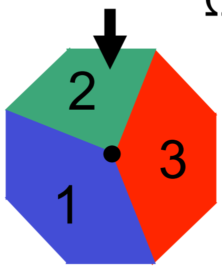
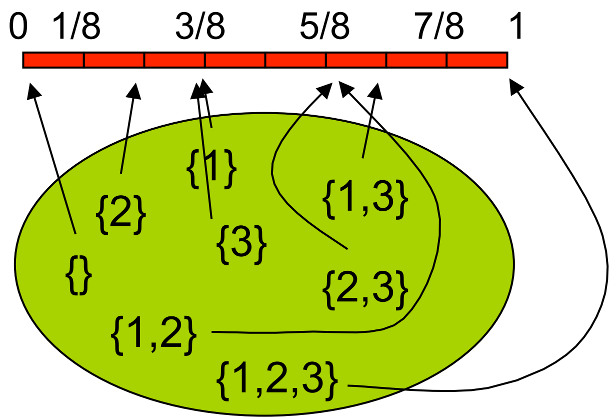
Voraussetzungen

Ein **Wahrscheinlichkeitsraum** ist ein Tripel (Ω,Σ,*P*). Ω bezeichnet die Menge aller Elementarereignissee, Σ eine Sigma-Algebra und *P* ein Wahrscheinlichkeitsmaß auf Σ. Aus P ergeben sich die Wahrscheinlichkeitsverteilungen für die Ereignisse aus Σ.



Ω = {1, 2, 3}



Beispiel: Ein Glücksrad mit Ergebnismenge Ω, Ereignisraum Σ (hier die Potenzmenge von Ω) und Wahrscheinlichkeitsmaß *P*.

Stochastische Problemstellungen

Im folgenden sind alle Wahrscheinlichkeitsverteilungen diskret und endlich, also sehr einfach. Kombinierte Gesamtverteilungen sind zum Teil nur implizit gegeben.

Es sollen folgende Problemstellungen betrachtet werden:

- Stochastic Satisfibility (SSAT)
- Dynamic Graph Reliability (DGR)
- Multi-Stage Stochastic Programming
- Optimal Control mit Dynamic Programming (später, Ende VL)

SSAT [Papadimitriou 1985]

Problem SSAT: Gegeben sei eine boolsche Formel C in CNF, mit den Variablen $x_1,\ldots,x_n$ (n gerade) und eine rationale Zahl $b \in [0,1]$.

Ist die Wahrscheinlichkeit für

$\exists x_1 \mathfrak{R} x_2 \exists x_3 \ldots \mathfrak{R} x_n : C(x_1, x_2, \ldots, x_n) = \text{TRUE}$

größer oder gleich 1/2?

$\mathfrak{R}$ ist ein stochastischer Quantor, der so quantifizierte Variablen mit Wahrscheinlichkeit ½ TRUE bzw. FALSE setzt.

Satz: SSAT ist PSPACE complete

SSAT

Beweis: Wir nehmen eine QSAT-Instanz, interpretieren deren Allquantifizierte Variablen als Random-Variablen, die mit Wahrscheinlichkeit ½ auf true oder false gesetzt werden und fügen eine Variable x_0 und eine Klausel (x_0) hinzu:

Ist die Wahrscheinlichkeit für

$\Re x_0 \exists x_1 \Re x_2 \exists x_3 \ldots \Re x_n : (x_0) \wedge C(x_1, x_2, \ldots, x_n) = \text{TRUE}$

größer oder gleich 1/2?

Mit Wahrscheinlichkeit ½ ist die CNF wegen x_0 false. Wenn also die SSAT-Antwort auf unsere Konstruktion "ja" ist, kann das nur daran liegen, dass es eine Gewinnstrategie für den Existenzspieler im ursprünglichen QSAT Problem gibt.

Dynamic Graph Reliability (DGR)

Problem DGR: Gegeben sei ein gerichteter Graph ohne Kreise, also ein DAG (Directed Acyclic Graph), G = (V,E). Zwei seiner Knoten s (source) und t (sink) seien speziell ausgezeichnet.

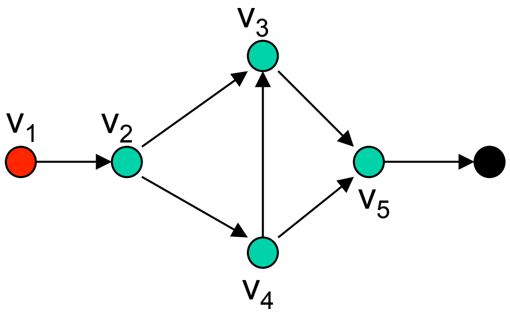
Frage: Wie ist die Startegie, die die Wahrscheinlichkeit das Ziel t zu erreichen maximiert, wenn

- man bei Knoten s startet

- p(e,v) die Wahrscheinlichkeit angibt, dass Kante e ausfällt, wenn wir Knoten v betreten, und der Graph somit während unserer Wanderung abhängig von unseren Entscheidungen und vom Zufall zerfällt?

Dynamic Graph Reliability (DGR)

Beispiel A:

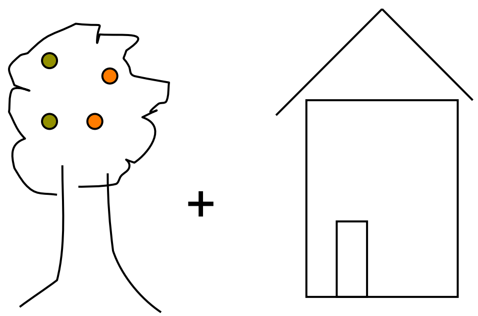


$p((v_3, v_5), v_3) = 0.5$ → 0.5
$p((v_4, v_5), v_3) = 0.9$ → 0.1
$p((v_3, v_5), v_4) = 0.5$ → 0.5
$p((v_4, v_5), v_4) = 0.6$ → 0.4

W., dass man ans Ziel kommt, wenn man über v_4 geht:

0.4 + 0.5 * 0.5 = 0.65

Beispiel B:



Frage: Baum fällen? Dann kann er nicht mehr umfallen. Bringt aber später auch keine Ernte.

-> Entscheidung beeinflusst Wahrscheinlichkeiten, deren Auswirkungen erst spät spürbar werden.

Dynamic Graph Reliability (DGR)

Hilfsproblem SSAT': Gegeben ist eine boolsche 3-SAT Formel mit

alternierendem ∃ - $\mathfrak{R}$' - Quantor-Präfix, sowie eine rationale Zahl b $\in$ [0,1].

In dieser Version:

- Die Wahrscheinlichkeit, dass eine der Belegungen true oder false für eine Randomvariable *unverfügbar* wird ist ½.
- Eine Existenzstrategie wählt eine Belegung für die Existenzvariable x_1, und dann wird bestimmt, welche Belegungen für x_2 verfügbar sind. Mit Wahrscheinlichkeit ¼ ist keine verfügbar, mit Wahrscheinlichkeit ¼ sind beide verfügbar, mit Wahrscheinlichkeit ¼ ist nur true verfügbar und mit Wahrscheinlichkeit ¼ nur false.
- Dann wählt die Strategie eine der verfügbaren Belegungen für x_2 etc. Wenn an einer $\mathfrak{R}$'-quantifizierten Variable keine Belegung verfügbar ist, hat der Existenzspieler sein Spiel verloren.

Dynamic Graph Reliability (DGR)

Hilfsproblem SSAT':

- **Die Frage:**

  Ist die Wahrscheinlichkeit für

  $$\exists\, x_1\, \mathfrak{R}'\, x_2\, \exists\, x_3 \ldots \mathfrak{R}'\, x_n : C(x_1, x_2, \ldots, x_n) = \text{TRUE}$$

  größer oder gleich b?

  ist PSPACE-vollständig.

Härte: Wir nehmen eine QSAT Formel und interpretieren die n vielen Allquantoren als $\mathfrak{R}'$-Quantoren; b setzen wir auf $(3/4)^n$. Mit Wahrscheinlichkeit $1-(3/4)^n$ geht ein Spiel für den Existenzspieler dadurch verloren, dass die SAT-Formel C "nicht erreicht" wird. Falls es nun eine Gewinnstrategie für den Existenzspieler gibt, die immer gewinnt, ist die Wahrscheinlichkeit, das C erfüllt wird gerade gleich $(3/4)^n$. Sonst ist sie kleiner.

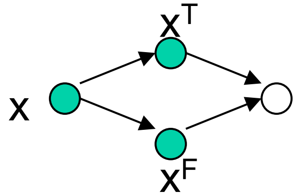
Dynamic Graph Reliability (DGR)

DGR ist PSPACE-schwer

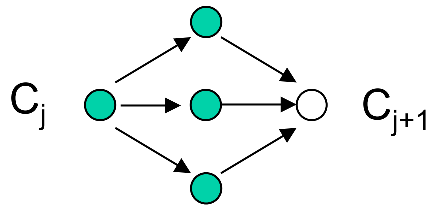
Beweis:

Sei (X,C,b) eine SSAT' Instanz mit $X=\{x_1,..,x_n\}$ der Variablenmenge, den Klauseln $C=\{C_1,..,C_m\}$ und der Schranke b.

- Für jede Variable x habe G 3 Knoten: x, x^T, x^F und die Kanten (x, x^T), (x, x^F)



- von x^T und x^F führen Kanten zum nächsten Variablenknoten.
- Für jede Klausel C_j haben wir 4 Knoten C_j, C_j^1, C_j^2, C_j^3. C_j^1, C_j^2, C_j^3 sind jeweils mit C_j und C_{j+1} verbunden.

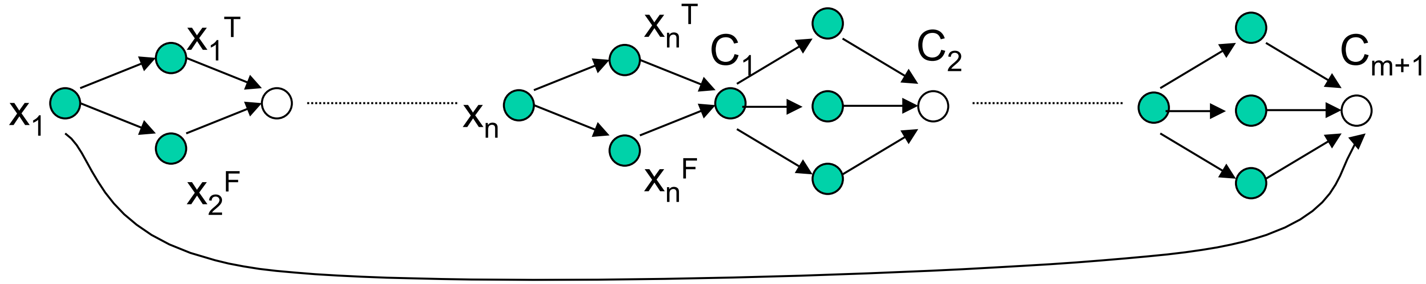


Dynamic Graph Reliability (DGR)

DGR ist PSPACE-schwer

Beweis Forts:

- Zusätzlich gibt es Kanten von x_n^T und x_n^F nach C_1 und eine Kante von x_1 nach C_{m+1}.



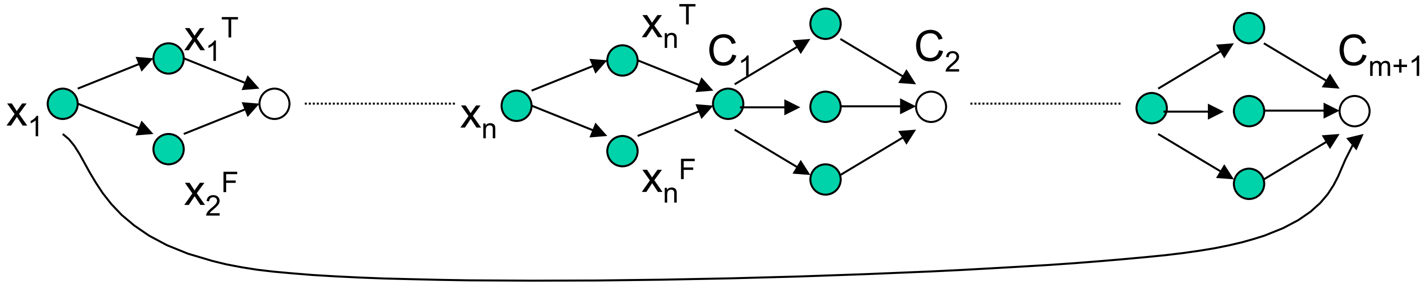
Definiere nun die Wahrscheinlichkeiten p(e,v):

- für jede stochastische Variable setzen wir $p((x,x^T),x) = p((x,x^F),x) = 0.5$.
- für alle Variablen x setzen wir: Wenn $e=(C_j,C_j^{i\in\{1,2,3\}})$ eine Kante ist, die zum Literal x im Klauselpart gehört: $p(e,x^F) = 1$. Wenn $e=(C_j,C_j^{i\in\{1,2,3\}})$ eine Kante ist, die zum Literal *not* x im Klauselpart gehört: $p(e,x^T) = 1$.
- $p((x_1,C_{m+1}),x_1) := 0.5 \cdot 1/(1-b) + 2^{-3n}$
- $p(e,v) = 0$ für alle anderen Paare (e,v).

Dynamic Graph Reliability (DGR)

DGR ist PSPACE-schwer

Beweis Forts:



Beste Strategie:
Wenn (x_1,C_{m+1}) existiert gehen wir dort entlang. Wir erreichen also das Ziel mit Wahrscheinlichkeit 1- $p((x_1,C_{m+1}),x_1)$ plus etwas Mehr, abhängig von der der Wahrscheinlichkeit, dass die Formel C erfüllt wird. Die Gesamtwahrscheinlichkeit, das Ziel zu erreichen ist über 0.5, g.d.w. die Wahrscheinlichkeit, dass die Formel erfüllbar wird, größer oder gleich b ist.

2-stufige Programme , Beispiel

Farmer – Problem

- 500 ha Land
- mindestens 200t Weizen und 240t Mais werden fürs Vieh benötigt.
- Verkaufspreis von Weizen/Mais = 170/150 Euro/t, für nicht benötigtes Getreide
- Einkaufspreis von Weizen/Mais = 238/210 Euro/t
- Verkaufspreis von Zuckerrüben: 36 Euro/t unter 6000t; 10 Euro ab mehr als 6000t (wegen Agrarregeln in der EU)
- Planzkosten Weizen/Mais/Zuckerrüben = 150/230/260 Euro/ha
- Ertrag in t/ha für Weizen/Mais/Zuckerrüben = 2.5/3/20

x_1 = Land für Weizen; x_2 = Land für Mais; x_3 = Land für Zuckerrüben;
w_1 = verkaufter Weizen in Tonnen; w_2 = verkaufter Mais in Tonnen; w_3 = verkaufte Zuckerüben zu gutem Preis; w_4 = verkaufte Zuckerüben zu schlechtem Preis;
y_1 = Tonnen gekaufter Weizen; y_2 = Tonnen gekaufter Mais

Stochastic Programming

2-stufige Programme , Beispiel

$$\min\ 150x_1 + 230x_2 + 260x_3 + 238y_1 - 170w_1 + 210y_2 - 150w_2 - 36w_3 - 10w_4$$

s.t.

$$x_1+x_2+x_3 \leq 500$$
$$2.5x_1 + y_1 - w_1 \geq 200$$
$$3x_2 + y_2 - w_2 \geq 240$$
$$w_3+w_4 \leq 20x_3$$
$$w_3 \leq 6000$$
$$x_1,x_2,x_3,y_1,y_2,w_1,w_2,w_3,w_4 \geq 0$$

Stochastic Programming

TECHNISCHE UNIVERSITÄT DARMSTADT

2-stufige Programme , Beispiel

Pflanzenart	Weizen	Mais	Zuckerrüben
Anzahl ha	120	80	300
Ertrag in Tonnen	300	240	6000
Verkauf in Tonnen	100	-	6000
Einkauf in Tonnen	-	-	-

Gewinn bei optimaler Lösung: 118.600 Euro

Unser Farmer ist verunsichert: Ertrag hängt doch sehr vom Wetter ab. Annahme, der Ertrag je ha erhöht / erniedrigt sich um 20%:

Pflanzenart	Weizen	Mais	Zuckerr
Anzahl ha	183.33	66.67	250
Ertrag in T.	550	240	6000
Verkauf in T.	350	-	6000
Einkauf in T.	-	-	-

max. Gewinn bei +20%: 167.667 Euro

Pflanzenart	Weizen	Mais	Zuckerr.
Anzahl ha	100	25	375
Ertrag in T.	200	60	6000
Verkauf in T.	-	-	6000
Einkauf in T.	-	180	-

max. Gewinn bei -20%: 59.950 Euro

Stochastic Programming

2-stufige Programme , Beispiel

Unser Farmer würde gerne flexibel reagieren können. Entscheidungen für das Land (x_1,x_2,x_3) müssen sofort gefällt werden, aber die anderen Entscheidungen hängen von den Erträgen je ha ab.

→ Bilde 3 Szenarios mit index 1,2,3. Jedes Szenario habe Eintrittswahrscheinlichkeit 1/3.

min $150x_1 + 230x_2 + 260x_3$

$- 1/3*(170w_{11}-238y_{11}+150w_{21}-210y_{21}+36w_{31}+10w_{41})$ Szenario 1

$- 1/3*(170w_{12}-238y_{12}+150w_{22}-210y_{22}+36w_{32}+10w_{42})$ Szenario 2

$- 1/3*(170w_{13}-238y_{13}+150w_{23}-210y_{23}+36w_{33}+10w_{43})$ Szenario 3

s.t. $x_1 + x_2 + x_3 \le 500$,

$3x_1 + y_{11} - w_{11} \ge 200$, $3.6x_2 + y_{21} - w_{21} \ge 240$, $w_{31} + w_{41} \le 24x_3$, $w_{31} \le 6000$,

$2.5x_1 + y_{12} - w_{12} \ge 200$, $3x_2 + y_{22} - w_{22} \ge 240$, $w_{32} + w_{42} \le 20x_3$, $w_{32} \le 6000$,

$2x_1 + y_{13} - w_{13} \ge 200$, $2.4x_2 + y_{23} - w_{23} \ge 240$, $w_{33} + w_{43} \le 16x_3$, $w_{33} \le 6000$,

$x, y, w \ge 0$

Stochastic Programming

TECHNISCHE
UNIVERSITÄT
DARMSTADT

2-stufige Programme, Beispiel

	Pflanzenart	Weizen	Mais	Zuckerrüben
1. Stufe	Anzahl ha	170	80	250
s=1 (+20%)	Ertrag in Tonnen	510	288	6000
	Verkauf in Tonnen	310	48	6000
	Einkauf in Tonnen	-	-	-
s=2	Ertrag in Tonnen	425	240	5000
	Verkauf in Tonnen	225	-	5000
	Einkauf in Tonnen	-	-	-
s=3 (-20%)	Ertrag in Tonnen	340	192	4000
	Verkauf in Tonnen	140	-	4000
	Einkauf in Tonnen	-	48	-

Erwarteter Gewinn bei optimaler Lösung: 108.390 Euro

Durchschnittlicher Gewinn der optimalen Einzellösungen: 115.406 Euro
Differenz = 7016 Euro ist „erwarteter Wert von perfekter Information“

Stochastic Programming

mehrstufige Programme, formal

$$\min z = c^1 x^1 + E_{\xi^2}\left[\min c^2(\omega)x^2(\omega^2) + \ldots + E_{\xi^H}\left[\min c^H(\omega)x^H(\omega^H)\right]\ldots\right]$$

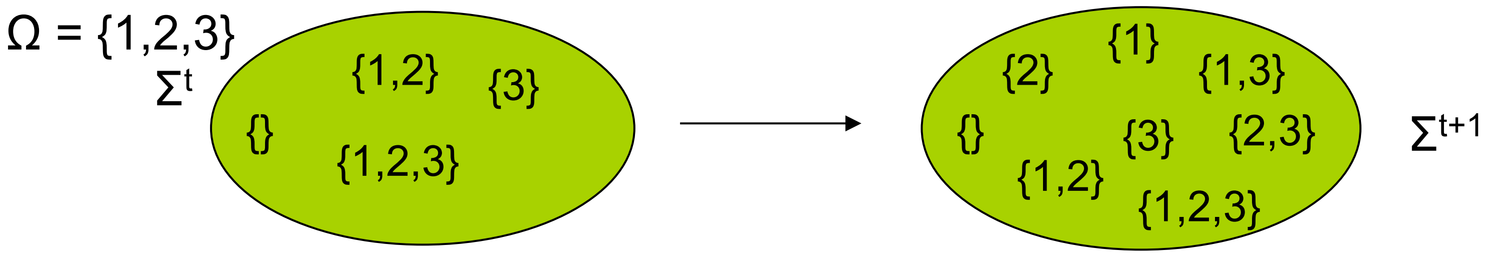
$$\begin{aligned}
s.t.\quad & W^1 x^1 = h^1, \\
& T^1(\omega)x^1 + W^2 x^2(\omega^2) = h^2(\omega), \\
& \vdots \\
& T^{H-1}(\omega)x^{H-1}(\omega^{H-1}) + W^H x^H(\omega^H) = h^H(\omega), \\
& x^1 \geq 0; x^t(\omega^t) \geq 0, t = 2,\ldots,H
\end{aligned}$$

Stochastic Programming

TECHNISCHE UNIVERSITÄT DARMSTADT

mehrstufige Programme, formal

c_1 Vektor in $\mathbb{Q}^{n_1}$, h_1 Vektor in $\mathbb{Q}^{m_1}$, $\xi^t(\omega) = (c^t(\omega), h^t(\omega), T_1^{t-1}(\omega), \ldots, T_{m_t}^{t-1}(\omega))$ ist ein Zufallsvektor, definiert auf (Ω, Σ^t, P), für alle $t = 2, \ldots, H$. Dabei ist $\Sigma^t \subseteq \Sigma^{t+1}$. W^t ist eine feste Matrix und spiegelt den festen Recourse wider.



Entscheidungen x hängen von der Historie bis zum Zeitpunkt t ab, die Historie bezeichnen wir mit ω^t

Stochastic Programming

TECHNISCHE UNIVERSITÄT DARMSTADT

2-stufige Programme , Beispiel, andere Schreibweise

$$\min 150x_1 + 230x_2 + 260x_3 + E_\xi(Q(x,\xi)$$
$$\text{s.t. } x_1+x_2+x_3 \leq 500$$

und

$$Q(x,s) = \min 238y_1 - 170w_1 + 210y_2 - 150w_2 - 36w_3 - 10w_4$$
$$\text{s.t.}$$
$$t_1(s)x_1 + y_1 - w_1 \geq 200$$
$$t_2(s)x_2 + y_2 - w_2 \geq 240$$
$$w_3 + w_4 \leq t_3(s)x_3$$
$$w_3 \leq 6000$$
$$x_1,x_2,x_3,y_1,y_2,w_1,w_2,w_3,w_4 \geq 0$$

- $t_i(s)$ repräsentiert den Ertrag von dem Getreide
- der Zufallsvektor $\xi=(t_1,t_2,t_3)$ besteht aus 3 Werten.
- ξ kann 3 verschiedene Werte annehmen, sagen wir ξ_1,ξ_2,ξ_3.
- ξ_1,ξ_2,ξ_3 repräsentieren $(t_1(1), t_2(1), t_3(1))$, $(t_1(2), t_2(2), t_3(2))$, $(t_1(3), t_2(3), t_3(3))$,

Stochastic Programming

TECHNISCHE UNIVERSITÄT DARMSTADT

mehrstufige Programme, deterministisches Äquivalent, Version 1

als dynamisches Programm:

$$Q^H(x^{H-1}, \xi^H(\omega)) = \min c^H(\omega) x^H(\omega)$$

$$s.t. \quad W^H x^H(\omega) = h^H(\omega) - T^{H-1}(\omega) x^{H-1},$$

$$x^H(\omega) \geq 0 \quad und$$

letzte Stufe

$$Q^t(x^{t-1}, \xi^t(\omega)) = \min c^t(\omega) x^x(\omega) + E_\xi\{Q(x^t, \xi)\}$$

$$s.t. \quad W^t x^t(\omega) = h^t(\omega) - T^{t-1}(\omega) x^{t-1},$$

$$x^t(\omega) \geq 0.$$

Stufen 2,...H-1

Der Wert, den wir suchen ist:

$$\min c^1 x^1 + E_\xi\{Q(x^1, \xi)\}$$

$$s.t. \quad W^1 x^1 = h^1, \quad x^1 \geq 0.$$

1. Stufe

Mehrstufige Programme, deterministisches Äquivalent, Version 2

Als (gemischt ganzzahliges) lineares Programm:

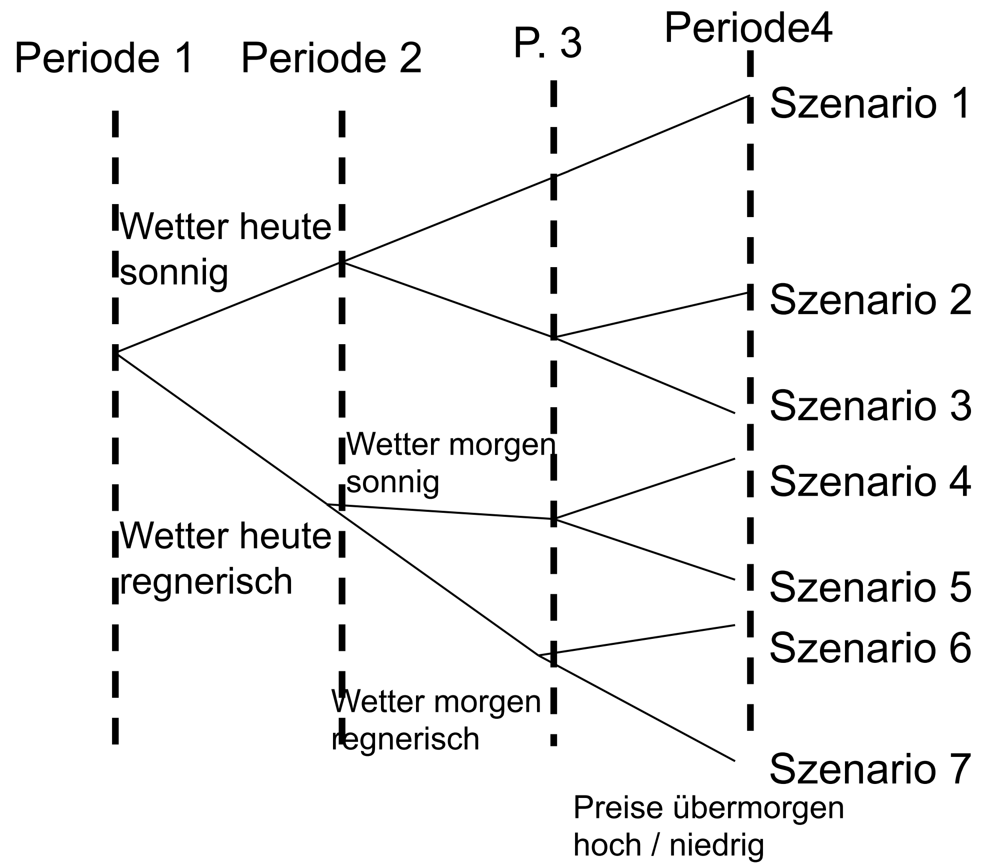
Annahmen:

- klare **Stufung** von Entscheidungsvariablen und Zufallsereignissen
- Zufallsereignisse **unabhängig** von unseren Entscheidungen
  (Das Wetter wird im Normalfall morgen sonnig oder regnerisch sein,
  egal, welche Entscheidung wir in unserem Optimierungsproblem treffen.)

  → essentielle Annahmen für stochastische Programme
  → erlaubt die Aufspaltung von Zufallsprozess und Entscheidungsprozeß
  → führt zu **Szenariobäumen**
- **endliche Anzahl von möglichen Realisierungen** für zukünftigen Ausgängen

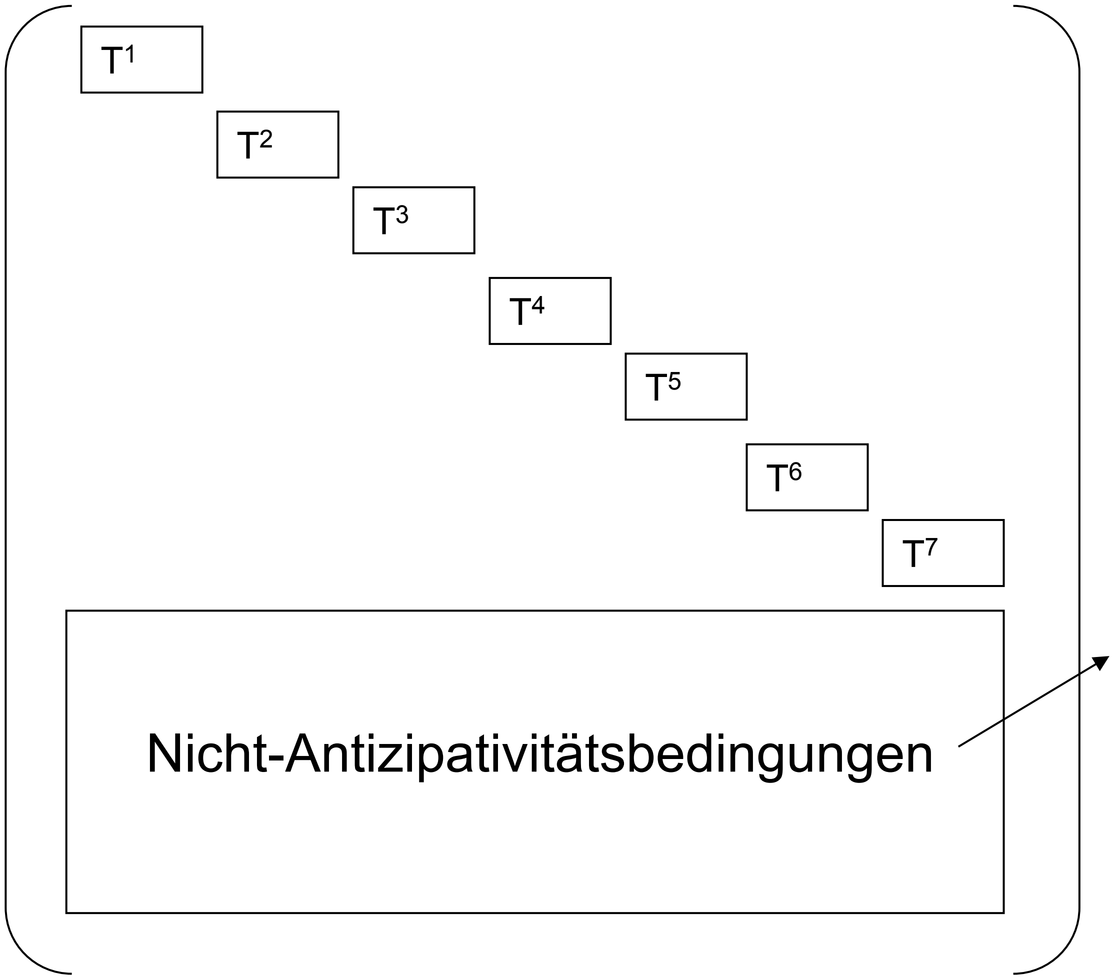
Mehrstufige Programme, deterministisches Äquivalent, Version 2

Szenariobaum:



Stochastic Programming

Matrix, andere Darstellung



Vervielfachung von Variablen führt zu entkoppelten Szenarien

Einige Variablen müssen in allen (manchen) anderen Szenarien gleich sein.

Stochastic Programming

Matrix