



IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

IKI 30320: Sistem Cerdas

Kuliah 8: (Deterministic) Game Playing

Ruli Manurung

Fakultas Ilmu Komputer
Universitas Indonesia

26 September 2007



Outline

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

- 1 Game playing
- 2 Strategi optimal
- 3 Bekerja cepat
 - Cutoff
 - Tree pruning
- 4 State of the art
- 5 Ringkasan



Outline

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

- 1 Game playing
- 2 Strategi optimal
- 3 Bekerja cepat
 - Cutoff
 - Tree pruning
- 4 State of the art
- 5 Ringkasan



Masalah menghadapi lawan

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

- State space search biasa: agent berinteraksi dengan environment (biasanya static & deterministic).
- Terkadang environment berisi agent lain: **cooperative**, **competitive**
- Melawan agent “musuh”: adversarial search \rightarrow game
- Latar belakang: **game theory** (matematika, ekonomi)



Jenis-jenis game

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat

Cutoff
Tree pruning

State of the
art

Ringkasan

	deterministic	stochastic
perfect information	chess, checkers, go, othello	monopoly, backgammon
imperfect information		bridge, poker, nuclear war

Dalam sejarah AI, *game* yang biasanya jadi bahan riset:

- Deterministic
- Perfect information
- 2 pemain
- Zero-sum game



Game sebagai search

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

- **State**: konfigurasi papan dan info pemain yang akan berjalan
- **Successor function**: mengembalikan list pasangan (move, state)
- **Terminal test**: menentukan apakah permainan sudah selesai (terminal state)
- **Utility function**: penilaian numerik terhadap terminal state. Mis: menang (+1), seri (0), kalah (-1).

Ke-4 hal ini mendefinisikan sebuah **game tree**.



Contoh game tree

IKI30320

Kuliah 8

26 Sep 2007

Ruli Manurung

Game playing

Strategi optimal

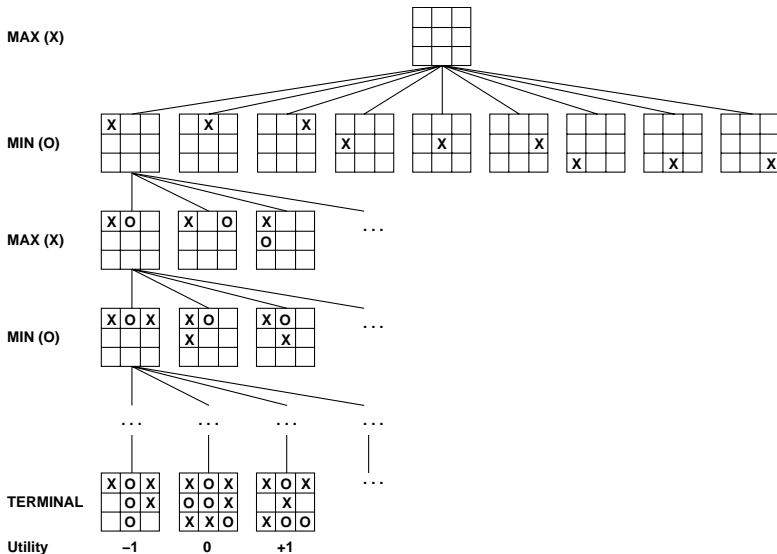
Bekerja cepat

Cutoff

Tree pruning

State of the art

Ringkasan





Outline

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

- 1 Game playing
- 2 Strategi optimal
- 3 Bekerja cepat
 - Cutoff
 - Tree pruning
- 4 State of the art
- 5 Ringkasan



Solusi optimal dalam sebuah game

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

- Andaikan sebuah permainan antara 2 pemain:
MAX (agent) dan MIN
- MAX jalan dulu, lalu MIN, dst. sampai game selesai
- 1 langkah = 2 ply (MAX jalan, MIN jalan)
- Kalau search biasa, cari path sehingga mencapai terminal state di mana MAX menang
- Tapi langkah MIN di luar kendali si agent MAX!
- Solusi berupa *contingent strategy* untuk setiap kemungkinan langkah MIN



Solusi optimal game 2 pemain: Algoritma Minimax

IKI30320

Kuliah 8

26 Sep 2007

Ruli Manurung

Game playing

Strategi optimal

Bekerja cepat

Cutoff

Tree pruning

State of the art

Ringkasan

MINIMAXVALUE(n)=

UTILITY(n)

jika n terminal

$\max_{s \in \text{Successor}(n)} \text{MINIMAXVALUE}(s)$

jika n node MAX

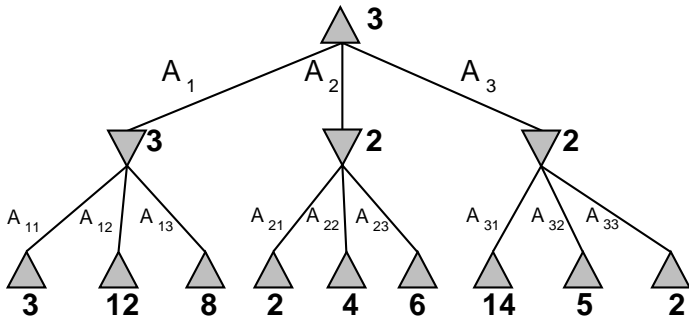
$\min_{s \in \text{Successor}(n)} \text{MINIMAXVALUE}(s)$

jika n node MIN

Ini **strategi optimal**, atau **perfect play**: memberikan hasil terbaik melawan musuh yang diasumsikan optimal.

MAX

MIN





Game dengan 3 (atau lebih) pemain

- Intinya sama dengan minimax: setiap pemain berlaku optimal
- Nilai setiap node berupa *vektor* dengan n nilai
Mis: untuk 3 pemain $A, B, C \rightarrow \langle v_A, v_B, v_C \rangle$
- Pada terminal state: nilai *utility* untuk setiap pemain
- Ternyata dengan mengikuti strategi optimal ini bisa muncul **alliansi**, mis: A & B sama-sama lemah, lawan C .

to move

A

(1, 2, 6)

B

(1, 2, 6)

(-1, 5, 2)

C

(1, 2, 6)

(6, 1, 2)

(-1, 5, 2)

(5, 4, 5)

A

(1, 2, 6)

(4, 2, 3)

(6, 1, 2)

(7, 4, -1)

(5, -1, -1)

(-1, 5, 2)

(7, 7, -1)

(5, 4, 5)



Algoritma Minimax

IKI30320

Kuliah 8

26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat

Cutoff
Tree pruning

State of the
art

Ringkasan

- Definisi algoritma ini **rekursif**, dengan *base case* pada *terminal state*
- Untuk menghitung MINIMAXVALUE pada *initial state*, harus *depth-first search* seluruh game tree!
- **Complete?** Ya, kalau game tree-nya *finite*
- **Optimal?** Ya, asumsi lawan musuh optimal juga. (Kalau tidak? “Lebih optimal”!)
- **Time complexity?** $O(b^m)$
- **Space complexity?** $O(bm)$ (atau $O(m)$ dgn. backtracking)



Algoritma Minimax

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

- Definisi algoritma ini **rekursif**, dengan *base case* pada *terminal state*
- Untuk menghitung MINIMAXVALUE pada *initial state*, harus *depth-first search* seluruh game tree!
- **Complete?** Ya, kalau game tree-nya *finite*
- **Optimal?** Ya, asumsi lawan musuh optimal juga. (Kalau tidak? “Lebih optimal”!)
- **Time complexity?** $O(b^m)$
- **Space complexity?** $O(bm)$ (atau $O(m)$ dgn. backtracking)

Teori sih OK...

Untuk catur: $b \approx 35$, $m \approx 100 \rightarrow$ pencarian strategi optimal berdasarkan Minimax tidak feasible!



Outline

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat

Cutoff
Tree pruning

State of the
art

Ringkasan

- 1 Game playing
- 2 Strategi optimal
- 3 Bekerja cepat**
 - Cutoff
 - Tree pruning
- 4 State of the art
- 5 Ringkasan



Keterbatasan sumber daya

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

- Biasanya dalam suatu permainan ada batasan waktu
- Andaikan ada agent bermain catur yang diberi 100 detik untuk “berpikir” tiap langkah.
- Mis. bisa memproses 10^4 node/detik
→ 10^6 node/langkah
- Kita bisa melakukan **aproksimasi** sbb.:
 - **Cutoff**: batasi depth yang diproses (\approx IDS), bisa juga **quiescence search**
 - **Evaluation function**: *prediksi* dari nilai utility function (tidak perlu sampai ke terminal state)


$$Eval(s) = w_1 f_1(s) + w_2 f_2(s) + \dots + w_n f_n(s) = \sum_{i=1}^n w_i f_i(s)$$

- $w_1 = 1$, f_1 = jumlah pion putih - jumlah pion hitam
- $w_2 = 3$, f_2 = jumlah gajah putih - jumlah gajah hitam





Perhatikan: nilai persisnya tidak penting

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat

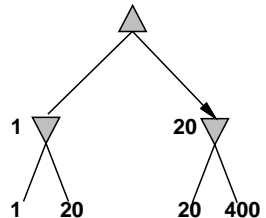
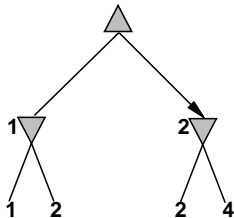
Cutoff
Tree pruning

State of the
art

Ringkasan

MAX

MIN



- Jika *Eval* diubah secara **monotonik**, hasil strategi tidak berubah
- Evaluation function pada game yang deterministic adalah fungsi *ordinal* (urutan/prioritas)



Melakukan search dengan cutoff

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

MINIMAXCUTOFF persis sama **MINIMAXVALUE**, kecuali:

Jika node n , **cutoff** \rightarrow $\text{EVAL}(n)$

- Untuk contoh catur: $b^m = 10^6$, $b = 35 \rightarrow m = 4$
- 4-ply lookahead \approx pemain manusia pemula
- 8-ply lookahead \approx pemain manusia “master”, catur komputer rata-rata
- 12-ply lookahead \approx Deep Blue, Garry Kasparov, ...



Pruning (memangkas) game tree

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat

Cutoff
Tree pruning

State of the
art

Ringkasan

- Kinerja MINIMAX masih bisa diperbaiki dengan **pruning** (memangkas) game tree.
- Prinsipnya: node (subtree) yang tidak mungkin mempengaruhi hasil akhir tidak perlu ditelusuri.
- Pruning demikian dilakukan oleh algoritma **alpha-beta pruning**



Contoh $\alpha - \beta$ pruning

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

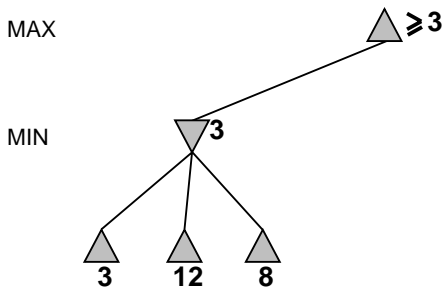
Strategi
optimal

Bekerja cepat

Cutoff
Tree pruning

State of the
art

Ringkasan





Contoh $\alpha - \beta$ pruning

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

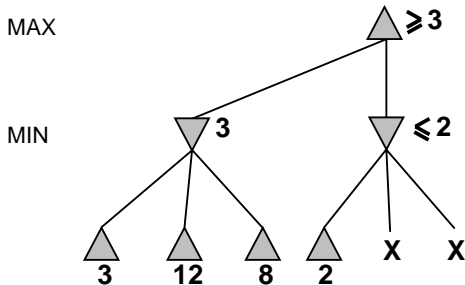
Strategi
optimal

Bekerja cepat

Cutoff
Tree pruning

State of the
art

Ringkasan





Contoh $\alpha - \beta$ pruning

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

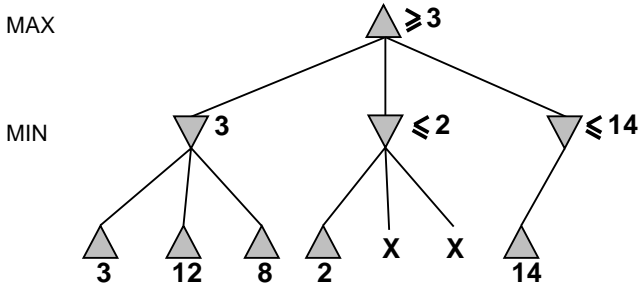
Strategi
optimal

Bekerja cepat

Cutoff
Tree pruning

State of the
art

Ringkasan





Contoh $\alpha - \beta$ pruning

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

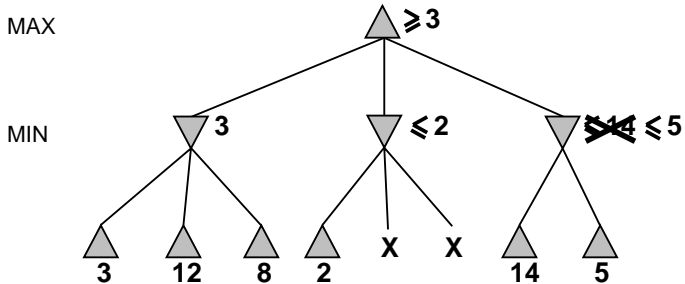
Strategi
optimal

Bekerja cepat

Cutoff
Tree pruning

State of the
art

Ringkasan





Contoh $\alpha - \beta$ pruning

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

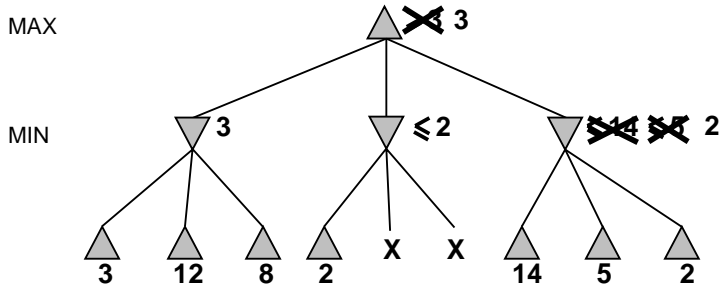
Strategi
optimal

Bekerja cepat

Cutoff
Tree pruning

State of the
art

Ringkasan



$$\begin{aligned}\text{MINIMAXVALUE}(\text{root}) &= \max(\min(3, 12, 8), \min(2, x, y), \min(14, 5, 2)) \\ &= \max(3, \min(2, x, y), 2) \\ &= 3\end{aligned}$$



Prinsip dasar $\alpha - \beta$ pruning

IKI30320

Kuliah 8

26 Sep 2007

Ruli Manurung

Game playing

Strategi optimal

Bekerja cepat

Cutoff
Tree pruning

State of the art

Ringkasan

MAX

MIN

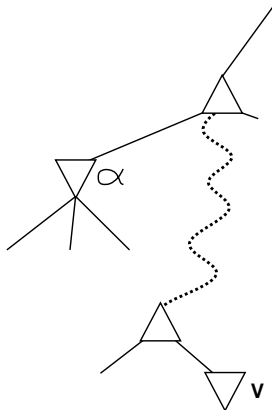
..

..

..

MAX

MIN



Pruning dengan α

α adalah nilai terbesar (terbaik untuk MAX) sementara yang sudah diketahui. Jika nilai $V < \alpha$, MAX tidak pernah akan memilihnya $\rightarrow V$ bisa dipangkas.

Pruning dengan β

β adalah nilai terkecil (terbaik untuk MIN) sementara yang sudah diketahui. Jika nilai $V > \beta$, MIN tidak pernah akan memilihnya $\rightarrow V$ bisa dipangkas.



Sifat **alpha-beta pruning**

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

- **Alpha-beta pruning** tidak mempengaruhi hasil akhir algoritma minimax
- Urutan penelusuran nilai mempengaruhi kinerja → coba pilih nilai yang “terbaik” dulu
- Dengan urutan yang ideal → $O(b^{m/2})$
 - Cutoff depth bisa 2x lipat
 - Untuk catur, lookahead 8-ply \approx tingkat pemain “master”



Outline

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

- 1 Game playing
- 2 Strategi optimal
- 3 Bekerja cepat
 - Cutoff
 - Tree pruning
- 4 State of the art**
- 5 Ringkasan



Hasil riset game yang deterministic

IKI30320

Kuliah 8

26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat

Cutoff
Tree pruning

State of the
art

Ringkasan

- **Checkers**: 1994, Chinook mengalahkan juara dunia selama 40 tahun, Marion Tinsley. Menggunakan database *endgame* untuk *perfect play* semua kemungkinan state dengan maks 8 keping (443 miliar state).
- **Catur**: 1997, Deep Blue mengalahkan juara dunia Gary Kasparov dalam pertarungan 6-ronde. Bisa memroses 200 juta node/detik, evaluation function canggih, dan metode lain sehingga lookahead 40-ply.
- **Othello**: Juara (manusia) menolak melawan komputer yang terlalu jago.
- **Go**: Juara (manusia) menolak melawan komputer yang terlalu jelek. Dalam *go*, $b > 300$.



Outline

IKI30320
Kuliah 8
26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat
Cutoff
Tree pruning

State of the
art

Ringkasan

- 1 Game playing
- 2 Strategi optimal
- 3 Bekerja cepat
 - Cutoff
 - Tree pruning
- 4 State of the art
- 5 Ringkasan



Ringkasan

IKI30320

Kuliah 8

26 Sep 2007

Ruli Manurung

Game playing

Strategi
optimal

Bekerja cepat

Cutoff
Tree pruning

State of the
art

Ringkasan

- Adversarial search adalah masalah search problem melawan agent musuh → *game*
- **Perfect play**: strategi optimal yang memberikan hasil terbaik, asumsi lawan musuh optimal.
- **Algoritma MINIMAX**: secara teoritis memberikan perfect play. Dalam kenyataannya terlalu mahal *computational cost*-nya
- **Aproksimasi** dengan MINIMAXCUTOFF: lihat m langkah ke depan, perkiraan *utility* dengan *evaluation function*
- **Pruning** (memangkas) tree dengan **alpha-beta pruning**
- Baca bab 6.1 - 6.3 buku Russell & Norvig (ada algoritma Minimax, Alpha-Beta)