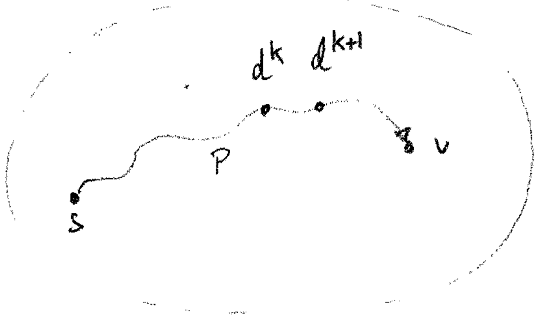
Beweis: Betrachte Zeitpunkt zu dem $d[v]$ gewählt wird

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

$\Rightarrow \exists \overbrace{\text{Weg}}^{\text{Pareto-opt.}} P$ von s nach v mit $\lambda(P) \lneq d[v]$



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

(*)

Seien $s = v_1, v_2, \ldots, v_m = v$ die Zwischenknoten auf dem Weg P und $d^1, d^2, \ldots, d^m$ die zugehörigen Pareto-optimalen Bewertungsvektoren, also $d^i = d^{i-1} + \lambda_{(v_{i-1}, v_i)}$

Sei d^k der letzte Vektor $\overset{\text{auf } P}{}$ 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 zur 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 (≤ 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 Fehler 1 – 20

für mehr Schritte deutlich schlechter!

4.3 Approximation Pareto-optimaler Wege

1. Gewichtete Summe der Kriterien (Jaffe 84)

Zulässigkeitsproblem

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

Transformation

Kürzeste Wege Problem bzgl $\lambda'(a) := \alpha_1 \lambda_1(a) + \dots + \alpha_r \lambda_r(a)$

Gewichte, die von $L_1, \dots, L_r$ abhängen

i.a. nur schlechte Approximationen

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

schlecht für $L_1 = 7 \quad L_2 = 8$

allgemein gilt [Theorem 95]

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

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

---

Dann ist $\max_k \lambda(P') \le r \cdot \max_k L_k$ (i.a. scharf)

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

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

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

$$\downarrow$$

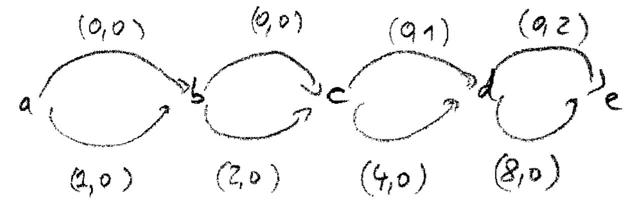
$$\lambda'(a) = \left(\lambda_1(a), \left\lfloor \frac{\lambda_2(a)}{\alpha} \right\rfloor, \dots, \left\lfloor \frac{\lambda_r(a)}{\alpha} \right\rfloor \right)^T \qquad \alpha > 1$$

$\Rightarrow$ pareto(v) wird kleiner!

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

Bsp: (weniger 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 $|d_k - d'_k| \leq \ell\,(\alpha - 1)$ $\quad k = 1, \dots, r$

ℓ: # Kanten von P'

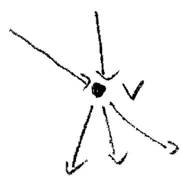
$\leftarrow$ pessimistisch aber i.A. scharf

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 an, die das Intervall $[0, (1+\varepsilon)\tau^*]$ äquidistant unterteilen (bis auf das letzte)



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

$\Rightarrow$

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

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
  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

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

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 $= \frac{\tau^*}{(n-1)/\varepsilon}$

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

$$\Rightarrow \tau(P^*) < \tau^* + \tau^* \cdot \varepsilon = (1+\varepsilon)\tau^* \underset{\uparrow}{\geq} \tau(P_\varepsilon)$$

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

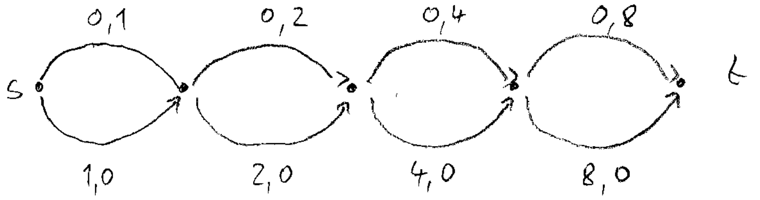
$\Rightarrow$ $\log((n-1)\tau_{max})$ viele Aufrufe von ε-Schubfach Dijkstra

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

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

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

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



$L = 7$

$\tau^* = 8$

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

← beste Weg