THE ANALYSIS OF ALPHA BETA PRUNING AND MTD(F) ALGORITHM TO DETERMINE THE BEST ALGORITHM TO BE IMPLEMENTED AT CONNECT FOUR PROTOTYPE

LUKAS TOMMY ET AL 2017 IOP CONF. SER.: MATER. SCI. ENG. 190 012044

PAPER REVIEW ASSIGNMENT

PENGANTAR KECERDASAN ARTIFISIAL - KELAS A - A25/26

BIMA PRIHASTO, S.SI., M.SI., PH.D.

PRESENTED BY:

Kelompok 8

Abdullah Adiwarman Wildan - 11231001

Bagus Nur Ardiansyah - 11231015

Rizky Irswanda Ramadhana - 11231089

Zakaria Fattawari - 11231092

Lisa Sapitri - 11241041

PERMASALAHAN

Game Connect Four memerlukan AI agar komputer dapat bermain layaknya manusia dengan baik dan benar dan cepat. Ada banyak algoritma yang bisa digunakan, tapi belum diketahui mana algoritma yang paling sesuai untuk Connect Four. Algoritma yang tidak cuma optimal utk menang, tapi juga tidak lambat pada search depth yang dalam

Algoritma yg sering dan dapat diimplementasikan pada board game seperti connect four seperti Algoritma seperti Greedy, Random, dan Brute Force yg cepat namun kurang optimal, sedangkan Minimax lebih optimal tapi lambat pada search depth yg dalam.

Terdapat 2 metode optimasi utama yang digunakan

Alpha-Beta Pruning

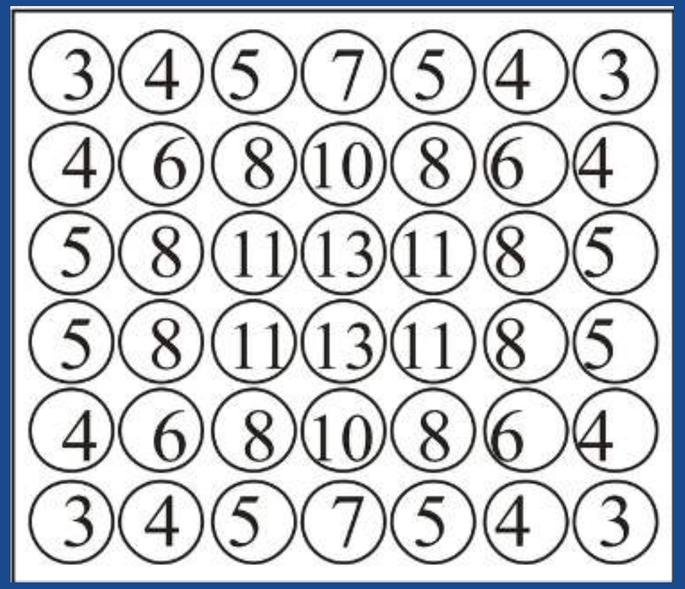
Melewati cabang-cabang pencarian yang tidak mengubah keputusan akhir dengan memanfaatkan nilai α (MAX) dan β (MIN)

Memory-enhanced Test Driver with node n and value f disingkat MTD(f)

Didasari oleh Alpha-Beta Pruning namun menerapkan pencarian zero-window untuk mengurangi search space beserta transposition table untuk menyimpan evaluation value dari board state.

Evaluation Function Design

AI memanfaatkan weighted holes agar dapat memprioritaskan gerakan terhadap posisi mana yang lebih menguntungkan.



*Weighted holes

HASIL

Eksperimen komparasi dijalankan dengan 12 kondisi berbeda.
(depth dan 1st move berbeda)

Depth	Average Execution Time (seconds)		Speed O:X Percentage	Number of *Leaf Node* (LN)		Number of LN O:X Percentage	MTD(*f*) Number of Pass	End Results (Win)
	X	O		X	O			
5	0,1317	0,1089	17,28%	12242	5451	55,47%	201	X(1), O(1)
6	0,3083	0,3089	(0,19%)	17885	8705	51,33%	255	X(1), O(1)
7	1,6981	0,6767	60,15%	175759	36906	79,00%	456	X(1), O(1)
8	2,3185	1,5027	35,19%	218777	73957	66,20%	715	X(1), O(1)
9	14,7149	3,0543	79,24%	1311365	143557	89,05%	478	X(1), O(1)
10	27,1850	5,0731	81,34%	3352837	327609	90,23%	590	Draw(2)
Sum of End Results				X win 5 times, O win 5 times, Draw 2 times				

X : AB Pruning O : MTD(f)

Kedua algoritma ini adalah pengembangan dari pencarian minimax dan menggunakan evaluation function yang sama, sehingga menghasilkan gerakan permainan yang sama.

MTD(f) lebih efisien dibandingkan AB pruning biasa, terutama jika depthnya semakin besar. Terlihat dari jumlah Leaf Node yang dievaluasi dan waktu eksekusinya.

Zero-window search yang digunakan di MTD(f) mengurangi search space lebih efektif dari pada AB Pruning biasa.

Transposition table juga membuat AB Pruning tidak perlu mengevaluasi ulang game state sebelumnya.

Di depth 6, MTD(f) hampir setara dengan AB Pruning karena di depth ini jumlah Leaf Node masih relatif sedikit dan belum menjustifikasi penggunaan algoritma MTD(f) yang lebih kompleks.

KESIMPULAN

Untuk Connect Four, MTD(f) terbukti lebih efisien dibantingkan Alpha-Beta Pruning biasa apabila jika search depth semakin besar. Efisiensinya bisa terlihat jelas dari jumlah Leaf Node yang dievaluasi serta waktu eksekusinya. Kalau search depthnya masih dangkal, performa keduanya kurang lebih sama.

Penggunaan zero-window search di MTD(f) mengurangi ruang pencarian lebih efektif daripada AB Pruning biasa.