Mathematik und *Spiel*
Ohne Glück zum Sieg

R. Verfürth

Fakultät für Mathematik
Ruhr-Universität Bochum

Bochum / 8. Oktober 2009

Überblick

Kategorisierung

Strategische Spiele

Bewertung einer Stellung

Berechnung des Wertes

Verbesserte Berechnung des Wertes

Aufwand

Gewinnstrategien und Entscheidbarkeit

Epilog

Literatur

Zwei Hauptaspekte

- Zufall
- Strategie

Bedeutung der Strategie

- Gering: Knobeln, Lotto, ...
- Mittel: Kniffel, Poker, Skat, Monopoly, ...
- Entscheidend: Othello, Mühle, Dame, Schach, Go, ...

Struktur

- Zwei Spieler ziehen abwechselnd.
- Zu jeder Stellung gibt es eine endliche Zahl möglicher Züge.
- Das Spiel hat endlich viele Endzustände (Sieg, Verlust, Unentschieden).
- Der Gewinn des einen Spielers ist der Verlust des anderen (Nullsummenspiel).
- Es gibt keine unendliche Sequenz von Stellungen.
- Der Zufall ist ausgeschlossen.

Fragen

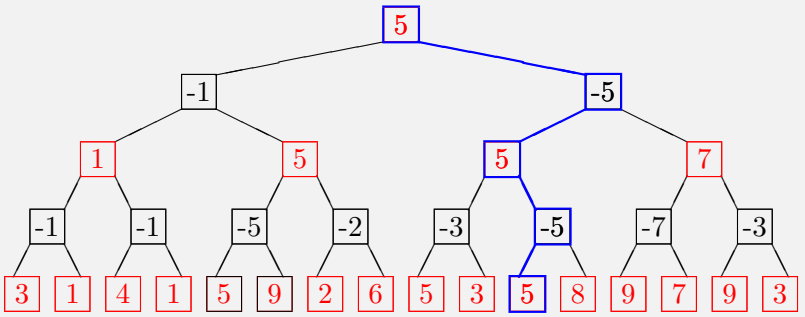
- Was ist der Wert einer gegebenen Stellung?
- Wie findet man einen optimalen Zug?
- Gibt es eine Gewinnstrategie?
- Ist das Spiel entscheidbar?

Wert einer Endstellung

- $F(p)$ beschreibe den Wert einer Stellung p aus Sicht des Spielers, der am Zug ist.
- $-F(p)$ ist dann der Wert aus Sicht des anderen Spielers.

$$F(p) = \begin{cases} \infty & \text{bei Sieg} \\ -\infty & \text{bei Niederlage} \\ 0 & \text{bei Unentschieden} \end{cases}$$

Suchbaum



Wert einer beliebigen Stellung

- Stellung p erlaube d zulässige Züge, die zu den Stellungen $p_1, \ldots, p_d$ führen.
- Wert von p soll durch $F(p)$ beschrieben werden.
- $F(p) = \max\{-F(p_1), \ldots, -F(p_d)\}$
- Rekursive Definition!
- Neg-Max Suche

Naive Realisierung

- Durchsuche den gesamten Suchbaum und werte F rekursiv aus.
- Ist nicht praktikabel wegen des zu großen Aufwandes.
- Verwende eine feste Suchtiefe oder einen Zeitmonitor.
- Liefert nur eine Näherung für F.

Naives Programm

```
int value(Position p) {
    if( endPosition(p) )
        return F(p);
    int v = - INF;
    while( nextChild(p, q) )
        v = max( v, -value(q) );
    return v;
}
```


Ziel

- Versuche frühzeitig zu erkennen, ob ein Zweig des Suchbaumes zu keinem besseren als dem bisherigen Ergebnis führt.
- Versuche, ineffektive Zweige möglichst nahe an der Wurzel (= aktuelle Stellung) abzuschneiden.
- Ziel ist eine größere Suchtiefe bei geringerer Rechenzeit.

α-β-pruning (Donald N. Knuth)

- to prune = beschneiden, ausdünnen, entlauben
- Suche eine Funktion $G(p, \alpha, \beta)$ mit:

$$G(p,\alpha,\beta) = \begin{cases} \alpha & \text{falls } F(p) \le \alpha, \\ F(p) & \text{falls } \alpha \le F(p) \le \beta, \\ \beta & \text{falls } \beta \le F(p). \end{cases}$$

- Aufruf an der Wurzel: $G(p, -\infty, \infty)$

Realisierung

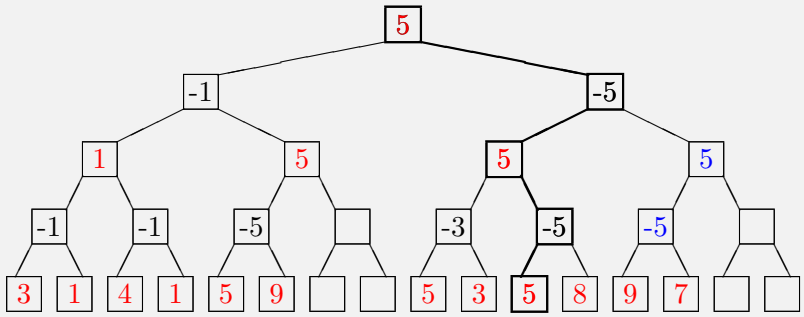
- Initialisiere den Wert mit $v = \alpha$.
- Breche die Maximierung ab, so bald $v \ge \beta$ ist.
- Setze auf dem nächsten Suchlevel

$$\alpha_{\text{neu}} = -\beta_{\text{alt}},$$
$$\beta_{\text{neu}} = -v.$$

Verbessertes Programm

```
int value(Position p, int alpha, int beta) {
    if( endPosition(p) )
        return F(p);
    int v = alpha;
    while( nextChild(p, q) && v < beta )
        v = max( v, -value(q, -beta, -v) );
    return v;
}
```


Suchbaum mit pruning



Spielunabhängige Ergebnisse

- Jeder andere Algorithmus benötigt mindestens den gleichen Aufwand.
- Bester Fall: Für jede Stellung führe der erste mögliche Zug zu einem Optimum, dann werden nur die optimalen Stellungen durchsucht.
- Schlechtester Fall: Es gibt immer eine Permutation der Stellungen, so dass alle Stellungen durchsucht werden.
- Typischer Fall: Aufwand ist $\left(d\frac{1}{\ln d}\right)^h$ mit h Zahl der Niveaus und d Zahl der möglichen Züge pro Stellung.

Spielabhängige Modifikationen

- Ad hoc Bewertung von Stellungen (ohne weitere Suche)
- Ausnutzen von Symmetrien
- Bibliotheken (Hash-Tabellen) bereits analysierter Stellungen
- Eröffnungsbibliotheken
- Endspielbibliotheken

Gewinnstrategien

- Spieler A hat eine Gewinnstrategie, wenn Spieler B auch bei optimalem Spiel einen Sieg von A nicht verhindern kann.
- Spieler A und B können nicht gleichzeitig eine Gewinnstrategie haben.

Remisstrategien

- Spieler A hat eine Remisstrategie, wenn Spieler B auch bei optimalem Spiel ein Remis nicht verhindern kann.
- Spieler A und B können gleichzeitig eine Remisstrategie haben.

Entscheidbarkeit

- Ein Spiel ist entscheidbar, wenn ein Spieler eine Gewinnstrategie hat oder beide Spieler eine Remisstrategie haben.
- Ein Spiel ist fair, wenn beide Spieler eine Remisstrategie haben.

Vermutlich faire Spiele

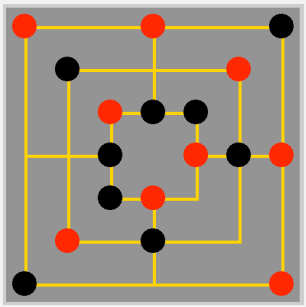
- 8×8 Othello
  (Verifikation würde das Durchsuchen von ca. 10^{37} Stellungen erfordern.)
- Schach
  (Verifikation würde das Durchsuchen von ca. 10^{43} Stellungen erfordern.)

Bekannte Ergebnisse

- 4×4 Othello: schwarz gewinnt.
- 6×6 Othello: schwarz gewinnt.
  (Feinstein 1993, 5 Wochen Workstation, 10^{10} Stellungen)
- Mühle ist fair.
  (Gasser-Nievergelt 1994, 3 Jahre PC-Cluster, Vorwärts-Rückwärts-Suche, 49 Zustandsräume à $10^6 - 10^{10}$ Stellungen, Ausnutzen von Symmetrien)
- Dame (Checkers Variante) ist fair.
  (Schaeffer et al 2007, Programm *Chinook*, Vorwärts-Rückwärts-Suche, ca. $5 \cdot 10^{20}$ Stellungen)

„High Noon“

- Wer zuerst zieht, verliert.



- Rot verliert nach 37 Zügen.
- Schwarz verliert nach 30 Zügen.

Grenzen: Go

- Zahl der Positionen 10^{170} (Schach: 10^{43})
- Keine ad hoc Bewertung von Stellungen möglich
- Suchbaumansatz selbst mit α-β-Pruning unmöglich
- Alternative Ansätze:
  - Monte-Carlo Methoden
  - Kombinatorische Optimierung
  - Mustererkennung

A quoi bon?

- Analyse großer Datenmengen
- Bewertung wirtschaftlicher Situationen (Nash-Gleichgewicht)
- Effiziente numerische Lösung partieller Differentialgleichungen

Literatur

J. Bewersdorff
Glück, Logik und Bluff
Vieweg, 2001

Handout dieses Vortrages
www.rub.de/num1
Link: Berichte

Bachelorarbeiten um Thema „Mathematik und Spiel“
www.rub.de/num1
Link: Abschlussarbeiten