

一种新的博弈树迭代向前剪枝搜索^{*}

孙若莹¹, 宫义山², 赵 刚¹

(1. 北京信息科技大学 信息管理学院, 北京 100192; 2. 沈阳工业大学 信息科学与工程学院, 沈阳 110870)

摘 要: 针对博弈树迭代加深搜索和向前剪枝搜索中误剪最佳分支的弱点, 利用向前剪枝搜索与预评估搜索间的双重迭代调用, 提出了一种新的博弈树迭代向前剪枝搜索方法. 预评估搜索通过节点排序及调整剪枝比率可以更加准确地选取排序在前的最佳分支, 进而使迭代向前剪枝搜索实现在预评估所保留的最佳分支方向进行深度搜索, 二者迭代相互调用以提高向前剪枝搜索的有效性 & 效率. 定性分析与中国象棋计算机博弈实验结果表明, 迭代向前剪枝搜索提高了实时行棋决策的效率 & 效果, 与 $\alpha\beta$ 剪枝搜索相比, 提高的搜索效率超过 160 倍, 同时取得了胜负比近 7 倍的博弈效果.

关 键 词: 人工智能; 博弈树搜索; $\alpha\beta$ 剪枝; 向前剪枝搜索; 迭代加深搜索; 评估函数; 中国象棋博弈; 实时行棋决策

中图分类号: TP 391 文献标志码: A 文章编号: 1000-1646(2017)03-0304-07

A new iterative forward-pruning search for game tree

SUN Ruo-ying¹, GONG Yi-shan², ZHAO Gang¹

(1. School of Information Management, Beijing Information and Technology University, Beijing 100192, China; 2. School of Information Science and Engineering, Shenyang University of Technology, Shenyang 110870, China)

Abstract: Aiming at the weakness of incorrectly pruning the optimal subtrees in the iterative deepening-search and forward-pruning search for game-tree, a new iterative forward-pruning search method for game tree was proposed through adopting the reciprocal iterative calls between the forward-pruning search and pre-estimation search. The optimal subtrees with high priority could be selected more exactly through ordering the node and adjusting the pruning ratio in the pre-estimation search, which made the iterative forward-pruning search realize the deep research in the direction of optimal subtrees preserved with the pre-estimation search, and both methods could iteratively call each other to improve the effectiveness and efficiency of forward-pruning search. The qualitative analysis and the experimental results of Chinese chess computer game show that the effectiveness and efficiency of real time move decision can be improved with the iterative forward-pruning search. Compared with the $\alpha\beta$ pruning search, the search efficiency gets improved more than 160 times, and the game effect with the win-lost ratio of nearly 7 times is obtained.

Key words: artificial intelligence; game-tree search; $\alpha\beta$ pruning; forward-pruning search; iterative-deepening search; evaluation function; Chinese chess game; real time move decision

众多博弈游戏深受人们喜爱, 计算机博弈模型使用博弈树搜索与评估函数相结合的方式实现

博弈游戏, 计算机博弈的基础是博弈树搜索技术, 博弈树搜索深度是决定其棋力高低的重要因

收稿日期: 2017-03-01.

基金项目: 国家自然科学基金资助项目(61572079); 北京市教委科技重点项目(KZ201411232036).

作者简介: 孙若莹(1963-), 女, 辽宁沈阳人, 副教授, 博士, 主要从事人工智能和电子商务等方面的研究.

^{*} 本文已于 2017-05-08 20:25 在中国知网优先数字出版. 网络出版地址: <http://www.cnki.net/kcms/detail/21.1189.T.20170508.2025.020.html>

素^[1-4]. 中国象棋和国际象棋具有许多类似之处, 但中国象棋计算机博弈的空间复杂度和博弈树复杂度比国际象棋略高, 按照每一棋局平均有 45 种可行着法、每局平均行棋 90 着计算, 其完全搜索博弈树的复杂度大约为 $10^{[5]}$. 二十世纪七十年代末我国学者开始了中国象棋计算机博弈的研究工作, 其研究工作的开展落后于国际象棋, 八十年代初开发出个人计算机象棋程序, 进而相继出现了中山大学的“纵马奔流”和东北大学的“棋天大圣”等象棋程序.

博弈树搜索技术与众多机器学习方法^[6-9]始终是人工智能领域的核心问题和研究热点, 直接关系着人工智能系统的性能和效率. 自 Claude Shannon 提出国际象棋计算机博弈方案以来, 博弈树搜索技术经历了从最大最小搜索到 $\alpha\beta$ 剪枝搜索, 再到启发式迭代加深搜索, 直至概率向前剪枝搜索等若干阶段, 人工智能领域研究者不断提出各种搜索技术, 提高博弈树搜索的效果和效率^[10-12]. 目前的主要博弈树搜索技术多数以迭代加深搜索及概率向前剪枝搜索为基础. 然而, 由于迭代加深搜索与向前剪枝搜索在向前剪枝过程中使用概率剪枝方法, 对于非终止状态存在一定概率剪掉最佳行棋, 可能导致错误行棋, 从而影响对弈结果.

本文深入讨论了迭代加深搜索和概率向前剪枝搜索算法, 在此基础上为进一步提高博弈树的搜索效率, 并针对计算局限导致博弈树搜索结果不确定性的弱点, 提出了迭代向前剪枝搜索算法, 并将其应用于中国象棋计算机博弈游戏, 定性地分析了所提高的搜索效率. 通过计算机对弈实验的结果验证迭代向前剪枝搜索算法的效率及效果.

1 相关工作

1.1 中国象棋计算机博弈基本流程

完整的中国象棋计算机博弈一般包括 6 个部分: 棋盘表示、局面评估函数、博弈树、搜索算法、开局库和残局库, 其状态演化流程示意图如图 1 所示.

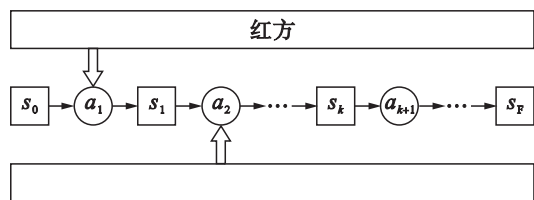


图1 象棋博弈状态演化过程

Fig.1 Evolution process of Chinese chess game state

棋局状态是在着法算子作用下演化的, 对应的状态转移方程为

$$\begin{cases} s_{k+1} = s_k a_{k+1} \\ s_0 = s(0) \end{cases} \quad (1)$$

式中: s_0 为象棋的初始局面; a_{k+1} 为第 $k+1$ 步的着法算子; s_{k+1} 为 $k+1$ 步后的棋局. 完整博弈展开后, 则有

$$s_F = s_0 a_1 a_2 \cdots a_F = s_0 A \quad (2)$$

式中: s_F 为终局, 或红胜、或黑胜、或和棋; $A = \{a_1, a_2, a_3, \dots, a_F\}$ 为着法序列, 记载着博弈过程, 其单数项记载着红方着法序列, 偶数项记载着黑方着法序列. 每步着法都是博弈者的决策, 博弈者通过前瞻若干步形成其当前决策. 图 2 给出了博弈思维过程的计算机实现示意图.

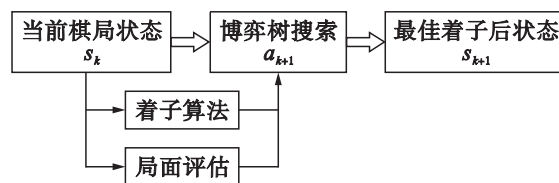


图2 计算机博弈思维过程

Fig.2 Thinking process in computer game

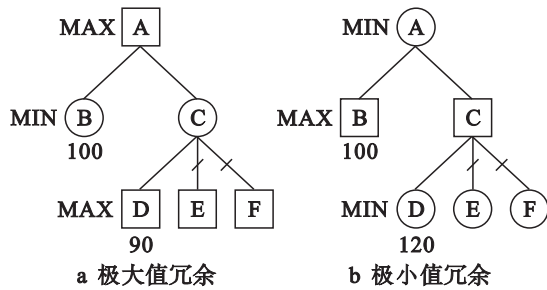
1.2 博弈树搜索

计算机象棋博弈按照图 2 “思维”方式在弈棋过程中展开一棵根在上、叶在下的博弈树. 博弈树搜索的任务是在弈棋当前局面下, 搜索符合象棋规则的着法, 得到一个当前局面下的最佳着子决策.

极大极小搜索是基于深度优先的博弈树搜索, 任一时刻只考虑树中某一路径上的节点. 在象棋博弈中, MAX 先行, 弈棋双方轮流着子, 直到博弈结束. 效用函数定义博弈者在终止状态 s_F 下的数值, 对应胜、平、负一般可以赋予数值 +1、0、-1.

极大极小搜索始终站在博弈 MAX 一方的立场上评估棋局, 有利于 MAX 的棋局给予较高估值, 不利于 MAX 的棋局给予较低估值. 弈棋过程中, MAX 行棋时选择估值极大的子节点着子, MIN 行棋时则选择估值极小的子节点扩展. 为避免极大极小搜索中总要检查哪一方取极大值、哪一方取极小值而造成执行不同的动作, 1975 年 Knuth 等提出了负极大值算法, 其搜索效果与极大极小搜索完全等效. 极大极小搜索一般需要检查 $O(b^m)$ 个节点, b 为搜索宽度, m 为截断深度.

在极大极小搜索过程中, 存在极大值冗余和极小值冗余 2 种明显的冗余现象, 图 3 给出了冗余示例. 图 3a 表示极大值冗余及对应的 α 剪枝, 图 3b 表示极小值冗余及对应的 β 剪枝. $\alpha\beta$ 剪枝搜索实现了对极大极小搜索中冗余现象的优化.

图3 α 剪枝和 β 剪枝Fig. 3 α -pruning and β -pruning

α - β 剪枝搜索在路径上回传的2个参数值: α 为到目前为止路径上发现的MAX的最佳选择; β 为到目前为止路径上发现的MIN的最佳选择.对于 d 层的节点 n , $\alpha-\beta(n) = \max\{\min_i(z_j)\}$, $j=1,2,\dots,b$, z_j 属于由 $d+1$ 层节点展开的、在 $d+2$ 层上对应的分支节点 $i=1,2,\dots,b$, $\min_i(z_j)$ 属于由 d 层节点 n 展开的、在 $d+1$ 层上对应的分支节点. α - β 剪枝搜索不断更新 α - β 值,当某个节

$$H_MAXMIN(s_i, d) = \begin{cases} E(s_i) & (\text{CUTOFF_TEST}(s_i, d, m) == \text{TRUE}) \\ \max\{H_MAXMIN(s_i + 1, d + 1)\} & (s_i \in \text{MAX_NODE}) \\ \min\{H_MAXMIN(s_i + 1, d + 1)\} & (s_i \in \text{MIN_NODE}) \end{cases} \quad (3)$$

式中: $\text{CUTOFF_TEST}(s_i, d, m) == \text{TRUE}$ 为节点 s_i 当前深度 d 到达截断深度 m (即 $d_m - d_c = m$, d_m 为当前节点深度; d_c 为评估节点深度); $E(\cdot)$ 为评估函数.

实时行棋决策根据博弈规则许可的时间决定截断深度 m ,常见的2种实时行棋策略分别为截断搜索和向前剪枝搜索.截断搜索中较为优秀的方法是迭代加深搜索,迭代加深搜索可以帮助行棋排序.逐层加深搜索比固定深度截断搜索更加适合于对弈过程的时间控制.逐层加深搜索在进行 d 层搜索时记录得到的最佳行棋,当进入 $d+1$ 层搜索时优先搜索 d 层记录下来的最佳行棋, d 层的最佳行棋与 $d+1$ 层的最佳行棋存在很大程度的相似, d 层搜索可被 $d+1$ 层搜索利用并提供启发信息.因此,迭代加深搜索比非迭代加深搜索效率更高.多数博弈程序都采用迭代加深搜索,并在迭代加深中加入置换表、历史表和杀手列表等启发信息.利用不同窗口搜索技术产生更好的行棋排序效果,不断提高搜索效率,为象棋博弈实时行棋决策提供有力的支撑.

实时行棋决策的另一种方法是向前剪枝搜索,又称为基于概率的剪枝方法,Buro将此技术应用于Othello取得了良好的效果.在搜索效率方面具有优势.该方法使用先验经验的统计信息,在一定程度上保护最佳行棋不被剪掉.这种算法首

点的值分别比目前的MAX的 α 或MIN的 β 值更差时,就剪去此节点剩下的分支. α - β 剪枝搜索如果首先调用评估函数检查到的后继节点是可能的最佳后继,则其最高效率只需调用评估函数检查 $O(b^{m/2})$ 个节点.如果 α - β 剪枝搜索后继状态采用随机顺序,则需检查的总节点约为 $O(b^{3m/4})$ 个. α - β 剪枝搜索节省了大量开销而不失穷尽搜索的本性.

极大极小搜索生成整个博弈空间, α - β 搜索剪裁掉其中的一大部分,然而,仍然需要搜索部分空间直到终止状态.一般博弈游戏通常要在合理的时间,如几分钟内确定行棋. Claude Shannon在1950年提出尽早截断搜索,将启发式评估函数用于评价搜索中的状态,有效地将非终止状态节点转变为截断终止节点,用启发式评估函数取代效用函数,用阶段测试取代终止测试.截断搜索的启发式评估函数的一般形式为

$$(\text{CUTOFF_TEST}(s_i, d, m) == \text{TRUE})$$

先通过浅层搜索计算得到节点的倒推估值,如果倒推估值在当前的 α - β 窗口之外,则对此棋面不进行更深的搜索,否则进行更深搜索以得到更好的 α - β 值.而利用浅层搜索对深层搜索进行估算所产生的开销相比深层搜索的开销而言可以忽略不计,从而节省大量搜索时间,进一步提高了搜索效率.然而,向前剪枝搜索与迭代加深搜索相同,可能剪掉非终止状态的最佳行棋,从而出现错误行棋,最终导致结果的不确定性.

2 迭代向前剪枝搜索

2.1 评估函数

评估函数在人工智能和模式识别领域有着广泛的应用.象棋行棋局面评估就是利用经验知识给棋局赋分,通过数值评估棋局的好坏.一般来说,象棋评估函数包括5个方面的要素,分别为子力估值、棋子位置估值、棋子灵活度估值、棋子配合估值、威胁与保护估值.每一方面估值又由许多值构成,如子力估值包括7类棋子的分值.棋子配合估值中担子炮、连环马等都需要考虑增加分值.评估函数的设计需要遵从3个基本原则:1)评估函数对终止状态的排序应该与真正的效用函数排序的结果一致,即胜局状态的评估值一定好于平局状态的评估值,平局状态的评估值一定好于负

局状态的评估值; 2) 评估函数的计算不能花费太长时间; 3) 对于非终止状态, 评估函数应该与获胜几率密切相关.

本文中的获胜几率并不是指几率博弈, 而是在搜索过程中绝大多数的截断节点只能发生在非终止状态, 从而导致搜索对这些状态的最后结果是不确定的. 引入这种不确定性的原因是计算局限性, 而不是信息受限.

按上述基本原则, 本文设计了子力估值 e_1 、棋子位置估值 e_2 、棋子灵活度估值 e_3 、棋子配合

估值 e_4 、威胁与保护估值 e_5 等 5 个部分, 表 1、2 分别给出了部分估值示例. 利用 5 个估值要素的线性加权 w_k 评估棋局状态 s_i , 评估函数为

$$E(s_i) = \sum_{k=1}^5 w_k e_k(s_i)$$

(4)

表 1 子力估值

Tab. 1 Estimating values for pieces

帅(将)	仕(士)	相(象)	车	马	炮(砲)	兵(卒)
10 000	20	20	165	70	70	8

表 2 车的位置估值

Tab. 2 Position values for rook

位置 1	位置 2	位置 3	位置 4	位置 5	位置 6	位置 7	位置 8	位置 9
11	15	6	15	6	15	6	15	11
10	15	12	16	10	16	12	15	10
10	11	11	12	10	12	11	11	10
2	10	2	10	4	10	2	10	2
2	10	2	10	2	10	2	10	2
2	12	2	15	2	15	2	12	2
2	6	2	6	2	6	2	6	2
3	7	3	7	3	7	3	7	3
4	8	4	2	2	2	4	8	4
-9	6	2	8	2	8	2	6	-9

2.2 iProbcut 搜索

向前剪枝搜索在效率上与其他搜索方法相比具有明显优势. 鉴于此, 本文提出了一种新的迭代向前剪枝搜索算法 iProbcut, 并将其应用于中国象棋博弈.

iProbcut 搜索的基本思想是利用向前剪枝搜索截断层 m 与预评估层 $m-l$ 节点间的迭代调用,

$$f(s_i, m) = \begin{cases} E(s_i) & (\text{CUTOFF_TEST}(s_i, m) == \text{TRUE}) \\ \max\{f(\arg \text{rank}_{(n, p_k)} g(s_j, m-l), m-1)\} & (s_j \in \text{MAX_NODE}) \\ \min\{f(\arg \text{rank}_{(n, p_k)} g(s_j, m-l), m-1)\} & (s_j \in \text{MIN_NODE}) \end{cases}$$

(5)

$$g(s_j, m-l) = \begin{cases} E(s_j) & (\text{CUTOFF_TEST}(s_j, l+1) == \text{TRUE}) \\ f(s_j, m-l) & (\text{其他}) \end{cases}$$

(6)

式中: $\arg \text{rank}_{(n, p_k)} g$ 为 f 调用 g 时对预评估层 $m-l$ 内节点实施预评估向前剪枝搜索, 并对其排序, 取最佳值对应的前 n 个节点, $\arg \text{rank}_{(n, p_k)}$ 为以概率 p_k 选取 \max 或 \min 值排序最佳的前 n 个节点; \max 和 \min 分别定义了对排序后的节点实施迭代向前剪枝搜索的评估值.

在迭代向前剪枝搜索函数 f 和预评估向前剪枝搜索函数 g 的相互迭代调用过程中, 对预评估层节点同样执行基于 $\alpha\beta$ 的向前剪枝, 对剪枝后的节点实施排序, 从而利用预评估排序得到的优

势节点进一步展开迭代向前剪枝搜索. 在迭代向前剪枝搜索过程中, 迭代向前剪枝搜索函数 f 通过调整预评估向前剪枝排序后的前 n 个节点的概率 p_k 取值系数 k , 对预评估向前剪枝搜索迭代过程中的远端节点增加选取概率, 从而达到在变化棋局时增加参与迭代向前剪枝搜索节点数量的目的, 以此增强 iProbcut 对应变化棋局的搜索效果, 降低非终止状态下误剪最佳行棋的几率, 提高向前剪枝搜索的效果. 因此, 在保证 iProbcut 搜索中的预评估向前剪枝搜索对比 $\alpha\beta$ 搜索更加高效的

同时,进一步保证对比迭代加深搜索实现更深层次的预评估,而且能够降低向前剪枝搜索在非终止状态下误剪最佳行棋的几率,实现提高向前剪枝搜索效率和效果的设计目标。

3 实验与分析

$\alpha\beta$ 剪枝搜索不失极大极小搜索对棋局评估的本性,在截断层数相同的情况下,其对弈效果优于迭代加深搜索或向前剪枝搜索。因此,本文在进行对弈结果的比较时,采用 iProbcut 迭代向前剪枝搜索与 $\alpha\beta$ 剪枝搜索的对弈实验。在搜索效率方面,通过比较各棋局调用节点评估函数的次数,验证 iProbcut 搜索的执行效率。

实验将 $\alpha\beta$ 剪枝搜索的截断层数 m 设为 6,为了在对等的条件下进行结果比较, iProbcut 搜索的截断层数 m 设为 8,预评估层数 $m-l$ 设为 2,也就是说, iProbcut 搜索中的 $\alpha\beta$ 搜索截断层数为 6,不使其优于 $\alpha\beta$ 搜索 p_k 的参数 k 随 $m-l$ 调整,由于预评估中存在排序剪枝,故而 iProbcut 搜索在其 $\alpha\beta$ 搜索截断层数 6 中保留的节点数目远少于 $\alpha\beta$ 搜索的节点数目。iProbcut 搜索与 $\alpha\beta$ 剪枝搜索在同一计算机上对弈 20 局,轮流采用相同的先手开局形式。实验运行环境为 Intel Pentium Dual 1.80 GHz 处理器,4 GB 内存,32 位 Windows 7 操作系统,Microsoft Visual Studio 2015 开发环境。本文测试的主要目的是验证 iProbcut 搜索的效果和效率,因此省略了图形可视化界面,采用 console 方式展现对弈棋局。棋局截图如图 4 所示。运行环境对本文实验结果并不产生影响,主要目的是说明 iProbcut 可在普通个人计算机上执行。

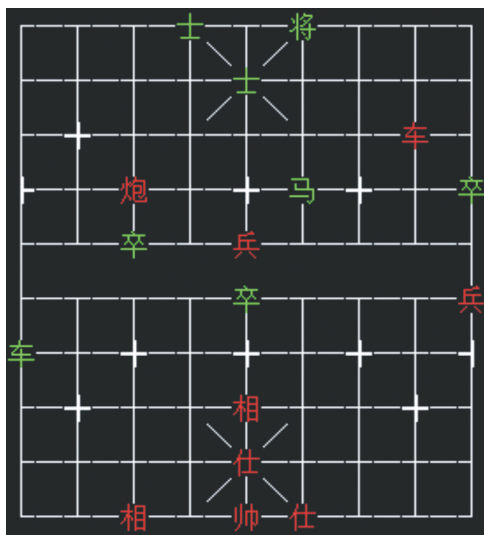


图 4 棋局截图

Fig. 4 Game screenshot

表 3 给出了上述设置下的对弈结果,其中 iProbcut 搜索胜 13 局、平 5 局、负 2 局,取得了很好的效果。表 3 中“终盘着数”表示每局双方一共用的着数,采用 300 着无胜负判和棋,其中胜局中的最多着数出现在局 12,220 着,最少着数为局 4,58 着,平均着数为 113 着。“节点数目倍数列 1”表示 iProbcut 搜索调用评估函数的次数与同一着中 $\alpha\beta$ 剪枝搜索调用评估函数次数的倍数值,“最大值”表示各局中最大的倍数值,“平均值”表示各局中倍数值的平均数。最大倍数值出现在平局的局 16 中,最小倍数值出现在胜局的局 12 中,平均倍数值约为 15,说明 iProbcut 搜索能够将调用评估函数次数的变化控制在稳定的区间内。本文中的 $\alpha\beta$ 剪枝搜索截断层数 m 为 6, iProbcut 搜索截断层数 m 为 8,中国象棋每着平均 45 分支,按此计算,截断层数为 8 与截断层数为 6 相比,其调用评估函数次数的倍数值平均约为 2 500 倍。可知 iProbcut 搜索与 $\alpha\beta$ 剪枝搜索调用评估函数次数的倍数值之比约为 15:2 500,说明 iProbcut 搜索在提高了对弈效果的同时具有良好的搜索效率。作为参考,“节点数目倍数列 2”给出了 6 层截断 $\alpha\beta$ 剪枝搜索调用评估函数次数与 iProbcut 搜索中 $\alpha\beta$ 剪枝搜索 6 层截断时调用评估函数次数的倍数值。

表 3 对弈结果

Tab. 3 Game results

棋局 序号	胜负 情况	终盘 着数	节点数目倍数列 1		节点数目倍数列 2	
			最大值	平均值	最大值	平均值
1	胜	129	27.4	10.7	4.8	1.3
2	胜	120	20.9	9.2	3.1	1.1
3	胜	97	32.9	14.9	2.7	1.1
4	胜	58	27.2	17.5	1.5	1.0
5	胜	75	23.4	13.0	1.6	1.0
6	平	300	45.5	17.1	2.5	1.1
7	胜	113	27.3	13.3	2.2	1.1
8	胜	112	26.3	15.6	1.8	1.1
9	负	90	20.8	11.3	2.1	1.0
10	平	300	27.9	15.0	5.4	1.1
11	胜	83	22.2	14.7	2.1	1.0
12	胜	220	20.1	10.8	3.3	1.0
13	负	76	21.4	14.6	2.5	1.1
14	平	300	105.1	22.8	4.0	1.1
15	平	300	23.2	11.8	2.1	1.1
16	平	300	131.5	18.3	2.4	1.1
17	胜	147	22.6	10.7	2.6	1.0
18	胜	182	30.0	9.8	2.7	1.0
19	胜	79	24.9	16.0	1.9	0.9
20	胜	62	22.8	16.0	1.8	1.0

图5给出了代表性棋局各回合中调用评估函数次数的倍数值,其中棋局4、7、12、16分别对应胜局中的最少着数局、最多着数局、平均着数局、以及调用评估函数次数的最大倍数值所出现的平局局16中。可以看出,胜局中调用评估函数次数的倍数值基本上是稳定的;而调用评估函数次数的倍数值的增大出现在双方进入对弈后期,增大 p_k 的系数 k 后,在最大限定时间内剪枝数目明显降低的阶段。图6给出了代表性棋局中iProbcut调用评估函数的次数,与图5相结合可以看出,调用评估函数次数的倍数值并不是随着调用次数的增大而增大,调用评估函数次数增大时,调用评估函数次数的倍数值变化基本平稳,调用评估函数次数的倍数值增大时,调用评估函数的次数相对数值不大且比较稳定。图6中棋局7与棋局12在对弈开局阶段出现调用评估函数近1亿次的大的波动,主要原因在于开局阶段 p_k 的系数 k 相对较小,出现概率性浅层迭代调用频次突增。对比图5可以看出,调用评估函数次数的倍数值保持稳定,满足实时行棋的要求。另外,调用评估函数次数及倍数值的大的波动主要出现在开局和残局阶段,主要原因在于本文主要目的是进行搜索效率及效果的比较,没有引入通常的开局库和残局库。代表性棋局中调用评估函数次数及倍数值的结果分析进一步说明了iProbcut搜索具有良好搜索效率。

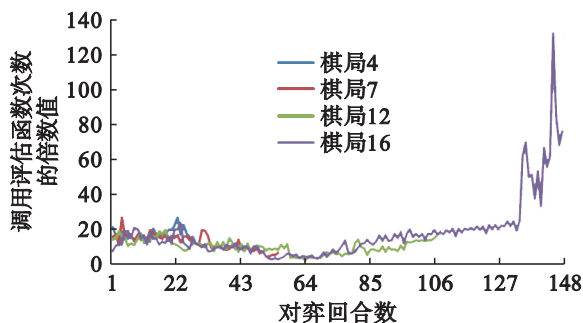


图5 调用评估函数次数的倍数值

Fig. 5 Times of number of evaluation function visited

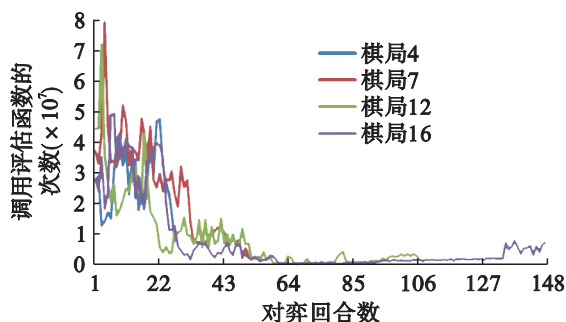


图6 iProbcut 调用评估函数的次数

Fig. 6 Number of evaluation function visited by iProbcut

4 结 论

本文基于向前剪枝搜索与预评估搜索间的双重迭代,提出了博弈树迭代向前剪枝搜索方法iProbcut。iProbcut通过迭代更新预评估节点的排序,动态增加预评估搜索中远端节点选取的概率,增大最佳节点的保留几率,改进了基于先验经验的向前剪枝搜索中非终止状态的最佳行棋剪枝策略,进一步加深了迭代向前搜索在有效方向上的搜索深度,故而提高了向前剪枝搜索的效果。iProbcut迭代向前剪枝搜索函数 f 和预评估向前剪枝搜索函数 g 的迭代调用,利用浅层搜索对深层搜索进行估算所产生的开销对比深层搜索的开销可以忽略不计的特性,迭代更新预评估节点,增加参与迭代向前剪枝搜索有效节点,进一步保证对比迭代加深搜索实现更深层次的预评估,从而提高向前剪枝搜索的效率。20局对等棋局的中国象棋对弈实验中,iProbcut对比 $\alpha\beta$ 剪枝搜索取得了13胜、5平、2负的良好效果。结果与分析表明,截断层数相同为8的情况下, $\alpha\beta$ 剪枝搜索调用评估函数次数大约是iProbcut搜索调用评估函数次数的160倍,iProbcut搜索提高了搜索效率,保证了搜索的稳定性,实现了提高迭代向前剪枝搜索效率和效果的目标。

参考文献(References):

- [1] 王亚杰,邱虹坤,吴燕燕,等. 计算机博弈的研究与发展[J]. 智能系统学报, 2016, 11(6): 788-798.
(WANG Ya-jie, QIU Hong-kun, WU Yan-yan, et al. Research and development of computer games [J]. CAAI Transactions on Intelligent System, 2016, 11(6): 788-798.)
- [2] 李学俊,王小龙,吴蕾,等. 六子棋中基于局部“路”扫描方式的博弈树生成算法[J]. 智能系统学报, 2015, 10(2): 267-272.
(LI Xue-jun, WANG Xiao-long, WU Lei, et al. Game tree generation algorithm based on local-road scanning method for connect 6 [J]. CAAI Transactions on Intelligent System, 2015, 10(2): 267-272.)
- [3] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search [J]. Nature, 2016, 529(7587): 484-489.
- [4] 李焕哲,吴志健,郭肇禄. 两阶段多峰优化算法求解纳什均衡[J]. 武汉大学学报(理学版), 2016, 62(5): 444-450.
(LI Huan-zhe, WU Zhi-jian, GUO Zhao-lu. Two-stage multimodel optimization algorithm for solving nash equilibrium [J]. Journal of Wuhan University (Natural Science), 2016, 62(5): 444-450.)
- [5] 蔡岫. 一种中国象棋机器博弈剪枝策略的改进方法[J]. 国外电子测量技术, 2016, 35(3): 47-49.
(CAI Shen. Improved pruning strategy of Chinese

- chess machine game [J]. Foreign Electronic Measurement Technology 2016 35(3):47-49.)
- [6] 王溪波,王彬,赵海,等. 基于 HOG 特征的优化区域模板匹配检测 [J]. 沈阳工业大学学报, 2016, 38(6):667-673.
(WANG Xi-bo, WANG Bin, ZHAO Hai, et al. Template matching detection for optimized region based on HOG features [J]. Journal of Shenyang University of Technology 2016 38(6):667-673.)
- [7] Nijssen J A M, Winands M H M. Search policies in multi-player games [J]. International Computer Chess Association Journal 2013 36(1):3-21.
- [8] 吕艳辉,宫瑞敏. 计算机博弈中估值算法与博弈训练的研究 [J]. 计算机工程, 2012, 38(11):163-166.
(LÜ Yan-hui, GONG Rui-min. Study on valuation algorithm and game training in computer game [J]. Computer Engineering 2012 38(11):163-166.)
- [9] 刘子正,卢超,张瑞友. 基于蒙特卡罗树搜索的“2048”游戏优化算法 [J]. 控制工程, 2016 23(4):550-555.
(LIU Zi-zheng, LU Chao, ZHANG Rui-you. An algorithm for the game of 2048 based on Monte Carlo tree search [J]. Control Engineering of China 2016 23(4):550-555.)
- [10] 张海峰,刘当一,李文新. 通用对弈游戏:一个探索机器游戏智能的领域 [J]. 软件学报, 2016 27(11):2814-2827.
(ZHANG Hai-feng, LIU Dang-yi, LI Wen-xin. General game playing: a research field for exploring machine intelligence in games [J]. Journal of Software, 2016 27(11):2814-2827.)
- [11] 岳金朋,冯速. 中国象棋 Alpha-Beta 搜索算法的研究与改进 [J]. 北京师范大学学报(自然科学版), 2009 45(2):156-160.
(YUE Jin-peng, FENG Su. Improvement on Alpha-Beta search algorithm in Chinese chess [J]. Journal of Beijing Normal University(Natural Science) 2009 45(2):156-160.)
- [12] 苏攀,王熙照,李艳. 基于不平衡学习的分类器博弈模型及其在中国象棋中的应用 [J]. 计算机研究与发展, 2011 48(5):841-847.
(SU Pan, WANG Xi-zhao, LI Yan. Modeling chess strategy by classifier based on imbalance learning and application in computer Chinese chess [J]. Journal of Computer Research and Development 2011 48(5):841-847.)
- (责任编辑:钟媛 英文审校:尹淑英)