

五子棋中 Alpha-Beta 搜索算法的研究与改进

程 宇, 雷小锋

(中国矿业大学计算机科学与技术学院, 江苏 徐州 221008)

摘 要: 对五子棋中 Alpha-Beta 搜索算法进行研究。依据五子棋的特点, 提出一种局部搜索的算法, 该算法可直接减少搜索的平均分枝因子。结合 Alpha-Beta 搜索算法效率与子节点着法顺序高度相关的特点, 给出静态评价启发以及迭代深化的方法优化着法顺序。实验结果表明, 该方法能提升 Alpha-Beta 搜索算法的效率。

关键词: 五子棋; Alpha-Beta 搜索算法; 局部搜索; 静态评价启发; 迭代深化; 着法顺序

Research and Improvement on Alpha-Beta Search Algorithm in Gobang

CHENG Yu, LEI Xiao-feng

(School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221008, China)

[Abstract] This paper researches Alpha-Beta search algorithm. According to the characteristics of Gobang, it proposes a local search method, thereby directly reducing the search node branch number; combined with the fact that Alpha-Beta algorithm efficiency is highly relevant to the child node order, this paper proposes the static evaluation heuristic and iterative deepening methods for optimization of move ordering. Experimental results show that this method enhances the efficiency of Alpha-Beta search algorithm.

[Key words] Gobang; Alpha-Beta search algorithm; local search; static evaluation heuristic; iterative deepening; move ordering

DOI: 10.3969/j.issn.1000-3428.2012.17.051

1 概述

人机博弈是人工智能的重要分支, 主要研究如何提高机器的智能水平的问题。提高机器的搜索能力将大大提高机器的智能, 人们在对人机博弈搜索算法研究中产生了大量的研究成果, 其中, 极大极小算法是其中最为基础的算法, 而改进极大极小算法所产生的 Alpha-Beta 搜索算法则是其中最重要的算法之一。Alpha-Beta 搜索算法的效率与平均分枝因子以及着法排列顺序息息相关^[1], 因此, 根据五子棋的特点对算法搜索的平均分枝因子和着法排列顺序进行改进成为提高 Alpha-Beta 搜索算法的效率的关键。为完善 Alpha-Beta 搜索算法, 本文对该算法进行研究, 提出改进的 Alpha-Beta 搜索算法, 即局部搜索、静态评价启发、迭代深化来提高 Alpha-Beta 搜索算法的效率。

2 极大极小算法

在人机博弈中, 搜索的过程是博弈程序就当前棋局对未来棋局的思考过程。这一思考过程和人类走棋的思考过程是一样的, 即当己方在所有着法中选择走一步后, 思考对方能有什么着法, 如此反复下去……, 这一过程就产生了一颗博弈树。在这一过程中, 博弈程序必须假定对方和己方一样的“聪明”, 也就是说己方在选择走最有利于自

己的着法的过程中, 对方也会选择最有利于它自己的着法, 这就是极大极小算法的过程, 顾名思义, 指在搜索的过程中存在取极大值一方和取极小值的另一方。为表述方便, 将取极大值的一方设为 max, 另一方设为 min。max 作为博弈程序一方, 选择价值极大的子节点走棋, min 方作为对方, 为了钳制 max 方, 选择价值极小的子节点走棋, 这就产生了一个极大极小过程。Shannon 在 1950 年首先提出了极大极小算法^[2], 从此奠定了计算机博弈的理论基础^[3-7]。

3 Alpha-Beta 搜索算法

3.1 Alpha-Beta 搜索算法原理

在极大极小算法的搜索过程中, 存在一定程度的数据冗余^[8]。一种称为极大值冗余, 如图 1 所示, 这是一棵极大极小树的某一部分, 节点下数字为该节点的值, 节点 B 的值为 20, 节点 D 的值为 15, 这里, C 为取极小值的 min 节点, 因此节点 C 的值将小于等于 15; 而节点 A 为取极大值 max 的节点, 因此 A 只可能取到 B 的值, 也就是说不再需要搜索 C 的其他子节点 E 和 F 的值就可以得出节点 A 的值。这样将节点 D 的后继兄弟节点减去称为 Alpha 剪枝。另一种称为极小值冗余, 如图 2 所示, 这也是一棵

作者简介: 程 宇(1987—), 男, 硕士研究生, 主研方向: 机器学习; 雷小锋, 副教授、博士

收稿日期: 2011-10-21 **修回日期:** 2011-12-26 **E-mail:** cyxhxm@126.com

极大极小树的某部分, 节点 B 的值为 10, 节点 D 的估值为 19, 这里, C 节点为取最大值 max 节点。因此, C 的值将大于等于 19; 节点 A 为取最小值的 min 节点, 因此 A 的值只能取 B 的值 10, 也就是说不再要求节点 C 的子节点 E 和 F 的值就可以得出节点 A 的值。这样将节点 D 的后继兄弟节点减去称为 Beta 剪枝。

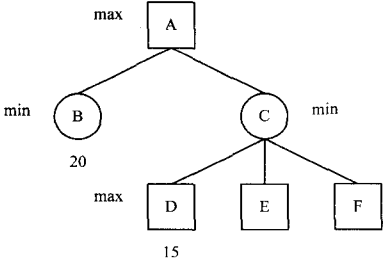


图1 极大值冗余的 Alpha 剪枝

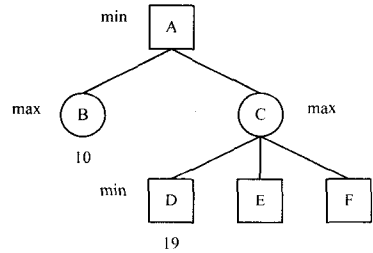


图2 极小值冗余的 Beta 剪枝

通过把 Alpha-Beta 剪枝的原理应用于极大极小算法中就形成了 Alpha-Beta 搜索算法。在搜索过程中 max 节点的当前最大值称为 α 值, min 节点当前最小值被称为 β 值。算法刚开始时取 α 值为 $-\infty$, 取 β 为 $+\infty$, 在搜索过程中 max 节点将搜索更大的值赋给 α 值, min 节点将搜索更小的值赋给 β 值。对于 max 节点, 当在搜索过程中出现了 α 值大于 β 值时, 就进行 Alpha 剪枝并返回 β 值, 如果没有 Alpha 剪枝出现, 则返回 α 值; 对于 min 节点, 当在搜索过程中出现了 α 值大于 β 值时, 就进行 Beta 剪枝并返回 α 值, 如果没有出现 Beta 剪枝, 则返回 β 值。

3.2 Alpha-Beta 搜索算法效率

由极大极小算法的原理可知, 如果将博弈搜索树平均分枝因子记为 b , 搜索深度记为 d , 那么极大极小算法搜索的总节点数为:

$$1+b+b^2+b^3+\cdots+b^d=b^d\frac{1-1/b^d}{1-1/b}\approx b^d$$
 (1)

如果使用 Alpha-Beta 搜索算法, 1975 年, Knuth 等证明在节点排列最为理想的情况下, 搜索的总节点数 n 为^[1]:

$$n=2b^{d/2}-1$$
 (2)

其中, d 为偶数。

$$n=b^{(d+1)/2}+b^{(d-1)/2}-1$$
 (3)

其中, d 为偶数。

这个数字大约是极大极小算法搜索节点数的平方根的 2 倍。由此可以看出, 减少平均分枝因子以及优化着法排列顺序将对 Alpha-Beta 搜索算法效率的提高起到非常大的作用。

4 改进的 Alpha-Beta 搜索算法

本节依据五子棋的特点, 分别提出了 3 种改进方法。局部搜索, 对棋盘进行局部化搜索, 减少平均分枝因子; 静态评价启发和迭代深化方法, 优化着法排列顺序。下文将详细阐述。

4.1 局部搜索

在五子棋博弈的搜索中, 所有空白位置都是合法的着法。对于 10×10 的棋盘, 这意味着棋局刚开始时, 博弈搜索的第 1 层的分枝因子为 $10\times 10-1=99$, 第 2 层为 98, 如果用 d 表示搜索深度, 那么 d 层的分枝因子为 $100-d$ 。如果不考虑减枝的情况, 那么搜索的总节点数, 由式(1)可以看出, 节点数将非常庞大。依据五子棋的特点, 通过仔细分析可以看出, 在这些大量的着法中, 其实很多着法都是没有必要进行搜索的。因为在五子棋中, 博弈的目的是阻止对方形成五连并尽量使己方形成五连, 在实践中, 发现最优走法应该是围绕在棋盘上已经形成的走法周围, 所以没有必要对整个棋盘中的空白位置都进行搜索, 而可以将棋盘进行剪切, 形成一个局部棋盘, 对局部棋盘上的空白位置进行搜索, 这样将大大减少平均分枝因子。

如果把棋盘看成是一个以左下角为坐标原点, 两条边线为坐标轴的坐标系, 对于每一个要进行搜索的节点, 设其为 P , P 中具有最小纵坐标的棋子的坐标为 \min_y , P 中具有最大纵坐标的棋子的坐标为 \max_y , P 中具有最小横坐标的棋子的坐标为 \min_x , P 中具有最大横坐标的棋子的坐标为 \max_x 。可以为 P 产生一个 $m\times n$ 的局部棋盘, 其中, $m=\max_y-\min_y+3$, $n=\max_x-\min_x+3$, 这个棋盘包含 P 中所有的棋子, 也就是说产生了一个等同于 P , 但是却减少了很多无用分枝着法的棋盘。对于 P 的着法的产生将在这个局部棋盘进行, 表 1 表明该方法的结果。

表1 采用局部搜索前后平均搜索节点数对比

搜索深度/层	平均搜索节点数		减少率/(%)
	局部搜索前	局部搜索后	
1	85	51	40
2	963	556	42
3	17 060	5 511	68
4	368 400	35 811	90
5	7 479 000	309 298	96

4.2 静态评价启发

由 Alpha-Beta 搜索算法原理可知, 越早搜索到较优着法, 那么剪枝就将越早得发生, Alpha-Beta 搜索算法的效率也就越高。静态评价启发是本文提出的一种用来优化着法顺序的启发方法, 它使得较优的走法能优先被搜索, 因此可以简单而有效地提高 Alpha-Beta 搜索算法的效率。

在五子棋博弈中, 当前节点最佳的着法可能不是在多层搜索的基础上的最佳的着法, 但是它往往是一个较优的着法。例如, 当前节点能产生一个形成活四的着法, 那么这个着法将是一个最优的着法, 不管还要进行多少层的搜索。因此, 对于每一个要进行搜索的节点, 设为 P , 其每一个着法为 m_i , 每一个着法 m_i 形成局面 P_i , 那么可以对 P_i 进行评估, 产生其评估值 v_i , 如果 P 是极大方, 则以 v_i

为关键字对 m_i 进行非递增排序; 如果 P 是极小方, 则以 v_i 为关键字对 m_i 进行非递减排序。最终把这个排序的着法序列 $\{m_i\}$ 作为 P 的着法搜索顺序, 表 2 表明了此方法的实验结果。

表 2 采用静态评价启发前后平均搜索节点数对比

搜索深度/层	平均搜索节点数		减少率/(%)
	静态评价启发前	静态评价启发后	
1	51	51	0
2	556	157	71
3	5 511	1 378	74
4	35 811	6 062	83
5	309 298	47 819	84

4.3 迭代深化

在搜索的过程中, 针对第 1 个要搜索的节点, 为了得到一个较好的着法作为第 1 个着法, 可以采用迭代深化的方法。当对一个节点进行深度为 d 的搜索时, 可以首先对其进行一次 $d-1$ 层搜索, 得出的最佳着法作为 d 层搜索的最先搜索的着法。由于两层相邻之间的节点比较相似, 因此这一着法很有可能便是最佳的着法或者是较优的着法。由此, 在搜索过程中, 将得到一个较高的剪枝效率^[9]。

在进行 d 层搜索时首先进行 $d-1$ 层的搜索看似多进行了一次搜索, 花费了更多的时间, 但实际上搜索将变得更加有效。文献[1]的实验表明, Alpha-Beta 剪枝搜索 d 层所需时间大约是 $d-1$ 层所需时间的 b 倍, 其中, b 为平均分枝因子。五子棋中平均分枝因子大约为取 $b=200$, 因此, 每多搜一层就会花上原先的 200 倍时间。所以, 对 $d-1$ 层的搜索大约只有进行 d 层搜索的 $1/200$, 这个代价并不大, 但却对 d 层的搜索提供了一个较优的着法的启发, 这使剪枝效率将大大提高, 表 3 表明了此方法的实验结果。

表 3 采用迭代深化前后平均搜索节点数对比

搜索深度/层	平均搜索节点数		减少率/(%)
	迭代深化前	迭代深化后	
1	51	51	0
2	157	118	24
3	1 378	944	31
4	6 062	5 249	13
5	47 819	39 381	17

5 实验结果与分析

选取 10×10 的棋盘, 进行了 15 步着法进行实验。分别在表 1 中对比了局部搜索前与局部搜索后的实验结果;

在表 2 中对比了局部搜索与采用静态评价启发方法的局部搜索的实验结果。最后将迭代深化和静态评价启发方法相结合, 即对第 1 个搜索节点首先进行迭代深化得到第 1 个着法, 然后通过静态评价启发计算其他着法, 表 3 中对比了此结合方法与仅采用静态评价启发的方法的实验结果。

通过实验可以看出, 局部搜索大大减少平均分枝因子, 从各层平均来看, 减少了需要搜索的节点数约 67.2%; 静态评价启发优化着法顺序, 导致大量剪枝的产生, 进一步减少搜索的节点数约 62.4%; 最后结合迭代深化, 更进一步减少了搜索的节点数约 17%, Alpha-Beta 搜索算法效率得到了大大的提高。

6 结束语

本文针对五子棋特点分别提出 3 种改进方法提高 Alpha-Beta 搜索算法的效率, 从实验结果可以看出这 3 种方法是非常有效的。但实验中搜索的层数只做到了 5 层, 这是因为当层数增加时, 平均搜索节点将大大增加, 实验过程将变得非常缓慢, 因此, 如何进一步加深搜索的层数将是下一步研究的方向。

参考文献

- [1] Knuth D E, Moore R W. An Analysis of Alpha-Beta Pruning[J]. Artificial Intelligence, 1975, 6(4): 293-326.
- [2] Shannon C E. Programming a Computer for Playing Chess[J]. Philosophical Magazine, 1950, 41(314): 256-275.
- [3] 王小春. PC 游戏编程[M]. 重庆: 重庆大学出版社, 2002.
- [4] 蔡自兴, 徐光祐. 人工智能及应用[M]. 北京: 清华大学出版社, 1996.
- [5] 姜 勇. 五子棋人机对战系统设计[D]. 成都: 电子科技大学, 2010.
- [6] 张仰森. 人工智能原理与应用[M]. 北京: 高等教育出版社, 2002.
- [7] 陆汝钤. 人工智能上册[M]. 北京: 科学出版社, 1989.
- [8] 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, 2009, 45(2): 156-160.
- [9] 董红安. 计算机五子棋博弈系统的研究与实现[D]. 济南: 山东师范大学, 2005.

编辑 索书志

(上接第 185 页)

- [5] Strunic S L, Rios-Gutierrez F, Alba-Flores R, et al. Detection and Classification of Cardiac Murmurs Using Segmentation Techniques and Artificial Neural Networks[C]//Proc. of IEEE Symposium on Computational Intelligence and Data Mining. [S. l.]: IEEE Press, 2007: 397-404.
- [6] 张 耀, 王耀南, 周博文. 异型瓶药液中可见异物的智能视觉检测机器人[J]. 仪器仪表学报, 2010, 31(5): 1058-1063.

- [7] 张天序, 刘 进. 目标不变矩的稳定性研究[J]. 红外与毫米波学报, 2004, 23(3): 197-200.
- [8] Hu Ming-Kuei. Visual Pattern Recognition by Moment Invariants[J]. IRE Trans. on Information Technology, 1962, 8(2): 179-187.

编辑 张正兴