Instandhaltungsstrategien für modulare monotone Multizustandssysteme

Doktoranden-Workshop Nordost 2014
Technische Universität Clausthal

Dipl.-Math. Michael Krause, M. Sc.

Abteilung für Betriebswirtschaftslehre, insb. Produktion und Logistik
Institut für Wirtschaftswissenschaft, TU Clausthal

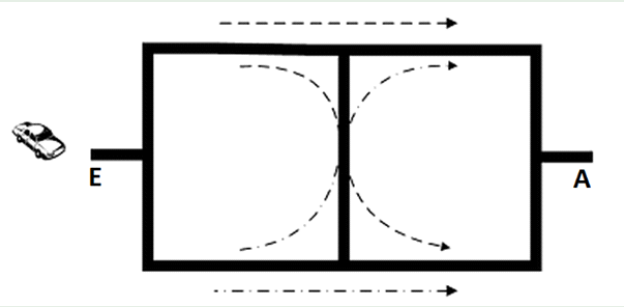
24. Mai 2014

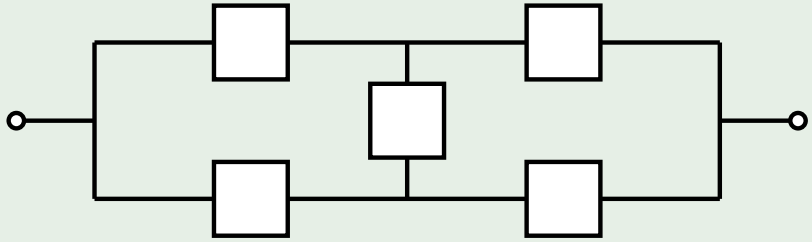
Agenda

Das Problem

- Gegeben:
  - System besteht aus **mehreren Komponenten**
  - Komponenten bzw. System haben **mehrere Zustände**
  - **Strukturfunktion**
  - (Stochastische) **Leistungsabnahmeprozesse** für Komponenten
  - **Instandhaltungskosten** für Komponenten
  - (Opportunitäts-)**Kostenrate** bei Systemleistungsabfall
- Gesucht: **Instandhaltungsstrategie** für jede einzelne Komponente, sodass Opportunitätskosten unter Einhaltung eines verfügbaren Budgets minimiert werden

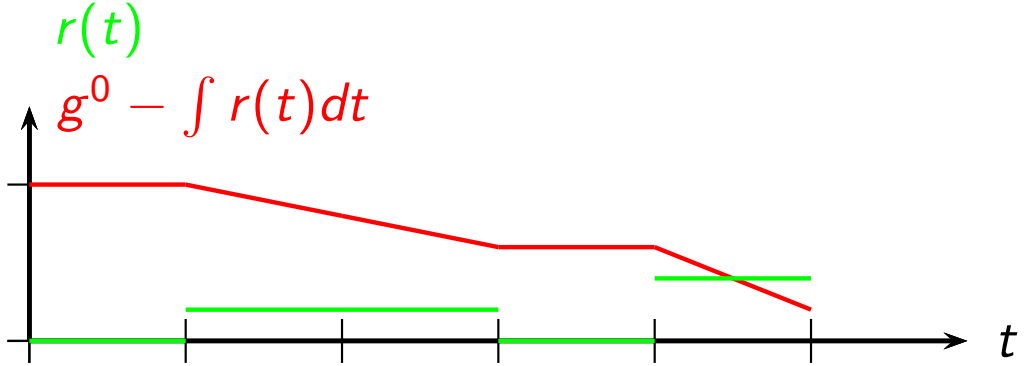
Beispiel





Komponenten

- Komponenten unterliegen **stochastischer beobachtbarer** Leistungsabnahme (**Verschleiß**)
- Für Zustand $s_j \in S_j$ der Komponente j sei $g_j(s_j)$ die zugehörige **Leistung**
- $\{R_j(t)|t \geq 0\}$ ist stochastischer **Verschleißprozess** von Komponente j
- Verteilungsfunktionen der Verschleißprozesse sind bekannt



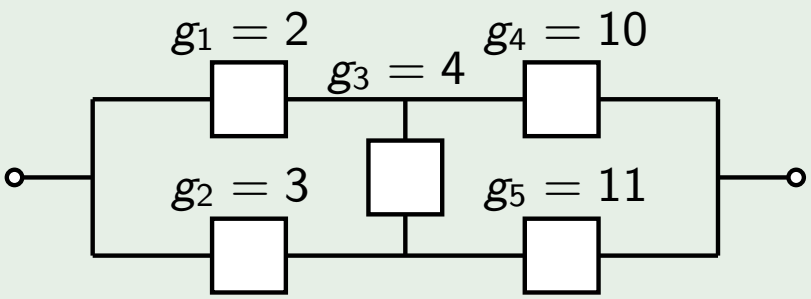
Strukturfunktion

- $S^n = S_1 \times \ldots \times S_n$: Menge möglicher Kombinationen von Zuständen aller n Komponenten; $\mathbf{s} \in S^n$: **Systemzustand**
- Abbildung der Komponentenleistung auf bewerteten Systemzustand (Systemleistung)

$$f(g_1(s_1), \ldots, g_n(s_n)) : \mathbf{g}(S^n) \to \mathbb{R}$$

die **Strukturfunktion** des Systems

Beispiel



Komponentenleistung:
aktuelle Belastbarkeit einer Straße in t Gewicht
Systemleistung:
maximal zulässiges Gewicht eines LKWs

$$f(g(s)) = \max_i \min_j \{\text{Belastbarkeit Straße } j \text{ über Weg } i\}$$
$$= \max\{\min\{2, 10\}, \min\{2, 4, 11\}, \min\{3, 11\}, \min\{3, 4, 10\}\} = 3$$

Begriffe

- Komponente j **relevant**, wenn es Zustände $\bar{s_{jk}}$ und $\bar{s_{jm}}$ mit $g_j(\bar{s_{jk}}) \neq g_j(\bar{s_{jm}})$ gibt, so dass

$$f(\ldots, g_j(\bar{s_{jk}}), \ldots) \neq f(\ldots, g_j(\bar{s_{jm}}), \ldots).$$

- System **monoton**, wenn gilt ...
  - alle Komponenten **relevant**
  - Strukturfunktion **komponentenweise monoton wachsend**, d. h. Instandhaltungsmaßnahme einer Komponente führt nicht zur Verschlechterung der Systemleistung

Beispiel

$f(g(s)) = \max\{\min\{2, 10\}, \min\{2, 4, 11\}, \min\{3, 11\}, \min\{3, 4, 10\}\} = 3$
$f(g(s)) = \max\{\min\{2, 10\}, \min\{2, 4, 11\}, \min\{4, 11\}, \min\{4, 4, 10\}\} = 4$

$s_{jt} = (g_{jt}, a_{jt})$, g_{jt} : Leistung von Komponente j zur Zeit t, a_{jt} : Alter von Komponente j zur Zeit t, x_{jt} : Investition in Komponente j zur Zeit t, B_0 : verfügbares Gesamtbudget, T : Planungshorizont

(Stochastisches) Dynamisches Programm (1/2)

$$\min_{x_{tj}} \quad \sum_{t=1}^{T} c \cdot (100\% - \mathbb{E}\{\min\{\hat{g}_{t+1,1}, \hat{g}_{t+1,2}\}\}) \tag{1}$$

$$\hat{g}_{t+1,j} = \hat{g}_{tj} - \hat{R}_{t+1,j}(\hat{g}_{tj}, a_{tj}) + \varphi(x_{tj}) \tag{2}$$

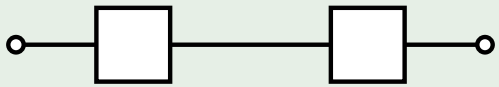
$$a_{t+1,j} = (1 + a_{tj}) \cdot \exp\left(-\frac{1}{c_j^a} x_{tj}\right) \qquad \forall t, j \tag{3}$$

$$B_{t+1} = B_t - \sum_j x_{tj} \qquad \forall t \tag{4}$$

- Außerdem: Initialisierung Anfangswerte für Komponentenleistung, -alter, Budget und Planungshorizont; Nicht-Negativität

(Stochastisches) Dynamisches Programm (2/2)

Serielles 2-Komponentensystem mit Strukturfunktion $\min\{g_1(s_1), g_2(s_2)\}$

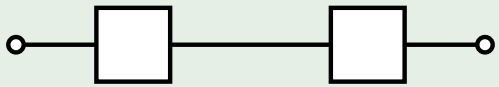


- Die Zielfunktion ist äquivalent zu

$$\max_{x_{tj}} \sum_{t=1}^{T} \mathbb{E}\{\min\{\hat{g}_{t+1,1}, \hat{g}_{t+1,2}\}\},$$

(Stochastisches) Dynamisches Programm (2/2)

Serielles 2-Komponentensystem mit Strukturfunktion $\min\{g_1(s_1), g_2(s_2)\}$



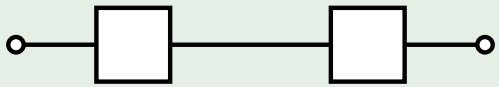
- Die Zielfunktion ist äquivalent zu

$$\max_{x_{tj}} \sum_{t=1}^{T} \mathbb{E}\{\min\{\hat{g}_{t+1,1}, \hat{g}_{t+1,2}\}\},$$

- $\hat{R}_{t+1,j}$ haben Erwartungswerte $\hat{g}_{t,j} \cdot \frac{a_{t,j}+1}{\frac{1}{c_j}T}$

(Stochastisches) Dynamisches Programm (2/2)

Serielles 2-Komponentensystem mit Strukturfunktion $\min\{g_1(s_1), g_2(s_2)\}$



- Die Zielfunktion ist äquivalent zu

$$\max_{x_{tj}} \sum_{t=1}^{T} \mathbb{E}\{\min\{\hat{g}_{t+1,1}, \hat{g}_{t+1,2}\}\},$$

- $\hat{R}_{t+1,j}$ haben Erwartungswerte $\hat{g}_{t,j} \cdot \frac{a_{t,j}+1}{\frac{1}{c_j}T}$
- Auf Stufe t des dynamischen Programms gilt:

$$\max_{x_{tj} \in X(S_t)} \underbrace{\mathbb{E}\{\min\{\hat{g}_{t+1,1}, \hat{g}_{t+1,2}\}\}}_{C_t(S_t, x_t)} + V_{t+1}(S_{t+1}(S_t, x_t, \hat{R}_{t+1}))$$

Bisherige Erkenntnisse

- Schon für zwei Perioden ist das zugehörige deterministische Problem nicht mehr analytisch lösbar
- Analytische Lösung nur für eine Stufe möglich
- Nur ein weiterer Schritt der Rückwärtsrechnung numerisch durchführbar
- Für mehr als zwei Perioden kann reine dynamische Programmierung nicht mehr verwendet werden

Herausforderungen

- Erhöhung der Anzahl der **Perioden**
- Bisher nur sehr einfache **Struktur** $\Rightarrow$ Übergang zu allgemeinen Serien-Parallel-Systemen mit höherer Dimension
- „Curse of Dimensionality“

Ausgewählte Literatur


Powell WB (2011)
Approximate Dynamic Programming - Solving the Curses of Dimensionality, 2nd edn
Wiley, Hoboken, NJ, USA


Bertsekas DP (2012)
Dynamic Programmin and Optimal Control - Vol II: Approximate Dynamic Programming
Athena Scientific, Nashua, NH, USA


Bertsekas DP, Tsitsiklis JN (1996)
Neuro-Dynamic Programming
Athena Scientific, Belmont, Mass., USA


Lisnianski A, Frenkel I, Ding Y (2010)
Multi-state System Reliability Analysis and Optimization for Engineers and Industrial Managers
Springer, London


Beichelt F (1993)
Zuverlässigkeits- und Instandhaltungstheorie
Teubner, Stuttgart

Ein einfacher ADP-Algorithmus (Quelle: i. A. a. Powell (2011), S. 128)

- Initialisierung
  - Initialisiere $\bar{V}_t^0(S_t)$ für alle möglichen Zustände S_t
  - Wähle einen Anfangszustand S_0^1
  - Setze $n := 1$

Ein einfacher ADP-Algorithmus (Quelle: i. A. a. Powell (2011), S. 128)

- Initialisierung
  - Initialisiere $\bar{V}_t^0(S_t)$ für alle möglichen Zustände S_t
  - Wähle einen Anfangszustand S_0^1
  - Setze $n := 1$
- **for** $t = 1, \ldots, T$ **do**
  - Generiere mehrere mögliche Realisationen von Zufallszahlen auf Stufe t: $\hat{\omega}$

Ein einfacher ADP-Algorithmus (Quelle: i. A. a. Powell (2011), S. 128)

- Initialisierung
  - Initialisiere $\bar{V}_t^0(S_t)$ für alle möglichen Zustände S_t
  - Wähle einen Anfangszustand S_0^1
  - Setze $n := 1$
- **for** $t = 1, \ldots, T$ **do**
  - Generiere mehrere mögliche Realisationen von Zufallszahlen auf Stufe t: $\hat{\omega}$
  - Löse:

$$\hat{v}_t^n = \max_{x_t \in \mathcal{X}_t^n} \left(C_t(S_t^n, x_t) + \sum_{\hat{\omega}} p_{t+1}(\hat{\omega}) \bar{V}_{t+1}^{n-1}(S_{t+1}) \right)$$

  Sei x_t^n die Lösung des Maximierungsproblems

Ein einfacher ADP-Algorithmus (Quelle: i. A. a. Powell (2011), S. 128)

- Initialisierung
  - Initialisiere $\bar{V}_t^0(S_t)$ für alle möglichen Zustände S_t
  - Wähle einen Anfangszustand S_0^1
  - Setze $n := 1$
- **for** $t = 1, \ldots, T$ **do**
  - Generiere mehrere mögliche Realisationen von Zufallszahlen auf Stufe t: $\hat{\omega}$
  - Löse:

$$\hat{v}_t^n = \max_{x_t \in \mathcal{X}_t^n} \left(C_t(S_t^n, x_t) + \sum_{\hat{\omega}} p_{t+1}(\hat{\omega}) \bar{V}_{t+1}^{n-1}(S_{t+1}) \right)$$

    Sei x_t^n die Lösung des Maximierungsproblems
  - Update $\bar{V}_t^{n-1}(S_t^n)$

Ein einfacher ADP-Algorithmus (Quelle: i. A. a. Powell (2011), S. 128)

- Initialisierung
  - Initialisiere $\bar{V}_t^0(S_t)$ für alle möglichen Zustände S_t
  - Wähle einen Anfangszustand S_0^1
  - Setze $n := 1$
- **for** $t = 1, \ldots, T$ **do**
  - Generiere mehrere mögliche Realisationen von Zufallszahlen auf Stufe t: $\hat{\omega}$
  - Löse:

$$\hat{v}_t^n = \max_{x_t \in \mathcal{X}_t^n} \left(C_t(S_t^n, x_t) + \sum_{\hat{\omega}} p_{t+1}(\hat{\omega}) \bar{V}_{t+1}^{n-1}(S_{t+1}) \right)$$

    Sei x_t^n die Lösung des Maximierungsproblems
  - Update $\bar{V}_t^{n-1}(S_t^n)$
  - Generiere eine mögliche Realisation von Zufallszahlen auf aktueller Stufe t: ω^n

Ein einfacher ADP-Algorithmus (Quelle: i. A. a. Powell (2011), S. 128)

- Initialisierung
  - Initialisiere $\bar{V}_t^0(S_t)$ für alle möglichen Zustände S_t
  - Wähle einen Anfangszustand S_0^1
  - Setze $n := 1$
- **for** $t = 1, \ldots, T$ **do**
  - Generiere mehrere mögliche Realisationen von Zufallszahlen auf Stufe t: $\hat{\omega}$
  - Löse:

$$\hat{v}_t^n = \max_{x_t \in \mathcal{X}_t^n} \left(C_t(S_t^n, x_t) + \sum_{\hat{\omega}} p_{t+1}(\hat{\omega}) \bar{V}_{t+1}^{n-1}(S_{t+1}) \right)$$

    Sei x_t^n die Lösung des Maximierungsproblems
  - Update $\bar{V}_t^{n-1}(S_t^n)$
  - Generiere eine mögliche Realisation von Zufallszahlen auf aktueller Stufe t: ω^n
  - Berechne Folgezustand $S_{t+1}^n(S_t^n, x_t^n, \omega^n)$
- $n := n + 1$, solange $n < N$

Update der Look-Up-Table

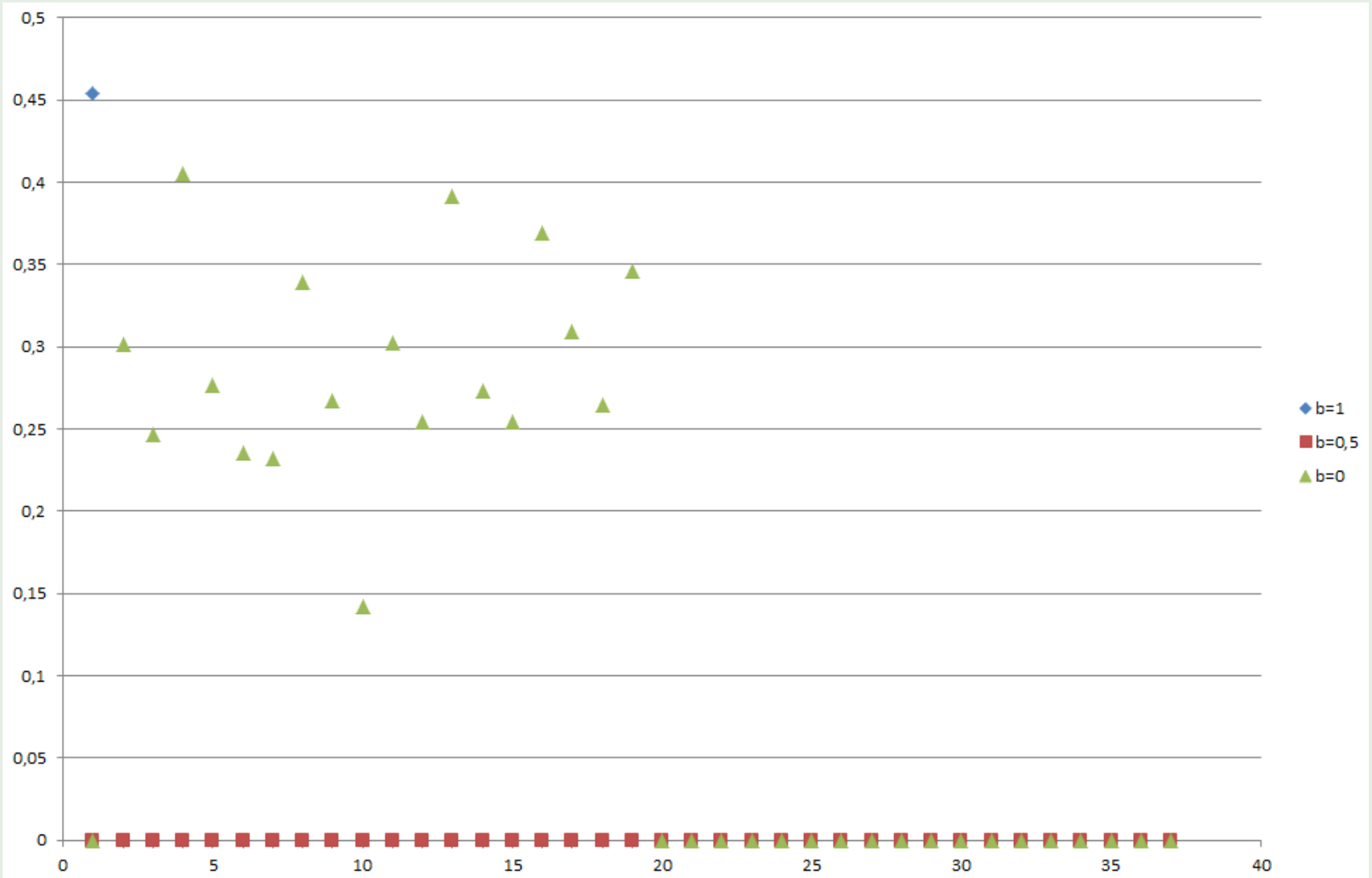
Update $\bar{V}_t^{n-1}(S_t^n)$:

$$\bar{V}_t^n(S_t) = \begin{cases} (1-\alpha_{n-1})\bar{V}_t^{n-1}(S_t^n) + \alpha_{n-1}\hat{v}_t^n, & \text{falls } S_t = S_t^n \\ \bar{V}_t^{n-1}(S_t), & \text{sonst} \end{cases}$$

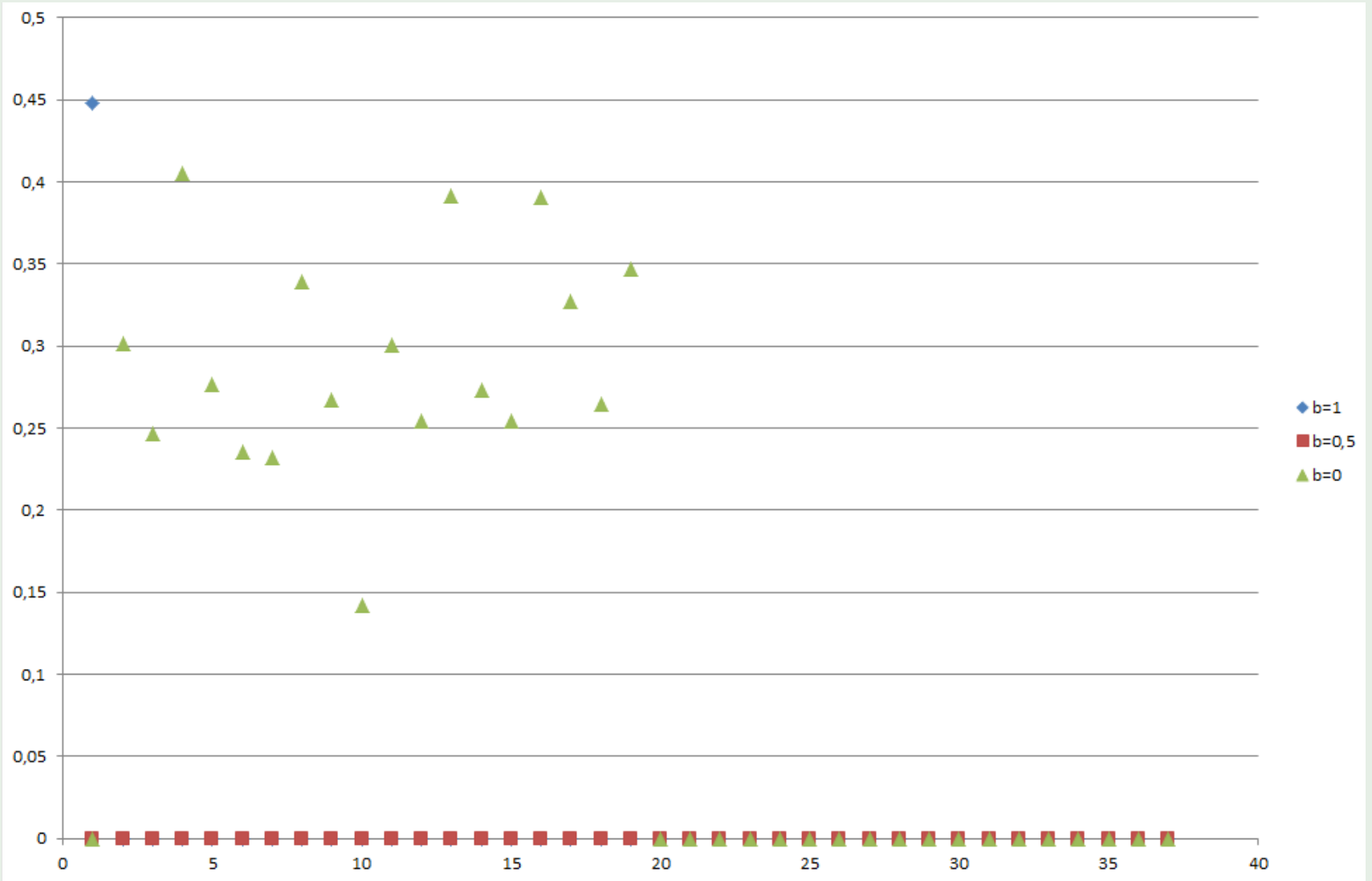
Nachteile der Diskretisierung

- In jedem Iterationsschritt wird pro Periode nur ein Eintrag der Look-Up-Table aktualisiert
- Anzahl der Iterationen im Vorhinein schwer abzuschätzen
- Wahl der Initialisierung der Look-Up-Table

Maximale Abweichung zur initialen Look-Up-Table (aggregiert, $n = 500.000$)



Maximale Abweichung zur initialen Look-Up-Table (aggregiert, $n = 1.000.000$)



Lösung

- Laufzeit für 1.000.000 Iterationen drei Stunden auf Standardrechner
- Vergleich mit deterministischer Lösung

	deterministisch, stetig		ADP, Diskretisierung = 0,1	
	$t = 1$	$t = 2$	$t = 1$	$t = 2$
x_{t1}	0,066	0	0,1	0
x_{t2}	0,759	0,175	0,8	0,1
ZF	1,559		1,448	

Neueste Erkenntnis

- Problem des „Curse of Dimensionality“ hat sich durch Diskretisierung nur verlagert
  - Feinere Diskretisierung lässt Größe der Look-Up-Table explodieren
  - Relativ starke Abhängigkeit von Initialisierung der Look-Up-Table

Neueste Erkenntnis

- Problem des „Curse of Dimensionality“ hat sich durch Diskretisierung nur verlagert
  - Feinere Diskretisierung lässt Größe der Look-Up-Table explodieren
  - Relativ starke Abhängigkeit von Initialisierung der Look-Up-Table

Nächste Schritte

- Abschätzung der Wertefunktion über neuronales Netz
- Abschätzung der Wertefunktion über parametrisierte Darstellung

Neueste Erkenntnis

- Problem des „Curse of Dimensionality“ hat sich durch Diskretisierung nur verlagert
  - Feinere Diskretisierung lässt Größe der Look-Up-Table explodieren
  - Relativ starke Abhängigkeit von Initialisierung der Look-Up-Table

Nächste Schritte

- Abschätzung der Wertefunktion über neuronales Netz
- Abschätzung der Wertefunktion über parametrisierte Darstellung

Ziel

- Entwicklung von und Vergleich mit heuristischen Instandhaltungsstrategien, z. B. auf der Grundlage von Importanzen

Vielen Dank für die Aufmerksamkeit.

Kontakt:

Michael Krause
Abteilung für Betriebswirtschaftslehre, insb. Produktion und Logistik
Institut für Wirtschaftwissenschaft
TU Clausthal
Julius-Albert-Straße 6, 38678 Clausthal-Zellerfeld
michael.krause@tu-clausthal.de
+49 (5323) 72-7617