第 4 卷第 4 期
1993 年 8 月
软件学报
JOURNAL OF SOFTWARE
Vol. 4, No. 4
Aug. 1993

面向目标的最佳 α—β 搜索策略及其在博弈问题中的应用

张幸儿　　潘征宇

(南京大学计算机科学系，南京 210008)　(南京紫金信息工业公司电脑分公司，南京 210009)

GOAL—ORIENTED OPTIMAL α—β SEARCH STRATEGY AND ITS APPLICATION TO THE GAME PROBLEMS

Zhang Xinger

(*Department of Computer Science, Nanjing University, Nanjing* 210008)

Pan Zhengyu

(*Computer Branch, Zijin Information Industry Corp., Nanjing* 210009)

Abstract The paper presents the formal description of the gobang and explores a strategy for solving the corresponding game problem, proposing a goal—oriented optimal α—β search strategy with the good unification of both the velocity and the precision.

摘要 本文给出五子棋的形式描述，并探讨相应博弈问题的求解策略，提出一种面向目标的最佳 α—β 搜索策略，以达到速度和精度的较好统一.

§1. 引　言

五子棋是在我国民间广为流传的一种双人对弈棋类，其特点是游戏规则简单，棋局变化却又层出不穷. 作为双人博弈问题，它属于人工智能博弈问题求解，也即求解能取胜的博弈走步序列.

本文给出五子棋的形式描述，并探讨相应博弈问题的求解策略，提出一种面向目标的最佳 α—β 搜索策略，以达到速度和精度的较好统一. 这种策略可供其它博弈问题求解参考.

§2. 博弈树搜索原理在五子棋中的应用

2.1 五子棋的形式描述

五子棋是一类双人完备博弈问题，由两位选手轮流依次走步（落子）. 一局棋的结束是

本文 1991 年 5 月收到. 作者张幸儿，副教授，主要研究领域为程序设计语言与编译，程序设计方法学，软件工程工具环境. 潘征宇，助理工程师，主要研究领域为应用软件开发.

其中一位胜(另一位则败)，或者双方平局. 得胜的标志是胜方在某直线上有己方的连续五子. 当无一方得胜而棋盘子满，则成平局.

让大小为 19×19 的五子棋棋盘，从上到下从左到右地对行列加以编号为 1，2，…，19. 落有黑子，白子与未落子的位置分别用颜色 BLACK，WHITE 与 GREEN 标志. 在每一棋步时的棋局反映了在此棋步时双方的落子情况，代表了一个问题状态. 一个问题状态就是一个具体的棋局，它可用数组数据结构来表示.

定义 1：初始状态 S_0 是未落子的空棋局：

$$\forall_{i,j} \in 1..19 : S_0[(i,j)] = \text{GREEN}$$

其中(i,j)代表棋局上的位置. 目标状态是某直线上落有连续同色五子的棋局.

由一切状态组成的状态集合记为 K，一切目标状态组成的集合记为 F，$F \subset K$.

五子棋仅一个落子算符 O，即在某位置上关于某颜色落子. 一次走步相应于应用一次落子算符，把一个状态转换成另一个新状态.

定义 2：落子算符 $O: K \times \text{colors} \times \sum \rightarrow K$，

其中 colors = {BLACK，WHITE}，$\sum = \{(x,y) \mid 1 \leqslant x, y \leqslant 19\}$.

例如，$S_{i+1} = O(S_i, \text{BLACK}, (x,y))$，它表示：在 S_i 相应的棋局中往位置(x，y)上落下一黑子，则状态 S_i 转换成后继状态 S_{i+1}. 显然，若 $S_i[(x,y)]$ IN colors 为真时，落子算符应用失败. 仅当 $S_i[(x,y)] = \text{GREEN}$ 时才能应用落子算符.

定义 3：五子棋求解问题是寻找一个状态序列 $S_1, S_2, \cdots, S_n$，使

$S_1 = O(S_0, \text{color}_1, P_1)$

$S_2 = O(S_1, \text{color}_2, P_2)$

…

$S_n = O(S_{n-1}, \text{color}_n, P_n)$

其中诸 P 是位置，$P_j \neq P_k (j \neq k)$，$\text{color}_j = \text{BLACK}$(j 为奇数)或 WHITE(j 为偶数)，$1 \leqslant j, k \leqslant 361$. 当 $n < 361$ 且 $P_n \in F$，则若 $\text{color}_n = \text{BLACK}$，黑方胜；但若 $\text{color}_n = \text{WHITE}$，则白方胜. 否则 $n \geqslant 361$，双方平局.

2.2 最佳 α—β 搜索策略

让计算机一方称为 MAX，游戏人一方称为 MIN. 约定最终目标是 MAX 取胜，采用的策略是试图使 MAX 取胜的策略.

上述求解过程在人工智能范畴内通过博弈树搜索过程解决. 博弈树中每个结点对应于一个状态，根结点对应于初始状态. 由初始状态结点达到目标状态结点的路径所对应的算符应用序列便是问题的解答. 鉴于对于五子棋的完整博弈树将包含 360! 个叶结点之多，自然采用 α—β 搜索(剪枝)策略，即在博弈树生长的同时完成对叶结点的静态估值与倒推值计算，进行 α—β 剪枝. 要达到最佳 α—β 搜索，必须让结点按一定顺序排列(生成). 但在一般情况下需生成结点后才可对结点排序，因此是难以实现最佳 α—β 搜索的. 然而，对于五子棋，引进棋局姿式的概念，使得实现最佳 α—β 搜索策略成为可能.

§3. NDWZQ 系统的设计与实现

3.1 设计思想

NDWZQ 系统用来进行五子棋游戏,如上所述采用 $\alpha—\beta$ 搜索策略. 为了 MAX 能在尽可能短的可容许的响应时间内确定走步,本系统采用了面向目标的最佳 $\alpha—\beta$ 搜索策略.

3.2 落子姿式定义

五子棋以某方在棋盘某直线位置上有相连的同色五子作为胜负标准,相连的同色子越接近五个,则越接近胜利. 显然如果某方出现有活四或双活三等情况时必胜无疑. 这时无需再按 $\alpha—\beta$ 过程进行搜索. 因此标志一些特殊的落子姿式是必要的. 现定义一直线上的如下几类落子姿式.

a2:表示活二,即两个同色子相连且两端无异色子阻挡的姿式名称;

a3:表示活三,即三个同色子相连且两端无异色子阻挡的姿式名称;

a5:表示目标姿式,也即至少有五个同色子相连的姿式名称;

b3:表示 a3 或 c3 有一端被异色子阻挡的姿式名称;

b4:表示 c4 有一端被异色子阻挡,或冲四,即相连四个同色子有一端被异色子阻挡的姿式名称;

c2:表示有两个同色子,中间隔一空位置,且两端无异色子阻挡的姿式名称;

c3:表示活嵌四,即有三个同色子,其中有两个相连,另一个与相连两子隔一空位置,且两端无异色子阻挡的姿式名称;

c4:表示嵌五,即相连五个位置上有四个同色子,它们不全相连,其中隔有一个空位置的姿式名称.

为了刻划棋局中落子姿式的状况,必要的是引进数组类型的变量来记录棋局中黑子与白子分别出现各种姿式的次数.

3.3 静态估值函数 f 的定义

静态估值函数 f 用来对博弈树的叶结点进行静态估值. NDWZQ 系统中进行静态估值的叶结点是按所允许最大深度扩展而得.

静态估值函数 f 的定义呈下形:

$$f(P)=\sigma\times maxscore-\sigma\times c+\delta\times g(staterecord(P,whatcolor))$$

其中 $\sigma=1$ 或 -1, $\delta=0$ 或 1, staterecord(P, whatcolor)用来计算棋局位置 P 处落下 whatcolor 颜色之子时所生成棋局姿式的状况. g 则是棋局各种姿式出现次数的线性函数,用来关于相应棋局的姿式状况,求得一个整数值. c 则在不同的姿式组合情况下取不同的值. 越接近目标姿式 a5, f 的绝对值越大,且当 whatcolor 为 MAX 方颜色时 f(P)取正值,为 MIN 方颜色时 f(P)取负值.

3.4 MAX 最佳优先走步位置的确定

确定 MAX 最佳优先走步位置的算法由算法 1 描述如下.

算法 1:面向目标的最佳 $\alpha—\beta$ 搜索算法

步骤 1:判别当前棋局中是否存在非常接近目标姿式的特殊姿式(如 b4 与 a3 等). 若存在,则确定落子位置,这时构成 MAX 方姿式 a5 或 b4,或者破坏 MIN 方姿式 b4 或 c4 等,转向步骤 5. 否则执行以下步骤.

步骤 2:判别当前棋局中是否存在其它姿式. 当不存在时转向步骤 3,否则对允许落子范围内相应结点有最高优先级的一切落子候选位置执行以下步骤.

步骤 2.1：若某位置处落子后生成的新棋局的姿式(组合)表明必胜无疑(如多次出现 b4 或 c4，或者同时出现 b4 与 c4)，则确定此位置落子，转向步骤 5. 否则执行以下步骤.

步骤 2.2：把当前棋局相应的结点作为当前叶结点，且让 α 为比一切可能的倒推值更小的最小值.

步骤 2.3：按落子后生成的新棋局的姿式(组合)变化及离达到目标姿式的程度，对结点(位置)赋以优先级，离达到目标姿式的程度相近者具有相同的优先级.

步骤 2.4：对粗略好处估值表明谁落子便对谁有利的一切位置的相应结点计算落 MAX 方之子时的倒推值，以超过 α 的倒推值作为 α 值，最终 α 值为最大倒推值，以相应于最大倒推值的结点所对应的位置为落子位置，转向步骤 5.

步骤 3：在允许落子范围内确定具有最大静态估值的落子位置，这时对于 MAX 方 a2 出现次数最多，转向步骤 5. 当不能确定这样的位置时，执行步骤 4.

步骤 4：以 MIN 方落子处的相邻位置为落子位置.

步骤 5：结束.

算法 1 基于离达到目标姿式的程度，对结点进行按优先级排序，仅对优先级最高的那些结点才进行倒推值计算，从而实现了最佳 α—β 搜索策略.

3.5 NDWZQ 系统的总体结构

NDWZQ 系统的总体结构是简单的，这里略过不作讨论.

§4. 面向目标的最佳 α—β 搜索策略对功效的改进

作为双人博弈的一方，MAX(计算机)必须在可容许的时间内作出走步决定. 必要的是采取各种可能的措施改进与提高博弈树搜索功效. 实现面向目标的最佳 α—β 搜索策略的算法 1 具体从下列四方面着手.

1) 限制最大深度

每次深度优先地计算倒推值时规定最大搜索深度为 2，即对每个叶结点至多相继扩展两次地计算倒推值.

2) 缩小落子范围与限制落子位置

五子棋盘可有 361 个落子位置，事实上，不论哪一方必将在有子处邻近落子. 因此可把落子区域缩小到由(X_{min}, Y_{min})，(X_{min}, Y_{max})，(X_{max}, Y_{min})与(X_{max}, Y_{max})四点连成的矩形区域，这里 X_{min} 与 Y_{min} 分别为已落子各位置的最小列号减 2 与最小行号减 2，而 X_{max} 与 Y_{max} 则分别为已落子各位置的最大列号加 2 与最大行号加 2(当然超过边界，则到边界止). 考虑到已落子位置不能再落子，以及当未落子的某位置其周围两行两列两对角线共 16 个位置处也未落子，该位置也不作为落子候选位置，需生成的结点数将进一步减少.

3) 粗略好处估值

对某未落子位置的相应结点的粗略好处估值，按照在此位置落子后呈现的姿式及姿式的变化情况而作出. 当有利于达到或接近目标姿式时，粗略好处估值则大. 仅当在假定 MAX 走步时对 MAX 有好处(即 MAX 粗略好处估值大于 MIN 粗略好处估值)或在假定 MIN 走步时对 MIN 有好处(即 MIN 粗略好处估值大于 MAX 粗略好处估值)才进行倒推值求值与 α—β 分析.

4) 实施最佳 α—β 搜索策略

考察算法 1 的各步骤，可见是按照根据离目标姿式的程度确定的优先级进行的，达到哪一级优先级便只进行到相应步骤. 算法 1 所进行的 α—β 搜索是否最佳，关键在步骤 2. 步骤 2 对允许落子范围内各个落子候选位置依据落子在此处时呈现的姿式(组合)的变化及离目标姿式的程度给相应结点以优先级. 仅具有最高优先级的那些结点才可能进行倒推值计算. 此时进一步仅对落子时粗略好处估值表明谁落子就对谁有利的那些位置相应结点进行倒推值计算.

这样便在进行倒推值计算之前对结点进行了排序，仅生成具有最高优先级的相应结点，从而实现了最佳 α—β 搜索.

概括起来，五子棋博弈问题，由于引进了姿式，尤其是目标姿式的概念，使实现最佳 α—β 搜索成为可能.

1) 目标姿式明确，任何一个姿式越接近目标姿式，则越接近胜利，因而相应结点被赋以的优先级越高.

2) 对任一棋步的棋局，总可确定其中呈现的姿式或姿式组合及姿式的变化，从而赋以相应的优先级.

3) 在任一棋步时可以在进行倒推值计算之前，对一切可落子位置落子时呈现的姿式(组合)确定优先级，仅对优先级最高的那些结点进行倒推值计算.

附录中给出人机对弈的一个实例，其中 MAX(计算机)一方先行，且 MAX 胜. 注意，在允许落子位置数一栏中列出了允许落子范围位置数与实际允许落子位置数两者.

附　录

棋步	最高优先级结点数	粗略好处值有利结点数	求 MAX 值次数	求 MIN 值次数	允许落子位置数	落子位置	棋步	最高优先级结点数	粗略好处值有利结点数	求 MAX 值次数	求 MIN 值次数	允许落子位置数	落子位置
1						(10,10)	28						(7,8)
2						(11,11)	29	10	7	160	2523	156,87	(10,8)
3	0	0	0	0	36,22	(9,10)	30						(13,11)
4						(11,10)	31	15	10	228	2729	156,87	(12,12)
5	6	3	36	324	42,32	(11,9)	32						(12,9)
6						(9,11)	33	10	7	219	2411	156,84	(9,7)
7	5	3	16	197	49,34	(10,11)	34						(8,6)
8						(10,9)	35	14	9	386	6633	168,90	(6,9)
9	8	6	39	316	49,40	(11,12)	36						(5,9)
10						(12,13)	37	14	9	400	8939	196,95	(8,8)
11	5	3	23	127	72,46	(12,11)	38						(6,10)
12						(10,13)	39	11	7	202	1018	196,95	(10,6)
13	3	3	61	308	72,48	(13,13)	40						(11,5)
14						(10,12)	41	15	9	208	2728	210,99	(9,8)
15	9	5	85	647	81,54	(12,10)	42						(11,8)
16						(11,13)	43	1	1	63	188	210,97	(9,5)
17	2	2	49	322	81,55	(12,14)	44						(7,11)
18						(9,13)	45	0	0	0	0	210,99	(8,12)
19	0	0	0	0	90,56	(8,13)	46						(4,8)
20						(8,10)	47	0	0	0	0	225,99	(3,7)
21	0	0	0	0	100,61	(7,9)	48						(13,6)
22						(8,14)	49	0	0	0	0	225,102	(14,5)
23	0	0	0	0	110,68	(7,15)	50						(9,6)
24						(14,12)	51	0	0	0	0	225,106	(11,7)
25	10	7	186	4233	132,76	(12,8)	52						(10,7)
26						(12,7)	53	0	0	0	0		(8,4)
27	9	7	218	2213	156,83	(8,9)							

算法 1 的功效与姿式组合优先级的划分紧密相关，如果在任一棋步具有相同最高优

先级的结点数减少到最少，则倒推值计算量将最少（剪枝最多）. 优先级的划分，在实践上需要经验与大量的试验，但在理论上，这总是可以做到的.

§5. 结　束　语

双人博弈问题中博弈树一般十分庞大，五子棋虽较简单，情况也如此，因此必须限制最大搜索深度，即向前只看若干步.

NDWZQ 系统针对五子棋的特殊情况，引进了姿式的概念，基于目标姿式明确，提出了面向目标的最佳 α—β 搜索策略，它有如下特点. 1）计算机能以可容许的响应时间作出反应，确定走步位置. 2）一般水平的游戏人不能轻易取胜.

NDWZQ 系统在 AST 286 型微机上实现，它具有良好的用户界面，使用方便，如同通常双人对弈，而且当计算机落子时伴有声音及光标提示，可谓有声有色. 实践表明本文提出的面向目标的最佳 α—β 搜索策略效果较好，可供其它博弈问题求解参考.

参考文献

1　傅京孙等，《人工智能及其应用》，清华大学出版社，1988 年.

2　陆汝钤，《人工智能》，科学出版社，1989 年.