

什么是 α - β 剪枝算法

IBM 的深蓝战胜国际象棋大师卡斯帕罗夫很大程度上要归功于 α - β 剪枝算法[2]，那么什么是 α - β 剪枝算法呢？我们从极小-极大过程开始讲起。

1. 极小-极大过程

我们先看看人是如何下棋的。人在下棋时首先根据当前局面考虑若干总可能的走法，再对每种可能的走法考虑对方会如何走，再考虑自己会如何应对……高手会这样往前看很多步，根据最后的局面判断哪种走法是最优的。换句话说，高手会选择那种即使对方正确应对的情况下，己方依然占据最大优势的走法，而不是把希望寄托在对方犯错上。人类棋手的这种思考方法可以用一个“极小极大过程”来描述，其中“极小”是指因为对方的正确应对，使己方收益最小，而“极大”是指假设对方让自己收益最小的前提下，通过己方走棋使自己收益最大。

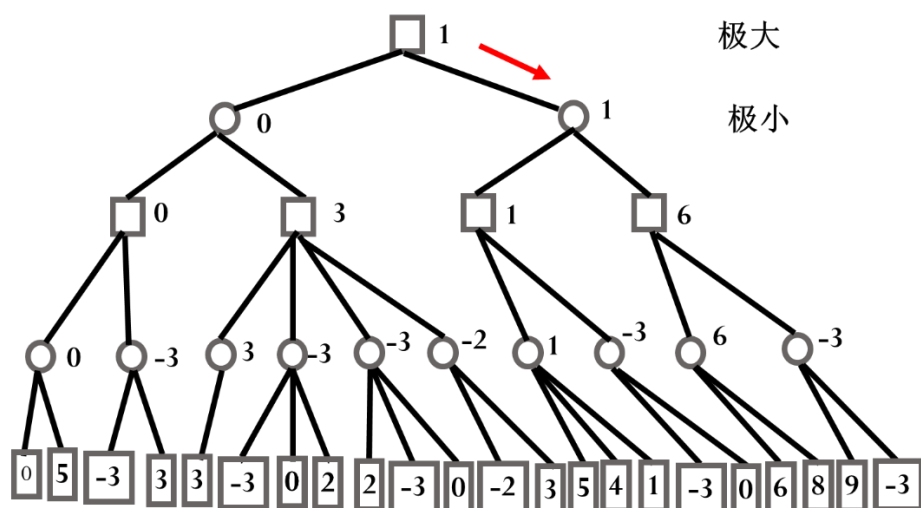


图 1: 极小-极大过程

如图 1 所示，其中方框表示轮到我方走棋，圆圈表示轮到对方走棋。最上面的方框表示当前棋局状态。从当前状态开始向下搜索 4 步，最下方的数字给出了四步之后的棋局得分，数字越大表示对我方越有利，数字越小表示对对方越有利（在深蓝系统中，这些得分来自于国际象棋大师总结的相关知识）。有了这棵搜索树，就可以自底向上倒推每个节点的得分。如何倒推呢？显然，在双方都不犯错的前提下，我方会选择得分更高的走法，对方会选择得分最低的走法。基于这一原则，圆圈节点的分值是其所有子节点的最小分值，而方框节点是所有子节点的最大分值。由此可自底向上得到所有节点的得分，如图 1 所示。

基于图 1 所示的各节点得分，可以看到在双方都没有出现错误的情况下，如果己方向左走将得到 0 分，而向右走的话，将得到 1 分。显然我方应该选择向右走，如红色箭头所示，这将保证无论对方如何应对，我方至少获得 1 分的优势，这就是极小-极大过程。本质上，这一

过程保证在最差情况下（对手完美走棋）选择对自己最有利的走法，因此可表现出稳定的棋力。极小-极大过程由香农于 1949 年提出[3]，是包括深蓝在内的众多计算机对奕机系统的基础。

2. α - β 剪枝

当搜索深度增加，极小-极大过程将产生规模庞大的搜索树，出现“组合爆炸”问题。据深蓝开发者估算，如果不做改进的话，即便每次走棋只往前考虑十步左右，每步棋也需要“思考”17 年。

如何解决这一问题呢？我们先看看人类棋手是如何处理的。众所周知，有经验的棋手在思考可能的走法时，并不是对每种可能性都平权考虑，而是根据自己的经验选择几种可能的走法进行尝试。计算机可以借鉴这一思路，在搜索的过程中减掉一些不必要的路径分枝，以提高搜索的效率，这一方案称为**剪枝**。利用极小-极大过程的特点（我方选择最大子节点，对方选择最小子节点），可以设计剪枝算法，在保证决策不变的前提下，去掉大量不必要的搜索路径。 **α - β 剪枝算法**正是这样一种算法。

我们假定节点是按照深度优先的方式边搜索边产生的，如图 2 所示。从根节点 s 开始，顺序产生 a、b、c 三个节点，达到了搜索深度后，计算 c 的两个子节点的得分。由于 c 是极小节点，所以两个子节点中最小得分 (0) 即为 c 的得分。由于 b 是极大节点，因此 b 的得分最小为 0。由 b 向下扩展生成节点 d 和 e，e 的得分为-3，而 d 又是极小节点，从而我们知道 d 的值最大为-3。到此我们会发现，b 的最小值为 0，d 的最大值为-3，因此节点 f 得分多少已经不重要了，所以 f 可以被剪掉，不需要生成。这样由 b 得分为 0，我们有 a 的最大值为 0。同样我们扩展 a 的另一个子节点，并向向下延伸到 k，得到 k 的分数为 3。由于 h 没有其他的子节点，我们推断极大节点 g 最小为 3。从前面我们知道极小节点 a 最大为 0，它的子节点 g 是极大节点，最小为 3，所以 g 的其他子节点得分是多少也不重要了，可以不需要生成。

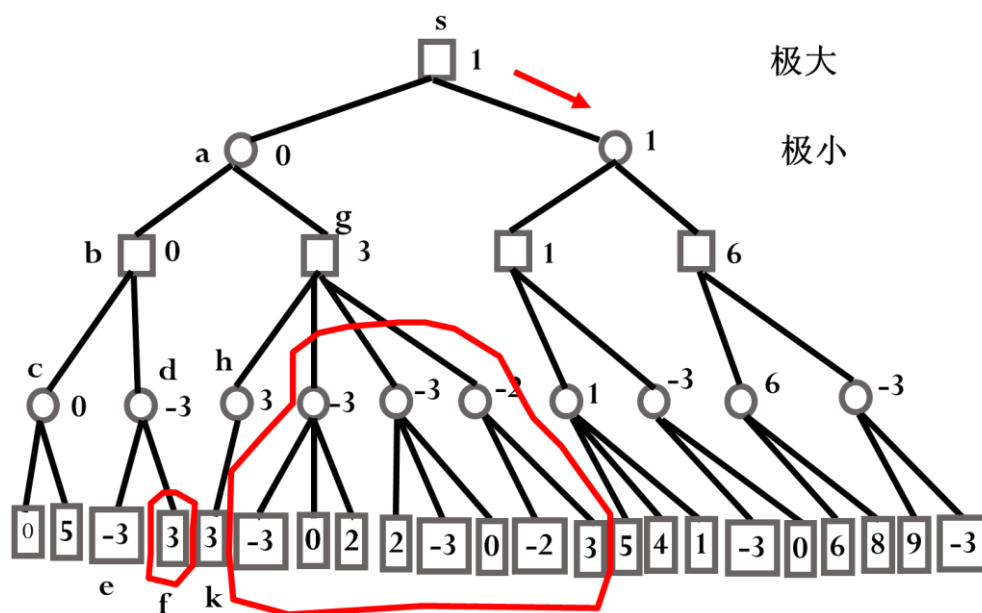


图 2: α - β 剪枝示意图

上面的例子可以总结成如下剪枝规则:

- 1, 后辈极小节点的值 \leq 祖先极大节点的值时, 发生剪枝, 称为 α 剪枝。
- 2, 后辈极大节点的值 \geq 祖先极小节点的值时, 发生剪枝, 称为 β 剪枝。

请注意, 这里发生剪枝的条件都是后辈跟祖先比较, 不只是跟父节点比, 只要有一个祖先满足剪枝条件就发生剪枝。

上述剪枝方法被称为 **α - β 剪枝算法**, 由人工智能创始人之一, 图灵奖获得者约翰-麦卡锡提出 (同期多位学者做过相关研究并独立提出了该算法) [1]。 α - β 剪枝算法显著提高了搜索效率, 允许在有限的时间内做更深的搜索, 从而得到更好的性能。需要强调的是, α - β 剪枝算法是一个“无损”算法, 即剪枝只会提高搜索效率, 不会影响走棋决策。

[1] McCarthy, John (27 November 2006). "Human Level AI Is Harder Than It Seemed in 1955". Retrieved 2006-12-20.

[2] Hsu F. IBM's deep blue chess grandmaster chips[J]. IEEE Micro, 1999, 19(2): 70-81.

[3] C.E. Shannon, "Programming a Computer for Playing Chess," Philosophical Magazine, Vol. 41, 1950, pp. 256 - 275.