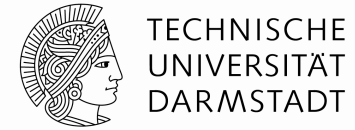

Computerschach

Grundlagen II

Spielbäume und Fehlerfilter



Untersuchung des Phänomens

Fragestellung: Was ist in diesen Spielbäumen z.B. des Schachspiels, das heuristische Spielbaumsuche so erfolgreich macht?

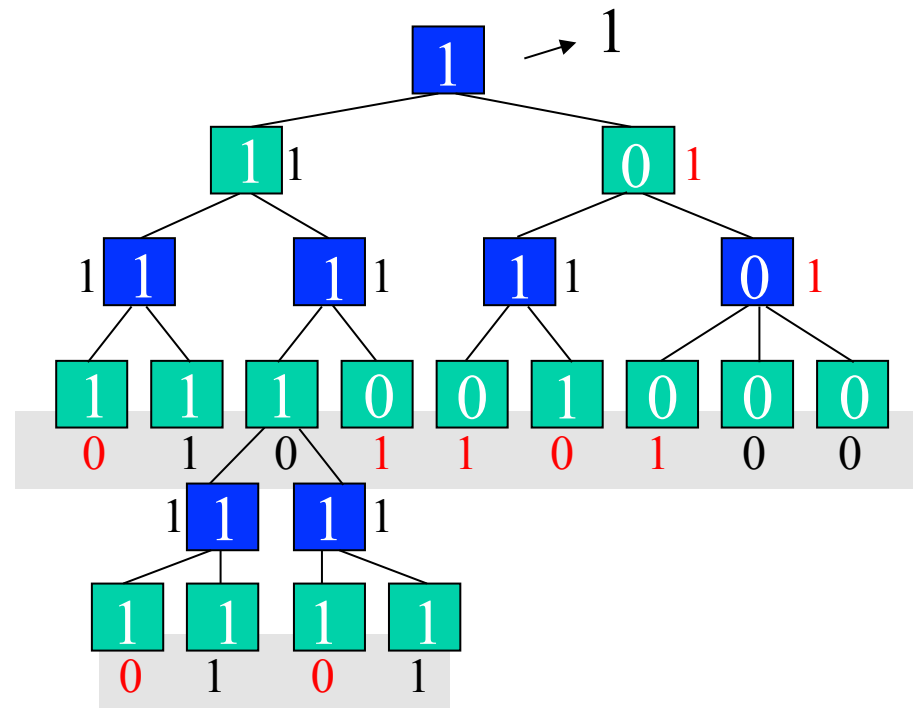
(Nau '79; Pearl '83; Schrüfer '86; Althöfer '88; Scheucher&Kaindl '89,
[Lorenz&Monien STACS '02, TCS '05](#))

Anwendung: Starkes Spiel gegen schwächere Gegner

([Lorenz ESA '04, ICGA Journal '06](#))

Fehleranalyse

Gegeben: **Spielbaum G**, jeder Knoten hat einen **'echten' Wert** 0|1; diese Werte gehorchen dem Minimax-Prinzip. So genannte **'heuristische Werte'** werden den Blättern von G zugewiesen, und diese heuristischen Werte werden genutzt, um heuristische Minimax-Werte für innere Knoten zu bestimmen.



Fehleranalyse

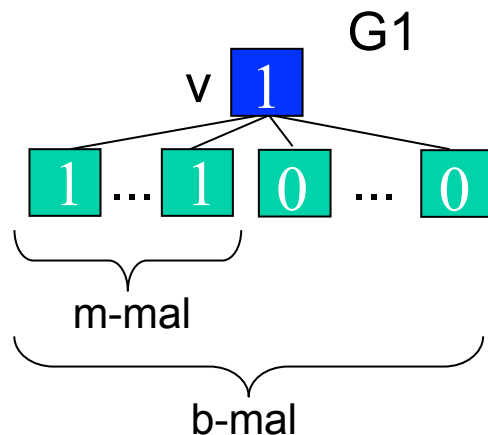
Verschiedene Fragestellungen sind möglich:

- Wie viele Fehler darf ich im günstigsten Fall machen?
wenn Spielbaum G b/t -uniform ist: $b^t - b^{\lfloor t/2 \rfloor}$
- Sei n die Anzahl der Blätter von G . Wie wirkt es sich aus, wenn man (ungefähr) k Fehler bei Blattbewertungen macht?
- Wie viele Fehler darf man an den Blättern im schlimmsten Fall machen?

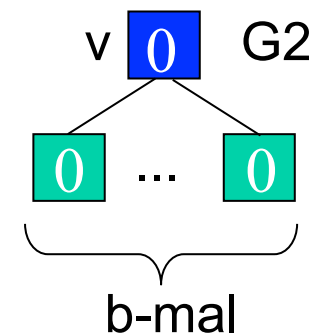
Modell I

- Fehler an Blättern werden mit Fehlerwahrscheinlichkeit \bar{p} ($= 1-p$) gemacht.

v_1, \dots, v_b seien die Nachfolger von v , $g_1(p), \dots, g_b(p)$ seien die Wahrscheinlichkeiten, dass die heuristischen Werte h_1, \dots, h_b der Knoten v_1, \dots, v_b gleich den echten Werten w_1, \dots, w_b sind. Dann ist die Wahrscheinlichkeit dafür, dass der heuristische Minimaxwert von v gleich dem echten Wert von v ist wie folgt:



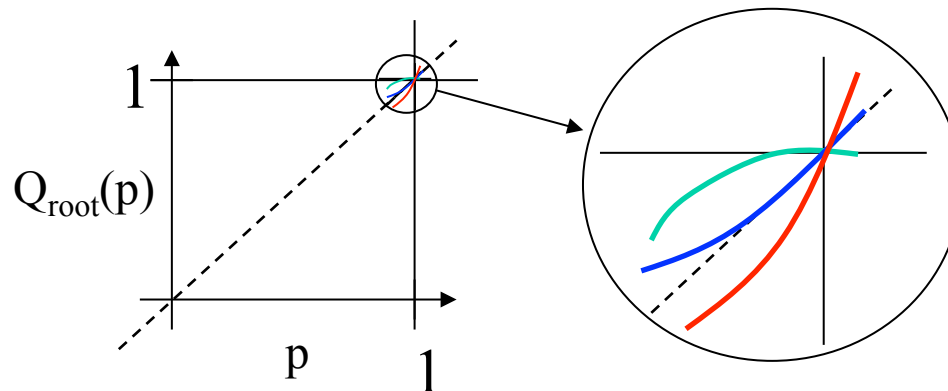
$$Q_v(p) = 1 - \left(\prod_{i=1}^m (1 - g_i(p)) \right) \cdot \prod_{i=m+1}^b g_i(p)$$



$$Q_v(p) = \prod_{i=1}^b g_i(p)$$

Modell I

- Fehler an Blättern werden mit Fehlerwahrscheinlichkeit \bar{p} ($= 1-p$) gemacht.
- Für jeden Knoten v des Spielbaums G , gibt es somit ein „Qualitätspolynom“ $Q_v(p)$, welches die Wahrscheinlichkeit dafür angibt, dass der echte und der heuristische Wert am Knoten v gleich sind.



- Intuitive Erklärung für besseres Spiel bei tieferer Suche:
Super-Bäume haben kleinere Fehlerwahrscheinlichkeiten an ihren Wurzeln,
zumindest wenn die Fehlerwahrscheinlichkeit am Blatt klein genug ist.
- Robustheit kann dann definiert werden mit Hilfe $Q_{\text{root}}^{(k)}(1)$, $k=1, \dots$

Modell I, Zusammenhang mit Average-Case Analyse

Sei G Spielbaum mit n Blättern und s der 0/1-Blattstring der echten Werte. s' sei der verfälschte String. p sei die Wahrscheinlichkeit dafür, einen heuristischen Blattwert korrekt zu erkennen. Die Anzahl korrekter heuristischer Blattbewertungen ist Binomialverteilt.

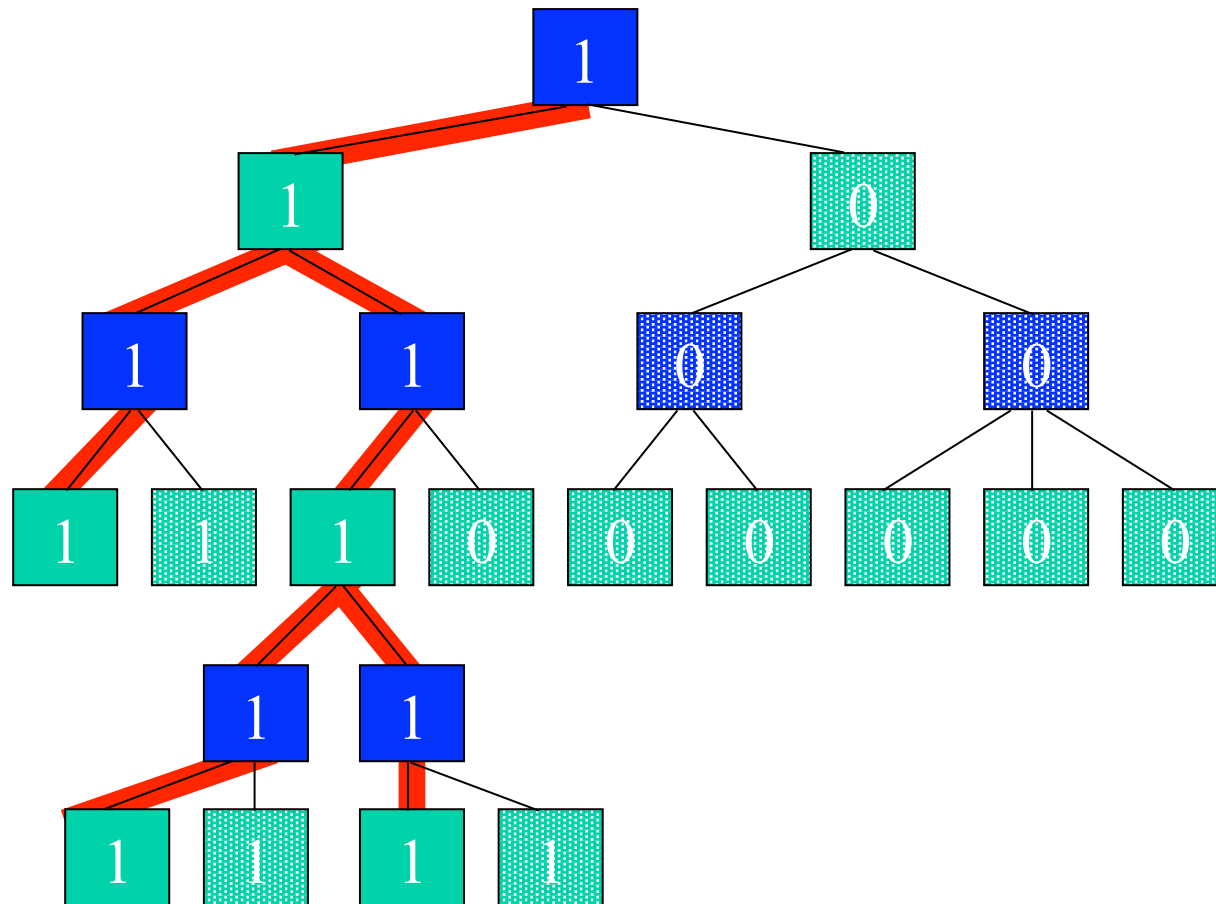
Man kann dann sagen: „Man macht ungefähr $n \cdot (1-p)$ Fehler“

$$\begin{array}{l} 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \quad -s \\ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \quad -s'_1 \\ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \quad -s'_2 \\ \dots \\ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \quad -s'_3 \\ \dots \end{array} \left. \begin{array}{l} \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} C_1 \rightarrow c_1 \\ \\ \\ C_2 \rightarrow c_2 \end{array} \quad \begin{array}{l} C_i \text{ sind Cluster, die Strings} \\ \text{mit genau } c_i \text{ korrekten} \\ \text{Bewertungen enthalten.} \end{array}$$

$Q_G(p) = \sum_{i=0}^n \text{Prob}(\text{heur. Wurzelwert ist korrekt} \mid \text{es gibt genau } i \text{ richtig klassifizierte Blätter}) \cdot \text{Prob}(\text{genau } i \text{ heur. Blattwerte sind korrekt})$

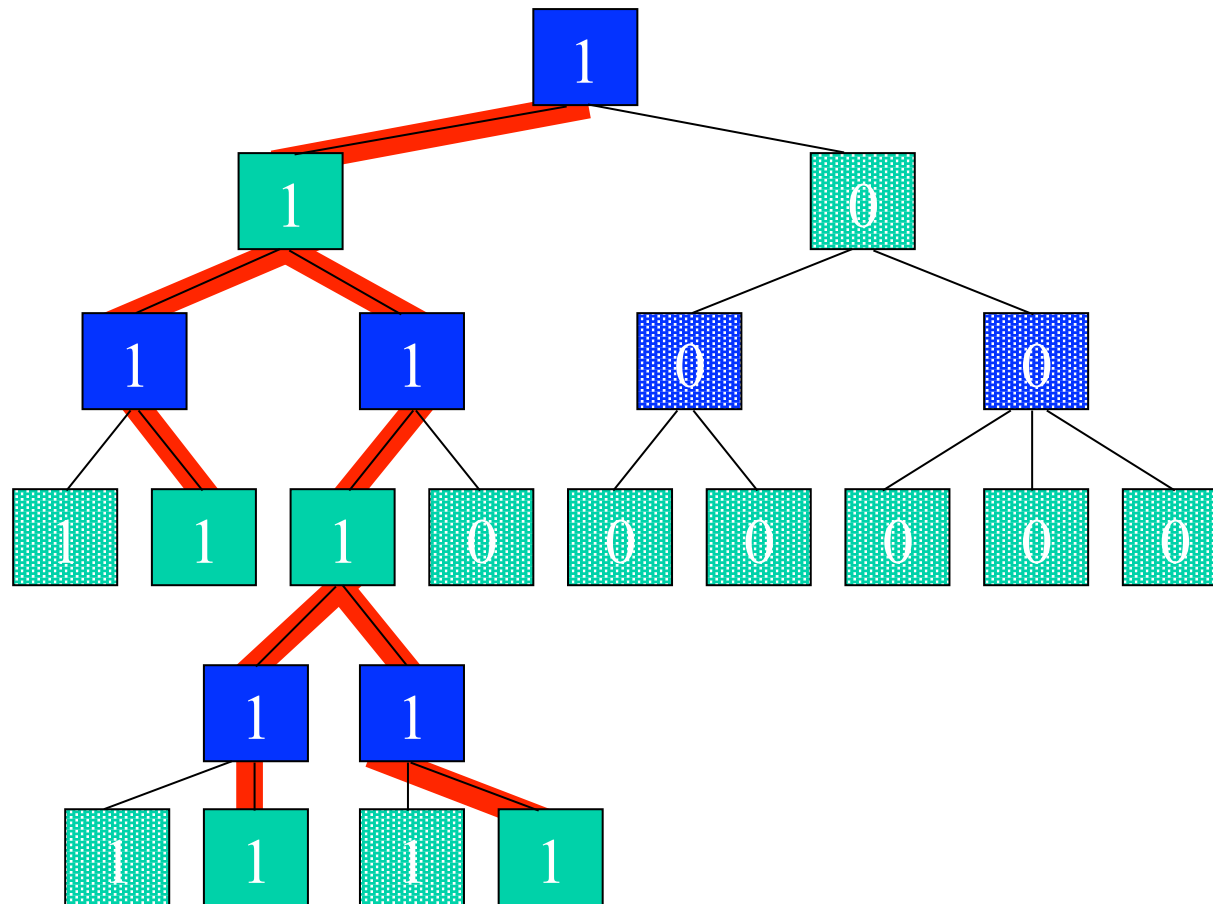
$$= \sum_{i=0}^n c_i \cdot |C_i| \binom{n}{i} p^i (1-p)^{n-i}$$

Modell II



Fehleranalyse

Modell II



Ergebnisse

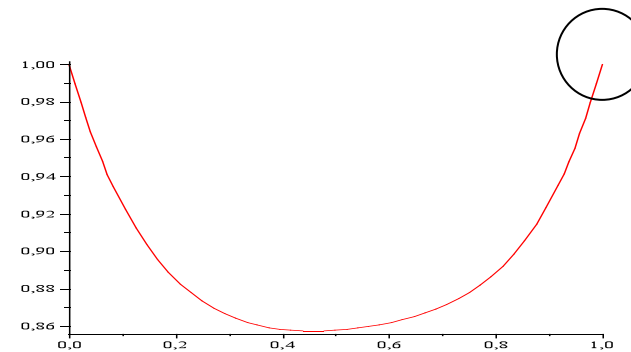
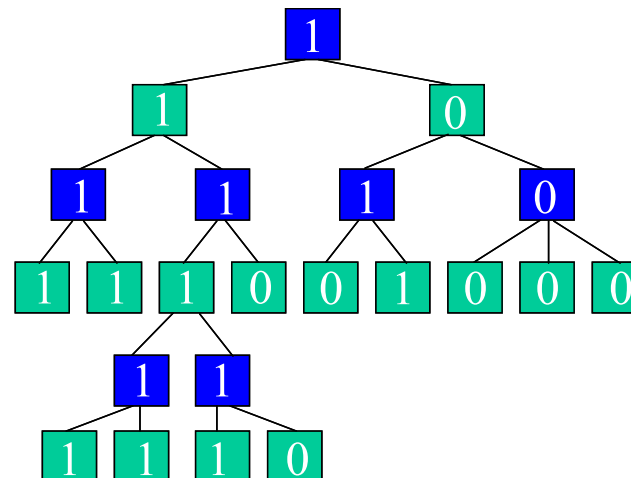
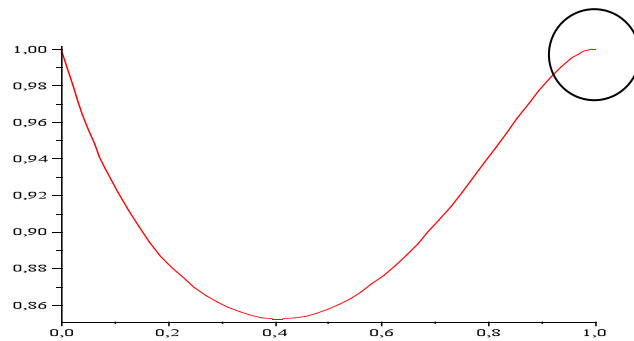
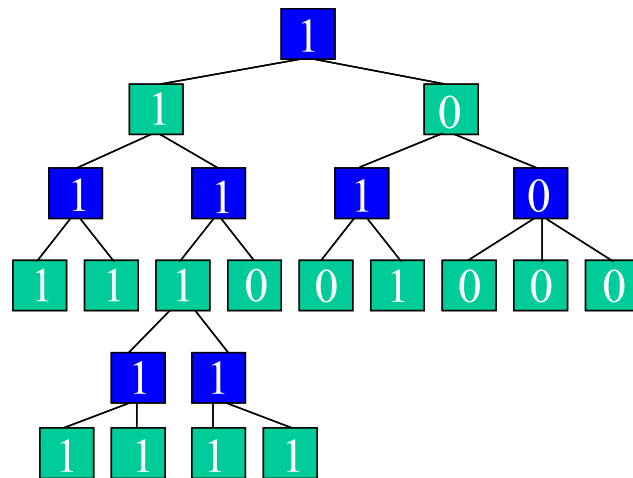
Lemma: $Q_{\text{root}}'(1) = 0$ oder $Q_{\text{root}}'(1) \geq 1$, wobei Q' erste Ableitung von Q

Lemma: Falls $Q_{\text{root}}'(1) \geq 1$, beschreibt $Q_{\text{root}}'(1)$ die Anzahl der Blätter, die den Wurzelwert durch einen Single-Flip ändern können.

Theorem: $Q_{\text{root}}'(1) = 0$ gilt g.d.w. G mindestens 2 blatt-disjunkte Strategien enthält, die den Wurzelwert belegen.

Fehleranalyse

Ergebnisse



Ergebnisse

Die zwei Modelle mit ihren Robustheitsmaßen sind äquivalent zueinander.

Es gibt $n+1$ blattdisjunkte Strategien in G , die alle den Wurzelwert von G belegen.

(Modell II)

\Leftrightarrow

$$Q_{\text{root}}^{(n)}(1) = Q_{\text{root}}^{(n-1)}(1) = \dots = Q_{\text{root}}^{(1)}(1) = 0$$

(Modell I)

Taylorreihenentwicklung $f(p) = f(1) + f'(1)(p-1) + \dots + (f^{(n)}(1)/n!)(p-1)^n + R_{n+1}(p)$ führt uns zu

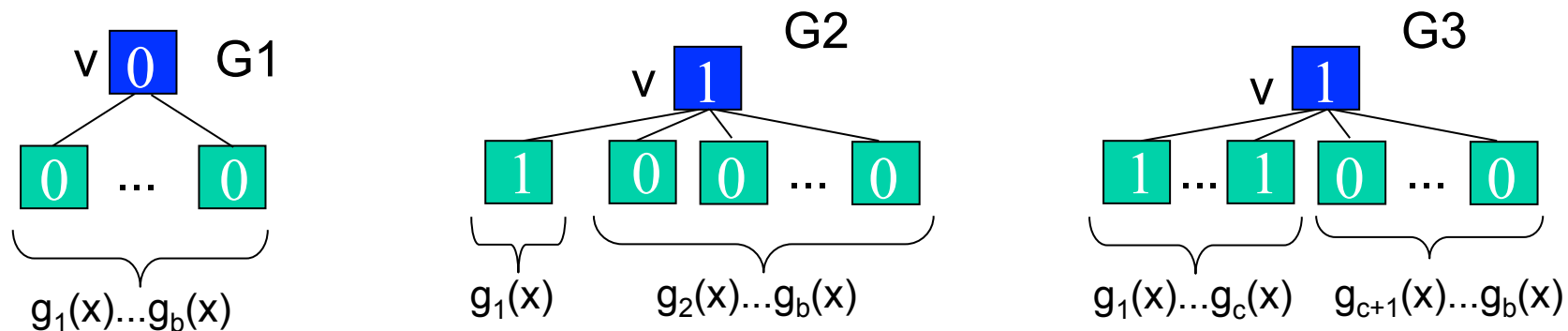
$|Q_{\text{root}}(p) - Q_{\text{root}}(1)| = O((1-p)^{n+1})$ g.d.w. es $n+1$ viele blatt-disjunkte Startegien gibt.

Fehleranalyse

ad Lemma: $Q_{\text{root}}'(1) = 0$ oder $Q_{\text{root}}'(1) \geq 1$, wobei Q' erste Ableitung von Q

ad Theorem: $Q_{\text{root}}'(1) = 0$ gilt g.d.w. G mindestens 2 blatt-disjunkte Strategien enthält, die den Wurzelwert belegen.

Betrachte die folgenden 3 Tiefe-1 Bäume:



Fehleranalyse

$$Q_{G1}(x) = g_1(x) \cdots g_b(x)$$

$$Q_{G2}(x) = 1 - (1 - g_1(x)) \cdot g_2(x) \cdots g_b(x)$$

$$Q_{G3}(x) = 1 - (1 - g_1(x)) \cdots (1 - g_c(x)) \cdot g_{c+1}(x) \cdots g_b(x)$$

$$Q'_{G1}(x) = \sum_{i=1}^b g_1(x) \cdots g'_i(x) \cdots g_b(x)$$

$$Q'_{G2}(x) = -(1 - g_1(x))' \cdot g_2(x) \cdots g_b(x) + \sum_{i=2}^b (1 - g_1(x)) \cdot g_2(x) \cdots g'_i(x) \cdots g_b(x)$$

$$Q'_{G3}(x) = - \sum_{i=1}^c (1 - g_1(x)) \cdots (1 - g_i(x))' \cdots (1 - g_c(x)) \cdot g_{c+1}(x) \cdots g_b(x) \\ - \sum_{i=c+1}^b (1 - g_1(x)) \cdots (1 - g_c(x)) \cdot g_{c+1}(x) \cdots g'_i(x) \cdots g_b(x)$$

$$Q'_{G1}(1) = g'_1(1) + \cdots + g'_b(1)$$

$$Q'_{G2}(1) = g'_1(1)$$

$$Q'_{G3}(1) = 0$$

Fehleranalyse

ad Theorem: Es gibt $n+1$ blattdisjunkte Strategien in G , die alle den Wurzelwert von G belegen. $\Leftrightarrow Q_{\text{root}}^{(n)}(1) = Q_{\text{root}}^{(n-1)}(1) = \dots = Q_{\text{root}}^{(1)}(1) = 0$

Allgemein läßt sich die n -te Ableitung eines Produkts von Polynomen darstellen als

$$\sum_{y_1 + \dots + y_b = n} a(y_1, \dots, y_b) g_1^{(y_1)}(x) \cdots g_b^{(y_b)}(x)$$

mit geeigneten $a(y_1, \dots, y_b) \in \mathbb{N}$.

Zu betrachten sind nun wieder die Ableitungen von Q_{G1} , Q_{G2} , Q_{G3} .

Annahmen:

(i) Für alle $i \leq n$ gilt: Für alle Spielbäume G gibt es i blattdisjunkte Strategien in G , die alle den Wurzelwert von G belegen. $\Leftrightarrow Q_G^{(i-1)}(1) = \dots = Q_G^{(1)}(1) = 0$

(ii) Für alle $G' \in \{G_1, G_2, G_3\}$ soll gelten: Es gibt n blattdisjunkte Strategien in G' , die alle den Wurzelwert von G' belegen **und** $Q_{G'}^{(n-1)}(1) = \dots = Q_{G'}^{(1)}(1) = 0$

(iii) Für alle $i \in \{1, \dots, n-1\}$ gilt: das Vorzeichen von $Q_G^{(i)}(1) = (-1)^{i-1}$

Bemerkung: Im folgenden machen wir Vorbetrachtungen für einen Induktionsbeweis über die Anzahl von blattdisjunkten Strategien und über die Höhe der Bäume. (i) und (iii) werden die Induktionsvoraussetzung bilden, und (ii) wird aus „ $Q_G^{(n)}(1) = \dots = Q_G^{(1)}(1) = 0$ “ oder aus „es gibt $n+1$ blattdisjunkte Strategien ...“ hergeleitet werden.

n-te Ableitung für G1:

$$Q_{G1}^{(n)}(x) = \sum_{y_1 + \dots + y_b = n} a(y_1, \dots, y_b) g_1^{(y_1)}(x) \cdots g_b^{(y_b)}(x)$$

Alle Summanden, die Ableitungen größer 0 aber kleiner als n enthalten, sind Null bei $x=1$, wegen Voraussetzung (ii). Da $g_i(1) = 1$ für alle i , gilt:

$$Q_{G1}^{(n)}(1) = \sum_{i=1}^b g_i^{(n)}(1)$$

n-te Ableitung für G2:

$$Q_{G2}^{(n)}(x) = (-1) \cdot \sum_{y_1 + \dots + y_b = n} a(y_1, \dots, y_b) (1 - g_1(x))^{(y_1)} \cdot g_2^{(y_2)}(x) \cdots g_b^{(y_b)}(x)$$

Mit Hilfe von (ii) sieht man, dass bei $x=1$ nur ein Summand ungleich 0 wird:

$$Q_{G2}^{(n)}(1) = g_1^{(n)}(1)$$

n-te Ableitung für G3:

$$Q_{G3}^{(n)}(x) = (-1) \cdot \sum_{y_1 + \dots + y_b = n} a(y_1, \dots, y_b) (1 - g_1(x))^{(y_1)} \cdots (1 - g_c(x))^{(y_c)} \cdot g_{c+1}^{(y_{c+1})}(x) \cdots g_b^{(y_b)}(x)$$

1. Fall $n < c$: Einer der ersten c Faktoren ist immer = 0 und es gibt, wegen der Definition von „Strategie“ $n+1$ blattdisjunkte Strategien unterhalb der Wurzel.

Also:

$$Q_{G3}^{(n)}(1) = 0$$

2. Fall $n = c$: Sei S_{y_1, \dots, y_b} ein beliebiger Summand von $Q_{G3}^{(n)}(x)$ bei $x=1$. Falls es ein l gibt mit $l \leq c$ und $(y_l = 0$ oder $y_l > 1)$, folgt $S_{y_1, \dots, y_b} = 0$, weil einer der ersten c Wurzelnachfolger (sei das k) liefert: $1 - g_k(1) = 0$. Falls es ein $l > c$ gibt mit $y_l > 0$, folgt ebenfalls sofort $S_{y_1, \dots, y_b} = 0$. Sonst gilt

$$S_{y_1, \dots, y_b}(1) = (-1)^c \cdot \prod_{i=1}^c g_i^{(1)}(1)$$

Vorzeichen: $(-1)(-1)^n \cdot k \cdot \prod_{i=1}^n g_i^{(1)}(1)$, für ein $k \in IN$

n-te Ableitung für G3:

3. Fall $n > c$: Sei S_{y_1, \dots, y_b} ein beliebiger Summand von $Q_{G3}^{(n)}(x)$ bei $x=1$.
- a) Falls es ein l gibt mit $l \leq c$ und $y_l = 0$, folgt $S_{y_1, \dots, y_b} = 0$
- b) Falls es ein l gibt mit $l > c$ und $y_l > 1$, gilt: $\sum_{i=1}^c y_i \leq n-1$. S_{y_1, \dots, y_b} hat die Form $(1-g_1(x))^{y_1} \cdots (1-g_c(x))^{y_c} \cdot X$, X eine reelle Zahl. Wegen Annahme (ii) gibt es n blattdisjunkte Strategien unter der Wurzel von G3. Wegen der Definition von Strategien ist die Summe der blattdisjunkten Strategien unter den ersten c Nachfolgern der Wurzel ebenfalls gleich n . Wir können also schließen, dass einer der ersten c Nachfolger mehr als y_i -viele blattdisjunkte Strategien unter sich hat. Mit Voraussetzung (i) folgt, dass ein $(1-g_i(x))^{(y_i)}$ an der Stelle $x=1$ zu 0 wird, für ein $i \in \{1, \dots, c\}$
- c) $\sum_{i=1}^c y_i = n$ und $y_i > 1$ für die ersten c Wurzelnachfolger

$$Q_{G3}^{(n)}(x) = - \sum_{y_1 + \dots + y_b = n} a(y_1, \dots, y_c, 0, \dots, 0) (1-g_1(x))^{(y_1)} \cdots (1-g_c(x))^{(y_c)}$$

n-te Ableitung für G3, 3. Fall:

$$Q_{G3}^{(n)}(x) = (-1)^{c+1} \sum_{y_1+\dots+y_c=n} a(y_1, \dots, y_c, 0, \dots, 0) g_1^{(y_1)}(x) \cdots g_c^{(y_c)}(x)$$

Vorzeichen:

Wegen Voraussetzung (iii) ist das Vorzeichen von

$$\text{sign}(Q_{G3}^{(n)}(1)) = (-1)(-1)^c \cdot \prod_{i=1}^c \text{sign}(g_i^{(y_i)}(1)), \text{ mit } \sum_{i=1}^c y_i = n$$

Sei $k_i = y_i - 1$, für $i = 1, \dots, c$. Somit ist $(-1)^{k_i}$ das Vorzeichen von $g_i^{(y_i)}(1)$. (vgl. (iii))

Da $\sum_{i=1}^c k_i = n - c$, folgt,

$$\text{sign}(Q_{G3}^{(n)}(1)) = (-1)(-1)^c \cdot \prod_{i=1}^c (-1)^{k_i} = (-1)^{n-1}$$

Induktion:

Annahme:

- (I) Für alle $i \leq n$ gilt: Für alle Spielbäume G gibt es i blattdisjunkte Strategien in G , die alle den Wurzelwert von G belegen. $\Leftrightarrow Q_G^{(i-1)}(1) = \dots = Q_G^{(1)}(1) = 0$
- (II) Für alle $i \in \{1, \dots, n-1\}$ gilt: das Vorzeichen von $Q_G^{(i)}(1) = (-1)^{i-1}$

Induktionsschritt ($n \rightarrow n+1$):

‘ \Leftarrow ’: Es gibt $n+1$ blattdisjunkte Strategien unter der Wurzel von G . Insbesondere gibt es n blattdisjunkte Strategien und mit (I) und (II) wissen wir, dass die Voraussetzungen (i)-(iii) erfüllt sind. Mit Hilfe einer inneren impliziten Induktion sehen wir, dass der Induktionsschritt bereits gemacht ist.

‘ \Rightarrow ’: Sei Q_G ein Qualitätspolynom. Seien $Q_G^{(n)}(1) = \dots = Q_G^{(1)}(1) = 0$. Offensichtlich gilt auch $Q_G^{(n-1)}(1) = \dots = Q_G^{(1)}(1) = 0$. Von (I) wissen wir, dass es n blattdisjunkte Strategien in G gibt, und (i)-(iii) sind erfüllt. Mit Hilfe einer impliziten Induktion über die Tiefe von G ist der Induktionsschritt fertig. Man muss allerdings beachten, dass für alle $i = 1, \dots, n$ gilt $\text{sign}(Q_G^{(i)}(1)) = (-1)^{i-1}$ ist.

Worst-Case Betrachtung:

Satz:

Sei G ein Spielbaum mit Wert 0 oder 1 an der Wurzel. Sei $oBdA$ die Wurzel ein MAX-Knoten. Dann sind folgende Aussagen äquivalent.

- es gibt c -viele blattdisjunkte Strategien unter der Wurzel von G , die beweisen, dass der Wert der Wurzel 0 (bzw. 1) ist.
- man muss mindestens c -viele Blattwerte, bezogen auf die echten Werte, verändern, damit der heuristische Minimaxwert der Wurzel falsch wird.

,=>' klar, denn mit der Veränderung eines Blattwertes kann man nur eine der blattdisjunkten Strategien „zerstören“.

,<=, Wir bauen per Induktion über die Baumtiefe t eine „Zerstörungsstrategie“, die mit c Änderungen die c blattdisjunkten Strategien zerstört.

Start: Sei $t = 1$. Der Baum besteht nur aus einem Knoten, damit gibt es nur eine Strategie und mit Änderung eines Blattes wird der Wurzelwert verfälscht.

Worst-Case Betrachtung:

, \leq , ...

Annahme: Für alle Tiefe-($t-1$)-Bäume gilt, wenn G genau c (c beliebig) blattdisjunkte Strategien enthält, die alle den Wert 0 (bzw. 1) der Wurzel beweisen, läßt sich der Wurzelwert mit Hilfe von c Blattwertänderungen verfälschen.

Schritt $t-1 \rightarrow t$: Betrachte die Wurzel eines Tiefe- t -Spielbaums.

- Ist der echte Wert 0, so gibt es für alle Nachfolger c blattdisjunkte Strategien, die den Wert 0 belegen. Mindestens 1 Nachfolger besitzt genau c solche Strategien. Auf den wenden wir die obige Annahme an und sind fertig.
- Ist der Wert der Wurzel 1, gibt es d viele Nachfolger, die ebenfalls den Wert 1 haben und die Summe der Anzahl blattdisjunkter Strategien unter diesen Wurzelnachfolgern ist gleich c . Mit Hilfe der Induktionsannahme zerstören wir alle diese Strategien.

Fehleranalyse

Fazit:

- Der Wert einer Stellung hängt primär nicht vom 'Bild' der Stellung ab, sondern von der Dynamik, die sich in ihr entwickelt.
- Man kann also 'Chancen' abschätzen.
- Wie die Chancen sind hängt von der Struktur des Spielbaums ab, der unter der aktuell betrachteten Stellung drunterhängt. Wie sieht der Baum aus, wie sind Gewinn, Verlust, Remis darin verteilt (nicht nur in ihrer Anzahl, sondern in ihrer Zusammensetzung)?

Quantifiziertes Lineares Programm (QLP)

- Vektor mit Variablen $x = (x_1, \dots, x_n) \in \mathbb{Q}^n$
- Obere und untere Schranken $l_i \in \mathbb{Q}^n$ und $u_i \in \mathbb{Q}^n$ mit $l_i \leq x_i \leq u_i$
- Matrix $A \in \mathbb{Q}^{m \times n}$
- Vektor $b \in \mathbb{Q}^m$
- Quantifiziererstring $Q = (q_1, \dots, q_n) \in \{\forall, \exists\}^n$ mit $q_i x_i$ für alle $1 \leq i \leq n$

Qlp $G := [Q : Ax \leq b]$ oder $G := [Q(x, y) : A(x, y)^T \leq b]$

Quantifizierte Lineare Programme:

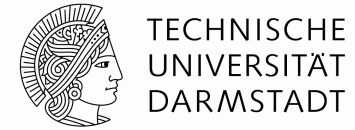
Beispiel: 3-dimensionales QLP

$$\forall x_1 \forall x_2 \exists x_3 : \begin{pmatrix} 10 & -4 & 2 \\ 10 & 4 & -2 \\ -10 & 4 & 1 \\ -10 & -4 & -1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \leq \begin{pmatrix} 0 \\ 4 \\ 12 \\ 8 \end{pmatrix}, \text{ with } x_1 \in [-1,0], x_2 \in [0,1], x_3 \in [-2,2]$$

QLP $\left[Q(x, y) : A(x, y)^T \leq b \right]$ als Zwei-Personen-Nullsummenspiel:

- Ein Spieler belegt die existenzquantifizierten Variablen (Existenzspieler)
- Ein Spieler belegt die allquantifizierten Variablen (Allspieler)
- Die Spieler belegen die Variablenblöcke abwechselnd, wie durch den Quantifiziererstring $Q(x, y)$ vorgegeben
- Ist ein Spieler am Zug bei Variable x_i , so kennt er die Belegung von x_1, \dots, x_{i-1}
- Gilt am Ende $A(x, y)^T \leq b$ gewinnt der Existenzspieler, sonst der Allspieler

Quantifizierte Lineare Programme



Fragestellung:

Existiert ein Algorithmus, der die Belegung von Variable x_i , bei Kenntnis der vorherigen Belegungen von x_1, \dots, x_{i-1} , berechnet, so dass der Existenzspieler das Spiel gewinnt, unabhängig davon wie der Allspieler agiert wenn er am Zug ist.

Komplexität spezieller QLP

Komplexität:

QIPs der Form

$$G : \exists x_1 \in \{a_1, b_1\} \forall y_1 \in \{l_1, u_1\} \dots \exists x_n \in \{a_n, b_n\} \forall y_n \in \{l_n, u_n\} \quad A \cdot \begin{pmatrix} x \\ y \end{pmatrix} \leq b, x \geq 0$$

sind PSPACE vollständig.

a) PSPACE-schwer:

Ausgehend von einem QSAT-Problem bilde für jede Klausel

$k=(l_{k1} \vee \dots \vee l_{kr})$ bilde eine Nebenbedingung der Form $L_{k1} + \dots + L_{kr} \geq 1$, wobei hier

$L_{ki} = x_j$, falls $l_{ki} = x_j$ nicht-negiert und Existenzquantifiziert,

$L_{ki} = 1-x_j$, falls $l_{ki} = x_j$ negiert und Existenzquantifiziert,

$L_{ki} = y_j$, falls $l_{ki} = y_j$ nicht-negiert und Allquantifiziert,

$L_{ki} = 1-y_j$, falls $l_{ki} = y_j$ negiert und Allquantifiziert.

Offenbar ist die SAT-Formel genau dann erfüllt, wenn das Nebenbedingungs-system nicht verletzt wird. Dies setzt sich über die Quantoren fort.

b) in PSPACE: Nutze den Alphabet-Algorithmus

Komplexität spezieller QLP

Komplexität von QLPs ist unbekannt. Jedoch:

Ein E-QLP ist ein QLP, bei dem erst die existenzquantifizierten Variablen kommen:

$$G : \exists x_1 \dots \exists x_n \forall y_1 \in \{l_1, u_1\} \dots \forall y_n \in \{l_n, u_n\} \quad A \cdot \begin{pmatrix} x \\ y \end{pmatrix} \leq b, x \geq 0$$

E-QLPs sind in P.

Beweis: Wir eliminieren zuerst alle Allvariablen.

Dass das in polynomieller Zeit geht, wird noch gezeigt.

Übrig bleibt ein LP.

Komplexität spezieller QLP

Komplexität von QLPs ist unbekannt. Jedoch:

Ein F-QLP ist ein QLP, bei dem erst die allquantifizierten Variablen kommen:

$$G : \forall y_1 \in \{l_1, u_1\} \dots \forall y_n \in \{l_n, u_n\} \exists x_1 \dots \exists x_n \quad A \cdot \begin{pmatrix} x \\ y \end{pmatrix} \leq b, x \geq 0$$

F-QLPs sind **coNP-vollständig**.

Beweis: Ein Problem P ist in coNP, wenn sein negiertes Problem in NP ist.
Wir bilden deshalb

$$\bar{G} : \exists y_1 \in \{l_1, u_1\} \dots \exists y_n \in \{l_n, u_n\} : \text{not} \left(\exists x_1 \dots \exists x_n \quad A \cdot \begin{pmatrix} x \\ y \end{pmatrix} \leq b, x \geq 0 \right)$$

Algorithmus: Rate y_1, \dots, y_n und prüfe, ob das Polyeder

$$\exists x_1 \dots \exists x_n \quad A \cdot x \leq b, x \geq 0$$

leer ist.

Komplexität spezieller QLP

Komplexität von QLPs ist unbekannt. Jedoch:

F-QLPs sind coNP-schwer.

Beweis: Nehme eine SAT-Formel und negiere sie:

$$\begin{aligned} & \exists x_1 \dots \exists x_n (l_{1,1} \vee \dots \vee l_{1,k_1}) \wedge (l_{2,1} \vee \dots \vee l_{2,k_2}) \wedge \dots \wedge (l_{m,1} \vee \dots \vee l_{m,k_m}) \\ & \Leftrightarrow \\ & \text{not} \left(\forall y_1 \dots \forall y_n (\bar{l}_{1,1} \wedge \dots \wedge \bar{l}_{1,k_1}) \vee (\bar{l}_{2,1} \wedge \dots \wedge \bar{l}_{2,k_2}) \vee \dots \vee (\bar{l}_{m,1} \wedge \dots \wedge \bar{l}_{m,k_m}) \right) \end{aligned}$$

Hieraus bilden wir folgendes QIP G:

$$\forall y_1 \dots \forall y_n \exists x_1 \dots \exists x_m :$$

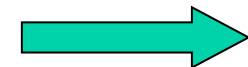
$$\bar{l}_{1,j} \geq x_1, \forall j \in \{1, \dots, k_1\}; \text{ wobei } \bar{l}_{1,j} = y_t \text{ oder } \bar{l}_{1,j} = 1 - y_t, \text{ mit passendem } t$$

$$\bar{l}_{2,j} \geq x_2, \forall j \in \{1, \dots, k_2\}; \text{ wobei } \bar{l}_{2,j} = y_t \text{ oder } \bar{l}_{2,j} = 1 - y_t, \text{ mit passendem } t$$

...

$$\bar{l}_{m,j} \geq x_m, \forall j \in \{1, \dots, k_m\}; \text{ wobei } \bar{l}_{m,j} = y_t \text{ oder } \bar{l}_{m,j} = 1 - y_t, \text{ mit passendem } t$$

$$x_1 + \dots + x_m \geq 1$$



Komplexität spezieller QLP

Komplexität von QLPs ist unbekannt. Jedoch:

F-QLPs sind coNP-schwer.

Beweis (Forts.):

Sei das QLP G' die LP-Relaxierung von QIP.

Behauptung: G hat eine Lösung, genau dann, wenn G' eine Lösung besitzt.

In beiden Fällen sollen die Allvariablen diskret sein, was in QIPs gegeben ist, und was nach vorigen Recherchen nicht die Lösbarkeit der QLP-Systeme ändert.

$G \Rightarrow G'$: klar. Wenn es gegen diskrete y -Variablen eine ganzzahlige Gewinnstrategie für X gibt, gibt es auch eine kontinuierliche Lösung.

$G' \Rightarrow G$:

Es gibt eine nicht-ganzzahlige Strategie \underline{x} für G' .

\Rightarrow mindestens eine x -Variable x_i wird in der Strategie \underline{x} auf einen Wert > 0 gesetzt.

\Rightarrow auf der „linken Seite“ der Ungleichungen, die x_i beschränken, sind alle Terme > 0 .

\Rightarrow alle diese Terme sind $= 1$, denn sie sind von der Form $y \geq x_i$ oder $1-y \geq x_i$.

\Rightarrow man kann x_i auf 1 setzen.

Quantifizierte Lineare Programme:

Beispiel mit Beschränkung auf Ganzzahligkeit / Variablengrenzen

- Knotentypen (Existenzknoten, Allknoten)
- Kantenbeschriftungen (Variablenbelegung)
- Spieldausgänge an Blättern
 - Existenzspieler gewinnt (+)
 - Allspieler gewinnt (-)

