sonst $d^m \lneq d[v]$ bereits bei v markiert wäre und $d(v)$ gestrichen wäre.

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

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

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

$\Rightarrow$ $\boxed{\text{Jeder markierte Vektor ist Pareto-optimal}}$

Analoger Beweis: jedes Pareto-Optimum wird im Algorithmus konstruiert □

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

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

$L_k(v) :=$ Maximale Weglänge (elementarer Weg) von s nach v bzgl λ_k

pareto(v) := # Pareto Optima in v

$$\boxed{\text{pareto}(v) \le \prod_{k=1}^{r} L_k(v)}$$

$\Rightarrow$ pareto(v) ist polynomiell in n, falls $\lambda_k(a)$ polynomiell in n und r fest ist $\Rightarrow$ Algorithmus polynomiell

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

$\lambda_k(a) \le L_0 \Rightarrow \text{pareto}(v) \le (L_0(n-1))^r$

erfüllt in Straßennetz ($\le$ 1000 m), $n \sim$ 10.000 in Berlin

$\boxed{\text{Anwendung bei CSP}}$ [Aneja, Aggarwal & Nair 1983]

min $\tau(P)$ unter $\ell(P) \le L$ P s.t. Weg

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

Pareto Dijkstra anwenden

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

$\Rightarrow$ liefert alle Pareto optima mit $\ell \le L$

$\Rightarrow$ lex.-kleinstes Pareto-optimum ist Optimallösung von CSPP
(Abbrechen falls t zum erstenmal erreicht)

Praktische Erfahrung:

2 Subgradientenschritte $\gtrsim$ Pareto Dijkstra (ungefähr gleich)

angenähert — optimal

max. Unterschied Faktor 1 – 20

für mehr Schritte deutlich schlechter!

4.3 Approximation Pareto-optimaler Wege

1. Gewichte Summe der Kriterien (Jaffe 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 bzgl $\lambda'(a) := \alpha_1 \lambda_1(a) + \dots + \alpha_r \lambda_r(a)$

Gewichte, die von $L_1, .., L_r$ abhängen

i.a. nur schlechte Approximationen

Bsp. $\alpha_1 \leq \alpha_2$ $\Rightarrow$ $(9,15)$ wird berechnet im Beispiel

schlecht für $L_1 = 7$ $L_2 = 8$

allgemein gilt [Thenne 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

Dann ist $\max\limits_k \lambda(P') \leq r \cdot \max\limits_k L_k$ (i.a. scharf)

in Bsp ist für $\alpha_1 = \alpha_2$ der Weg P' über die oberen Kanten ein kürzester

mit $\max\limits_k \lambda(P') = 15 \leq 2 \cdot \max\{7,8\}$

2. Einfache Skalierung der Längen [Theune 95]

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

$$\Downarrow$$

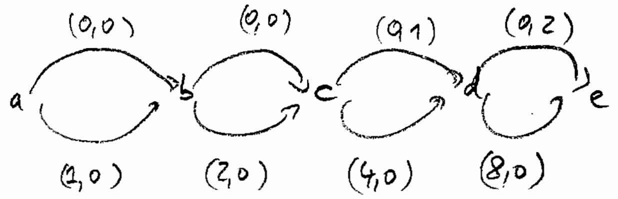
$$\lambda'(a) = \left(\lambda_1(a), \left\lfloor \frac{\lambda_2(a)}{\alpha} \right\rfloor, \ldots, \left\lfloor \frac{\lambda_r(a)}{\alpha} \right\rfloor \right)^T \qquad \alpha > 1$$

$\Rightarrow$ pareto(v) wird kleiner!

Skalierung erhält Ordnung $\Rightarrow$ Berechnete pareto-optimale Wege bzgl λ' sind "nahezu" pareto-optimal bzgl. λ

Bsp: (wenige Pareto optima)

$\alpha = 4$



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)

zugeh. 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 aller Pareto Optima X

Genauer:

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

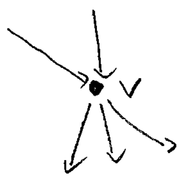
$\leftarrow$ pessimistisch aber i.A. scharf

Bsp $d = (9,6) \rightsquigarrow d' = (8,7)$
$(13,2) \rightsquigarrow (12,3)$

3. Ein voll-polynomiales Approximationsschema [Hassin 92]

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

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)



(τ, ℓ)
1,60
2,59
5,41
6,39
⋮
35,21
36,20
40,13

$\Rightarrow$

	(τ, ℓ)
	5,59
5	
	10,39
10	
	⋮
35	
	20,40
40	
	43,13

$(1+\varepsilon)\tau^* = 43$

In 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: <u>ε-Schubfach Dijkstra</u>

- Gehe vor wie beim verallgemeinerten Dijkstra Algorithmus
- Trick: gebe pro Schubfach nur ein Label weiter:
  das bzgl. ℓ-Wert beste Label (sofern $\ell \le L$)

  runde den τ-Wert des Labels auf die obere Grenze des Schubfaches
- Wähle am Ende den Weg mit kleinstem τ-Wert

$\Downarrow$

Gefundener Weg ist zulässig bzgl. Längenschranke L
(aber i.a. nicht optimal)

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

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

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

P^* ist elementar $\Rightarrow$ $\leq n-1$ Kanten. Der ε-Dijkstra macht bei P^* pro Kante Rundungsfehler im τ-Wert $\leq$ Intervall-Länge $= \dfrac{\tau^*}{(n-1)/\varepsilon}$

$\Rightarrow$ Gesamtfehler $\leq (n-1)\dfrac{\tau^*}{(n-1)/\varepsilon} = \tau^* \cdot \varepsilon$ bzgl. P^*

$\Rightarrow$ $\tau(P_\varepsilon) \leq \tau(P^*) \leq \tau^* + \tau^* \cdot \varepsilon = (1+\varepsilon)\tau^*$

da P_ε kürzester Weg im ε-Dijkstra

Problem: Kennen τ^* nicht

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

Kein längenbeschränkter Weg für aktuelles τ gefunden $\Rightarrow$ rechts weitersuchen

Weg gefunden $\Rightarrow$ links weitersuchen

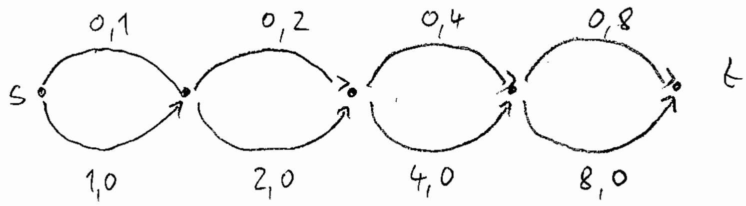
$\Rightarrow$ $\log((n-1)\tau_{max})$ viele Aufrufe von ε-Schubfach Dijkstra

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

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

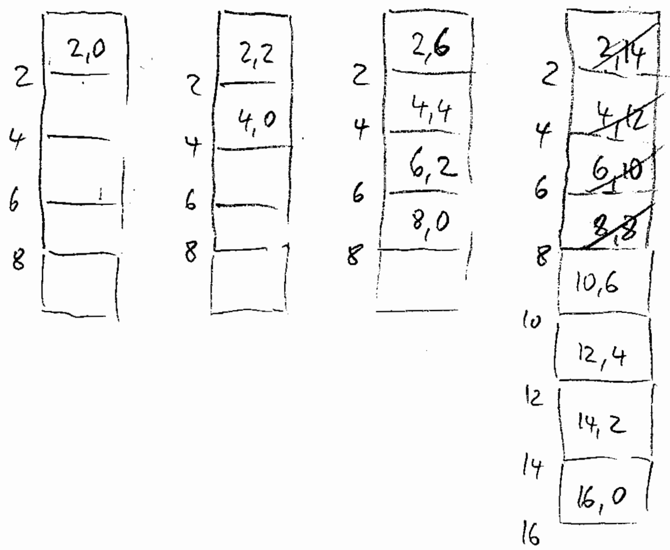
$\Rightarrow$ Intervall-Länge $\dfrac{\tau^*}{(n-1)/\varepsilon} = \dfrac{8}{4} = 2$

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



$L = 7$

$\tau^* = 8$



violate length bound $L = 7$

← beste Weg (approx.)

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

nutzt Weg-basierte IP-Formulierung und LP-Relaxation

0,1 Variable x_p für jeden st-Weg $\quad \tau_p := \sum_{a \in P} \tau_a, \; \ell_p := \sum_{a \in P} \ell_a$

$\Rightarrow$ Primales IP:

$$\min \sum_P \tau_p x_p \qquad \leftarrow \tau_p \text{ minimieren}$$

$$\text{unter } \sum_P \ell_p x_p \le L \qquad \leftarrow \text{Längenschranke eingehalten}$$

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

LP Relaxierung:

$$\min \sum \tau_p x_p$$

$$v \qquad \text{unter } -\sum_P \ell_p x_p \ge -L$$

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

$$x_p \ge 0 \qquad \leftarrow x_p \le 1 \text{ tritt im Optimum automatisch ein, da } \tau_p \ge 0$$

Duales LP:

$$\max \; u - Lv$$

$$\text{unter } u - \ell_p v \le \tau_p \quad \forall P$$

$$v \ge 0, \; u \text{ nicht vorzeichenbeschränkt}$$

Ersetzung $v \to -v$ ergibt

$$\max \; u + Lv$$

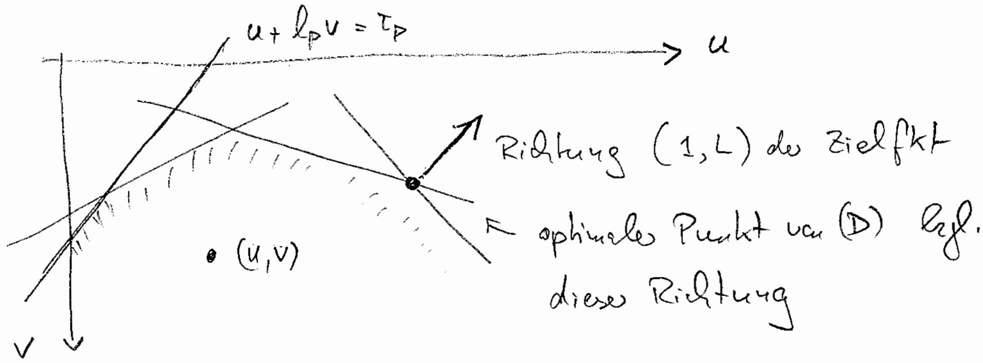
$$\text{unter } u + \ell_p v \le \tau_p \quad \forall P \qquad (D)$$

$$v \le 0 \quad u \text{ nicht vorzeichenbeschränkt}$$

Standardinterpretation von (D) als 2-dim LP

Nebenbedingungen = Halbebenen

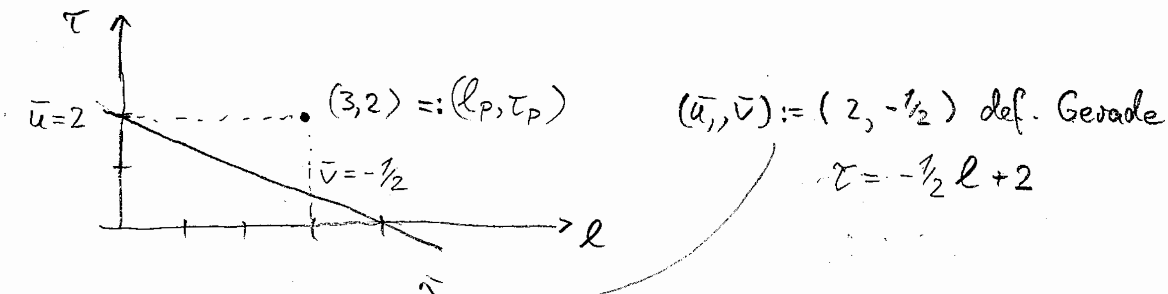
Werte der der Dualvariablen (u,v) = Punkte in der Ebene



andere, geometrisch duale Interpretation:

- Paar (u,v) als Gerade $\tau = v \cdot l + u$ in der (l,τ) Ebene auffassen
- Nebenbedingung $u + l_p v \leq \tau_p \;\hat{=}\;$ Punkt $(l_p, \tau_p) \;\hat{=}\;$ 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, τ_p) liegt oberhalb der Geraden $\tau = \bar{v}\, l + \bar{u}$



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

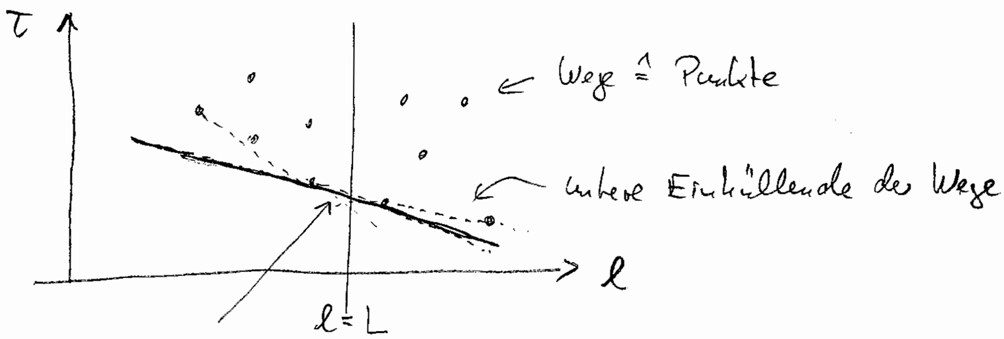
$$\Leftrightarrow \quad 2 \geq \underbrace{-1/2 \cdot 3 + 2}_{0,5}$$

- 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, 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 Schnittpkt hat den Wert $v^* L + u^*$ = Zielfkt. von (D)



Konstruktion dieser optimalen Geraden erfolgt mit dem <u>Hüllenansatz</u>

Konstruktion der unteren Einhüllenden bei $\ell = L$

1. Berechne 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 für optimalen Weg

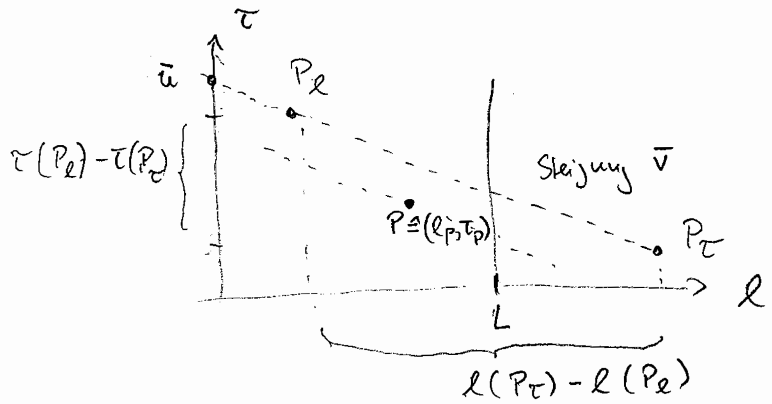
   Berechne kürzesten Weg $P_{\tau_{min}}$ bzgl τ

   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 $\tau = \bar{v} \cdot \ell + \bar{u}$ mit

   // Steigung $\bar{v} = \dfrac{\tau(P_\tau) - \tau(P_\ell)}{\ell(P_\tau) - \ell(P_\ell)} < 0$



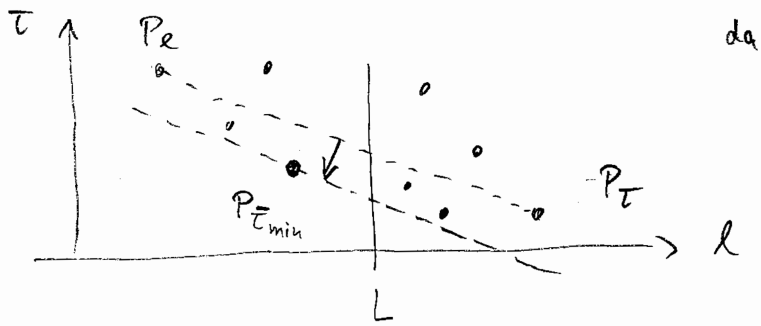
2. Prüfe ob es Punkte unterhalb der Geraden gibt.

Ein solcher Punkt entspricht einem Weg P mit

$$\bar v \ell_p + \bar u > \tau_p \iff \tau_p - \bar v \cdot \ell_p < \bar u$$

und kann durch kürzeste Wege-Berechnung bzgl $\underbrace{\tau_a - \bar v \ell_a}_{\geq 0 \text{ wegen } \bar v \leq 0} =: \bar\tau_a$ berechnet werden

Graphisch entspricht dies einer Parallelverschiebung der Geraden auf einen extremalen Punkt (dem kürzesten Weg) da $\bar v$ (Steigung) unverändert



Sei $P_{\bar\tau_{min}}$ der neue Weg/Punkt bzgl. $\bar\tau_a := \tau_a - \bar v \ell_a$

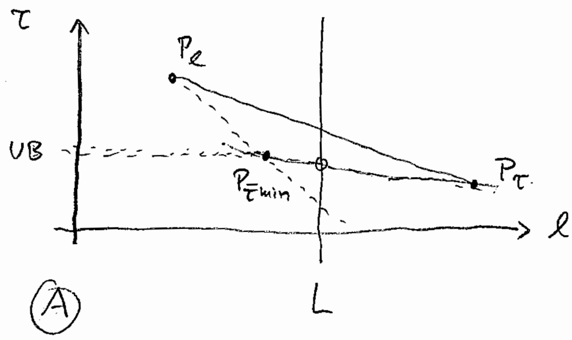
if $\bar\tau(P_{\bar\tau_{min}}) = \bar u \Rightarrow P_{\bar\tau_{min}}$ liegt auf der alten Geraden

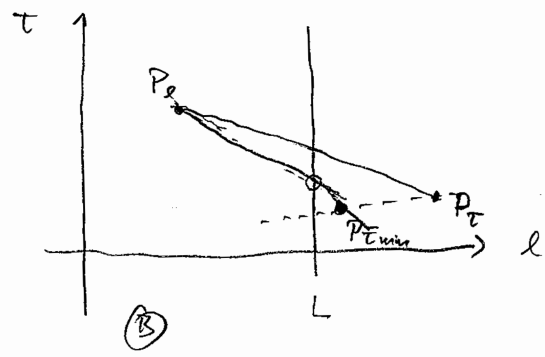
$\Rightarrow$ alte Gerade ist optimal,

Schnittpunkt mit $\ell = L$ ist Optimum

else Weitersuchen in 3

3. Aktualisiere die alte Gerade durch eine der beiden Geraden durch P_ℓ und $P_{\bar\tau_{min}}$ oder $P_{\bar\tau_{min}}$ und P_τ





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

hängt offenbar von der Lage des Punktes $P_{\bar{\tau}_{min}}$ ab:

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

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

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

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

(2) setze $P_\tau := P_{\bar{\tau}_{min}}$, UB kann nicht aktualisiert werden da $\ell(P_{\bar{\tau}_{min}}) > L$

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

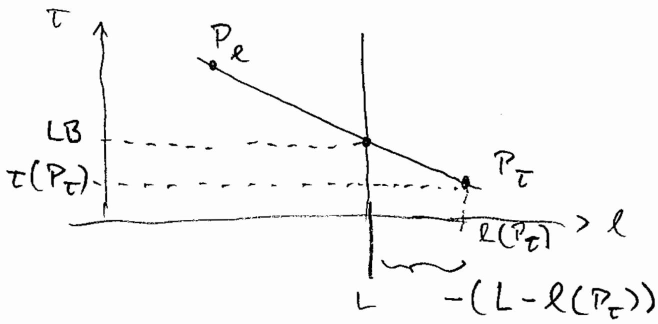
↑ nicht längenbeschränkt

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

Untere Schranke LB an das Optimum des IP ergibt sich aus

$$LB = \bar{v} \cdot (L - \ell(P_\tau)) + \tau(P_\tau)$$

mit $\bar{v}$ = Steigung der finalen Gerade $= \dfrac{\tau(P_\tau) - \tau(P_\ell)}{\ell(P_\tau) - \ell(P_\ell)}$



Untere Schranke LB an OPT des CSP

=

Opt.-Wert der LP-Relaxation

=

Opt.-Wert der Lagrange-Relaxation, da relaxiertes Problem in der Lagrange Relaxation ganzzahlige Ecken hat (ist kürzeste Wege Problem, in Kantenformulierung vollständig unimodular)

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

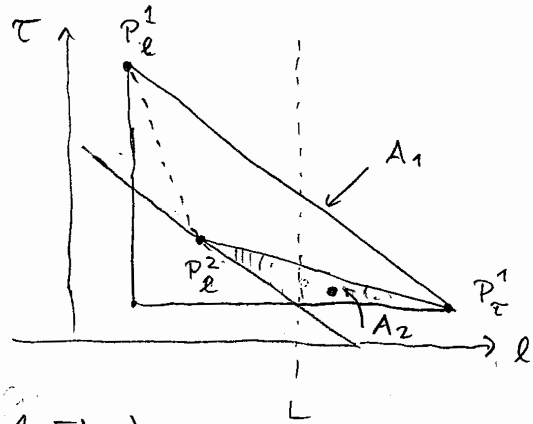
4.4 SATZ (Mehlhorn & Ziegelmann 00): Der Hüllenansatz arbeitet korrekt und hat eine Laufzeit von

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

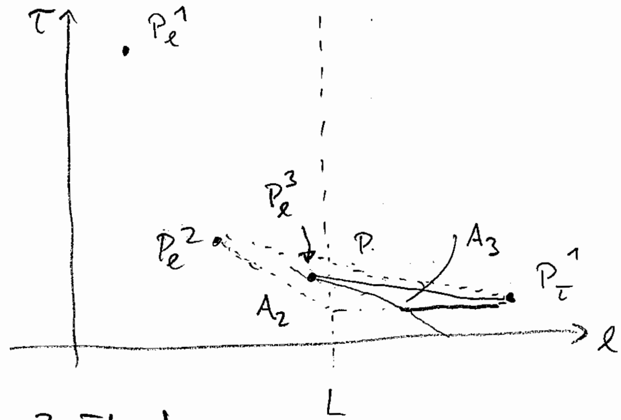
mit $\tau_{max} := \max_a \tau_a$, $\ell_{max} := \max_a \ell_a$

Beweis: Korrektheit klar aus den vorangegangenen Überlegungen

zur Laufzeit: geht über die Größe der "unerforschten" Gebiete von noch möglichen Punkten unterhalb der momentanen Geraden



1. Iteration



2. Iteration

Zu Beginn können unentdeckte Punkte im Dreieck A_1. Nach Parallelverschiebung der Geraden $\overline{P_\ell^1 P_\tau^1}$ zum Punkt P_ℓ^2 können unentdeckte Punkte nur noch im Dreieck A_2 liegen

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

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

und minimal $\frac{1}{2}$, da alle Daten (⇒ Punkte) ganzzahlig sind

~~Für~~ Iterationen i und $i+1$ des Hüllenansatzes gilt

$$A_{i+1} \leq \frac{1}{4} A_i \qquad \boxed{\text{AUFGABE 4.4}}$$

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

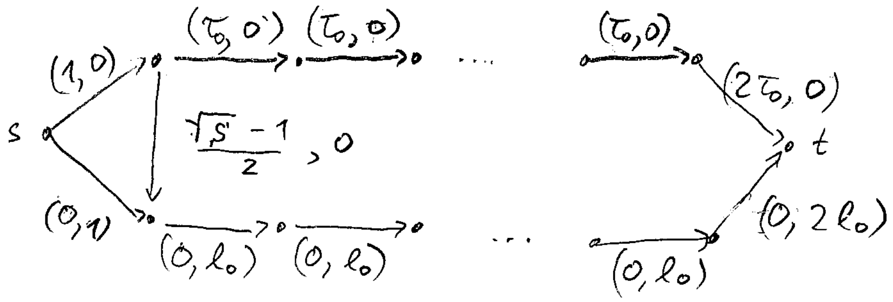
$\Rightarrow O(\log(n \ell_{max} \tau_{max}))$ Iterationen $\square$

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

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

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

4.5 BEISPIEL:



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

$n = \#$ Knoten ≥ 4 gerade

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

Hüllenansatz 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 $\Omega(\sqrt{n \cdot \tau_0})$ Approximationsgüte (ergibt sich aus $\frac{OPT}{LB}$)

$\boxed{\text{AUFGABE 4.5}}$

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

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

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

4.5 Anwendung auf das Constrained System Optimum

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

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

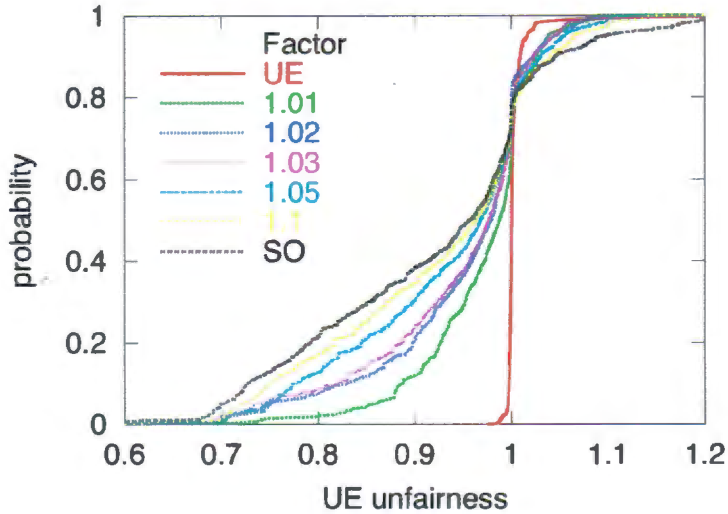
$f_P \geq 0$

→ lineares Problem im Abstiegsverfahren zerfällt in k CSP-Probleme

Ergebnisse: Paper Jahn, M., Schulz, Stier → www

CPP gelöst mit Pareto Dijkstra

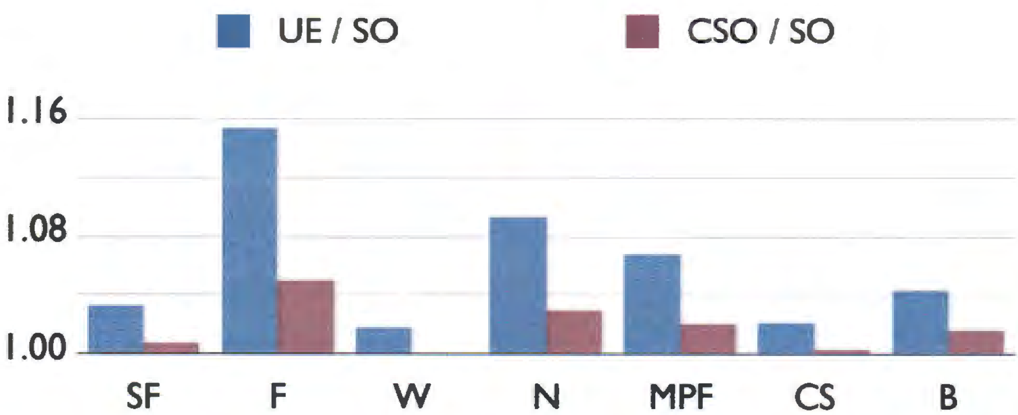
Analysis of fairness



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

in SO mehr als 5%

Results



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

B = Berlin