

博弈树启发式搜索的 α - β 剪枝技术研究

张聪品, 刘春红, 徐久成

ZHANG Cong-pin, LIU Chun-hong, XU Jiu-cheng

河南师范大学 计算机与信息技术学院 智能信息处理实验室, 河南 新乡 453007

Key Laboratory for Intelligent Information Processing, College of Computer and Information Technology, Henan Normal University, Xinxian, Henan 453007, China

ZHANG Cong-pin, LIU Chun-hong, XU Jiu-cheng. Research on alpha-beta pruning of heuristic search in game-playing tree. Computer Engineering and Applications, 2008, 44(16): 54-55.

Abstract: The game playing is an important domain of heuristic search, and its procedure is represented by a special and/or tree. Alpha-beta pruning is always used for problem solving by searching the game-playing-tree. In this paper, the plan which child nodes are inserted into game-playing-tree from large value of estimation function to small one when the node of no receiving fixed ply depth is expanded is proposed based on alpha-beta pruning. It improves the effect of search.

Key words: game playing; heuristic search; alpha-beta pruning

摘要: 博弈是启发式搜索的一个重要应用领域, 博弈的过程可以用一棵博弈搜索树表示, 通过对博弈树进行搜索求解问题的解, 搜索策略常采用 α - β 剪枝技术。在深入研究 α - β 剪枝技术的基础上, 提出在扩展未达到规定深度节点时, 对扩展出的子节点按照估价函数大小顺序插入到搜索树中, 从而在 α - β 剪枝过程中剪掉更多的分枝, 提高搜索效率。

关键词: 博弈; 启发式搜索; α - β 剪枝

DOI: 10.3778/j.issn.1002-8331.2008.16.016 文章编号: 1002-8331(2008)16-0054-02 文献标识码: A 中图分类号: TP18

1 引言

博弈是一类富有智能行为的竞争活动, 如下棋、打牌、战争等, 是启发式搜索的一个重要应用领域, 最简单的一种博弈是双人完备信息博弈。双人完备信息博弈^[1]指两位选手对垒, 轮流走步, 每一方不仅知道对方已经走过的棋步, 而且还能估计出对方未来的走步, 对弈的结果是一方赢, 另一方输, 或者双方和局。本文利用双人完备信息博弈作为研究对象, 为复杂的博弈问题提供参考基础。

双人完备信息博弈过程可以使用一棵特殊的“与或树”表示, 这种与或树被称为博弈树, 通过搜索博弈树实现问题求解。博弈树有下面一些特点:

(1) 博弈的初始状态是初始节点。

(2) 博弈树中的“或”节点和“与”节点是逐层交替出现的。自己一方扩展的节点之间是或的关系, 对方扩展的节点之间是“与”的关系。双方轮流扩展节点。

假设博弈的一方为 A, 另一方为 B。博弈过程的每一步, 供 A 和 B 选择的行动方案都可能有多种。从 A 方的观点看, 可供自己选择的那些行动方案之间是“或”的关系, 原因是主动权掌握在自己手里, 选择哪个方案完全是由自己决定的; 而对那些可供 B 选择的行动方案之间则是“与”的关系, 原因是主动权

掌握在 B 的手里, 任何一个方案都有可能被 B 选中, A 必须防止那种对自己最为不利情况的发生。

(3) 整个博弈过程始终站在某一方的立场上, 所有能使自己一方获胜的终局都是本原问题, 相应的节点是可解节点; 所有使对方获胜的终局都是不可解节点。

求解问题时一般采用搜索博弈树的方法, 利用完整的博弈树进行分析是不现实的, 可行的办法是只生成一定深度的博弈树, 找出当前最好的行动方案, 在此之后, 再在已经选定的分枝上扩展一定深度, 再选择最好的行动方案, 如此进行下去, 直到分出胜负或者和局。每次生成博弈树的深度, 当然是越大越好, 但由于受计算机内存空间的限制, 博弈树的深度根据实际情况而定, 但每次扩展节点时必须至少扩展一层或节点一层与节点。

博弈树的搜索包括盲目搜索和启发式搜索, 盲目搜索效率太低, 解决实际问题时一般不予考虑。常用的启发式搜索有极大极小搜索法和 α - β 剪枝技术。极大极小搜索过程将搜索树的产生和位置评估完全分开, 只有生成规定深度的博弈搜索树后, 才利用估价函数对位置进行评估, 算法的效率比较低。如果在生成博弈搜索树的过程中就对位置评估 (到达了规定的深度, 但博弈搜索树还没有完全生成), 从而剪去一些没用的分枝, 提高算法效率, 这种技术称为 α - β 剪枝过程, 它的搜索效率

基金项目: 河南省自然科学基金 the Natural Science Foundation of Henan Province of China under Grant No.0511012500; 河南省高校新世纪优秀人才支持计划 the New Century Excellent Talent Foundation of Henan Province of China。

作者简介: 张聪品 (1968-), 女, 副教授, 主要研究方向: 人工智能理论、编译技术; 刘春红 (1969-), 女, 讲师, 主要研究方向: 计算机网络; 徐久成 (1963-), 教授, 主要研究方向: 人工智能理论。

收稿日期: 2007-12-12 修回日期: 2008-03-24

比极大极小搜索法^[4]有了很大提高。为了进一步提高博弈树的搜索效率, 本文在研究 α - β 剪枝过程的基础上, 提出了一种改进的方案。

2 α - β 剪枝技术

α - β 剪枝技术把生成后继和倒推值估计结合起来, 及时剪掉一些无用的分枝, 以此提高搜索效率。在搜索过程中, 首先使搜索树的某一部分达到最大深度, 计算出或节点的估价函数值^[9] (称为 β 值) 或者是与节点的估计值 (估价函数值称为 α 值)。对任意一个节点, 当其某一个后继节点的最终值给定时, 就可以确定该节点的上界 (对与节点而言) 或者下界 (对或节点而言)。随着搜索的深入, 按照或节点的 β 值永不上升, 与节点的 α 值永不下降的规则, 不断修改个别节点的 α 值或 β 值。

搜索过程中 α - β 剪枝的规则如下^[9], 剪枝过程如图 1 所示:

(1) 任何或节点 n 的 β 值大于或等于它先辈节点的 α 值, 则 n 以下的分枝可停止搜索, 并令节点 n 的倒推值为 α 。这种剪枝称为 β 剪枝。

(2) 任何与节点 n 的 α 值小于或等于它先辈节点的 β 值, 则 n 以下的分枝可停止搜索, 并令节点 n 的倒推值为 β 。这种剪枝称为 α 剪枝。

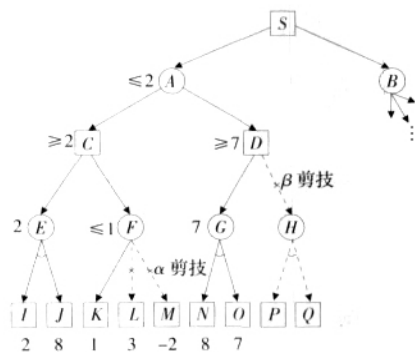


图1 α - β 剪枝过程

3 α - β 剪枝技术的改进

很明显, α - β 剪枝的搜索效率与最先生成的节点的 α 值和最终倒推值之间的近似程度有关。初始节点最终倒推值将等于某个叶节点的静态估值, 如果在搜索过程中, 第一次就遇到这个节点, 则剪枝数最大, 搜索效率最高。最理想情况下, 即与节点先扩展出最小估值的后继节点, 或节点先扩展出最大估值的后继节点, α - β 搜索生成深度为 d 的叶节点数目只有极大极小过程所生成的深度为 $d/2$ 的博弈树的节点数^[9]。反之, 最糟糕的情况下, α - β 剪枝退化成极大极小搜索。

在 α - β 剪枝过程中, 既然子节点的排列顺序影响着根节点的搜索效率, 调整子节点的排列顺序, 就成了提高搜索效率的一种思路。

α - β 剪枝技术在搜索过程中, 首先使搜索树的某一部分达到最大深度, 再计算节点的估计值, 本文在扩展节点过程中, 对于还未到规定深度的节点, 先对其子节点按照估价函数进行初步排序, 具体方案如下:

(1) 扩展节点时, 先判断是否到达规定的深度, 如果没有到达规定的搜索深度, 计算出每个子节点的估价函数值, 对或节点按照函数值从大到小的顺序依次插入搜索树中, 对与节点, 先取估价函数值的相反数, 再按照相反数从大到小排列。取反的目的是可以使用同一个排序过程。如果到达了规定的搜索深

度, 则按照 α - β 剪枝技术进行搜索。

(2) 在搜索过程中, 把本次搜索出的最佳格局存储在一个表中。这样, 在以后扩展节点时遇到该格局时, 可以参考表中各子节点的排列顺序, 不仅可以提高速度, 而且还可以纠正由于估价函数信息不完备带来的偏差。

下面以一字棋游戏作为研究对象, 把改进的 α - β 剪枝技术应用到搜索中。一字棋游戏如图 2 所示的 9 个空格, 由 A、B 二人对弈, 轮到谁走棋谁就在空格上放上自己的一只棋子, 谁先使自己的棋子构成三子一线, 谁就取得了胜利。设 A 的棋子用 \times 表示, B 的棋子用 \circ 表示, 估价函数为: 设棋局为 P , $\phi(P) = \phi_+(P) - \phi_-(P)$, 其中 $\phi_+(P)$ 表示棋局 P 上有可能使 \times 成为三子一线的数目, $\phi_-(P)$ 表示棋局 P 上有可能使 \circ 成为三子一线的数目, 如果 P 是 A 的必胜棋局, 则 $\phi(P)$ 是无穷大, 如果 P 是 B 的必胜棋局, 则 $\phi(P)$ 是无穷小。



图2 一字棋

在扩展节点生成子节点时, 按照从左到右, 从上到下的顺序走棋, 图 3 给出了每次扩展两层节点, A 的第一著走棋生成的博弈树, 从图中可以看到, 如果采用极大极小搜索策略, 生成了 16 个节点, 使用 α - β 剪枝生成 12 个节点, 如果采用改进的 α - β 剪枝技术, 对 S_0 进行扩展时, 对扩展出的子节点 S_1, S_2, S_3 按照估价函数值进行排序, 下次再扩展节点时就会先扩展 S_2 , 当扩展到规定的深度时开始剪枝, 仅生成了 7 个节点。

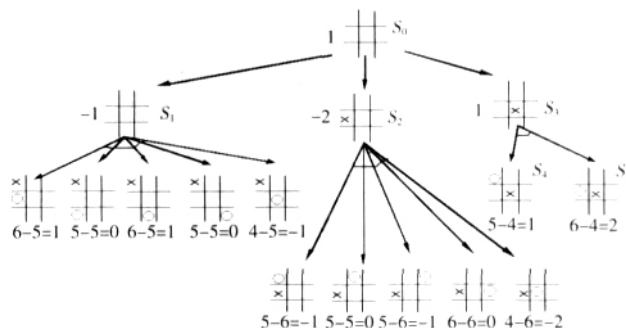


图3 一字棋的极大极小搜索生成的节点

在实验中, 每次扩展四层节点时, 对于 A 的第一著走棋不同的搜索策略生成的节点数如表 1 所示。

表 1 采用不同的搜索策略所生成的节点数

搜索采用的方法	生成的节点数
极大极小搜索	462
α - β 剪枝	121
改进的 α - β 剪枝	50

随着每次扩展层数的增加, 被剪掉的分枝也会越来越多, 排序虽然花费了部分时间, 但是在搜索过程中会节约更多的时间。

另外, 每次搜索结束后, 把最优格局存储下来, 在以后的搜索过程中极有可能遇到该格局, 这样可以直接利用相关的子节点信息。

4 结论

该文在深入研究 α - β 剪枝技术的基础上, 提出了一种对 (下转 97 页)

所描述算法建立路由所需的开销情况。这三条曲线分别是 $M=1, 2, 3$ 的情况。 $K=1$ 且 $M=1$ 可以看作是 GPSR 的开销情况。当 K 为 1 时, 锚点之间只有 1 跳, 找到的第二、三条路径比较曲折, 因此开销较大。本算法主要开销在于 K 跳范围内路由表的维护, 而由 K 的增大带来的建立路由开销并不大。另一方面, 分段维护 K 跳路由带来路由稳定性提升, 因此是值得的。

图 9、图 10 分别表示在小规模和大规模环境中, 平均路由带宽请求成功率。单跳的带宽请求成功率均值 p 分别取 0.95、0.90、0.85。为了体现移动自组网的特性, 如果两节点距离比较远, 单段成功率值就取得比均值略小一些, 反之则稍大一些。 K 邻域值取 3。可以看出, 不论是小规模环境还是大规模环境, 整个路径的带宽请求成功率都随着 M 值显著上升。这表明该算法是可行的。

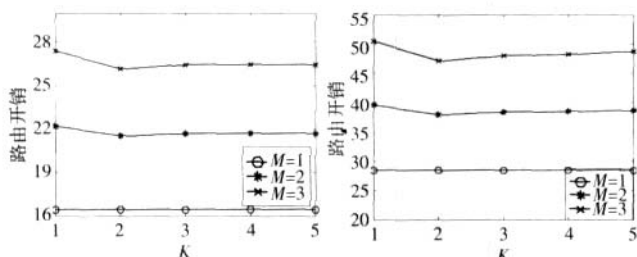


图 7 小规模环境路由开销

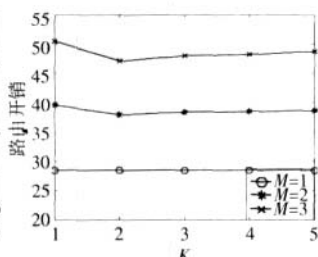


图 8 大规模环境路由开销

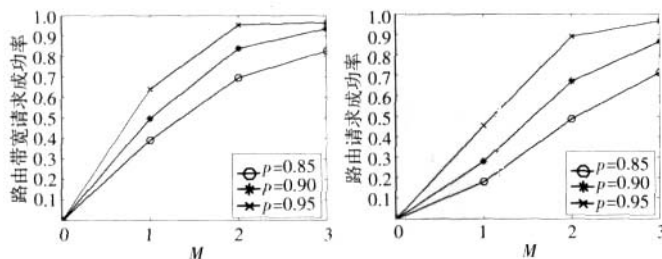


图 9 小规模环境路由请求成功率

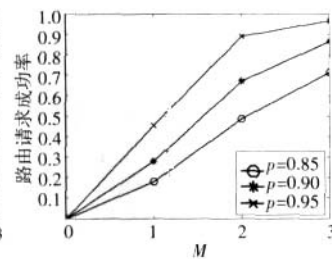


图 10 大规模环境路由请求成功率

5 结束语

分段式节点不相交多路径 QoS 路由协议 (NDQ-SSR) 减少了路由管理开销和路由维护的代价, 有效地提高了网络的扩展性、可靠性和抗毁性等。通过理论分析和模拟验证, 表明该协议能满足多 QoS 约束。

参考文献:

- [1] Perkins C E, Bhagwat P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers[C]//Proceed-

ings of the SIGCOMM 1994, Conference on Communications Architectures, Protocols and Applications, London, UK, August 1994: 234-244.

- [2] Johnson D B, Maltz D A. Dynamic source routing in Ad Hoc wireless networks[M]//Mobile Computing. [S.l.]: Kluwer Academic Publishers, 1996: 153-181.
- [3] Perkins C E, Royer E M. Ad-hoc On-Demand distance vector routing[C]//Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, Massachusetts Ave. NW Washington, DC, USA: IEEE Computer Society, 1999: 90-100.
- [4] Karp B, Kung H T. GPSR: Greedy Perimeter Stateless Routing for wireless networks[C]//Proceedings of the ACM Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000), August 2000: 243-254.
- [5] Liao W, Sheu J, Tseng Y. GRID: a fully location-aware routing protocol for mobile ad hoc networks[J]. Telecommunication Systems, 2001, 18: 37-60.
- [6] Wang Guo-jun, Zhang Li-fan, Cao Jian-nong. Logical location-based routing with hole-shadowing in large-scale MANETs[C]//Proceedings of the 2006 IEEE International Conference on Communications (ICC 2006), Istanbul, Turkey, June 2006: 3560-3565.
- [7] Cao Jian-nong, Zhang Li-fan, Wang Guo-jun, et al. SSR: Segment-by-Segment Routing in large-scale MANETs[C]//Proceedings of the Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2006), Vancouver, Canada, October 2006: 216-225.
- [8] Lee Sung-Ju, Gerla M. Split multipath routing with maximally disjoint paths in ad hoc networks[C]//IEEE International Conference on Communications, 2001, 10: 3201-3205.
- [9] Xiao Wen-dong, Soong B H, Law C L, et al. QoS routing protocol for ad hoc networks with mobile backbones[C]//IEEE International Conference on Networking, Sensing and Control, 2004, 2: 1212-1217.
- [10] Lee Sung-Ju, Gerla M. AODV-BR: backup routing in ad hoc networks[C]//Proceedings of IEEE WCNC 2000, Chicago, IL, Sep 2000: 1311-1316.
- [11] Hassanein H S, Du H, Yeh C-H. Robust route establishment in high mobility MANETs[C]//International Computer Engineering Conference (ICENCO), Cairo, Egypt, December 2004: 74-80.
- [12] Ayyash M, Alzoubi K, Alsobu Y. Preemptive quality of service infrastructure for wireless mobile ad hoc networks[C]//Proceedings of the IEEE/ACM International Wireless Communications and Mobile Computing Conference (IWCMC '06), Vancouver, British Columbia, Canada, 2006: 707-712.

(上接 55 页)

- 剪枝技术进行改进的方案。由于子节点的排列顺序极大地影响了剪枝的数目, 本文提出在扩展节点时, 当未到达规定的搜索深度时, 使用估价函数对扩展出的子节点进行排序, 对或节点按照从大到小的顺序插入到博弈搜索树中, 对与节点按照从小到大的顺序插入到博弈搜索树中, 当到达了规定的搜索深度时, 则进行 - 剪枝过程。另外, 把每次搜索过程中得到的最优格局存储下来, 供以后搜索参考。最后, 以一字棋游戏作为研究对象, 对该方案进行了验证, 实验结果表明, 随着扩展节点层数的增加, 剪掉的分支会快速增加, 因此改进的 - 剪枝技术提高了搜索效率。

参考文献:

- [1] Cohen P R, Feigenbaum E A. The handbook of artificial intelligence[M]. New Jersey: Addison Wesley, 1982: 45-80.
- [2] Clancy W J. Heuristic classification[J]. Artificial Intelligence, 1985, 27: 289-350.
- [3] 王永庆. 人工智能原理与方法[M]. 西安: 西安交通大学出版社, 2003: 290-292.
- [4] Luger G F. Artificial intelligence structures and strategies for complex problem solving [M]. 5th ed. Beijing: China Machine Press, 2006: 110-118.
- [5] 王文杰, 叶世伟. 人工智能原理与应用[M]. 北京: 人民邮电出版社, 2004: 53-61.