Optimale Kontrolle von Markovschen Entscheidungsproblemen

Carolin Gruber und Torsten Templin

Humbodlt Universität zu Berlin

26. Januar 2011

Das Markovsche Entscheidungsmodell besteht aus Folgenden Komponenten:

- **Zustandsraum** S
- **Aktionenraum** $A = \cup_{s \in s} A_s$
- **Zeithorizont** T
- Familie von **Übergangsmatrizen** $(P(a))_{a \in A}$ wobei $p_{ij}(a) = p(j|s, a) \quad \forall i, j \in S$
- **Kostenfunktion** $c : S \times A \to \mathbb{R}$
- für $T < \infty$ ist $c_T(s)$ die Kosten zum Endzeitpunkt, an dem keine Entscheidung mehr getroffen wird
- **Entscheidungsregel** $u_t : \{S \times A\}^{t-1} \times S \to A$
- **Steuerung/Strategie** $\pi = (u_1, \ldots u_{T-1})$

Das Markovsche Entscheidungsmodell besteht aus Folgenden Komponenten:

- **Zustandsraum** S
- **Aktionenraum** $A = \cup_{s \in s} A_s$
- **Zeithorizont** T
- Familie von **Übergangsmatrizen** $(P(a))_{a \in A}$ wobei $p_{ij}(a) = p(j|s, a) \quad \forall i, j \in S$
- **Kostenfunktion** $c : S \times A \to \mathbb{R}$
- für $T < \infty$ ist $c_T(s)$ die Kosten zum Endzeitpunkt, an dem keine Entscheidung mehr getroffen wird
- **Entscheidungsregel** $u_t : \{S \times A\}^{t-1} \times S \to A$
- **Steuerung/Strategie** $\pi = (u_1, \ldots u_{T-1})$

Das Markovsche Entscheidungsmodell besteht aus Folgenden Komponenten:

- **Zustandsraum** S
- **Aktionenraum** $A = \cup_{s \in s} A_s$
- **Zeithorizont** T
- Familie von **Übergangsmatrizen** $(P(a))_{a \in A}$ wobei $p_{ij}(a) = p(j|s, a) \quad \forall i, j \in S$
- **Kostenfunktion** $c : S \times A \to \mathbb{R}$
- für $T < \infty$ ist $c_T(s)$ die Kosten zum Endzeitpunkt, an dem keine Entscheidung mehr getroffen wird
- **Entscheidungsregel** $u_t : \{S \times A\}^{t-1} \times S \to A$
- **Steuerung/Strategie** $\pi = (u_1, \ldots u_{T-1})$

Das Markovsche Entscheidungsmodell besteht aus Folgenden Komponenten:

- **Zustandsraum** S
- **Aktionenraum** $A = \cup_{s \in s} A_s$
- **Zeithorizont** T
- Familie von **Übergangsmatrizen** $(P(a))_{a \in A}$ wobei $p_{ij}(a) = p(j|s, a) \quad \forall i, j \in S$
- **Kostenfunktion** $c : S \times A \to \mathbb{R}$
- für $T < \infty$ ist $c_T(s)$ die Kosten zum Endzeitpunkt, an dem keine Entscheidung mehr getroffen wird
- **Entscheidungsregel** $u_t : \{S \times A\}^{t-1} \times S \to A$
- **Steuerung/Strategie** $\pi = (u_1, \ldots u_{T-1})$

Das Markovsche Entscheidungsmodell besteht aus Folgenden Komponenten:

- **Zustandsraum** S
- **Aktionenraum** $A = \cup_{s \in s} A_s$
- **Zeithorizont** T
- Familie von **Übergangsmatrizen** $(P(a))_{a \in A}$ wobei $p_{ij}(a) = p(j|s, a) \quad \forall i, j \in S$
- **Kostenfunktion** $c : S \times A \to \mathbb{R}$
- für $T < \infty$ ist $c_T(s)$ die Kosten zum Endzeitpunkt, an dem keine Entscheidung mehr getroffen wird
- **Entscheidungsregel** $u_t : \{S \times A\}^{t-1} \times S \to A$
- **Steuerung/Strategie** $\pi = (u_1, \ldots u_{T-1})$

Das Markovsche Entscheidungsmodell besteht aus Folgenden Komponenten:

- **Zustandsraum** S
- **Aktionenraum** $A = \cup_{s \in s} A_s$
- **Zeithorizont** T
- Familie von **Übergangsmatrizen** $(P(a))_{a \in A}$ wobei $p_{ij}(a) = p(j|s, a) \quad \forall i, j \in S$
- **Kostenfunktion** $c : S \times A \to \mathbb{R}$
- für $T < \infty$ ist $c_T(s)$ die Kosten zum Endzeitpunkt, an dem keine Entscheidung mehr getroffen wird
- **Entscheidungsregel** $u_t : \{S \times A\}^{t-1} \times S \to A$
- **Steuerung/Strategie** $\pi = (u_1, \ldots u_{T-1})$

Das Markovsche Entscheidungsmodell besteht aus Folgenden Komponenten:

- **Zustandsraum** S
- **Aktionenraum** $A = \cup_{s \in s} A_s$
- **Zeithorizont** T
- Familie von **Übergangsmatrizen** $(P(a))_{a \in A}$ wobei $p_{ij}(a) = p(j|s, a) \quad \forall i, j \in S$
- **Kostenfunktion** $c : S \times A \to \mathbb{R}$
- für $T < \infty$ ist $c_T(s)$ die Kosten zum Endzeitpunkt, an dem keine Entscheidung mehr getroffen wird
- **Entscheidungsregel** $u_t : \{S \times A\}^{t-1} \times S \to A$
- **Steuerung/Strategie** $\pi = (u_1, \ldots u_{T-1})$

Das Markovsche Entscheidungsmodell besteht aus Folgenden Komponenten:

- **Zustandsraum** S
- **Aktionenraum** $A = \cup_{s \in s} A_s$
- **Zeithorizont** T
- Familie von **Übergangsmatrizen** $(P(a))_{a \in A}$ wobei $p_{ij}(a) = p(j|s, a) \quad \forall i, j \in S$
- **Kostenfunktion** $c : S \times A \to \mathbb{R}$
- für $T < \infty$ ist $c_T(s)$ die Kosten zum Endzeitpunkt, an dem keine Entscheidung mehr getroffen wird
- **Entscheidungsregel** $u_t : \{S \times A\}^{t-1} \times S \to A$
- **Steuerung/Strategie** $\pi = (u_1, \ldots u_{T-1})$

Man unterscheidet zwischen

- **Markovschen Strategien** $\pi \in \Pi^M$ wobei $u_t : S \to A$ nur vom aktuellen Zustand s_t abhängt
- **Vergangenheitsabhängigen Strategien** $\pi \in \Pi^H$ wobei $u_t : \{S \times A\}^{t-1} \times S \to A$ von der gesamten Vergangenheit $h_t = (s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t)$ abhängt

Falls $u_t = u$ für alle t dann ist π eine **stationäre Strategie**.

Formulierung als Stochastischer Prozess

- $\Omega = \{S \times A\}^{T-1} \times S$ oder $\Omega = \{S \times A\}^{\infty}$
- $\mathfrak{A} = \mathfrak{B}(\Omega)$
- $X_t = s_t$, $Y_t = a_t$ und $Z_t = h_t$
- Anfangsverteilung $(\lambda_s)_{s \in S}$
- Strategie π induziert die Verteilung P^π auf $(\Omega, \mathfrak{B}(\Omega))$ durch
  - $P^\pi(\{X_0 = s\}) = \lambda_s$
  - $P^\pi(X_{t+1} = s \,|\, Z_t = h_t,\ Y_t = u_t(h_t)) = p(s \,|\, s_t,\ u_t(h_t))$

Formulierung als Stochastischer Prozess

- $\Omega = \{S \times A\}^{T-1} \times S$ oder $\Omega = \{S \times A\}^{\infty}$
- $\mathfrak{A} = \mathfrak{B}(\Omega)$
- $X_t = s_t$, $Y_t = a_t$ und $Z_t = h_t$
- Anfangsverteilung $(\lambda_s)_{s \in S}$
- Strategie π induziert die Verteilung P^π auf $(\Omega, \mathfrak{B}(\Omega))$ durch
  - $P^\pi(\{X_0 = s\}) = \lambda_s$
  - $P^\pi(X_{t+1} = s \,|\, Z_t = h_t,\ Y_t = u_t(h_t)) = p(s \,|\, s_t,\ u_t(h_t))$

Formulierung als Stochastischer Prozess

- $\Omega = \{S \times A\}^{T-1} \times S$ oder $\Omega = \{S \times A\}^{\infty}$
- $\mathfrak{A} = \mathfrak{B}(\Omega)$
- $X_t = s_t$, $Y_t = a_t$ und $Z_t = h_t$
- Anfangsverteilung $(\lambda_s)_{s \in S}$
- Strategie π induziert die Verteilung P^{π} auf $(\Omega, \mathfrak{B}(\Omega))$ durch
  - $P^{\pi}(\{X_0 = s\}) = \lambda_s$
  - $P^{\pi}(X_{t+1} = s \,|\, Z_t = h_t,\ Y_t = u_t(h_t)) = p(s \,|\, s_t,\ u_t(h_t))$

Formulierung als Stochastischer Prozess

- $\Omega = \{S \times A\}^{T-1} \times S$ oder $\Omega = \{S \times A\}^{\infty}$
- $\mathfrak{A} = \mathfrak{B}(\Omega)$
- $X_t = s_t$, $Y_t = a_t$ und $Z_t = h_t$
- Anfangsverteilung $(\lambda_s)_{s \in S}$
- Strategie π induziert die Verteilung P^π auf $(\Omega, \mathfrak{B}(\Omega))$ durch
  - $P^\pi(\{X_0 = s\}) = \lambda_s$
  - $P^\pi(X_{t+1} = s \,|\, Z_t = h_t,\ Y_t = u_t(h_t)) = p(s \,|\, s_t,\ u_t(h_t))$

Formulierung als Stochastischer Prozess

- $\Omega = \{S \times A\}^{T-1} \times S$ oder $\Omega = \{S \times A\}^{\infty}$
- $\mathfrak{A} = \mathfrak{B}(\Omega)$
- $X_t = s_t$, $Y_t = a_t$ und $Z_t = h_t$
- Anfangsverteilung $(\lambda_s)_{s \in S}$
- Strategie π induziert die Verteilung P^π auf $(\Omega, \mathfrak{B}(\Omega))$ durch
  - $P^\pi(\{X_0 = s\}) = \lambda_s$
  - $P^\pi(X_{t+1} = s \,|\, Z_t = h_t,\ Y_t = u_t(h_t)) = p(s \,|\, s_t,\ u_t(h_t))$

Formulierung als Stochastischer Prozess

- $\Omega = \{S \times A\}^{T-1} \times S$ oder $\Omega = \{S \times A\}^{\infty}$
- $\mathfrak{A} = \mathfrak{B}(\Omega)$
- $X_t = s_t$, $Y_t = a_t$ und $Z_t = h_t$
- Anfangsverteilung $(\lambda_s)_{s \in S}$
- Strategie π induziert die Verteilung P^π auf $(\Omega, \mathfrak{B}(\Omega))$ durch
  - $P^\pi(\{X_0 = s\}) = \lambda_s$
  - $P^\pi(X_{t+1} = s \,|\, Z_t = h_t,\ Y_t = u_t(h_t)) = p(s \,|\, s_t,\ u_t(h_t))$

Formulierung als Stochastischer Prozess

- $\Omega = \{S \times A\}^{T-1} \times S$ oder $\Omega = \{S \times A\}^{\infty}$
- $\mathfrak{A} = \mathfrak{B}(\Omega)$
- $X_t = s_t$, $Y_t = a_t$ und $Z_t = h_t$
- Anfangsverteilung $(\lambda_s)_{s \in S}$
- Strategie π induziert die Verteilung P^π auf $(\Omega, \mathfrak{B}(\Omega))$ durch
  - $P^\pi(\{X_0 = s\}) = \lambda_s$
  - $P^\pi(X_{t+1} = s \,|\, Z_t = h_t,\ Y_t = u_t(h_t)) = p(s \,|\, s_t,\ u_t(h_t))$

Definition (erwartete Kosten unter π)

$$V^{\pi}(s) := \mathbb{E}_s^{\pi}\left[\sum_{t=0}^{T-1} c(X_t, Y_t) + c_T(X_T)\right]$$

Gesucht ist eine Strategie π^* unter der die erwarteten Kosten minimal sind.

Definition (optimale Strategie)

π^* die

$$V^{\pi^*}(s) \leq V^{\pi}(s) \quad s \in S \qquad \forall \pi \in \Pi^H$$

erfüllt

Definition (erwartete Kosten unter π)

$$V^{\pi}(s) := \mathbb{E}_s^{\pi}\left[\sum_{t=0}^{T-1} c(X_t, Y_t) + c_T(X_T)\right]$$

Gesucht ist eine Strategie π^* unter der die erwarteten Kosten minimal sind.

Definition (optimale Strategie)

π^* die

$$V^{\pi^*}(s) \leq V^{\pi}(s) \quad s \in S \qquad \forall \pi \in \Pi^H$$

erfüllt

Definition (erwartete Kosten unter π)

$$V^{\pi}(s) := \mathbb{E}_s^{\pi}\left[\sum_{t=0}^{T-1} c(X_t, Y_t) + c_T(X_T)\right]$$

Gesucht ist eine Strategie π^* unter der die erwarteten Kosten minimal sind.

Definition (optimale Strategie)

π^* die

$$V^{\pi^*}(s) \leq V^{\pi}(s) \quad s \in S \qquad \forall \pi \in \Pi^H$$

erfüllt

Definition (Wertefunktion)

$$V^*(s) := \inf_{\pi \in \Pi^H} V^\pi(s) \quad s \in S$$

Bemerkung

- *Es gilt, dass* $V^*(s) = \min_{\pi \in \Pi^H} V^\pi(s)$, *da wir annehmen, dass* A *und* S *endlich sind.*
- $V^{\pi^*}(s) = V^*(s)$ *für jeden Startwert* $s \in S$
- *optimale Strategie bevor der Startwert bekannt ist: minimiere*

$$\sum_{s \in S} V^\pi(s) \lambda_s$$

Um die erwarteten Kosten einer Strategie $\pi \in \Pi^H$zu berechnen nutze

Algorithmus (policy evaluation)

1. $V_T^\pi(h_T) := c_T(s_T)$
2.

$$V_t^\pi(h_t) := c(s_t, u_t(h_t)) + \sum_{j \in S} p(j \mid s_t, u_t(h_t)) V_{t+1}^\pi(h_t, u_t(h_t), j) \quad (1)$$

Satz

Für $V_t^\pi(h_t)$ aus obigem Algorithmus gilt:

$$V_t^\pi(h_t) = \mathbb{E}^\pi \left[\sum_{n=t}^{T-1} c(X_n, Y_n) + c_T(X_T) \,\middle|\, Z_t = h_t \right] \tag{2}$$

und

$$V_0^\pi(s) = V^\pi(s) \qquad s \in S \tag{3}$$

Beweis:

Durch Induktion. Offensichtlich gilt
$V_T^\pi(h_T) = c_T(s_T) = \mathbb{E}^\pi\left[c_T(X_T) \,|\, Z_T = h_T\right]$ *für* t *gilt:*

$$V_t^\pi = c(s_t, u_t(h_t)) + \underbrace{\sum_{j \in S} p(j \,|\, s_t, u_t(h_t)) V_{t+1}^\pi(h_t, u_t(h_t), j)}_{\mathbb{E}^\pi\left[V_{t+1}^\pi(h_t, u_t(h_t), X_{t+1}) \,\middle|\, Z_t = h_t\right]}$$

$$\overset{IV}{=} c(s_t, u_t(h_t)) +$$

$$\mathbb{E}^\pi\left[\mathbb{E}^\pi\left[\sum_{n=t+1}^{T-1} c(X_n, Y_n) + c_T(X_T) \,\middle|\, Z_{t+1} = h_{t+1}\right] \,\middle|\, Z_t = h_t\right]$$

Definition

$$V_t^*(h_t) = \min_{\pi \in \Pi^H} V_t^\pi(h_t)$$

ist das Minimum der erwarteten Gesamtkosten über alle Strategien $\pi \in \Pi^H$ *ab dem Zeitpunkt* t

Algorithmus (Bellman Gleichungen)

1. $V_T(h_T) = c_T(s_T)$
2. $V_t(h_t) = \min_{a \in A} \left[c(s_t, a) + \sum_{j \in S} p(j \mid s_t, a) V_{t+1}(h_t, a, j) \right]$

Satz

Sei V_t aus den Bellman Gleichungen, dann gilt

1. $V_t(h_t) = V_t^*(h_t) \quad \forall h_t$
2. $V_0(s) = V^*(s) \quad s \in S$

Beweis:

1. *zu zeigen*

$$V_t(h_t) \leq V_t^*(h_t)$$

äquivalent zu

$$V_t(h_t) \leq V_t^{\pi}(h_t) \quad \forall \pi \in \Pi^H$$

2. *zu zeigen: für beliebiges* $\epsilon > 0$ *existiert eine Strategie* $\hat{\pi}$ *mit* $V_t(h_t) + (T-t)\epsilon \geq V_t^{\hat{\pi}}(h_t) \geq V_t^*(h_t)$
3. *Aus 1. und 2. folgt*

$$V_t^* \overset{def}{\leq} V_t^{\pi} \overset{Schritt\ 2}{\leq} V_t + (T-t)\epsilon \overset{Schritt\ 1}{\leq} V_t^* + (T-t)\epsilon$$

Satz (Identifiktion der optimalen Strategie)

Sei V_t^* Lösung der Bellman Gleichungen und $\pi^* := (u_0^*, \ldots, u_{T-1}^*) \in \Pi^H$ erfüllte

$$c(s_t, u_t^*(h_t)) + \sum_{j \in S} p(j \mid s_t, u_t^*(h_t)) V_{t+1}^*(h_t, u_t^*(h_t), j)$$

$$= \min_{a \in A} \left[c(s_t, a) + \sum_{j \in S} p(j \mid s_t, a) V_{t+1}^*(h_t, a, j) \right]$$

dann gilt:

1. $V_t^{\pi^*}(h_t) = V_t^*(h_t)$
2. $V^{\pi^*}(s) = V^*(s)$

Beweis:

Durch Induktion

1. *Induktionsanfang ist klar. Induktionsschritt :*

$$
\begin{aligned}
V_t^*(h_t) &= \min_{a \in A} \left[c(s_t, a) + \sum_{j \in S} p(j \mid s_t, a) V_{t+1}^*(h_t, a, j) \right] \\
&\overset{\text{def von } \pi}{=} c(s_t, u_t^*(h_t)) + \sum_{j \in S} p(j \mid s_t, u_t^*(h_t)) V_{t+1}^*(h_t, u_t^*(h_t), j) \\
&\overset{IV}{=} c(s_t, u_t^*(h_t)) + \sum_{j \in S} p(j \mid s_t, u_t^*(h_t)) V_{t+1}^{\pi^*}(h_t, u_t^*(h_t), j) \\
&= V_t^{\pi^*}(h_t)
\end{aligned}
$$

2. *Es gilt*

$$V^*(s) \overset{BE}{=} V_0^*(s) \overset{1}{=} V_0^{\pi^*}(s) \overset{PE}{=} V^{\pi^*}(s)$$

Bemerkung

- *Definiere nun nur noch die optimale Strategie so, dass gilt:*

$$u_t^*(h_t) \in \operatorname{argmin} \left[c(s_t, a) + \sum_{j \in S} p(j \mid s_t, a) V_{t+1}^*(h_t, a, j) \right]$$

- *Schließlich zeigen wir noch im letzten Satz, dass für das MDP eine optimale Markovsche Strategie existiert.*

Satz

Sei $V_t^*(h_t)$ Lösung der Bellman Gleichungen. Dann gilt

$$\forall t = 1, \ldots, T \quad \textit{hängt } V_t^*(h_t) \textit{ nur von } s_t \textit{ ab.}$$

und es existiert eine optimale Strategie $\pi \in \Pi^M$

Das bedeutet: $V^*(s) = \min_{\pi \in \Pi^H} V^\pi(s) = \min_{\pi in \Pi^M} V^\pi(s) \quad s \in S$

Beweis:

Durch Induktion

Algorithmus (Backward Induction)

1. $V_T^*(s_T) = c_T(s_T)$
2. $V_t^* = \min_{a \in A} \left[c(s_t, a) + \sum_{j \in S} p(j \mid s_t, a) V_{t+1}^*(s_{t+1}) \right]$
3. $A_{s,t}^* = argmin \left[c(s_t, a) + \sum_{j \in S} p(j \mid s_t, a) V_{t+1}^*(s_{t+1}) \right]$

Mit $u_t^*(s_t) \in A_{s,t}^* \quad \forall s_t \in S$ und $\pi^* = (u_0^*(s_0), \ldots, u_{T-1}^*(s_{T-1}))$ gilt dann, dass π^* optimale Strategie ist und $\pi^* \in \Pi^M$

Kontrollproblem

Gegeben sei:

- *ein endlicher* **Zustandsraum** S,
- *ein endlicher* **Aktionsraum** A,
- *eine Familie von* **Übergangsmatrizen** $(P(a))_{a \in A}$,
- *eine* **Kostenfunktion** $c : S \times A \to \mathbb{R}_+$ *und*
- *eine Klasse zulässiger Strategien* $\pi = (u_n)_{n=0,1,2,\ldots}$ *mit* $u_n : S^{n+1} \to A$.

Gesucht ist

1. Wertefunktion $V^* \in \mathbb{R}_+ \cup \{+\infty\}$ mit

$$V^*(s) = \inf_\pi V^\pi(s) = \inf_\pi \mathbb{E}_s^\pi \left[\sum_{n=0}^{\infty} c(X_n, u_n(X_0, ..., X_n)) \right]$$

2. und eine optimale Strategie π^*, für die gilt

$$V^*(s) = V^{\pi^*}(s) \quad \forall s \in S,$$

Gesucht ist

1. Wertefunktion $V^* \in \mathbb{R}_+ \cup \{+\infty\}$ mit

$$V^*(s) = \inf_\pi V^\pi(s) = \inf_\pi \mathbb{E}_s^\pi \left[\sum_{n=0}^{\infty} c(X_n, u_n(X_0, ..., X_n)) \right]$$

2. und eine optimale Strategie π^*, für die gilt

$$V^*(s) = V^{\pi^*}(s) \quad \forall s \in S,$$

Was ist der Unterschied zum zeitendlichen Fall?

- Es existiert kein Endzeitpunkt, bei dem man eine Rückwärtsiteration starten kann.

Kann man die Ergebnisse des zeitendlichen Falls trotzdem nutzen?

- Ja, indem man nun aber Grenzwerte betrachtet.

Was ist der Unterschied zum zeitendlichen Fall?

- Es existiert kein Endzeitpunkt, bei dem man eine Rückwärtsiteration starten kann.

Kann man die Ergebnisse des zeitendlichen Falls trotzdem nutzen?

- Ja, indem man nun aber Grenzwerte betrachtet.

Was ist der Unterschied zum zeitendlichen Fall?

- Es existiert kein Endzeitpunkt, bei dem man eine Rückwärtsiteration starten kann.

Kann man die Ergebnisse des zeitendlichen Falls trotzdem nutzen?

- Ja, indem man nun aber Grenzwerte betrachtet.

Was ist der Unterschied zum zeitendlichen Fall?

- Es existiert kein Endzeitpunkt, bei dem man eine Rückwärtsiteration starten kann.

Kann man die Ergebnisse des zeitendlichen Falls trotzdem nutzen?

- Ja, indem man nun aber Grenzwerte betrachtet.

Definition

Zu einem Steuerungsproblem auf dem Zustandsraum S, mit Aktionen aus A, einer Kostenfunktion $c : S \times A \to \mathbb{R}_+$ und Übergangsmatrizen $(P(a))_{a\in A}$ definieren wir die Folge von Funktionen $(V_n)_{n=0,1,2,\ldots}$, $V_n : S \to \mathbb{R}_+$ durch

$$V_0(s) = \inf_{a\in A} c(s,a), \tag{4}$$

$$V_{n+1}(s) = \inf_{a\in A} \left\{ c(s,a) + \sum_{s'\in S} p_{s,s'}(a) V_n(s') \right\}. \tag{5}$$

Lemma

Für die in (**??**) *definierte Folge* $(V_n)_{n=1,2,\ldots}$ *gilt*

$$V_n(s) \leq V_{n+1}(s).$$

Dies erlaubt uns die Definition des Grenzwertes

$$V_\infty := \lim_{n\to\infty} V_n \in \mathbb{R}_+ \cup \{\infty\}.$$

Lemma

Es existiert eine stationäre Steuerung $u : S \to A$, sodass

$$V_\infty(s) = c(s, u(s)) + \sum_{s' \in S} p_{s,s'}(u(s)) V_\infty(s') \tag{6}$$

für alle $s \in S$ gilt.

Satz

Es gelten die folgenden Aussagen:

(i) $V_n(s) \overset{n\to\infty}{\to} V^*(s)\ \forall s \in S.$

(ii) *Gilt für eine stationäre Steuerung* u^*

$$c(s, u^*(s)) + \sum_{s'} p_{s,s'}(u^*(s)) V^*(s') =$$

$$\inf_{a\in A} \left\{ c(s,a) + \sum_{s'} p_{s,s'}(a) V^*(s') \right\}$$

für alle $s \in S$*, dann ist* u^* *optimale, d.h.*

$$V^{u^*}(s) = V^*(s) \ \ \forall s \in S.$$

Value Iteration

Teil (i) des letzten Satzes liefert einen Algorithmus zur approximativen Berechnung der Wertefunktion V^*. Der Algorithmus startet in $V_0(s) = \inf_{a \in A} c(s,a)$ und die n-te Approximation bestimmt sich durch

$$V_{n+1}(s) = \inf_{a \in A} \left\{ c(s,a) \; + \; \sum_{s' \in S} p_{s,s'}(a) V_n(s') \right\}.$$

Dieses Verfahren zur näherungsweisen Bestimmung der Wertefunktion wird als *Value Iteration* bezeichnet.

Policy Improvement

Satz

Für eine vorgegebene stationäre Steuerung $w : S \to A$ definieren wir iterativ eine Folge von stationären Steuerungen durch

$$w_0 := w,\ w_{n+1}(s) := argmin_{a \in A} \left\{ c(s,a) + \sum_{s' \in S} p_{s,s'}(a) V^{w_n}(s') \right\}.$$

Dann erhalten wir:

(i) $V^{w_{n+1}} \leq V^{w_n}$.

(ii) *In endlich vielen Schritten erreicht der Algorithmus die Abbruchbedingung $w_{n+1} = w_n := w^*$ und die stationäre Steuerung w^* ist optimal.*

Problem im unendlichen Modell:

- **Beschränktheit der Gesamtkostenkosten**. Im allgemeinen sind die erwarteten Gesamtkosten über eine unendlich lange Zeit nicht beschränkt.

Lösung:

- Diskontierung der Kosten über die Zeit.

Problem im unendlichen Modell:

- **Beschränktheit der Gesamtkostenkosten**. Im allgemeinen sind die erwarteten Gesamtkosten über eine unendlich lange Zeit nicht beschränkt.

Lösung:

- Diskontierung der Kosten über die Zeit.

Definition (Das Diskontierte Modell)

Für einen Diskontierungsfaktor $\alpha \in (0,1)$ *betrachten wir die diskontierten erwarteten Kosten aus einer Steuerungssequenz* π

$$V_\alpha^\pi(s) = \mathbb{E}_s^\pi \left[\sum_{n=0}^{\infty} \alpha^n c(X_n, u_n(X_n)) \right]$$

und die optimalen diskontierten Kosten

$$V_\alpha^*(s) = \inf_\pi V_\alpha^\pi(s).$$

Wie im nicht-diskontierten Fall definiert man weiterhin die mit $\alpha \in (0,1)$ diskontierte Folge $(V_{n,\alpha})_{n=0,1,\ldots}$, wobei $V_{0,\alpha} = V_0$ und

$$V_{n+1,\alpha}(s) = \inf_a \left\{ c(s,a) + \alpha \sum_{s'} p_{s,s'}(a) V_{n,\alpha}(s'). \right\}.$$

Für eine gegebene stationäre Steuerung w definiert man außerdem die diskontierte Folge stationärer Steuerungen $w_0^\alpha := u$,

$$w_{n+1}^\alpha(s) := argmin_{a \in A} \left\{ c(s,a) + \alpha \sum_{s'} p_{s,s'}(a) V_\alpha^{w_n}(s') \right\}.$$

Satz

Wie im nicht diskontierten Fall erhält man die analogen Aussagen

(i) $V_{n,\alpha}(s) \nearrow V_\alpha^*(s)$ *für* $n \to \infty$ $\forall s \in S$.

(ii) *Ist* u^* *eine stationäre Steuerung, welche für alle Zustände* s *den Term*

$$c(s,a) + \alpha \sum_{s'} p_{s,s'}(a) V_\alpha^*(s')$$

minimiert, dann ist u^* *optimale im Sinne von* $V_\alpha^{u*} = V_\alpha^*$.

(iii) *Für jede stationäre Ausgangssteuerung* w *gilt die Konvergenz*

$$V_\alpha^{w_n^\alpha}(s) \searrow V_\alpha^*(s) \quad \text{für } n \to \infty, \quad \forall s \in S.$$

Bemerkung zu Value Iteration

Lemma

Ist der Abstand $\|V_{n+1,\alpha} - V_{n,\alpha}\|_\infty \leq \delta$, *dann*

(i) *verhält sich der Abstand von* V_n *zum optimalen Wert entsprechend*

$$\|V_\alpha^* - V_{n,\alpha}\|_\infty \leq \frac{\|V_{n+1,\alpha} - V_{n,\alpha}\|_\infty}{1-\alpha} \leq \frac{\delta}{1-\alpha}$$

(ii) *und der Abstand des Wertes der Steuerung* u_n *zum optimalen Wert entsprechen*

$$\|V_\alpha^* - V_\alpha^{u_n}\|_\infty \leq \frac{2\|V_{n+1,\alpha} - V_{n,\alpha}\|_\infty}{1-\alpha} \leq \frac{2\delta}{1-\alpha}.$$

Bemerkung zu Value Iteration

Für einen maximalen Fehler von $\varepsilon > 0$ wähle also n so, dass $\|V_{n+1,\alpha} - V_{n,\alpha}\| < (1-\alpha)\varepsilon$.

Bemerkung zu Policy Improvement

In jedem Iterationsschritt müssen die erwarteten diskontierten Kosten einer Steuerung V_α^u berechnet werden. Die ist analytisch i.A. nicht möglich. Für $c_{max} = \max_{s \in S, a \in A} c(s,a)$ gilt

$$\begin{aligned} V_\alpha^u(s) &= \mathbb{E}_s^u \left[\sum_{n=0}^{\infty} \alpha^n c(X_n, u(X_n)) \right] \\ &\leq \underbrace{\mathbb{E}_s^u \left[\sum_{n=0}^{k} \alpha^n c(X_n, u(X_n)) \right]}_{=: V_{\alpha,k}^u(s)} + c_{max} \frac{\alpha^{k+1}}{1-\alpha} \end{aligned}$$

Wählt man das k also hinreichen groß, so kann man $V_{\alpha,k}^u(s)$ zur Approximation für $V_\alpha^u(s)$ nutzen und produziert dadurch nur einen maximalen gewählten Fehler.

Berechnung der $V^u_{\alpha,k}(s)$

1. Die $V^u_{\alpha,k}(s)$ kann man direkt berechnen, da man durch die Potenzen von $P(u)$ die Verteilungen der X_n kennt.

2. Dank des Gesetzes der Großen Zahlen kann man auch hinreichend viele Realisierungen von $\sum_{n=0}^{k} \alpha^n c(X_n, u(X_n))$ simulieren und diese dann mitteln. Dies gibt zwar nur eine Approximation des Erwartungswertes, kann jedoch bei großen Problemen praktischer sein.

Berechnung der $V^u_{\alpha,k}(s)$

1. Die $V^u_{\alpha,k}(s)$ kann man direkt berechnen, da man durch die Potenzen von $P(u)$ die Verteilungen der X_n kennt.

2. Dank des Gesetzes der Großen Zahlen kann man auch hinreichend viele Realisierungen von $\sum_{n=0}^{k} \alpha^n c(X_n, u(X_n))$ simulieren und diese dann mitteln. Dies gibt zwar nur eine Approximation des Erwartungswertes, kann jedoch bei großen Problemen praktischer sein.

Wir betrachten nun das Folgende Modell des Lagerbestandsproblem.

- **M**: Lagerkapazität
- **c**: Kosten für den Ankauf von Ware
- **l**: Lagerkosten
- **pr**: Verkaufspreis
- **d**: Dichtevektor der zufälligen Nachfrage

Wir betrachten nun das Folgende Modell des Lagerbestandsproblem.

- **M**: Lagerkapazität
- **c**: Kosten für den Ankauf von Ware
- **l**: Lagerkosten
- **pr**: Verkaufspreis
- **d**: Dichtevektor der zufälligen Nachfrage

Wir betrachten nun das Folgende Modell des Lagerbestandsproblem.

- **M**: Lagerkapazität
- **c**: Kosten für den Ankauf von Ware
- **l**: Lagerkosten
- **pr**: Verkaufspreis
- **d**: Dichtevektor der zufälligen Nachfrage

Wir betrachten nun das Folgende Modell des Lagerbestandsproblem.

- **M**: Lagerkapazität
- **c**: Kosten für den Ankauf von Ware
- **l**: Lagerkosten
- **pr**: Verkaufspreis
- **d**: Dichtevektor der zufälligen Nachfrage

Wir betrachten nun das Folgende Modell des Lagerbestandsproblem.

- **M**: Lagerkapazität
- **c**: Kosten für den Ankauf von Ware
- **l**: Lagerkosten
- **pr**: Verkaufspreis
- **d**: Dichtevektor der zufälligen Nachfrage

Im Lagerbestandsmodell minimieren wir nicht die Kosten sondern wir maximieren den Gewinn. Aus diesem Grund ersetzen wir die im Theorieteil $c(s, a)$ genannten Kostenfunktion (c wie cost)durch die Gewinnfunktion $r(s, a)$ (r wie Reward) und min durch max. Die Gewinnfunktion sowie die Übergangsmatrix $P(a)$ errechnen sich wie folgt:

Übergangsmatrix $P(a)$

$$p(j \mid s,a) = \begin{cases} 0 & \text{für } j > s+a \\ d_{s+a-j} & \text{für } a+s \geq j > 0 \\ P(D \geq s+a) = 1 - \sum_{n=0}^{a+s-1} d_n & \text{für } j = 0 \end{cases}$$

Gewinn in Abängigkeit des folgenden Zustandes

$$r(j,s,a) = \begin{cases} pr(s+a-j) - c(a) - l(s+a) & \forall j \leq a+s \\ -\infty & \forall j > a+s \end{cases}$$

Also gilt für den erwarteten Gewinn

$$r(s,a) = \sum_{j \in S} r(j,s,a) p(j \mid s,a)$$

Übergangsmatrix $P(a)$

$$p(j \mid s,a) = \begin{cases} 0 & \text{für } j > s+a \\ d_{s+a-j} & \text{für } a+s \geq j > 0 \\ P(D \geq s+a) = 1 - \sum_{n=0}^{a+s-1} d_n & \text{für } j = 0 \end{cases}$$

Gewinn in Abängigkeit des folgenden Zustandes

$$r(j,s,a) = \begin{cases} pr(s+a-j) - c(a) - l(s+a) & \forall j \leq a+s \\ -\infty & \forall j > a+s \end{cases}$$

Also gilt für den erwarteten Gewinn

$$r(s,a) = \sum_{j \in S} r(j,s,a) p(j \mid s,a)$$

Übergangsmatrix $P(a)$

$$p(j \mid s,a) = \begin{cases} 0 & \text{für } j > s+a \\ d_{s+a-j} & \text{für } a+s \geq j > 0 \\ P(D \geq s+a) = 1 - \sum_{n=0}^{a+s-1} d_n & \text{für } j = 0 \end{cases}$$

Gewinn in Abängigkeit des folgenden Zustandes

$$r(j,s,a) = \begin{cases} pr(s+a-j) - c(a) - l(s+a) & \forall j \leq a+s \\ -\infty & \forall j > a+s \end{cases}$$

Also gilt für den erwarteten Gewinn

$$r(s,a) = \sum_{j \in S} r(j,s,a) p(j \mid s,a)$$

Wir haben bei der Implementierung einfache Kosten- und Lagerfunktionen benutzt.

1. lineare Funktionen ohne Fixkosten:
   $r(s+a-j) = pr*(s+a-j) - c*a - l*(s+a)$
2. lineare Funktionen mit Fixkosten:
   $r(s+a-j) = pr*(s+a-j) - F(a \neq 0) - c*a - l*(s+a)$
3. Wurzelfunktion ohne Fixkosten:
   $r(s+a-j) = pr*(s+a-j) - c*a - l*\sqrt{(s+a)}$
4. Wurzelfunktion mit Fixkosten:
   $r(s+a-j) = pr*(s+a-j) - F(a \neq 0) - c*a - l*\sqrt{(s+a)}$

Simulationen für den endlichen Fall

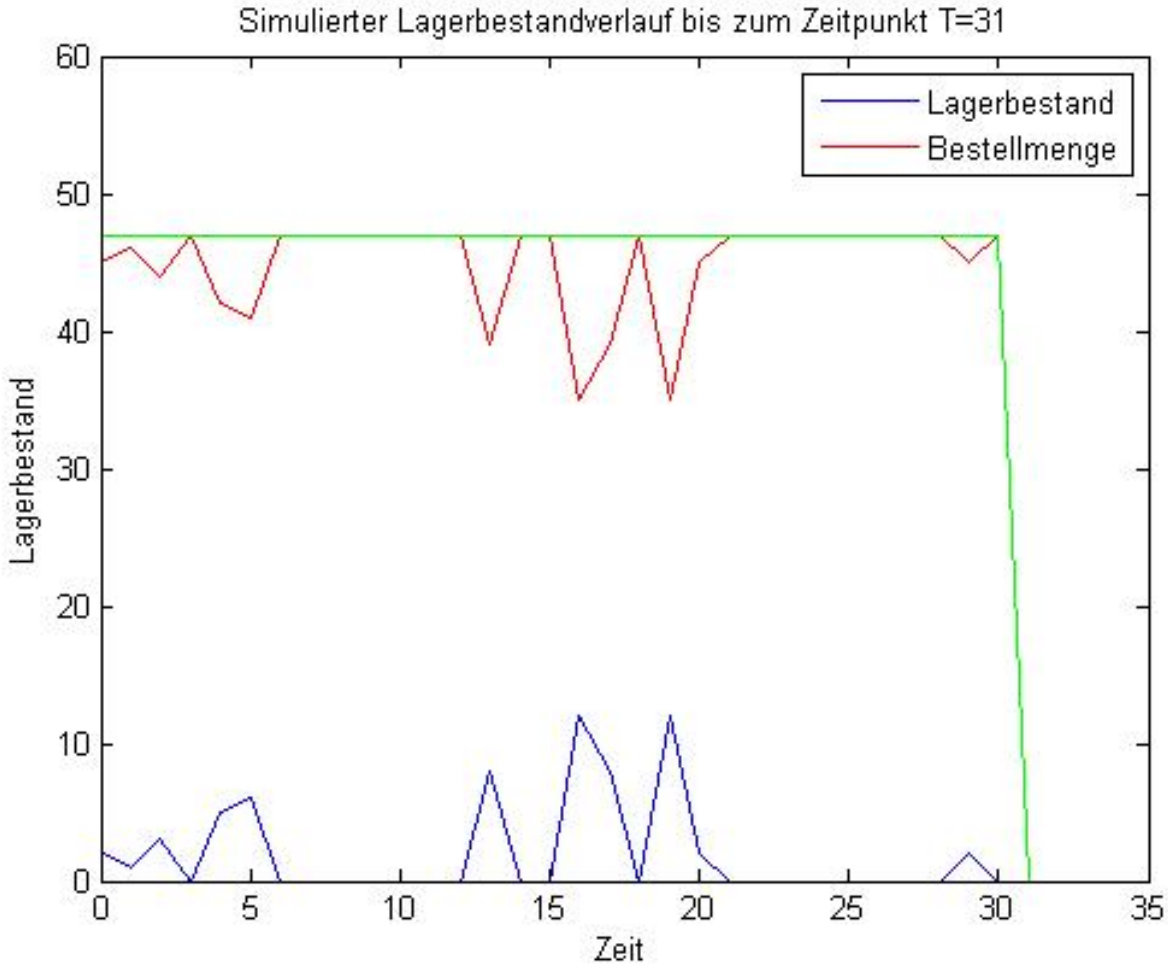

Abbildung: lineare Kosten ohne Fixkosten Poissonverteilt mit $\lambda = 50$, max. Lagerbestand 100, p=4, c=1, l=2

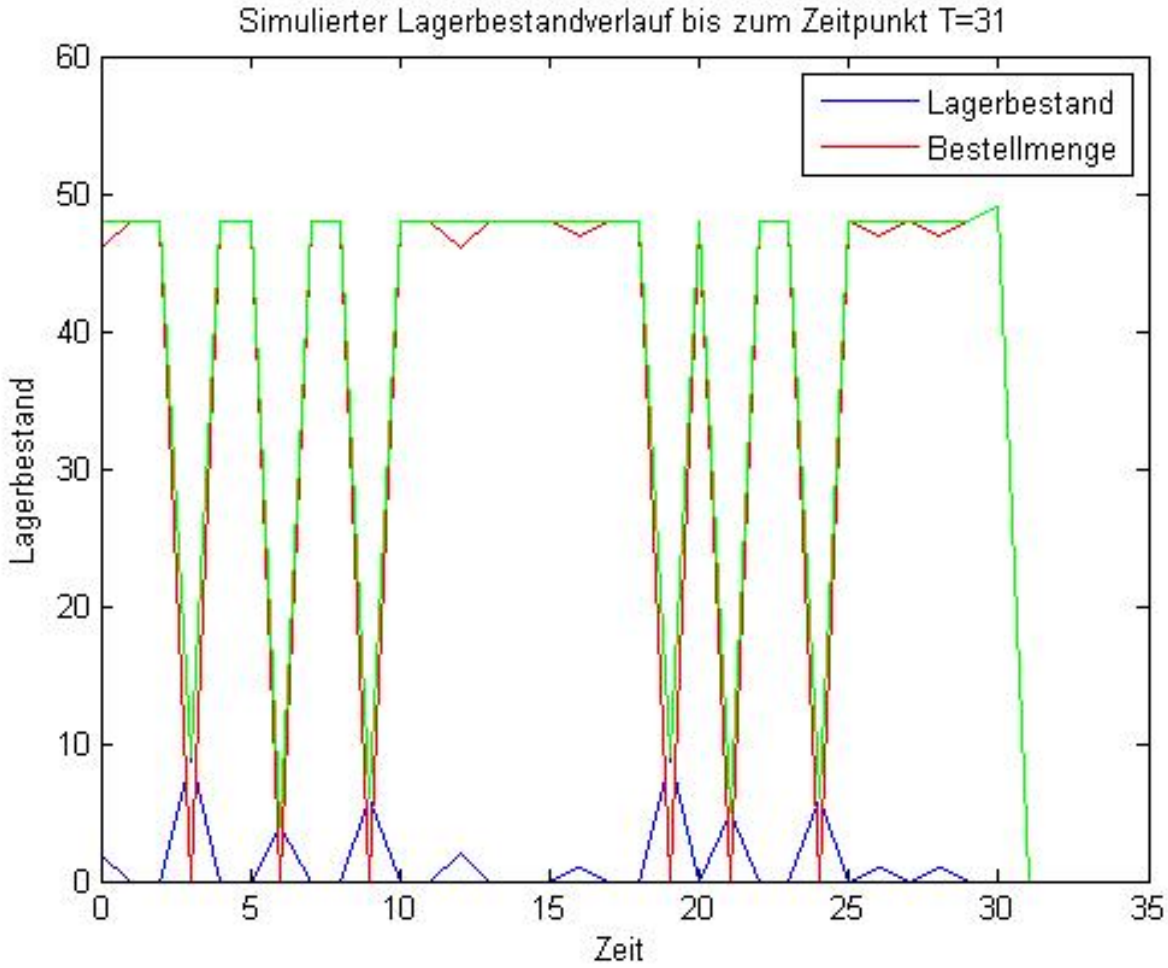

Abbildung: lineare Kosten mit Fixkosten(40), Poissonverteilt mit $\lambda = 50$, max. Lagerbestand 100, p=4, c=1, l=2

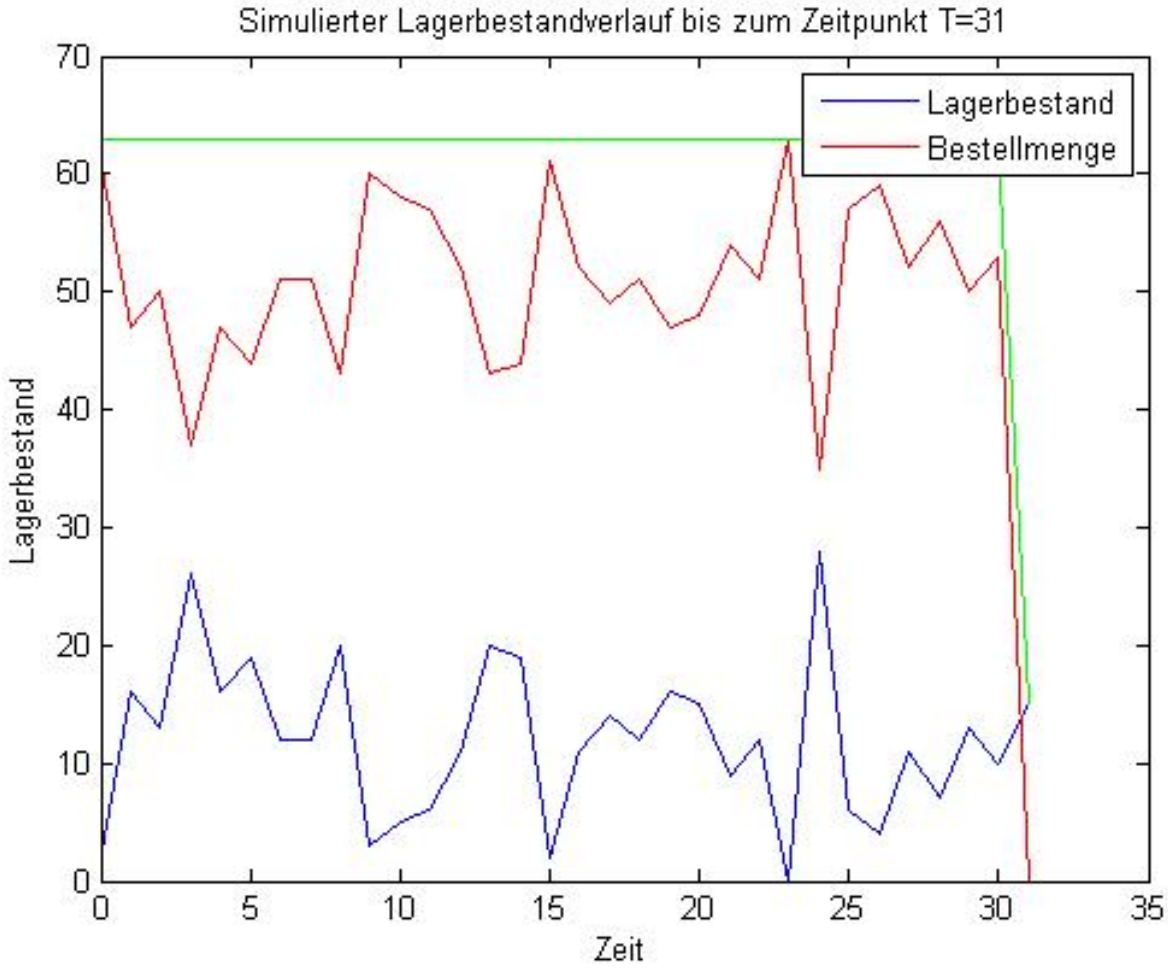

Abbildung: Wurzelfunktion ohne Fixkosten, Poissonverteilt mit $\lambda = 50$, max. Lagerbestand 100, p=4, c=1, l=2

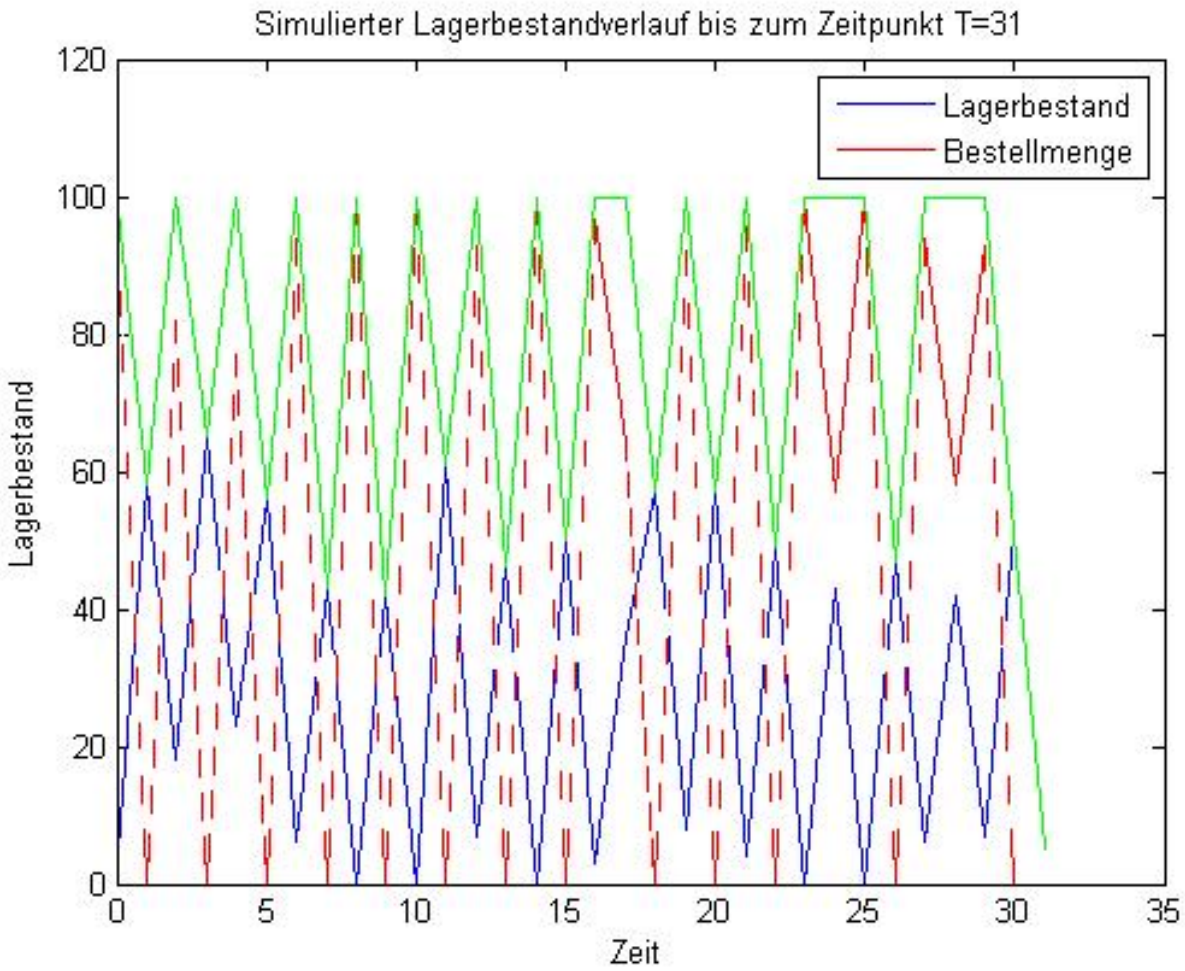

Abbildung: Wurzelfunktion mit Fixkosten(40), Poissonverteilt mit $\lambda = 50$, max. Lagerbestand 100, p=4, c=1, l=2

Wir stellen folgendes fest

- Ohne Fixkosten: optimalen Lagerbestand
  - Linear: unterhalb des EW
  - Wurzel:: über dem EW
- Mit Fixkosten : Bestellgrenze (nur wenn der Bestand unter einem bestimmten Wert liegt wird bestellt)
  - Linear: ca. bis zum optimalen Lagerbestand
  - Wurzel: bis zur Kapazitätsgrenze
- Preis erhöhen: gleiche Struktur
- Verteilung verändern: gleiche Struktur
- optimale Strategie fast stationär.

Zeitabhängige Gewinnfunktion:

$$r(s+a-j) = (pr+z*((\frac{T-t}{T}))^2)*(s+a-j)-F(a \neq 0)-c*a-l*\sqrt{(s+a)}$$

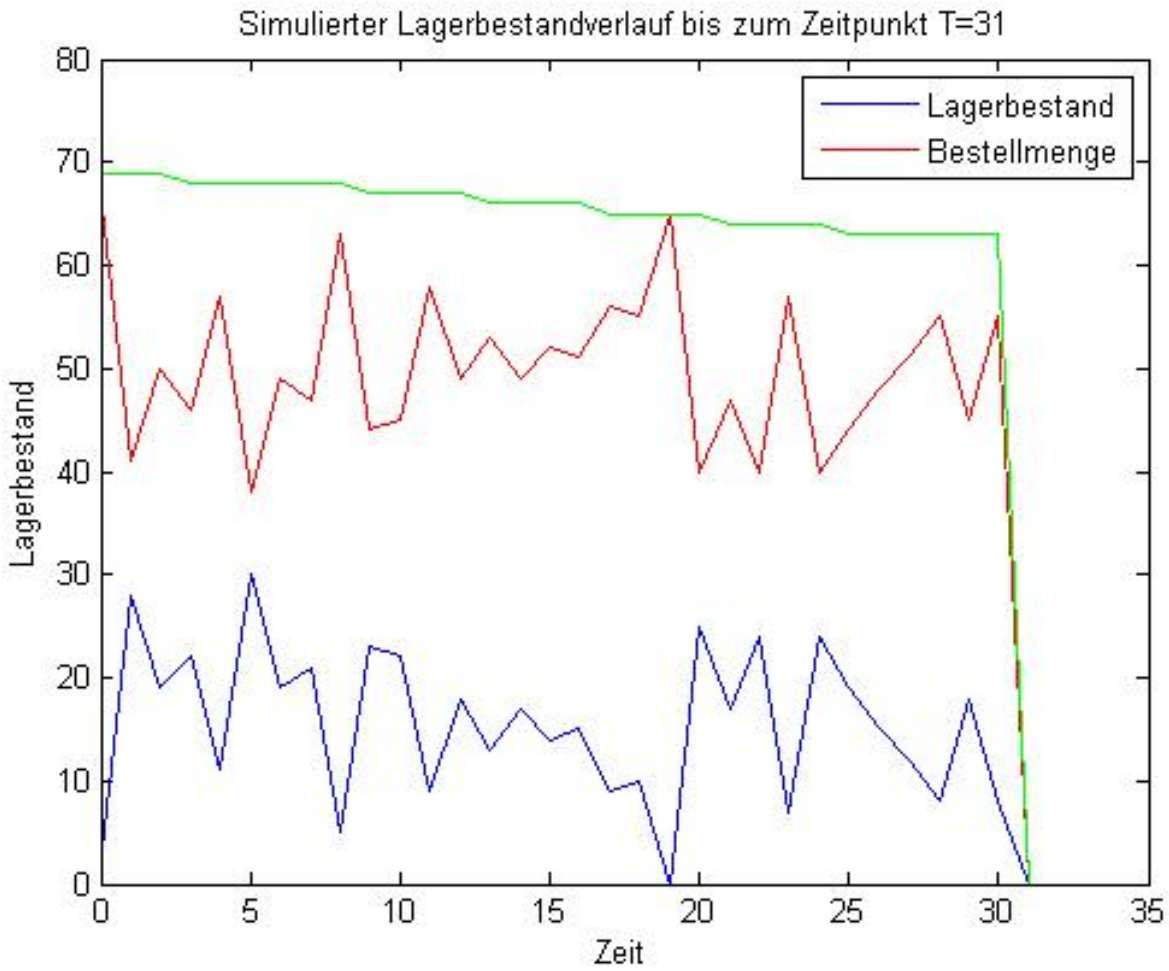

Abbildung: Zeitabhängige Gewinnfunktion, max. Lagerbestand 100, $\lambda = 50$, p=4, z=20, c=1, I=2, Wurzel- Kosten.

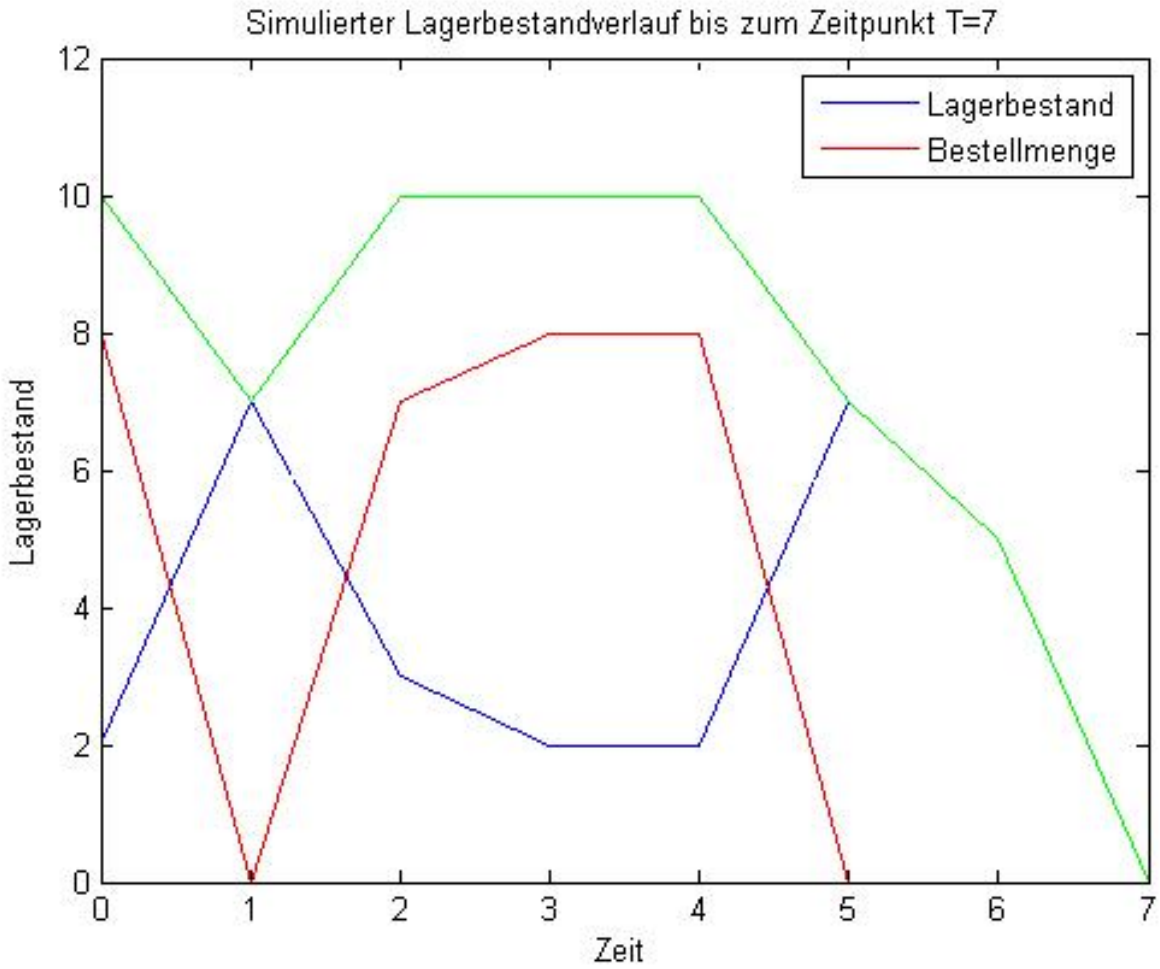

Abbildung: Wochenübersicht bei zeitabhängiger Gewinnfunktion, Fixkosten (4) und Wurzelfunktion als Lagerkosten. max. Lagerbestand 10, $\lambda = 5$, p=4,z=20, c=1, I=2

Zustand	t=0	t=1	t=2	t=3	t=4	t=5	t=6
0	10	10	10	10	10	10	6
1	9	9	9	9	9	9	5
2	8	8	8	8	8	8	4
3	7	7	7	7	7	7	0
4	6	6	6	6	6	6	0
5	5	5	5	5	0	0	0
6	4	4	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0

Tabelle: optimale Strategie zur vorhergehenden Abbildung

Simulationen für den unendlichen Fall

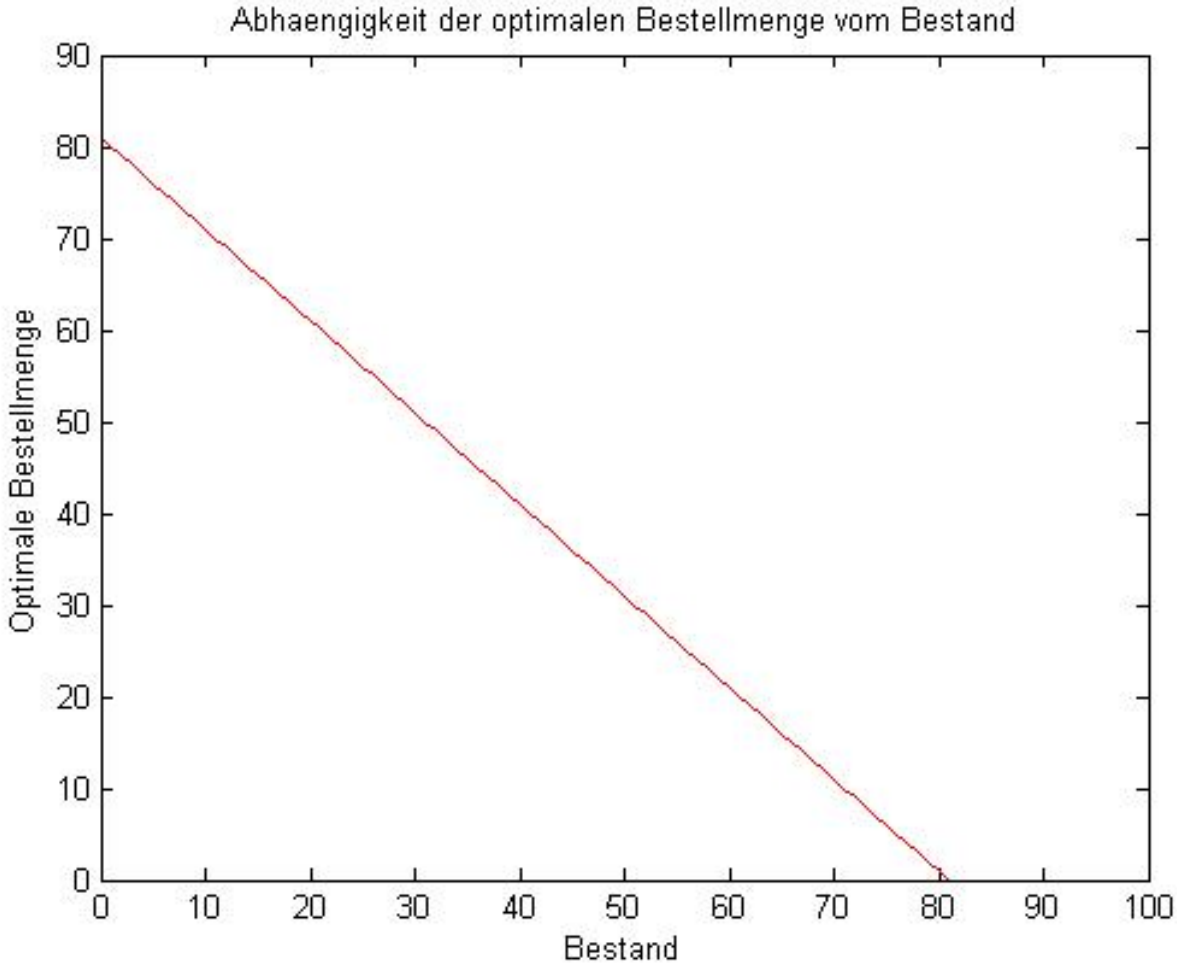

Abbildung: Optimle Steuerung für max. Lagerbestand 100, Poissonnachfrage mit $\lambda = 80$, $p = 4$, $c = 1$, $l = 2$, $p = 0.9$ und Wurzel-Kosten

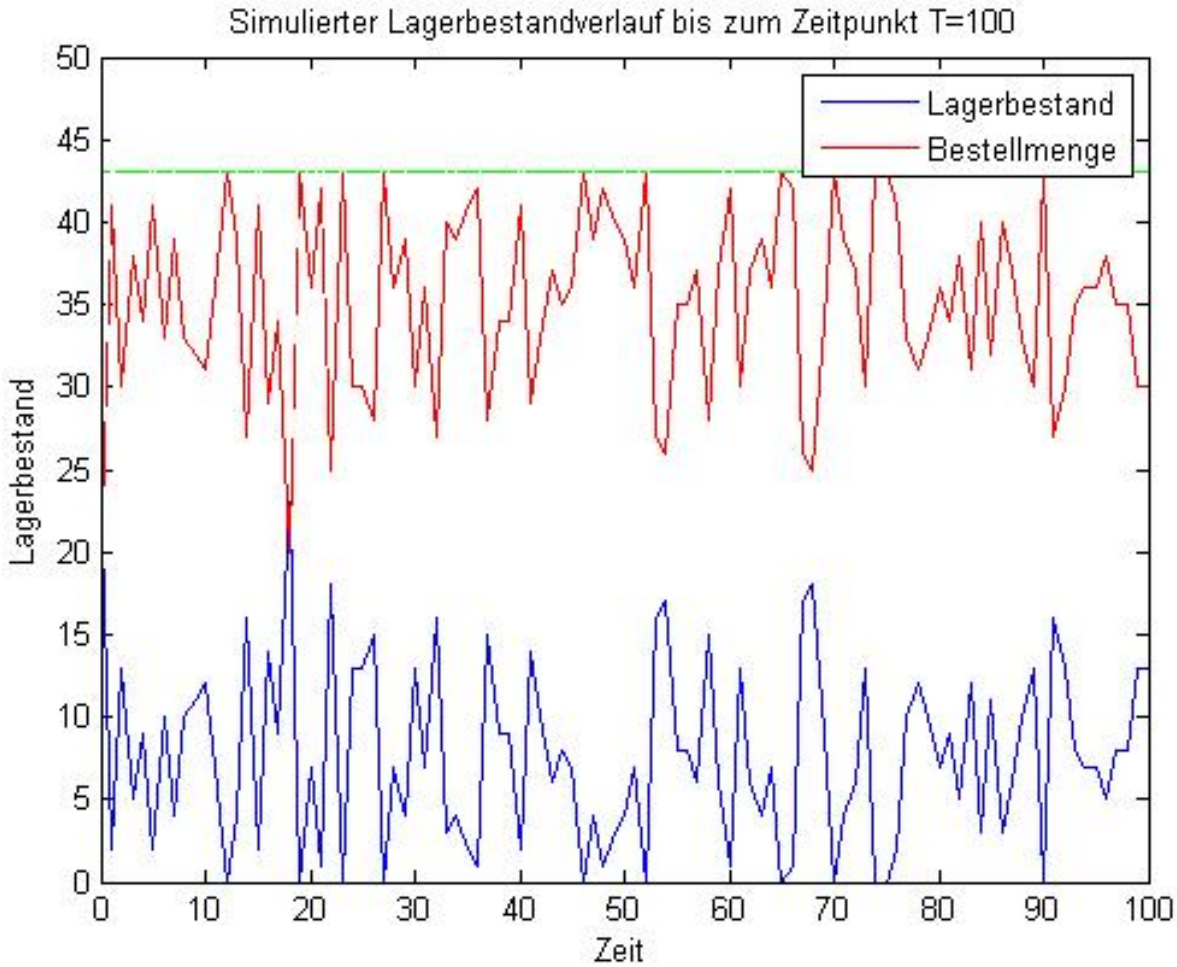

Abbildung: Simulierte Trajektorie für max. Lagerbestand 50, Poissonnachfrage mit $\lambda = 35$, $p = 4$, $c = 1$, $l = 2$, $p = 0.9$ und Wurzel-Kosten.

Parameter\Kosten	linear	Wurzel	logarithmisch
max. Lager 50, $\lambda = 35$ $p = 4, c = 1, l = 2$ $\alpha = 0.9$, $\varepsilon = 0.5$	61	72	72
max. Lager 50, $\lambda = 35$ $p = 4, c = 1, l = 2$ $\alpha = 0.98$, $\varepsilon = 0.5$	394	452	455
max. Lager 150, $\lambda = 105$ $p = 4, c = 1, l = 2$ $\alpha = 0.9$, $\varepsilon = 0.5$	70	81	81
max. Lager 50, $\lambda = 35$ $p = 4, c = 1, l = 2$ $\alpha = 0.9$, $\varepsilon = 0.005$	104	115	116

Tabelle: Anzahl der Iterationen n bei der *Value Iteration* abhängig vom Parametern und dem maximalen Fehler ε. Wähle n so, dass $\|V_{n+1,\alpha} - V_{n,\alpha}\| < (1-\alpha)\varepsilon$.

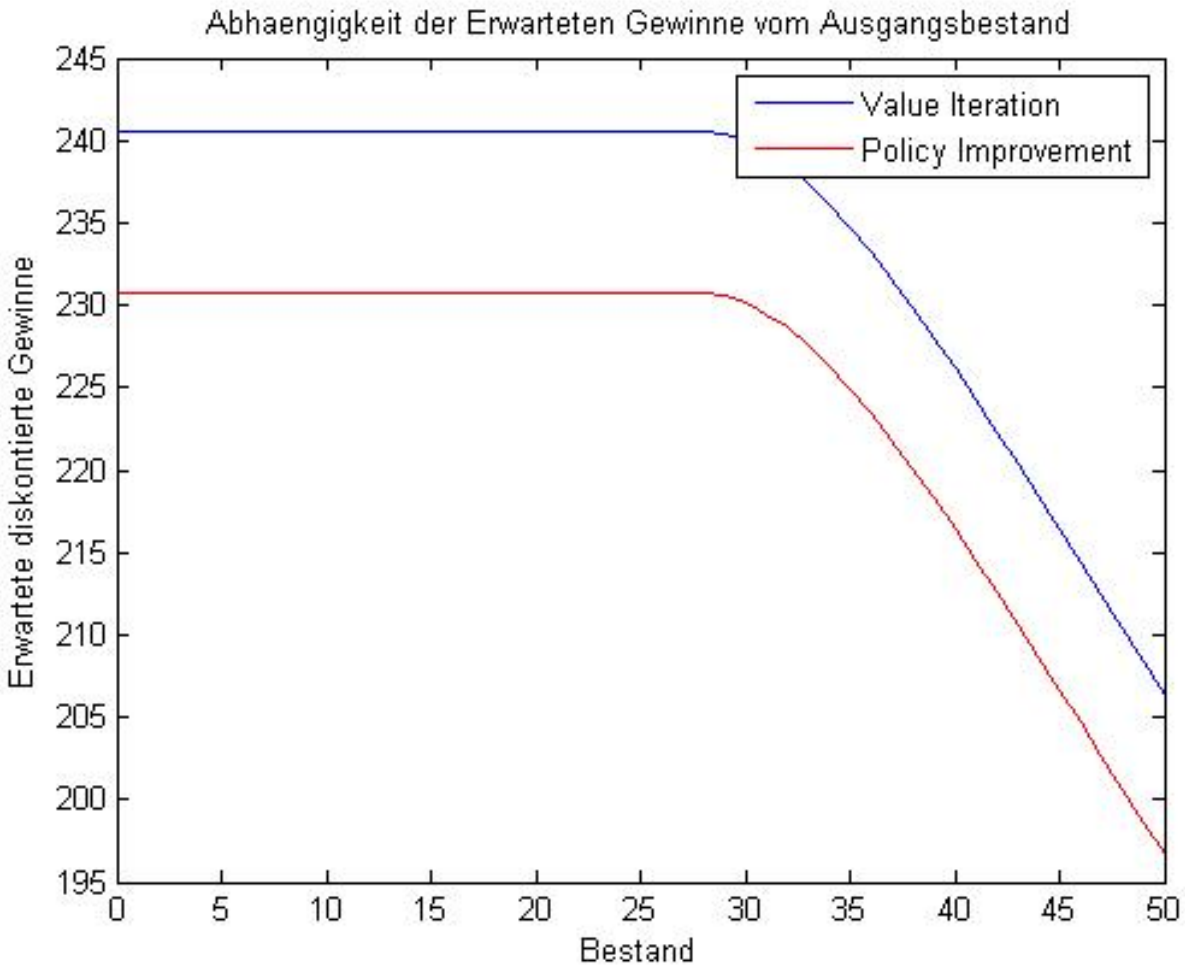

Abbildung: Vergleich *Value It.* und *Policy Im.* zur Berechnung der Wertefunktion bei max. Lagerbestand 50, Poissonnachfrage mit $\lambda = 35$, $p = 4$, $c = 1$, $l = 2$, $p = 0.9$, linearen Kosten und **30 Summanden**.

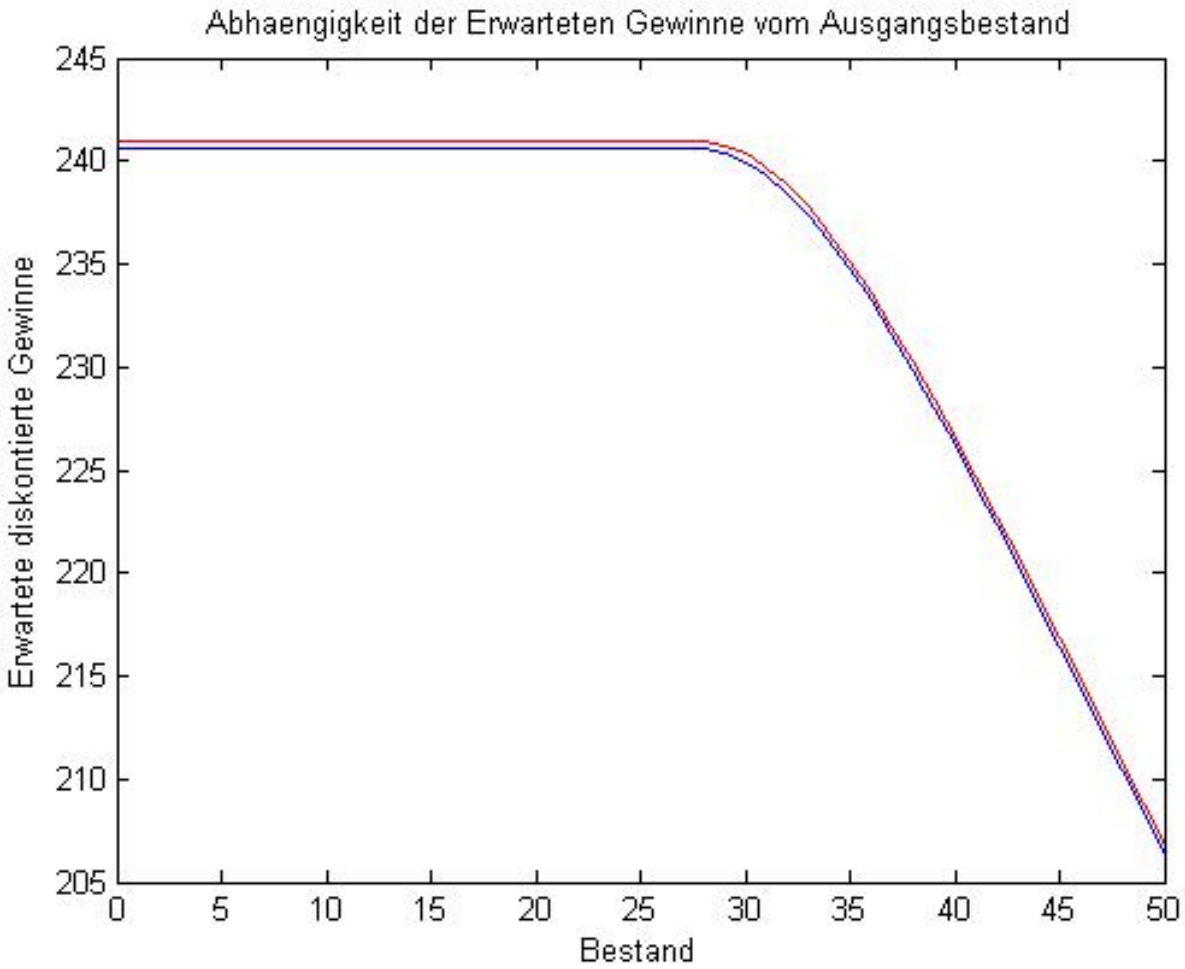

Abbildung: Vergleich *Value It.* und *Policy Im.* zur Berechnung der Wertefunktion bei max. Lagerbestand 50, Poissonnachfrage mit $\lambda = 35$, $p = 4$, $c = 1$, $l = 2$, $p = 0.9$, linearen Kosten und **100 Summanden**.