§5 Anwendung der Politik-Iteration im Taxibetrieb und beim Ersetzen von Autos

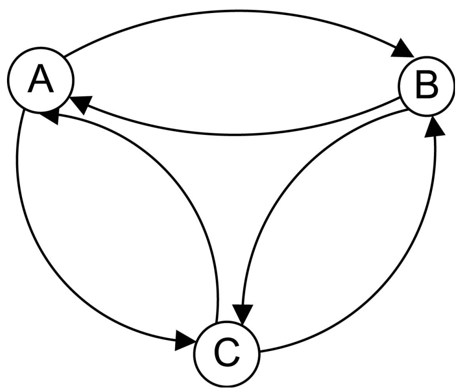
5.1 Beispiel – Taxibetrieb

Fassen wir das Problem eines Taxichauffeurs ins Auge, dessen Gebiet drei Städte A, B und C umfasst. Wenn er sich in der Stadt A befindet, hat er drei mögliche Strategien:

1. Er fährt planlos durch die Gegend in der Hoffnung, einen Fahrgast, von dem er angehalten wird, aufzunehmen.

2. Er fährt zum nächsten Taxistand und wartet, bis er an die Reihe kommt.

3. Er zieht sich zurück und wartet auf eine Fernmeldung.



Befindet er sich in der Stadt C, hat er die gleichen drei Strategien, ist er jedoch in der Stadt B, fällt die letzte Strategie dahin, weil es in dieser Stadt keine Funkzentrale gibt. Für eine bestimmte Stadt und eine bestimmte Strategie bestehen Wahrscheinlichkeiten dafür, dass die nächste Fahrt zu einer der drei Städte A, B oder C führt, und die entsprechenden Erlöse in Geldeinheiten, die mit einer derartigen Fahrt verbunden sind, sind bekannt. Diese Erlöse stellen das Einkommen der Fahrt nach Abzug aller nötigen Unkosten dar.

Zum Beispiel müssen im Falle der Strategien 1 und 2 die Kosten für das Umherfahren und den Weg zum nächsten Taxistand bei der Berechnung der Erlöse berücksichtigt werden. Die Übergangswahrscheinlichkeiten und die Erlöse hängen von der Strategie ab, weil durch die verschiedenen Strategien ganz verschiedene Kundenkreise erfasst werden.

Wenn wir den Aufenthalt in den Städten A, B und C mit den Zuständen 1, 2 und 3 identifizieren, ergibt sich die folgende Tabelle.

Zustand i	Stratwgie k	Wahrsch. p_{ij}^k, $j = 1, 2, 3$	Gewinn c_{ij}^k $j = 1, 2, 3$	Zu erwartender unmittelbarer Erlös $q_i^k = \sum_{j=1}^N p_{ij}^k c_{ij}^k$
1	1	$\left(\frac{1}{2}\ \frac{1}{4}\ \frac{1}{4}\right)$	$(10\ 4\ 8)$	8
1	2	$\left(\frac{1}{16}\ \frac{3}{4}\ \frac{3}{16}\right)$	$(8\ 2\ 4)$	2.75
1	3	$\left(\frac{1}{4}\ \frac{1}{8}\ \frac{5}{6}\right)$	$(4\ 6\ 4)$	4.25
2	1	$\left(\frac{1}{2}\ 0\ \frac{1}{2}\right)$	$(14\ 0\ 18)$	16
2	2	$\left(\frac{1}{16}\ \frac{7}{8}\ \frac{1}{16}\right)$	$(8\ 16\ 8)$	15
3	1	$\left(\frac{1}{4}\ \frac{1}{4}\ \frac{1}{2}\right)$	$(10\ 2\ 8)$	7
3	2	$\left(\frac{1}{8}\ \frac{3}{4}\ \frac{1}{8}\right)$	$(6\ 4\ 2)$	4
3	3	$\left(\frac{3}{4}\ \frac{1}{16}\ \frac{3}{16}\right)$	$(4\ 0\ 8)$	4.5

Der Erlös wird in irgendeiner beliebigen Geldeinheit angegeben. Die Zahlen in dieser Tabelle sind eher wegen der rechnerischen einfachheit als aus irgendeinem anderen Grund gewählt worden.

Zu Beginn des Entscheidungsprozesses setzen wir v_1, v_2 und $v_3 = 0$, so dass in der Politik-Verbesserung zuerst die Politik, die den zu erwartenden unmittelbaren Erlös maximiert, ausgewählt wird. Bei der Prüfung der q_i^k sehen wir, dass

$$\max_k \left[q_1^k\right] = \max\left\{q_1^1,\ q_1^2,\ q_1^3\right\} = \min\{8,\ 2.75,\ 4.25\} = 8 \Rightarrow d(1) = 1$$

$$\min_k \left[q_2^k\right] = \min\left\{q_2^1,\ q_2^2\right\} = \min\{16,\ 15\} = 16 \Rightarrow d(2) = 1$$

$$\min_k \left[q_3^k\right] = \min\left\{q_3^1,\ q_3^2,\ q_3^3\right\} = \min\{7,\ 4,\ 4.5\} = 7 \Rightarrow d(3) = 1,$$

d.h. dass diese Politik in jedem Zustand aus der ersten Strategie besteht. Mit anderen Worten, der Vektor **d**, der die Politik angibt und dessen i-tes Element die Entscheidung im i-ten Zustand ist, wird zu

$$\mathbf{d} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

oder die Politik besteht darin, in jedem Fall umherzufahren.

Schritt 0. Die der Anfangspolitik **d** entsprechenden Übergangswahrscheinlichkeiten und zu erwartenden unmittelbaren Erlöse sind

$$\mathbf{d} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{P} = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{pmatrix}, \mathbf{q} = \begin{pmatrix} 8 \\ 16 \\ 7 \end{pmatrix}.$$

Schritt 1. Nun folgt die Wertbestimmung, und wir lösen die Gleichungen

$$v_i = q_i + \sum_{j=1}^{N} p_{ij}\, v_j - g, \quad i \in E = \{1, 2, 3\}.$$

In diesem Fall haben wir

$$v_1 = 8 + \frac{1}{2}\, v_1 + \frac{1}{4}\, v_2 + \frac{1}{4}\, v_3 - g,$$

$$v_2 = 16 + \frac{1}{2}\, v_1 + 0\, v_2 + \frac{1}{2}\, v_3 - g,$$

$$v_3 = 7 + \frac{1}{4}\, v_1 + \frac{1}{4}\, v_2 + \frac{1}{2}\, v_3 - g.$$

Wir setzen willkürlich $v_3 = 0$, lösen diese Gleichungen und erhalten

$$v_1 = 1.33,\ v_2 = 7.47,\ v_3 = 0,\ g = 9.2$$

```
Solve[{v1 == 8 + 1/2 v1 + 1/4 v2 - g, v2 == 16 + 1/2 v1 - g, 0 == 7 + 1/4 v1 + 1/4 v2 - g},
  {v1, v2, g}] // N
```

```
{{v1 → 1.33333, v2 → 7.46667, g → 9.2}}
```

Mit der Politik des Umherfahrens wird der Chauffeur durchschnittlich $g = 9.2$ Einheiten pro Fahrt erhalten.

Schritt 2. In einer neuerlichen Politik-Verbesserung berechnen wir für alle i und k die Grössen

$$q_i^k + \sum_{j=1}^{N} p_{ij}^k\, v_j$$

wie das aus nächster Tabelle hervorgeht.

Für den **Zustand 1**

$$\max_k \left[q_1^k + \sum_{j=1}^{N} p_{1j}^k\, v_j\right] =$$
$$\max\left\{q_1^1 + p_{11}^1\, v_1 + p_{12}^1\, v_2 + p_{13}^1\, v_3,\ q_1^2 + p_{11}^2\, v_1 + p_{12}^2\, v_2 + p_{13}^2\, v_3,\ q_1^3 + p_{11}^3\, v_1 + p_{12}^3\, v_2 + p_{13}^3\, v_3\right\} =$$

$$\max\{10.53, 8.43, 5.52\} = 10.53 \Rightarrow d(1) = 1$$

Für den **Zustand 2**

$$\max_k \left[q_2^k + \sum_{j=1}^{M} p_{2j}^k\, v_j\right] = \max\left\{q_2^1 + p_{21}^1\, v_1 + p_{22}^1\, v_2 + p_{23}^1\, v_3,\ q_2^2 + p_{21}^2\, v_1 + p_{22}^2\, v_2 + p_{23}^2\, v_3\right\} =$$

$$\max\{16.67, 21.62\} = 21.62 \Rightarrow d(2) = 2$$

Für den **Zustand 3**

$$\max_k \left[q_3^k + \sum_{j=1}^{M} p_{3j}^k\, v_j\right] =$$
$$\max\left\{q_3^1 + p_{31}^1\, v_1 + p_{32}^1\, v_2 + p_{33}^1\, v_3,\ q_3^2 + p_{31}^2\, v_1 + p_{32}^2\, v_2 + p_{33}^2\, v_3,\ q_3^3 + p_{31}^3\, v_1 + p_{32}^3\, v_2 + p_{33}^3\, v_3\right\} =$$

$$\max\{9.20, 9.77, 5.97\} = 9.77 \Rightarrow d(3) = 2$$

Zustand i	Strategie k	$q_i^k + \sum_{j=1}^N p_{ij}^k v_j$
1	1	10.53 ←
1	2	8.43
1	3	5.52
2	1	16.67
2	2	21.62 ←
3	1	9.20
3	2	9.77 ←
3	3	5.97

Wir sehen, dass für $i = 1$ die Grösse in der rechten Spalte maximiert wird für $k = 1$. Für $i = 2$ oder 3 wird sie maximiert, wenn $k = 2$ ist. Also gilt für unsere neue Politik

$$\mathbf{d} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}.$$

Das heißt, dass der Chauffeur umherfahren soll, wenn er sich in der Stadt A befindet. Ist er in der Stadt B oder C, soll er zum nächsten Taxistand fahren.

Da die zwei aufeinanderfolgende Politiken nicht übereinstimmen, muss das Zyklus wiederholt werden.

Schritt 0. Wir haben nun

$$\mathbf{d} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}, \mathbf{P} = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{16} & \frac{7}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \end{pmatrix}, \mathbf{q} = \begin{pmatrix} 8 \\ 15 \\ 4 \end{pmatrix}.$$

Schritt 1. Nun gehen wir wieder zur Wertbestimmung über und lösen die Gleichungen

$$v_1 = 8 + \frac{1}{2} v_1 + \frac{1}{4} v_2 + \frac{1}{4} v_3 - g,$$
$$v_2 = 15 + \frac{1}{16} v_1 + \frac{7}{8} v_2 + \frac{1}{16} v_3 - g,$$
$$v_3 = 4 + \frac{1}{8} v_1 + \frac{3}{4} v_2 + \frac{1}{8} v_3 - g.$$

Wiederum mit $v_3 = 0$ erhalten wir

$$v_1 = -3.88,\ v_2 = 12.85,\ v_3 = 0,\ g = 13.15$$

```
Solve[{v1 == 8 + 1 / 2 v1 + 1 / 4 v2 - g, v2 == 15 + 1 / 16 v1 + 7 / 8 v2 - g,
   0 == 4 + 1 / 8 v1 + 3 / 4 v2 - g}, {v1, v2, g}] // N
```

```
{{v1 → -3.87879, v2 → 12.8485, g → 13.1515}}
```

Man beachte, dass g sich wie gewünscht von 9.2 auf 13.15 erhöht hat, so dass der Chauffeur im Durchschnitt13.15 Einheiten pro Fahrt verdient.

Schritt 2. In nächster Tabelle ist eine zweite Politik-Verbesserung dargestellt.

Zustand i	Strategie k	$q_i^k + \sum_{j=1}^{N} p_{ij}^k v_j$
1	1	9.27
1	2	12.14 ←
1	3	4.89
2	1	14.06
2	2	26.06 ←
3	1	9.24
3	2	13.10 ←
3	3	2.39

Die neue Politik ist somit

$$\mathbf{d} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}.$$

Der Chauffeur soll somit zum nächsten Stand fahren, gleichgültig in welcher Stadt er sich befindet.

Schritt 0. Bei dieser Politik ist

$$\mathbf{d} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}, \mathbf{P} = \begin{pmatrix} \frac{1}{16} & \frac{3}{4} & \frac{3}{16} \\ \frac{1}{16} & \frac{7}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \end{pmatrix}, \mathbf{q} = \begin{pmatrix} 2.75 \\ 15 \\ 4 \end{pmatrix}.$$

Schritt 1. Gehen wir in die Wertbestimmung, dann erhalten wir das Gleichungssystem

$$v_1 = 2.75 + \frac{1}{16} v_1 + \frac{3}{4} v_2 + \frac{3}{16} v_3 - g,$$
$$v_2 = 15 + \frac{1}{16} v_1 + \frac{7}{8} v_2 + \frac{1}{16} v_3 - g,$$
$$v_3 = 4 + \frac{1}{8} v_1 + \frac{3}{4} v_2 + \frac{1}{8} v_3 - g.$$

Mit $v_3 = 0$ ergibt sich folgende Lösung für diese Gleichungen

$$v_1 = -1.18,\ v_2 = 12.66,\ v_3 = 0,\ g = 13.34$$

```
Solve[{v1 == 2.75 + 1/16 v1 + 3/4 v2 - g, v2 == 15 + 1/16 v1 + 7/8 v2 - g,
    0 == 4 + 1/8 v1 + 3/4 v2 - g}, {v1, v2, g}] // N
```

```
{{v1 → -1.17647, v2 → 12.6555, g → 13.3445}}
```

Man beachte, dass bei g eine kleine, aber deutliche Zunahme (13.15 zu 13.34) stattgefunden hat. Jedoch haben wir bis jetzt noch keinen Beweis dafür, die optimale Politik gefunden zu haben.

Schritt 2. Die nächste Politik-Verbesserung wird in nächster Tabelle gezeigt.

Zustand i	Strategie k	$q_i^k + \sum_{j=1}^N p_{ij}^k v_j$
1	1	10.58
1	2	12.17 ←
1	3	5.54
2	1	15.41
2	2	24.42 ←
3	1	9.87
3	2	13.34 ←
3	3	4.41

Die neue Politik ist

$$\mathbf{d} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix},$$

doch diese ist gleich der vorhergehenden Politik, so dass der Prozess konvergiert und g sein Maximum, nähmlich $g = 13.34$ erreicht hat.

Der Taxichauffeuer muss in jeder Stadt zum nächsten Stand fahren. Bei dieser Politik wird ein durchschnittlicher Erlös von 13.34 Einheiten pro Fahrt erziehlt, beinahe die Hälfte mehr als bei der Politik, bei welcher ständig umhergefahren wird, und die durch Maximierung der zu erwartenden unmittelbaren Erlöse bestimmt wurde.

Die Berechnungen sind in der Tabelle zusammengefasst.

v_1	0	1.33	−3.88	−1.18
v_2	0	7.47	12.85	12.66
v_3	0	0	0	0
g	−	9.20	13.15	13.34
d (1)	1	1	2	2
d (2)	1	2	2	2
d (3)	1	2	2	2

Man beachte, dass die optimale Politik, immer zu einem Stand zu fahren, in Bezug auf die unmittelbaren Erlöse die schlechteste ist. Man könnte somit auch sagen, dass der Taxichauffeur, wenn er den besten Weg beschreiten will, nicht nur den Fahrpreis einer Fahrt berücksichtigen muss, sondern auch den Bestimmungsort der Fahrt wegen der davon abhängigen Aussicht auf andere Fahrten. Ein erfahrener Taxichauffeur wird die Richtigkeit solcher Überlegungen bestätigen. Sehr oft ergibt sich im sequentiellen Entscheidungsprozess, dass eine Taube auf dem Dach doch besser ist als ein Sperling in der Hand.

Die Politik-Verbesserung der Tabelle für zweite Politik-Verbesserung gibt uns Gelegenheit, die Gleichung (4.6.8) zu kontrollieren. Die Politik ändert sich von der Politik A

$$\mathbf{d} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix},$$

zur Politik B, dargestellt durch

$$\mathbf{d} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}.$$

Die Grössen γ_i, deliniert durch die Gleichung (4.6.2), sind aus der Tabelle für zweite Politik-Verbesserung ersichtlich. Sie sind die Differenzen zwischen den Testgrössen jeder Politik. Wir finden

$$\gamma_1 = 12.14 - 9.27 = 2.87,$$

während

$$\gamma_2 = \gamma_3 = 0$$

ist, weil die Entscheidungen in den Zuständen 2 und 3 in beiden Politiken A und B die gleichen sind.

Die Anwendung der Gleichungen aus dem Satz 1.2.5 auf die Übergangsmatrix für die Politik B ergibt die Grenzwahrscheinlichkeiten

$$\pi_1 = \frac{1}{16}\,\pi_1 + \frac{1}{16}\,\pi_2 + \frac{1}{8}\,\pi_3,$$
$$\pi_2 = \frac{3}{4}\,\pi_1 + \frac{7}{8}\,\pi_2 + \frac{3}{4}\,\pi_3,$$
$$\pi_3 = \frac{3}{16}\,\pi_1 + \frac{1}{16}\,\pi_2 + \frac{1}{8}\,\pi_3$$

```
Solve[{pi1 == 1/16 pi1 + 1/16 pi2 + 1/8 pi3, pi2 == 3/4 pi1 + 7/8 pi2 + 3/4 pi3,
   pi1 + pi2 + pi3 == 1}, {pi1, pi2, pi3}] // N
```

```
{{pi1 → 0.0672269, pi2 → 0.857143, pi3 → 0.0756303}}
```

d.h.

$$\pi_1 = 0.0672,\ \pi_2 = 0.8571,\ \pi_3 = 0.0756.$$

Aus der Gleichung (4.6.8) erhalten wir dann

$$g^{\Delta} = \sum_{i=1}^{N} \pi_i^B\,\gamma_i = \pi_1\,\gamma_1 + \pi_2\,\gamma_2 + \pi_3\,\gamma_3 = 0.0672 \cdot 2.87 = 0.19.$$

Die Änderung der Politik von A zu B müsste somit eine Gewinnerhöhung von 0.19 Einheiten nach sich ziehen. Da

$$g^A = 13.15 \text{ und } g^B = 13.34,$$

ist unsere Voraussage richtig.

5.2 Das Ersatzproblem

Das soeben behandelte Beispiel der Politik-Iterationsmethode ist etwas weit entfernt vom Problembereich der Praxis. Es wäre daher äusserst interessant, die Methode auf ein Problem anzuwenden, welches für die Industrie von grösserer Wichtigkeit ist. Als Beispiel einer solchen praktischen Anwendung wurde das Ersatzproblem gewählt. Das Problem setzt sich mit der Frage auseinander, wann ein Kapitalgut ersetzt werden soll, das mit der Zeit an Wert verliert.

Wenn wir jetzt eine Maschine seit einiger Zeit besitzen, fragen wir uns, ob wir Sie behalten oder verkaufen sollen. Und wenn wir sie verkaufen, wie alt darf die Maschine sein, die wir zu kaufen gedenken?

Um eine bessere Vorstellung zu haben, wollen wir das Autoersatzproblem über eine Zeitspanne von 10 Jahren untersuchen. Alle 3 Monate prüfen wir die Situation und entscheiden, ob wir unseren jetzigen Wagen behalten oder verkaufen. Der Zustand des Systems i wird durch das Alter des Wagens in Einheiten von drei Monaten dargestellt, d.h.

$$i \in E = \{1,\ 2,\ \ldots,\ 40\}$$

i kann also Werte von 1 bis 40 annehmen. Um die Zahl der Zustande zu begrenzen, wird ein Wagen im Alter von 40 für immer ein Wagen von 40 bleiben (wir betrachten ihn als völlig abgenutzt). Die verfügbaren Strategien in jedem Zustand sind die folgenden:

Die erste Strategie, k = l, ist die, den Wagen für ein weiteres Vierteljahr zu behalten.

Die anderen Strategien, $k > 1$, bedeuten den Kauf eines Autos vom Alter $k-2$, wobei es sich bei $k-2$ um eine Zahl bis 39 handeln darf.

Wir haben dann $|E| = 40$ Zustände mit je $|A| = 41$, das heisst, dass es 41^{40} mögliche Politiken gibt. Die folgenden Daten sind gegeben :

C_i – die Kosten beim Kauf eines Autos vom Alter $i \in E$.

T_i – der Verkehrswert eines Autos vom Alter $i \in E$.

E_i – die zu erwartenden Betriebskosten eines Wagens vom Alter i bis zum Alter $i + 1$.

p_i – die Wahrscheinlichkeit dafür, dass ein Auto vom Alter i das Alter $i+1$ erreicht, ohne dass dabei unerschwingliche Reparaturkosten entstehen.

Die hier definierte Wahrscheinlichkeit ist nötig, um die Zahl der Zustände zu begrenzen. Ein Wagen irgendeines Alters, der Totalschaden hat, wird gleich dem Zustand 40 zugeführt. Es ist klar, dass

$$\mathrm{p}_{40} = 0$$

ist.

Die grundlegenden Gleichungen, die das System im Zustand $i \in E$ beherrschen sind die folgenden:

Wenn $k = 1$ (behalte den jetzigen Wagen), so gilt

$$\mathrm{v}_i = -E_i + \mathrm{p}_i\, \mathrm{v}_{i+1} + (1 - \mathrm{p}_i)\, \mathrm{v}_{40} - g.$$

Wenn $k > 1$ (Umtausch gegen einen $k-2$ Quartale alten Wagen), dann ist

$$\mathrm{v}_i = T_i - C_{k-2} - E_{k-2} + \mathrm{p}_{k-2}\, \mathrm{v}_{k-1} + \left(1 - \mathrm{p}_{k-2}\right) \mathrm{v}_{40} - g.$$

Es ist einfach, diese Gleichungen in unseren früheren Bezeichnungen auszudrücken, zum Beispiel

$$q_i^k = -E_i \text{ für } k = 1$$

$$q_i^k = T_i - C_{k-2} - E_{k-2} \text{ für } k > 1$$

$$p_{i\,j}^k = \begin{cases} \mathrm{p}_i & \text{für } j = i+1 \\ 1 - \mathrm{p}_i & \text{für } j = 40 \\ 0 & \text{für alle anderen } j \end{cases} \qquad \text{für } k = 1$$

$$p_{i\,j}^k = \begin{cases} \mathrm{p}_{k-2} & \text{für } j = k-1 \\ 1 - \mathrm{p}_{k-2} & \text{für } j = 40 \\ 0 & \text{für alle anderen } j \end{cases} \qquad \text{für } k > 1.$$

Die Daten unseres Problems sind in der nächsten Tabelle eingetragen und in der Figur graphisch dargestellt.

Die Unstetigkeiten in den Kaufpreis- und Verkehrswertfunktionen wurden eingeführt, um die Auswirkungen des Erscheinens neuer Modelle zu charakterisieren.

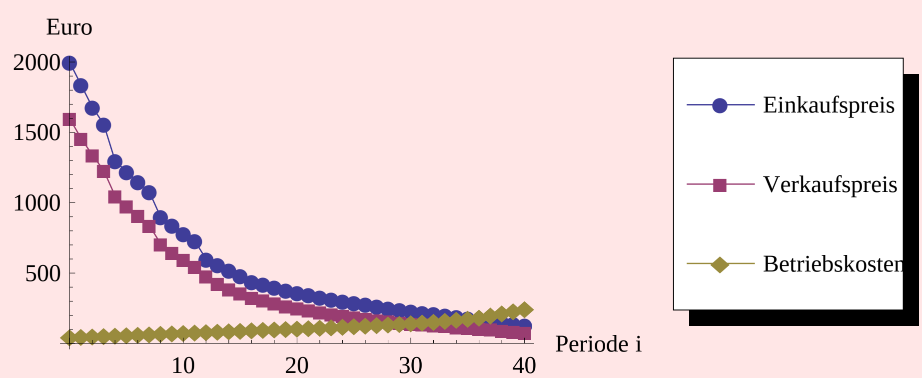
```
Needs["PlotLegends`"]
data = {{Alter i, Ci, Ti, Ei, pi}, {0, 2000, 1600, 50, 1.000},
   {1, 1840, 1460, 53, 0.999}, {2, 1680, 1340, 56, 0.998}, {3, 1560, 1230, 59, 0.997},
   {4, 1300, 1050, 62, 0.996}, {5, 1220, 980, 65, 0.994}, {6, 1150, 910, 68, 0.991},
   {7, 1080, 840, 71, 0.988}, {8, 900, 710, 75, 0.985}, {9, 840, 650, 78, 0.983},
   {10, 780, 600, 81, 0.980}, {11, 730, 550, 84, 0.975}, {12, 600, 480, 87, 0.970},
   {13, 560, 430, 90, 0.965}, {14, 520, 390, 93, 0.960}, {15, 480, 360, 96, 0.955},
   {16, 440, 330, 100, 0.950}, {17, 420, 310, 103, 0.945}, {18, 400, 290, 106, 0.940},
   {19, 380, 270, 109, 0.935}, {20, 360, 255, 112, 0.930}, {21, 345, 240, 115, 0.925},
   {22, 330, 225, 118, 0.919}, {23, 315, 210, 121, 0.910}, {24, 300, 200, 125, 0.900},
   {25, 290, 190, 129, 0.890}, {26, 280, 180, 133, 0.880}, {27, 265, 170, 137, 0.865},
```

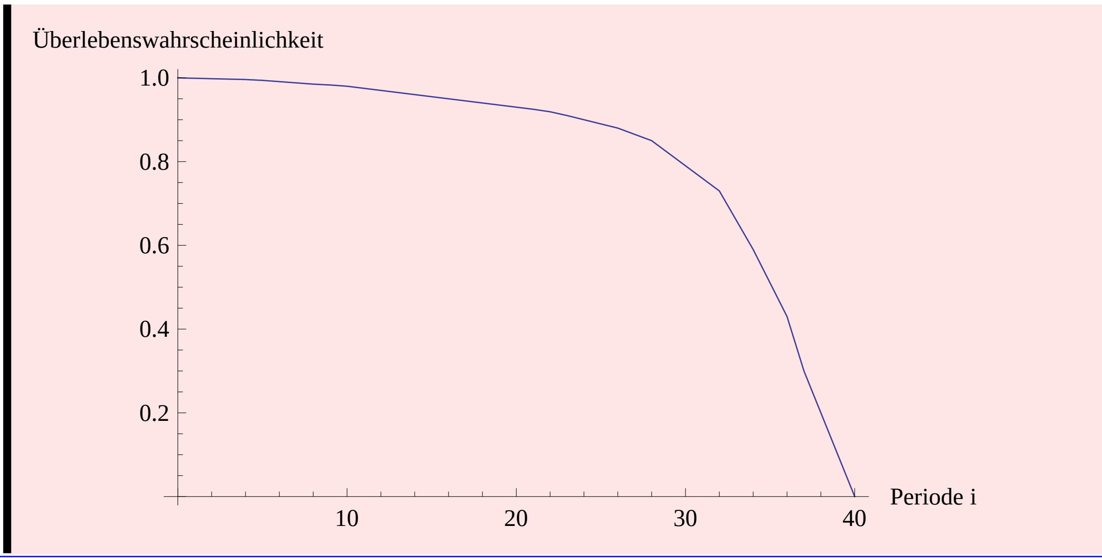
```
   {28, 250, 160, 141, 0.850}, {29, 240, 150, 145, 0.820}, {30, 230, 145, 150, 0.790},
   {31, 220, 140, 155, 0.760}, {32, 210, 135, 160, 0.730}, {33, 200, 130, 167, 0.660},
   {34, 190, 120, 175, 0.590}, {35, 180, 115, 182, 0.510}, {36, 170, 110, 190, 0.430},
   {37, 160, 105, 205, 0.300}, {38, 150, 95, 220, 0.200}, {39, 140, 87, 235, 0.100},
   {40, 130, 80, 250, 0}};
TableForm[Rest[data]];

For[i = 0, i ≤ 40, {c[i] = data[[i + 2]][[2]], t[i] = data[[i + 2]][[3]],
   e[i] = data[[i + 2]][[4]], p[i] = data[[i + 2]][[5]]}; i++];
cvalue[i_] := Module[{liste}, liste = {i, c[i]}];
tvalue[i_] := Module[{liste}, liste = {i, t[i]}];
evalue[i_] := Module[{liste}, liste = {i, e[i]}];
pvalue[i_] := Module[{liste}, liste = {i, p[i]}];
clist = Table[cvalue[i], {i, 0, 40}];
tlist = Table[tvalue[i], {i, 0, 40}];
elist = Table[evalue[i], {i, 0, 40}];
plist = Table[pvalue[i], {i, 0, 40}];
TableForm[data]
ListPlot[{clist, tlist, elist}, Joined → True, AxesLabel → {"Periode i", "Euro"},
 PlotLegend → {"Einkaufspreis", "Verkaufspreis", "Betriebskosten"},
 PlotMarkers → Automatic, LegendPosition → {1.1, -0.4}]

ListPlot[plist, PlotJoined → True,
 AxesLabel → {"Periode i", "Überlebenswahrscheinlichkeit"}]
```

41 Alter	C_{41}	T_{41}	e_{41}	p_{41}
0	2000	1600	50	1.
1	1840	1460	53	0.999
2	1680	1340	56	0.998
3	1560	1230	59	0.997
4	1300	1050	62	0.996
5	1220	980	65	0.994
6	1150	910	68	0.991
7	1080	840	71	0.988
8	900	710	75	0.985
9	840	650	78	0.983
10	780	600	81	0.98
11	730	550	84	0.975
12	600	480	87	0.97
13	560	430	90	0.965
14	520	390	93	0.96
15	480	360	96	0.955
16	440	330	100	0.95
17	420	310	103	0.945
18	400	290	106	0.94
19	380	270	109	0.935
20	360	255	112	0.93
21	345	240	115	0.925
22	330	225	118	0.919
23	315	210	121	0.91
24	300	200	125	0.9
25	290	190	129	0.89
26	280	180	133	0.88
27	265	170	137	0.865
28	250	160	141	0.85
29	240	150	145	0.82
30	230	145	150	0.79
31	220	140	155	0.76
32	210	135	160	0.73
33	200	130	167	0.66
34	190	120	175	0.59
35	180	115	182	0.51
36	170	110	190	0.43
37	160	105	205	0.3
38	150	95	220	0.2
39	140	87	235	0.1
40	130	80	250	0





Das Autoersatzproblem wurde mittels der Politik-Iterationsmethode in sieben Iterationsschritten gelöst. Die Folge der Politiken, Gewinne und Werte zeigt die nächste Tabelle.

Die optimale Politik, die im siebenten Iterationsschritt herauskam, ist folgende: Wenn wir im Besitze eines Autos sind, das älter als 0.5 Jahr aber weniger als 6.5 Jahre alt ist, behalten wir es. Haben wir ein Auto irgend eines anderen Alters, verkaufen wir es und kaufen einen 3 Jahre alten Wagen. Dieses Ergebnis scheint sich sehr gut mit unserer intuitiven Meinung über das Verhalten des Autobesitzers zu decken. Man beachte, dass ein 3 oder 6 Monate alter Wagen gegen einen 3 Jahre alten Wagen eingetauscht werden sollte. Liegt das Alter unseres Wagens jedoch zwischen 6 Monaten und 6| Jahren, sollten wir ihn behalten. Diese Richtlinien ermöglichen uns, den «3 bis 6.5 — Zyklus» einzuführen; ist der Zyklus einmal eingeführt, so bewegt sich das Alter unseres Autos immer zwischen 3 und 6.5 Jahren[1].

5.2.1 Bemerkung: Wenn alle Autobesitzer diese Strategie befolgen würden, würde dies natürlich für die Automobilindustrie ein Chaos bedeuten. Woher würden wir alle die 3 Jahre alten Autos nehmen? Nach Ökonomischen Gesetzen würde sich für solche Autos der Preis erhöhen und zwar so stark, dass die « 3 bis 6.5» − Politik nicht mehr optimal wäre. In der obigen Analyse muss angenommen werden, dass es genug Leute gibt, die aus psychologischen Gründen Autos kaufen, und die sogenannten «vernünftigen» Käufer nur von unwesentlicher Bedeutung sind.

Mit Befriedigung stellen wir fest, dass das Programm nach jedem Iterationsschritt verlangt, dass wir, wenn wir einen Austausch vornehmen wollen, einen Wagen kaufen müssen, dessen Alter unabhängig vom Alter unseres jetzigen Wagens ist. Dies würde auch die Logik in einer solchen Situation vorschreiben.

Wenn wir uns an unsere optimale Politik halten, dann werden wir einen Wagen behalten, bis er 6.5| Jahre alt ist, und dann einen 3 Jahre alten kaufen. Nehmen wir jedoch an, unser Wagen sei 4 Jahre alt und ein Freund biete uns seinen 1 Jahr alten Wagen zum Tausch gegen unseren zu einem Betrag X an. Sollen wir das Angebot annehmen? Um diese Frage zu beantworten, müssen wir unsere Werte ansehen.

In jeder der lterationsschritte wurde der Wert des Zustandes 40, aus rechnerischem Grunde, gleich Null gesetzt. Tabelle zeigt auch die Werte bei der besten Politik, wenn der Wert im Zustand 40 gleich 80 Dollars, dem Verkaufswert eines Wagens bei diesem Alter, gesetzt wird. Wird das ausgeführt, so stellt für jeden, der die optimale Politik befolgt, jedes v_i, $i \in E$, den Wert eines Wagens vom Alter i dar. Um die soeben gestellte Frage zu beantworten, müssen wir den Wert eines 1-Jahr alten Wagens,

$$v_4 = 1152 \ \text{Dollars},$$

mit dem Wert eines 4-Jahre alten Wagens,

$v_{16} = 422$ Dollars,

vergleichen. Wenn der verlangte Preis X kleiner als

$v_4 - v_{16} = 730$ Dollars

ist, können wir den Handel eingehen, andernfalls nicht. Es ist natürlich nicht nötig, v_{40} von Null verschieden anzunehmen, um dieses Problem zu lösen; doch der Wert v_{40} = 80 Dollars gibt den übrigen Werten eine sowohl absolute als auch eine relative Bedeutung.

Die Zahlen in der Entscheidungskolonne bedeuten den Tausch gegen einen Wagen mit einem Alter von soviel Perioden. Ein K heisst: Behalte den Wagen. Die Werte und Gewinne sind in Dollars angegeben.

```
numstates = 40;
For[i = 0, i ≤ numstates, {c[i] = data[[i + 2]][[2]], t[i] = data[[i + 2]][[3]],
   e[i] = data[[i + 2]][[4]], p[i] = data[[i + 2]][[5]]}; i++];
```

```
initialpolicy[state_] := Module[{result},
  max = -10 000;
  cost[1][state] = -e[state];
  For[k = 2, k ≤ numstates + 1, cost[k][state] = t[state] - c[k - 2] - e[k - 2]; k++];
  If[cost[1][state] > max, {max = cost[1][state], policy = 1}];
  For[k = 2, k ≤ numstates + 1,
   If[cost[k][state] > max, {max = cost[k][state], policy = k}]; k++];
  result = policy
 ]
Table[initialpolicy[state], {state, 0, 40}]
```

```
{38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38,
 38, 38, 38, 38, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
```

```
improvedpolicy[state_] := Module[{result},
  max = -10 000;
  If[cost[1][state] + p[state] v[state + 1] + (1 - p[state]) * v[40] > max,
   {max = cost[1][state] + p[state] v[state + 1] + (1 - p[state]) * v[40], policy = 1}];
  For[k = 2, k ≤ numstates + 1,
   If[cost[k][state] + p[k - 2] v[k - 1] + (1 - p[k - 2]) v[40] > max,
    {max = cost[k][state] + p[k - 2] v[k - 1] + (1 - p[k - 2]) v[40], policy = k}]; k++];
  result = policy
 ]
```

```
model := Module[{result},
  eps = 0.001;
  flag = 0;
  iter1 = 0;
  iter2 = 0;
  For[j = 0, j ≤ numstates, {d[j] = initialpolicy[j], v1[j] = 0, v[j] = 0}; j ++];
  While[flag ≠ 1,
   {delta = 1.0,
    iter1 = iter1 + 1,
    iter2 = 0,
    While[delta > eps,
     {iter2 = iter2 + 1,
      max1 = 0,
      If[d[numstates] == 1, g = -e[numstates]],
      If[d[numstates] > 1,
       g = t[numstates] - c[d[numstates] - 2] - e[d[numstates] - 2] +
         p[d[numstates] - 2] v[d[numstates] - 1]],
      For[i = 0, i ≤ numstates,
        { v[numstates] = 0,
         If[((d[i] == 1) && (i < numstates)), v1[i] = -e[i] + p[i] v[i + 1] - g],
         If[((d[i] == 1) && (i == numstates)), v1[i] = -e[i] - g],
        If[d[i] > 1, v1[i] = t[i] - c[d[i] - 2] - e[d[i] - 2] + p[d[i] - 2] v[d[i] - 1] -
           g],
        If[d[numstates] > 1, v1[numstates] = 0],
        deltav[i] = Abs[v1[i] - v[i]],
        If[deltav[i] ≥ max1, max1 = deltav[i]]
        }; i ++]
       For[i = 0, i ≤ numstates, v[i] = v1[i]; i ++],
      delta = max1
     }],
    flag = 1,
    For[i = 0, i ≤ numstates, {If[d[i] == 1, dd[i] = "K", dd[i] = d[i] - 2]}; i ++];
    tables[iter1] = Table[{state, dd[state], v[state]}, {state, 0, 40}],
    gvalue[iter1] = {iter1, Abs[g]},
    For[i = 0, i ≤ numstates,
     { d1[i] = improvedpolicy[i],
       If[d1[i] ≠ d[i], flag = 0],
      d[i] = d1[i]
     }; i ++]
   }
  ];
  For[i = 0, i ≤ numstates, {If[d[i] == 1, dd[i] = "K", dd[i] = d[i] - 2]}; i ++];
  result = {Table[{state, dd[state], v[state]}, {state, 0, 40}], g, iter1, iter2}
 ]
```

```
Print["Mittlere Gewinne pro Zeiteinheit g=", model[[2]]]
Print["Anzahl der notwendigen Iterationsschritten = ", model[[3]]]
Print["Optimale Entscheidung, Iteration iter1=7"]
TableForm[model[[1]],
 TableHeadings ->
  {None, {"Zustand i", "Optimale Entscheidung d(i)",
    "Relative Wertfunktion Vi"}}]
Print["Entscheidung, Iteration iter1=1"]
TableForm[tables[1],
 TableHeadings ->
  {None, {"Zustand i", "Entscheidung d(i)", "Relative Wertfunktion Vi"}}]
Print["Entscheidung, Iteration iter1=2"]
TableForm[tables[2],
 TableHeadings ->
  {None, {"Zustand i", "Entscheidung d(i)", "Relative Wertfunktion Vi"}}]
Print["Entscheidung, Iteration iter1=3"]
TableForm[tables[3],
 TableHeadings ->
  {None, {"Zustand i", "Entscheidung d(i)", "Relative Wertfunktion Vi"}}]
Print["Entscheidung, Iteration iter1=4"]
TableForm[tables[4],
 TableHeadings ->
  {None, {"Zustand i", "Entscheidung d(i)", "Relative Wertfunktion Vi"}}]
Print["Entscheidung, Iteration iter1=5"]
TableForm[tables[5],
 TableHeadings ->
  {None, {"Zustand i", "Entscheidung d(i)", "Relative Wertfunktion Vi"}}]
Print["Entscheidung, Iteration iter1=6"]
TableForm[tables[6],
 TableHeadings ->
  {None, {"Zustand i", "Entscheidung d(i)", "Relative Wertfunktion Vi"}}]
```

```
Mittlere Gewinne pro Zeiteinheit g=-150.946
```

```
Anzahl der notwendigen Iterationsschritten = 7
```

```
Optimale Entscheidung, Iteration iter1=7
```

Zustand i	Optimale Entscheidung d(i)	Relative Wertfunktion V_i
0	12	1520.
1	12	1380.
2	12	1260.
3	K	1160.66
4	K	1071.93
5	K	986.933
6	K	906.426
7	K	830.958
8	K	760.134
9	K	694.607
10	K	632.413
11	K	573.946
12	K	520.
13	K	470.159
14	K	424.055
15	K	381.364
16	K	341.8
17	K	306.162
18	K	273.244
19	K	242.87
20	K	214.892
21	K	189.19
22	K	165.669
23	K	144.421
24	K	125.797
25	K	110.946
26	12	100.
27	12	90.
28	12	80.
29	12	70.
30	12	65.
31	12	60.
32	12	55.
33	12	50.
34	12	40.
35	12	35.
36	12	30.
37	12	25.
38	12	15.
39	12	7.
40	12	0

Entscheidung, Iteration iter1=1

Zustand i	Entscheidung d(i)	Relative Wertfunktion V_i
0	36	1513.61
1	36	1373.61
2	36	1253.61
3	36	1143.61
4	36	963.607
5	36	893.607
6	36	823.607
7	36	753.607
8	36	623.607
9	36	563.607
10	36	513.607
11	36	463.607
12	36	393.607
13	36	343.607
14	36	303.607
15	36	273.607
16	36	243.607
17	36	223.607
18	36	203.607
19	36	183.607
20	36	168.607
21	K	875.927
22	K	801.002
23	K	727.968
24	K	658.206
25	K	592.451
26	K	529.721
27	K	469.001
28	K	411.562
29	K	355.955
30	K	306.043
31	K	260.813
32	K	218.176
33	K	175.583
34	K	140.277
35	K	110.64
36	K	83.607
37	K	54.9
38	K	33.
39	K	15
40	K	0

Entscheidung, Iteration iter1=2

Zustand i	Entscheidung d(i)	Relative Wertfunktion v_i
0	20	1520.
1	20	1380.
2	20	1260.
3	20	1150.
4	20	970.
5	20	900.
6	20	830.
7	20	760.
8	20	630.
9	20	570.
10	20	520.
11	20	470.
12	20	400.
13	20	350.
14	20	310.
15	20	280.
16	20	250.
17	20	230.
18	20	210.
19	20	190.
20	K	280.
21	K	213.019
22	20	145.
23	20	130.
24	20	120.
25	20	110.
26	20	100.
27	20	90.
28	20	80.
29	20	70.
30	20	65.
31	20	60.
32	20	55.
33	20	50.
34	20	40.
35	20	35.
36	20	30.
37	20	25.
38	20	15.
39	20	7.
40	20	0

Entscheidung, Iteration iter1=3

Zustand i	Entscheidung d(i)	Relative Wertfunktion v_i
0	19	1520.
1	19	1380.
2	19	1260.
3	19	1150.
4	K	1036.63
5	K	939.951
6	K	847.599
7	19	760.
8	K	695.441
9	K	617.262
10	K	542.039
11	19	470.
12	19	400.
13	K	574.998
14	K	520.787
15	K	470.153
16	K	422.738
17	K	379.263
18	K	338.439
19	K	300.
20	K	263.702
21	K	229.314
22	K	196.622
23	K	165.598
24	K	136.439
25	19	110.
26	19	100.
27	19	90.
28	19	80.
29	19	70.
30	19	65.
31	19	60.
32	19	55.
33	19	50.
34	19	40.
35	19	35.
36	19	30.
37	19	25.
38	19	15.
39	19	7.
40	19	0

Entscheidung, Iteration iter1=4

Zustand i	Entscheidung d(i)	Relative Wertfunktion V_i
0	12	1520.
1	12	1380.
2	12	1260.
3	12	1150.
4	12	970.
5	12	900.
6	12	830.
7	12	760.
8	12	630.
9	12	570.
10	12	520.
11	12	470.
12	K	520.
13	K	463.842
14	K	411.159
15	K	361.548
16	K	314.633
17	K	271.116
18	K	229.674
19	12	190.
20	12	175.
21	12	160.
22	12	145.
23	12	130.
24	12	120.
25	12	110.
26	12	100.
27	12	90.
28	12	80.
29	12	70.
30	12	65.
31	12	60.
32	12	55.
33	12	50.
34	12	40.
35	12	35.
36	12	30.
37	12	25.
38	12	15.
39	12	7.
40	12	0

Entscheidung, Iteration iter1=5

Zustand i	Entscheidung d(i)	Relative Wertfunktion v_i
0	12	1520.
1	12	1380.
2	12	1260.
3	12	1150.
4	K	1002.62
5	K	917.243
6	K	836.21
7	12	760.
8	K	760.542
9	K	694.915
10	K	632.619
11	K	574.05
12	K	520.
13	K	470.052
14	K	423.837
15	K	381.028
16	K	341.339
17	K	305.568
18	K	272.506
19	K	241.976
20	K	213.824
21	K	187.929
22	K	164.193
23	K	142.702
24	K	123.793
25	K	108.603
26	K	97.2499
27	12	90.
28	12	80.
29	12	70.
30	12	65.
31	12	60.
32	12	55.
33	12	50.
34	12	40.
35	12	35.
36	12	30.
37	12	25.
38	12	15.
39	12	7.
40	12	0

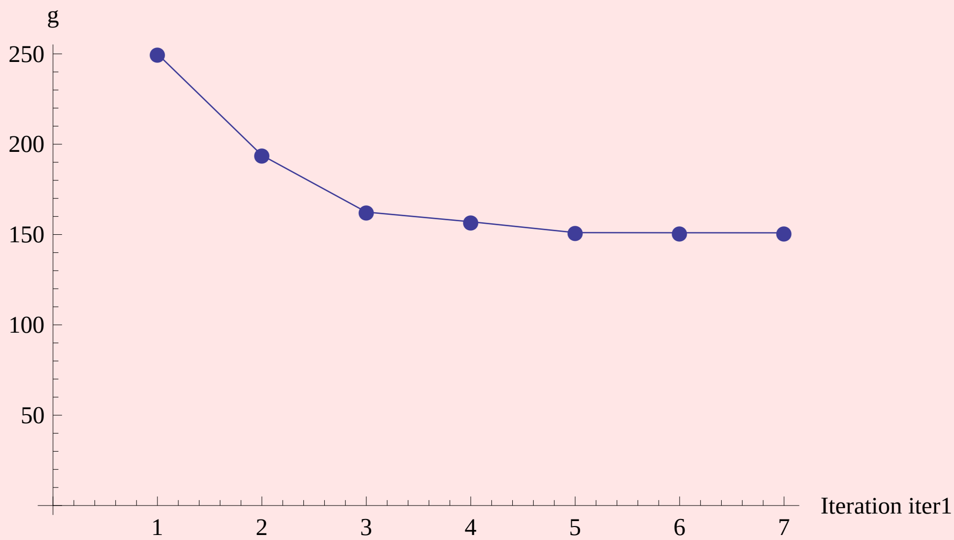
Entscheidung, Iteration iter1=6

Zustand i	Entscheidung d(i)	Relative Wertfunktion v_i
0	12	1520.
1	12	1380.
2	12	1260.
3	12	1150.
4	K	1072.26
5	K	987.218
6	K	906.67
7	K	831.162
8	K	760.298
9	K	694.731
10	K	632.496
11	K	573.988
12	K	520.
13	K	470.116
14	K	423.967
15	K	381.229
16	K	341.614
17	K	305.921
18	K	272.945
19	K	242.508
20	K	214.46
21	K	188.68
22	K	165.072
23	K	143.727
24	K	124.988
25	12	110.
26	12	100.
27	12	90.
28	12	80.
29	12	70.
30	12	65.
31	12	60.
32	12	55.
33	12	50.
34	12	40.
35	12	35.
36	12	30.
37	12	25.
38	12	15.
39	12	7.
40	12	0

Wird die optimale Politik befolgt, so betragen die jährlichen Fahrkosten ca. 604 Dollars (4 x 150.95 Dollars). Befolgt man die Politik der Maximierung der unmittelbaren Erlöse, wie im ersten Iterationsschritt, belaufen sich die jährlichen Kosten auf 1000 Dollars. Daraus geht hervor, dass eine Politik, welche die künftigen Erlöse maximiert, gegenüber einer, die nur den unmittelbaren Erlös maximiert, eine jährliche Ersparnis von fast 400 Dollars einbringt. Die Abnahme der Kosten pro Periode im Verlauf der Iteration zeigt nächste Figure. Der Gewinn nähert sich dem optimalen Wert ungefähr exponentiell.

```
liste = Table[gvalue[iter1], {iter1, 1, 7}]
ListPlot[liste, PlotJoined → True, AxesLabel → {"Iteration iter1", "g"},
 AxesOrigin → {0, 0}, PlotMarkers → Automatic]
```

```
{{1, 250}, {2, 193.894}, {3, 162.439},
 {4, 157.074}, {5, 151.05}, {6, 150.988}, {7, 150.946}}
```



Man beachte, dass die Gewinne in den letzten drei Iterationsschritten so nahe beieinanderliegen, dass in der Praxis die entsprechenden Politiken als gleichwertig betrachtet werden können.

Die Tatsache, dass der Kauf eines 3 Jahre alten Wagens der beste ist, geht schon aus dem Ilerationsschritt 4 hervor. Die Diskontinuität beim Erscheinen eines neuen Modells, die nach 3 Jahren noch feststellbar ist, ist ohne Zweifel verantwortlich für diese besondere Wahl.

Das in diesem Abschnitt beschriebene Ersatzproblem ist typisch für eine grosse Klasse von Ersatzproblemen in der Industrie. Die Einreibung dieser Probleme in den Rahmen der Politik-Iterationsmethode verlangt nur ein gründliches Kennen der Eigenheiten dieser Probleme und einige Vorsicht in der Wahl einer passenden Formulierung.