4. Metrische Aufgabensysteme (Metrical Task Systems)

Eiscremeproblem:

Maschine M kann zwei Sorten Eis produzieren.

Sorte	Kosten
Vanille	1
Schokolade	2

- nur eine Sorte gleichzeitig
- Kosten für Sortenwechsel betragen 1.

Ohne Maschine kann Eis mit folgenden Kosten produziert werden:

Sorte	Kosten
Vanille	2
Schokolade	4

Bestellung einer Einheit Eis (Vanille oder Schokolade

$\leadsto$

- Verwendung von M (eventuell mit Sortenwechsel)
- ohne Verwendung von M.

Frage:

Wie soll bei einer Folge von Bestellungen das Eis produziert werden?

Ziel dabei ist, die Kosten zu minimieren.

Das Eiscremeproblem ist ein Beispiel für ein metrisches Aufgabensystem. Formal ist ein metrisches Aufgabensystem wie folgt definiert:

Sei $M = (S, d)$ ein metrischer Raum. Einer Aufgabe J ist eine Abbildung $J : S \mapsto \mathbb{R}^+$ zugeordnet. Dabei bezeichnet $J(x)$, $x \in S$ die Kosten für die Erledigung der Aufgabe J im Zustand x.

Sei T eine endliche Menge von Aufgaben. Dann heißt das Paar (M, T) <u>metrisches Aufgabensystem</u> (MTS).

Seien

- $x_0 \in S$ der Startzustand
- $\bar{J} = J_1 \, J_2 \, \ldots \, J_m$ eine Aufgabenfolge und
- $\bar{x} = x_0, x_1, x_2, \ldots, x_m$, $x_i \in S$ ein (Service) Schedule

Definiere

$$\mathrm{cost}(x_0, \bar{J}, \bar{x}) = \sum_{i=1}^{m} \left[d(x_{i-1}, x_i) + J_i(x_i) \right]$$

und

$$\mathrm{opt}(x_0, \bar{J}) = \min_{\bar{x}} \left\{ \mathrm{cost}(x_0, \bar{J}, \bar{x}) \right\}$$

<u>Ziel</u>:

Entwicklung von Online-Strategien für MTS.

Sei $ALG : S \times T^* \to S$ eine Abbildung.
Die Abbildung ALG berechnet, gegeben einen Startzustand x_0, eine Aufgabenfolge $\bar{\sigma}$ einen Zustand $ALG(x_0, \bar{\sigma}) \in S$.

Seien $\bar{\sigma} = \sigma_1 \sigma_2 \dots \sigma_m$ und $x_i = ALG(x_0, \sigma_1 \dots \sigma_i)$ für $1 \le i \le m$. Die Folge $x_0, x_1, x_2, \dots, x_m$ heißt der <u>von ALG produzierte Schedule</u>.

Die <u>Kosten</u> $cost_{ALG}(x_0, \bar{\sigma})$ von ALG auf $x_0, \bar{\sigma}$ sind definiert durch

$$cost_{ALG}(x_0, \bar{\sigma}) := cost(x_0, \bar{\sigma}, \bar{x}),$$

wobei $\bar{x}$ der von ALG produzierte Schedule ist.

ALG heißt <u>c-competitive mit Startfunktion</u> $\alpha : S \to \mathbb{R}$, falls $\forall x_0 \in S \ \forall \bar{\sigma} \in T^*$ gilt:

$$cost_{ALG}(x_0, \bar{\sigma}) \le c \cdot opt(x_0, \bar{\sigma}) + \alpha(x_0)$$

Die kleinste Konstante c mit ALG ist c-competitive heißt <u>competitives Verhältnis</u> von ALG.

Ein <u>metrisches Dienstsystem</u> (MSS) ist ein Paar (M, R), wobei $M = (S, d)$ ein endlicher metrischer Raum und R eine Familie von nichtleeren Teilmengen von S sind. Ein Element von R heißt <u>Anfrage</u>. Falls $\rho \in R$ angefragt wird, dann wird ein Server zu einem Punkt in ρ bewegt, wo er dann die Anfrage bearbeitet. (M, R) heißt <u>unbeschränkt</u>, falls R alle nichtleeren Teilmengen von S enthält.

Übung:
Zeigen Sie, dass MSS eine Verallgemeinerung des k-Server-Problems ist.

Analog zum k-Server-Problem können wir auch bei metrischen Aufgabensystemen Arbeitsfunktionen definieren um dann mit Hilfe dieser Online-Algorithmen zu entwickeln.

Seien (M, T) ein metrisches Aufgabensystem, x_0 ein Startzustand und $\bar{J} = J_1 J_2 \dots J_m$ eine Aufgabenfolge. Die zu x_0 und $\bar{J}$ assoziierte <u>Arbeitsfunktion</u> $w : S \to \mathbb{R}$ ist wie folgt definiert:

$$w(x) := \min_{\bar{x}} \{ cost(x_0, \bar{J}, \bar{x}) + d(x_m, x) \},$$

wobei $\bar{x} = x_0 x_1 \dots, x_m$. D.h., $w(x)$ sind die minimalen Kosten zur Erledigung von $\bar{J}$, gestartet in x_0 und beendet in x.

Bemerkung:

Ursprünglich wurden Arbeitsfunktionen für metrische Aufgabensysteme erfunden und fanden danach ihre Anwendung beim k-Server-Problem.

Beobachtung:

Die optimalen Kosten opt$(x_0, \bar{\sigma})$ zur Erledigung von $\bar{\sigma}$ gestartet in x_0 ergeben sich aus

$$\mathrm{opt}(x_0, \bar{\sigma}) = \min\{\, w(x) \mid x \in S \,\}.$$

Zu Aufgabenfolgen assoziierte Arbeitsfunktionen heißen erreichbar.

Bemerkung:

Beim k-Server-Problem haben wir gesehen, dass die Arbeitsfunktion sämtliche Information enthält, die der Arbeitsfunktionenalgorithmus benötigt, um den nächsten Schritt zu entscheiden. Auch hier werden wir einen Arbeitsfunktionenalgorithmus entwickeln, für den dann auch entsprechendes erfüllt sein wird.

Ziel:

Entwicklung von Werkzeugen zum schrittweisen Update der Arbeitsfunktion.

Zum Erreichen des Zieles ist zunächst nützlich, Eigenschaften von erreichbaren Arbeitsfunktionen herauszuarbeiten.

Eigenschaften:

- $w(x) - w(y) \leq d(x,y) \quad \forall\, x,y \in S$

- <u>Bezeichnungen:</u>

  w wird <u>gestützt</u> von $K \subseteq S$, falls $\forall x \in S$ $\exists y \in K$ mit $w(x) = w(y) + d(y,x)$.

  Eine <u>Stützmenge</u> für w ist eine kleinste Teilmenge $K \subseteq S$, die w stützt.

  Eine Arbeitsfunktion, die von einem einzelnen Punkt in S gestützt wird, heißt <u>Kegel</u>.

  χ_x bezeichnet den von $\{x\}$ gestützten Kegel mit $w(x) = 0$.

  Sei x_0 der Startzustand. Dann ist χ_{x_0} <u>die Startarbeitsfunktion.</u>

  Sei $X \subseteq S$. χ_X bezeichnet den <u>verallgemeinerten Kegel</u> auf X. D.h., diejenige Arbeitsfunktion w, für die gilt:

  $$X \text{ stützt } w \text{ und } w(x) = 0 \quad \forall x \in X.$$

- <u>Update der Arbeitsfunktion</u> (Startzustand x_0)

  - Startarbeitsfunktion χ_{x_0}

  - Arbeitsfunktion w, $\sigma \in T$. Dann bezeichnet $w \wedge \sigma$ folgende Arbeitsfunktion:

  $$\underset{\substack{\uparrow \\ \text{Updateoperator}}}{w \wedge \sigma}(x) := \min_{y \in S} \{ w(y) + \sigma(y) + d(y,x) \} \quad \forall x \in S.$$

Sei $\overline{J} = J_1 J_2 \dots J_m$ eine Aufgabenfolge. Dann schreiben wir

$$w \wedge \overline{J} := w \wedge J_1 \wedge J_2 \wedge \dots \wedge J_m.$$

Lemma 4.1

$$w \wedge J(x) = w(y) + J(y) + d(y,x)$$

$$\Longrightarrow$$

$$w \wedge J(y) = w(y) + J(y).$$

Beweis:

$\forall\, z \in S$ gilt wegen $w \wedge J(x) \leq w(z) + J(z) + d(z,x)$

$$w(y) + J(y) + d(y,x) \leq w(z) + J(z) + d(z,x)$$

$$\Longleftrightarrow \quad w(y) + J(y) \leq w(z) + J(z) + d(z,x) - d(y,x)$$

$$\underset{\Delta\text{-Ungl.}}{\leq} w(z) + J(z) + d(z,y)$$

Also folgt aus der Definition von $w \wedge J(y)$

$$w \wedge J(y) = w(y) + J(y).$$

■

Mit Hilfe des Lemmas 3.1 lässt sich unmittelbar folgendes Korollar beweisen:

Korollar 4.1

Falls x Element der Stützmenge von $w \wedge J$ ist, dann gilt:

$$w \wedge J(x) = w(x) + J(x).$$

Beweis:

Annahme: $w \wedge J(x) \neq w(x) + J(x)$

$\Longrightarrow$

$\exists\, y \in S \setminus \{x\}: \quad w \wedge J(x) = w(y) + J(y) + d(y,x)$

Lemma .1 $\Longrightarrow$

$w \wedge J(y) = w(y) + J(y)$

Also gilt

$$(*) \quad w \wedge J(x) = w \wedge J(y) + d(y,x)$$

Somit gilt für alle $z \in S$:

$$\begin{aligned} w \wedge J(x) + d(x,z) &= w \wedge J(y) + \underbrace{d(y,x) + d(x,z)}_{\geq d(y,z)} \\ &\geq w \wedge J(y) + d(y,z) \\ &= w(y) + J(y) + d(y,z) \\ &\underset{\text{Def.}}{\geq} w \wedge J(z) \end{aligned}$$

Also gilt für alle $z \in S$ mit

$w \wedge J(z) = w \wedge J(x) + d(x,z)$

auch

$$(**) \quad w \wedge J(z) = w \wedge J(y) + d(y,z)$$

Wegen (*) gilt (Beachte $y \neq x$)

$$w \wedge J(y) \neq w \wedge J(x) + d(x,y).$$

$\Rightarrow$ in der Stützmenge K existiert $u \neq x$ mit

$$w \wedge J(y) = w \wedge J(u) + d(u,y)$$

In (**) eingesetzt ergibt dies

$$\begin{aligned}
w \wedge J(z) &= w \wedge J(u) + \underbrace{d(u,y) + d(y,z)}_{\geq d(u,z)} \\
&\geq w \wedge J(u) + d(u,z) \\
&= w(u') + J(u') + d(u',u) + d(u,z) \quad \text{für ein } u' \in S \\
&\geq w(u') + J(u') + d(u',z) \\
&\underset{\text{Def.}}{\geq} w \wedge J(z)
\end{aligned}$$

Also gilt

$$w \wedge J(z) = w \wedge J(u) + d(u,z)$$

Also gilt

$$K \setminus \{x\} \text{ stützt } w \wedge J.$$

Dies ist ein Widerspruch dazu, dass K eine Stützmenge von $w \wedge J$ ist.

■

Eine auf Arbeitsfunktionen basierende Strategie sieht wie folgt aus:

gegeben: $x \in S$, aktuelle Arbeitsfunktion w und eine neue Aufgabe $J \in T$.

Abbildung $A : S \times \Omega \times T \to S$, wobei

- $\Omega = \{ w \mid w$ erreichbare Arbeitsfunktion $\}$
- $A(x, w, J) = y$ ist derjenige Zustand, in den das System geht um J zu erledigen.

D.h., die Abbildung A definiert einen auf Arbeitsfunktionen basierenden Online-Algorithmus.

Ziel:
Beweis, dass für ein beliebiges MTS ein optimaler auf Arbeitsfunktionen basierender Online-Algorithmus existiert.

Satz 4.1
Sei ALG ein beliebiger Online-Algorithmus für ein MTS (M, T), der c-competitive mit Startfunktion $\alpha : S \to \mathbb{R}^+$ ist. Dann existiert ein auf Arbeitsfunktionen basierender

Online-Algorithmus ALG', der auch c-competitive mit Startfunktion α ist.

Beweis:

Ziel:
Konstruktion der Abbildung ALG'. D.h., Definition der Werte $ALG'(x, w, \mathfrak{J})$.

Für alle $x_0 \in S$ und alle $\bar{\mathfrak{J}} \in T^*$ definiere

$$L(x_0, \bar{\mathfrak{J}}) := cost_{ALG}(x_0, \bar{\mathfrak{J}}) - c \cdot opt(x_0, \bar{\mathfrak{J}})$$

Da ALG c-competitive mit Startfunktion α ist, folgt aus der Definition

$$L(x_0, \bar{\mathfrak{J}}) \leq \alpha(x_0).$$

Wir wollen nun erreichbare Paare (x, w) charakterisieren.

Sei $\forall x \in S$ $\forall$ Arbeitsfunktionen w

$$F(x,w) := \{ (x_0, \bar{\mathfrak{J}}) \mid ALG(x_0, \bar{\mathfrak{J}}) = x \text{ und } \chi_{x_0} \wedge \bar{\mathfrak{J}} = w \}$$

Falls $F(x,w) \neq \emptyset$, dann heißt das Paar (x, w) ALG-erreichbar.

Für ALG-erreichbare (x, w) definieren wir

$$\phi(x,w) = \inf_{(x_0, \bar{\mathfrak{J}}) \in F(x,w)} \sup_{\bar{\sigma} \in T^*} \{ L(x_0, \bar{\mathfrak{J}}\bar{\sigma}) - L(x_0, \bar{\mathfrak{J}}) \}$$

$$\mathrm{cost}_{ALG'}(x_0, \bar{\jmath})$$

$$= \sum_{i=1}^{m} \left(d(x_{i-1}, x_i) + \jmath_i(x_i) \right)$$

$$= \sum_{i=1}^{m} \Big[d(x_{i-1}, x_i) + \jmath_i(x_i)$$

$$+ \left(\phi(x_i, \chi_{x_0} \wedge \jmath_1 \dots \jmath_i) - \phi(x_{i-1}, \chi_{x_0} \wedge \jmath_1 \dots \jmath_{i-1}) \right) \Big]$$

$$+ \phi(x_0, \chi_{x_0}) - \underbrace{\phi(x_m, \chi_{x_0} \wedge \bar{\jmath})}_{\geq 0}$$

$$\underset{\text{Beh.}}{\leq} \sum_{i=1}^{m} c \cdot \Big[\min_{z \in S} \chi_{x_0} \wedge \jmath_1 \wedge \dots \wedge \jmath_i(z)$$

$$- \min_{z \in S} \chi_{x_0} \wedge \jmath_1 \wedge \dots \wedge \jmath_{i-1}(z) \Big]$$

$$+ \phi(x_0, \chi_{x_0})$$

$$\leq c \cdot \min_{z \in S} \chi_{x_0} \wedge \bar{\jmath}(z) \quad + \phi(x_0, \chi_{x_0})$$

$$\leq c \cdot \mathrm{opt}(x_0, \bar{\jmath}) \quad + \alpha(x_0)$$

Es verbleibt noch der Beweis der Behauptung.

Bew. d. Beh.:

Wähle $(x_0, \bar{\jmath}) \in F(x, w)$ mit

$$\phi(x, w) = \sup_{\bar{\sigma} \in T^*} \left\{ L(x_0, \bar{\jmath}\bar{\sigma}) - L(x_0, \bar{\jmath}) \right\}$$

Dann gilt:

Beobachtung:

- $\phi(x,w) \geq 0$, da $\bar{\sigma} = \varepsilon$ gewählt werden kann.
- $\phi(x_0, \chi_{x_0}) \leq \alpha(x_0)$, da $\bar{\sigma} = \varepsilon$ gewählt werden kann.

Beh.:

Falls (x,w) ALG-erreichbar, dann existiert für alle $\sigma \in T$ ein y in S, so dass

$$d(x,y) + \sigma(y) + \phi(y, w \wedge \sigma) - \phi(x,w)$$
$$\leq C \cdot [\min_z w \wedge \sigma(z) - \min_z w(z)]$$

$\square$

Bevor wir die Behauptung beweisen, führen wir den Beweis des Satzes zu Ende.

Sei (x,w) ALG-erreichbar und $\sigma \in T$. Wir definieren dann

$$ALG'(x,w,\sigma) := y,$$

wobei $y \in S$ mit $d(x,y) + \sigma(y) + \phi(y, w \wedge \sigma)$ minimal.

Für $x_0 \in S$ und $\bar{\sigma} = \sigma_1 \sigma_2 \ldots \sigma_m$ erhalten wir dann

$$(*)\quad \Phi(x,w) \geq \sup_{\bar\sigma' \in T^*} \{ L(x_0, \bar\jmath\,\jmath\,\bar\sigma') - L(x_0, \bar\jmath) \},$$

da die Festlegung der ersten Aufgabe in $\bar\sigma$ die Auswahl höchstens einschränkt.

Sei

$$y = ALG(x_0, \bar\jmath\,\jmath).$$

Konstruktion $\Rightarrow$

$$(**)\quad \phi(y, w \wedge \jmath) \leq \sup_{\bar\sigma \in T^*} \{ L(x_0, \bar\jmath\,\jmath\,\bar\sigma) - L(x_0, \bar\jmath\,\jmath) \}$$

$\Rightarrow$

$$d(x,y) + \jmath(y) + \phi(y, w \wedge \jmath) - \phi(x,w)$$

$$\underset{(*)\text{ u. }(**)}{\leq} d(x,y) + \jmath(y) + \sup_{\bar\sigma \in T^*} \{ L(x_0, \bar\jmath\,\jmath\,\bar\sigma) - L(x_0, \bar\jmath\,\jmath) \}$$

$$- \sup_{\bar\sigma \in T^*} \{ L(x_0, \bar\jmath\,\jmath\,\bar\sigma) - L(x_0, \bar\jmath) \}$$

$$= d(x,y) + \jmath(y) - \big(L(x_0, \bar\jmath\,\jmath) - L(x_0, \bar\jmath) \big)$$

$$= d(x,y) + \jmath(y)$$

$$- \Big[\big(cost_{ALG}(x_0, \bar\jmath\,\jmath) - c \cdot opt(x_0, \bar\jmath\,\jmath) \big)$$

$$- \big(cost_{ALG}(x_0, \bar\jmath) - c \cdot opt(x_0, \bar\jmath) \big) \Big]$$

$$\underset{\text{Wahl von } y}{=} c \cdot \big(\underbrace{opt(x_0, \bar\jmath\,\jmath)}_{= \min_z w \wedge \jmath(z)} - \underbrace{opt(x_0, \bar\jmath)}_{= \min_z w(z)} \big)$$

4.1 Durchschnittseigenschaft von metrischen Dienstsystemen

Sei (M, R) ein metrisches Dienstsystem. Ein Online-Algorithmus für (M, R) heißt träge (lazy), falls er niemals seinen Zustand ändert, wenn der aktuelle Zustand Element der Nachfragemenge ist.

Lemma 4.2

Sei ALG ein Online-Algorithmus für ein MSS (M, R). Dann existiert ein träger Online-Algorithmus ALG' für (M, R), dessen Kosten niemals größer als die von ALG sind.

Beweis:

Wir beweisen das Lemma, indem wir mit Hilfe von ALG den trägen Online-Algorithmus ALG' konstruieren. Sei

$$\mathrm{ALG}'(x_0, \varepsilon) = x_0$$

$$\mathrm{ALG}'(x_0, \bar{\varrho}\varrho) = \begin{cases} \mathrm{ALG}'(x_0, \bar{\varrho}) & \text{falls } \mathrm{ALG}'(x_0, \bar{\varrho}) \in \varrho \\ \mathrm{ALG}(x_0, \bar{\varrho}\varrho) & \text{sonst.} \end{cases}$$

ALG' arbeitet wie ALG mit der Ausnahme, dass ALG' den Zustand nicht ändert, wenn dieser Element der Anfrage ist. Beim nächsten Mal, wenn dies der Fall ist, geht ALG' in denselben Zustand wie ALG. Die Behauptung folgt nun aus der Δ-Ungleichung.

$\blacksquare$

Folgender Satz ist gerade die Durchschnittseigenschaft von metrischen Dienstsystemen.

Satz 4.2

Sei (M, R) ein MSS. Sei

$$R' = \{ \varrho_1' \cap \varrho_2' \mid \varrho_1', \varrho_2' \in R \text{ und } \varrho_1' \cap \varrho_2' \neq \emptyset \}.$$

Dann hat (M, R) genau dann einen c-competitiven Online-Algorithmus, wenn (M, R') solchen besitzt.

Beweis:

Folgende Behauptung impliziert den Satz:

Beh. 1:

Sei (M, R) ein MSS. Seien $\varrho_1', \varrho_2' \in R$ mit $\emptyset \neq \varrho_1' \cap \varrho_2' \notin R$. Sei $R' = R \cup \{\varrho_1' \cap \varrho_2'\}$. Dann existiert genau dann ein c-competitiver Online-Algorithmus für (M, R), wenn es einen solchen für (M, R') gibt.

Bew. d. Beh.:

„$\Leftarrow$"

trivial

„$\Rightarrow$"

Sei ALG ein träger Online-Algorithmus für (M,R).

<u>Ziel</u>:

Konstruktion eines Online-Algorithmus ALG' für (M,R'), dessen competitives Verhältnis nicht schlechter ist, als das von ALG für (M,R).

<u>Idee</u>:

Gegeben eine beliebige Anfragefolge $\bar{\varrho}$ für (M,R') definieren wir gleichzeitig das Verhalten von ALG' auf $\bar{\varrho}$ und eine Anfragefolge $\bar{\tau}$ für (M,R) und zeigen, dass das competitive Verhältnis von ALG' bzgl. $\bar{\varrho}$ nicht schlechter ist, als das von ALG bzgl. $\bar{\tau}$.

Sei $\bar{\varrho} = \varrho_1 \varrho_2 \cdots \varrho_m$. Für $1 \le i \le m$ definieren wir in Abhängigkeit von ϱ_i eine Anfragefolge $\bar{\tau}_i$ für (M,R) und das Verhalten von ALG' bzgl. ϱ_i; d.h., den Zustand x_i', in dem ALG' die Anfrage ϱ_i erledigt.

Wir unterscheiden zwei Fälle:

<u>1. Fall</u>: $\varrho_i \in R$

Wir definieren dann:

- $\bar{\sigma}_i := \varrho_i$

- Sei x_i der Zustand, in dem ALG die Anfrage ϱ_i erledigt. Setze

  $x_i' := x_i$.

<u>2. Fall</u> $\varrho_i \notin R \Rightarrow \varrho_i = \varrho_1' \cap \varrho_2'$

Wir definieren dann

- $\bar{\sigma}_i := (\varrho_1' \varrho_2')^N$, wobei

  $N \geq \frac{D}{d_{min}} + 1$ ganzzahlig mit

  D ist der Diameter bzgl. $M = (S, d)$ und d_{min} ist die kleinste Distanz in M.

- Sei x_i der Zustand, in dem ALG sich nach Abarbeitung der Anfragefolge $\bar{\sigma}_i$ befindet.

  Falls $x_i \in \varrho_1' \cap \varrho_2'$, dann setze

  $x_i' := x_i$.

  Andernfalls setze

  $x_i' \in \varrho_1' \cap \varrho_2'$ beliebig.

$\square$

Als nächstes werden wir die Kosten der beiden Algorithmen analysieren.

Seien

$cost_i'$ die Kosten von ALG' nach $\rho_1 \rho_2 \dots \rho_i$

$cost_i$ die Kosten von ALG nach $\bar{\jmath}_1 \bar{\jmath}_2 \dots \bar{\jmath}_i$.

Aus folgender Behauptung folgt direkt die Beh. 1:

Beh. 2:

i) $opt(x_0, \bar{\jmath}) \leq opt(x_0, \bar{\rho})$

ii) $cost_i' \leq cost_i - d(x_i, x_i') \quad 1 \leq i \leq m$

Bew. d. Beh. 2:

i) folgt direkt, da eine beliebige träge Lösung für $\bar{\rho}$ auch eine träge Lösung für $\bar{\jmath}$ ist.

Wir beweisen ii) durch Induktion über i.

Für $i = 0$, d.h., keine Anfrage ist bisher erfolgt, gilt die Behauptung offensichtlich.

Sei $i > 0$.

Annahme:
Die Behauptung gilt für $i-1$.

$i-1 \rightsquigarrow i$:

Wir unterscheiden zwei Fälle:

1. Fall: $x_i = x_i'$

Dann gilt:

$$\begin{aligned} cost_i' &= cost_{i-1}' + d(x_{i-1}', x_i') \\ &\leq cost_{i-1} - d(x_{i-1}, x_{i-1}') + d(x_{i-1}', x_i') \\ &\leq cost_{i-1} + d(x_{i-1}, x_i') \\ &= cost_i \\ &= cost_i - \underbrace{d(x_i, x_i')}_{=0} . \end{aligned}$$

2. Fall: $x_i \neq x_i'$

Dann gilt: $\varrho_i = \varrho_1' \cap \varrho_2'$.

Seien $v_1, v_2, \ldots, v_{2N} = x_i$ diejenige Zustände, die ALG zur Abarbeitung von $\bar{\sigma}_i = (\varrho_1' \varrho_2')^N$ in dieser Reihenfolge annimmt.

Falls

$$v_\ell \in \varrho_i \quad \text{für ein } \ell \in \{1, 2, \ldots, 2N\},$$

dann impliziert die Trägheit von ALG

$$v_\ell = v_{\ell+1} = \ldots = v_{2N} = x_i .$$

Dies ist ein Widerspruch zu $x_i \neq x_i'$.

Also gilt $v_\ell \notin \varrho_i \ \forall \ \ell \in \{1, 2, \ldots, 2N\}$.

$\Rightarrow$

$$v_\ell \in \begin{cases} \rho_1' \setminus \rho_2' & \text{falls } \ell \text{ ungerade} \\ \rho_2' \setminus \rho_1' & \text{falls } \ell \text{ gerade} \end{cases}$$

$\Rightarrow$

ALG bezahlt für $\bar{J}_i$

$$\geq 2N \cdot d_{min} > 2D.$$

Falls $\rho_{i-1} = \rho_i$, dann bezahlt ALG' für ρ_i nichts und die Behauptung folgt offensichtlich aus der Induktionsvoraussetzung.

Somit verbleibt noch der Fall

$$\rho_{i-1} \neq \rho_i = \rho_1' \cap \rho_2'.$$

Konstruktion $\Rightarrow$

$$x_{i-1}' = x_{i-1}.$$

$\Rightarrow$

$$\begin{aligned} cost_i' &= cost_{i-1}' + d(x_{i-1}', x_i') \\ &\leq cost_{i-1}' + D \\ &\leq cost_{i-1} - \underbrace{d(x_{i-1}, x_{i-1}')}_{=0} + D \\ &\leq cost_i - D \\ &\leq cost_i - d(x_i, x_i') \end{aligned}$$

4.2 Der Arbeitsfunktionenalgorithmus

Satz .1 zeigt, dass für ein beliebiges MTS stets ein auf Arbeitsfunktionen basierender optimaler Online-Algorithmus existiert. Jedoch ist der Beweis des Satzes 3.1 nicht konstruktiv, so dass sich direkt folgende Frage aufdrängt:

Frage:

Ist es möglich, für ein beliebiges MTS einen auf Arbeitsfunktionen basierenden optimalen Online-Algorithmus explizit anzugeben?

Idee:

Verfahre lokal am kostengünstigsten.

Folgender Algorithmus implementiert obige Idee:

Algorithmus WFA

Online-Regel:

Annahme:

WFA ist im Zustand s, w ist die aktuelle Arbeitsfunktion und J die zu erledigende Aufgabe.

neuer Zustand:

$WFA(s, w, J) = x'$, wobei

i) $x' \in$ Stützmenge von $w \wedge J$ und

ii) $d(s, x') + w \wedge J(x') = \min_{x \in S} \{ d(s,x) + w \wedge J(x) \}$.

Falls mehrere solche Zustände existieren, dann kann unter diesen ein beliebiger Zustand gewählt werden.

Bemerkung:

Lemma .1 $\Rightarrow$

Für das gewählte x' gilt:

$$w \wedge J(x') = w(x') + J(x').$$

Frage:

Ist WFA wohldefiniert? D.h., gibt es stets ein Zustand x' in der Stützmenge von $w \wedge J$ mit $d(s,x') + w \wedge J(x') = \min_{x \in S} \{ d(s,x) + w \wedge J(x) \}$?

Folgendes Lemma bejaht diese Frage.

Lemma 4.3

Für alle s, w, J existiert in der Stützmenge von $w \wedge J$ ein Zustand x' mit

$$d(s,x') + w \wedge J(x') = \min_{x \in S} \{ d(s,x) + w \wedge J(x) \}.$$

Beweis:

Betrachte $y \in S$ mit

$$d(s,y) + w \wedge J(y) = \min_{x \in S} \{ d(s,x) + w \wedge J(x) \}.$$

Wähle x' aus der Stützmenge von $w \wedge \mathcal{T}$ mit

$$w \wedge \mathcal{T}(y) = w \wedge \mathcal{T}(x') + d(x', y)$$

Dann gilt:

$$d(s, x') + w \wedge \mathcal{T}(x') = d(s, x') - d(x', y) + w \wedge \mathcal{T}(y)$$

$$= d(s, x') \underbrace{- d(y, x') - d(s, y)}_{\leq\ -d(s, x')} + d(s, y) + w \wedge \mathcal{T}(y)$$

$$\leq d(s, y) + w \wedge \mathcal{T}(y).$$

$\Rightarrow$

$$d(s, x') + w \wedge \mathcal{T}(x') = \min_{x \in S} \{ d(s, x) + w \wedge \mathcal{T}(x) \}.$$

■

<u>Ziel:</u> Analyse von WFA.

Seien

s_0	Startzustand
$\overline{\mathcal{T}} = \mathcal{T}_1 \mathcal{T}_2 \ldots \mathcal{T}_m$	Aufgabenfolge
$\overline{s} = s_1 s_2 \ldots s_m$	Zustände, in denen WFA die Aufgaben erledigt.

$w_0 = \chi_{s_0}$

$w_i = w_0 \wedge \mathcal{T}_1 \mathcal{T}_2 \ldots \mathcal{T}_i$.

Dann gilt:

$$(*) \quad cost_{WFA}(s_0, \overline{\mathcal{T}}) = \sum_{i=1}^{m} d(s_{i-1}, s_i) + \mathcal{T}_i(s_i).$$

Ferner gilt:

$$w_i(s_i) + d(s_{i-1}, s_i) = \min_{x \in S} \{ w_i(x) + d(s_{i-1}, x) \}$$

$$\leq w_i(s_{i-1}) + \underbrace{d(s_{i-1}, s_{i-1})}_{= 0}$$

$$= w_i(s_{i-1})$$

$$\Longleftrightarrow \quad d(s_{i-1}, s_i) \leq w_i(s_{i-1}) - w_i(s_i).$$

In (*) eingesetzt erhalten wir:

$$(**) \quad cost_{WFA}(s_0, \bar{\sigma}) \leq \sum_{i=1}^{m} w_i(s_{i-1}) - w_i(s_i) + \sigma_i(s_i)$$

Korollar .1 $\Rightarrow$

$$w_i(s_i) = w_{i-1}(s_i) + \sigma_i(s_i)$$

$$\Longleftrightarrow \quad -w_{i-1}(s_i) = -w_i(s_i) + \sigma_i(s_i)$$

In (**) eingesetzt erhalten wir:

$$cost_{WFA}(s_0, \bar{\sigma}) \leq \sum_{i=1}^{m} w_i(s_{i-1}) - w_{i-1}(s_i)$$

$$= \sum_{i=1}^{m} \left[w_i(s_{i-1}) - w_{i-1}(s_{i-1}) + w_i(s_i) - w_{i-1}(s_i) \right]$$

$$+ w_0(s_0) - w_m(s_m).$$

Seien

$$\nabla(w, J) := \max_{x \in S} \{ w \wedge J(x) - w(x) \}$$

und für $\overline{J} = J_1 J_2 \cdots J_m$

$$\nabla(w, \overline{J}) := \sum_{i=1}^{m} \max_{x \in S} \{ w \wedge J_1 J_2 \cdots J_i(x) - w \wedge J_1 \cdots J_{i-1}(x) \}$$

Dann gilt

$$cost_{WFA}(s_0, \overline{J})$$

$$\leq 2 \nabla(\chi_{s_0}, \overline{J}) - opt(s_0, \overline{J}).$$

Also haben wir folgendes Lemma bewiesen:

<u>Lemma 4.4</u>

Seien s_0 der Startzustand eines MTS und $\overline{J}$ eine Aufgabenfolge. Dann gilt:

$$cost_{WFA}(s_0, \overline{J}) \leq 2 \nabla(\chi_{s_0}, \overline{J}) - opt(s_0, \overline{J}).$$

<u>Ziel:</u>

Herleitung einer oberen Schranke für $\nabla(\chi_{x_0}, \overline{J})$.

Betrachte hierzu folgende Potentialfunktion:

$$\phi(w) = - \sum_x w(x).$$

$\Rightarrow$

$$-ND \leq \phi(\chi_{s_0}) \leq 0 \quad \text{und}$$

$$\Phi(\underline{w}) \geq -N\left(\min\{w(x) \mid x \in S\} + D\right),$$

wobei N die Anzahl der Zustände in S ist.

Ferner gilt:

$$\nabla(w, \mathfrak{I}) + \Phi(w \wedge \mathfrak{I}) - \Phi(\underline{w})$$

$$= \max_{x \in S}\{w \wedge \mathfrak{I}(x) - w(x)\} - \sum_{y \in S}[w \wedge \mathfrak{I}(y) - w(y)]$$

$$\leq 0.$$

Also gilt:

$$\nabla(\chi_{s_0}, \bar{\mathfrak{I}}) + \Phi(\chi_{s_0} \wedge \bar{\mathfrak{I}}) - \Phi(\chi_{s_0}) \leq 0$$

$$\Longleftrightarrow \quad \nabla(\chi_{s_0}, \bar{\mathfrak{I}}) \leq \underset{\leq 0}{\Phi(\chi_{s_0})} - \underbrace{\Phi(\chi_{s_0} \wedge \bar{\mathfrak{I}})}_{\geq -N(\min_{x \in S}\{\chi_{s_0} \wedge \bar{\mathfrak{I}}(x)\} + D)}$$

$$\leq N\left(\min_{x \in S}\{\chi_{s_0} \wedge \bar{\mathfrak{I}}(x)\} + D\right)$$

$$= N \cdot \min_{x \in S}\{\chi_{s_0} \wedge \bar{\mathfrak{I}}(x)\} + N \cdot D$$

$$= N \cdot \operatorname{opt}(s_0, \bar{\mathfrak{I}}) + ND$$

Konstante

Zusammen mit Lemma 4.4 haben wir somit folgenden Satz bewiesen:

Satz 4.3

Für jedes MTS mit N Zuständen ist WFA $(2N-1)$ - competitive.

Übung:

Zeigen Sie, dass WFA für jedes MSS mit N Zuständen $(N-1)$ - competitive ist.

Frage:

Wie gut ist obige obere Schranke?

Ziel:

Beweis einer unteren Schranke.

Hierzu sei ALG ein deterministischer Online-Algorithmus für ein MTS mit metrischen Raum $M = (S, d)$. Sei $\varepsilon > 0$ eine positive reele Zahl. Dann bezeichnet ADV_ε denjenigen Gegenspieler, der folgende Aufgabenfolge $\bar{J} = J_1 J_2 \ldots$ konstruiert:

$\forall s \in S$

$$J_i(s) = \begin{cases} \varepsilon & \text{falls } ALG(J_1 J_2 \ldots J_{i-1}) = s \\ 0 & \text{sonst} \end{cases}$$

D.h., ALG bezahlt zur Erledigung einer Aufgabe ε, falls ALG im vorangegangenen Zu=

stand bleibt und 0 sonst.

Satz 4.4

Sei ALG ein beliebiger deterministischer Online-Algorithmus für ein MTS mit metrischen Raum $M = (S, d)$, wobei $|S| = N$. Sei $\varepsilon > 0$ eine positive reele Zahl. Sei $\overline{J}_m = J_1 J_2 \dots J_m$ die Aufgabenfolge der Länge m, die ADV_ε konstruiert. Dann gilt

$$\limsup_{m \to \infty} \frac{ALG(\overline{J}_m)}{OPT(\overline{J}_m)} \geq \frac{2N-1}{1+2\varepsilon}$$

Beweis:

Annahme:
ALG ändert seinen Zustand k-mal ($k \leq m$).

Seien $s_0, s_1, \dots, s_k$ die Zustände von ALG in dieser Ordnung. Dann gilt:

Kosten von ALG bzgl. $\overline{J}_m$:

- $(m-k) \cdot \varepsilon$ plus
- $\sum_{i=1}^{k} d(s_{i-1}, s_i)$

Ziel:
Herleitung einer oberen Schranke für $OPT(\overline{J}_m)$.

Idee:
- Betrachte eine endliche Menge B von Offline-Algorithmen.

- Berechne $\sum_{\sigma \in B} cost(\sigma(\bar{J}_m)) = B(\bar{J}_m)$

$$\Rightarrow \quad OPT(\bar{J}_m) \leq \frac{B(\bar{J}_m)}{|B|}.$$

<u>Durchführung</u>:

Seien

q der Zustand von ALG während $[i, i+1)$

$q_1, q_2, \ldots, q_{N-1}$ andere Zustände in S.

B enthält genau $2N-1$ Offline-Algorithmen, welche für $i = 1, 2, \ldots, m-1$ folgende Invariante erfüllen:

<u>Invariante</u>:

Während $[i, i+1)$ sind exakt zwei Offline-Algorithmen im Zustand q_j für $j = 1, 2, \ldots, N-1$.
Genau ein Algorithmus ist im Zustand q.

$\Rightarrow$

B enthält $2N-1$ Algorithmen.

<u>Realisierung</u>:

ALG ändert seinen Zustand von q nach q'

$\Rightarrow$

Genau einer der beiden Algorithmen in Zustand q' ändert seinen Zustand nach q.

Berechnung von $B(\bar{J}_m)$:

a) Jedesmal, wenn ALG seinen Zustand ändert, führt exakt ein Offline-Algorithmus den symmetrischen Übergang durch.

b) Genau ein Offline-Algorithmus in B ist stets im selben Zustand wie ALG.

b) $\Rightarrow$

Gesamtkosten, wenn ALG ε bezahlt $\qquad (m-k)\varepsilon$

Wenn ALG nicht ε bezahlt, dann bezahlen genau zwei Offline-Algorithmen ε. Also betragen die Gesamtkosten hierfür

$$2k \cdot \varepsilon$$

Insgesamt erhalten wir:

$$B(\bar{J}_m) = \underbrace{(m-k)\varepsilon + \sum_{i=1}^{k} d(s_{i-1}, s_i)}_{ALG(\bar{J}_m)} + 2k\varepsilon$$

$$\Rightarrow \quad OPT(\bar{J}_m) \leq \frac{2k\varepsilon + ALG(\bar{J}_m)}{2N-1}$$

Wegen

$$ALG(\bar{J}_m) \geq \sum_{i=1}^{k} d(s_{i-1}, s_i)$$

$$\geq k \cdot d_{min}$$

skalieren, so dass $d_{min} \geq 1$.

$$\geq k$$

folgt

$$2k\varepsilon \leq 2\varepsilon \cdot ALG(\bar{J}_m).$$

$\Longrightarrow$

$$(2N-1)\, OPT(\bar{J}_m) \leq 2\varepsilon\, ALG(\bar{J}_m) + ALG(\bar{J}_m)$$
$$= (1+2\varepsilon)\, ALG(\bar{J}_m)$$

$\Longrightarrow$

$$\limsup_{m\to\infty} \frac{ALG(\bar{J}_m)}{OPT(\bar{J}_m)} \geq \frac{2N-1}{1+2\varepsilon}.$$

■