

机器博弈中搜索算法的研究

廖景亮, 陈冬强

(泉州师范学院 数学与计算机学院 福建 泉州 362000)

【摘要】: 机器博弈是人工智能领域中一个重要的组成部分, 计算机智能和人类棋手之间展开了长达几十年的竞赛。本文首先简要介绍机器博弈的基本理论和机器博弈系统的一般构成, 然后对几种经典的机器博弈搜索算法及其特点的进行了讨论说明。

【关键词】: 机器博弈; 搜索算法; α - β 搜索; 增强算法

1. 引言

计算机科学的一个重要分支就是人工智能的发展, 而棋类游戏的机器博弈又是人工智能发展的一个重要方向。国内外许多知名学者和知名科研机构都曾经涉足这方面的研究, 历经半个多世纪, 到目前为止已经取得了许多惊人的成就。1997年IBM的“深蓝”战胜了国际象棋世界冠军卡斯帕罗夫, 惊动了世界。除此之外, 加拿大阿尔伯塔大学的奥赛罗(黑白棋)程序 Logistello 和西洋跳棋程序 Chinook 也相继确定为二人、零和、完备信息游戏世界冠军, (所谓二人、零和、完备信息就是在游戏过程的任意时刻只有博弈双方两个人, 任意一方对当前的局面的信息的了解都是足够、充分和公平的, 即不存在概率性的信息, 并且游戏进行到最后必定能分出胜负)。对围棋、中国象棋、桥牌、扑克等许多种其它种类游戏博弈的研究也正在进行中。机器博弈的核心技术是博弈搜索算法, 这也是机器博弈研究的热点。

2. 机器博弈的基本思想

现今机器博弈的核心思想并不复杂, 实际上就是对博弈树节点的估值过程和对博弈树搜索过程的结合。在博弈进行时, 站在两个博弈玩家一方的角度上, 可以计算并构造出一棵博弈树。这棵博弈树的最高节点, 即根节点就是当前时刻所在进行的棋局, 其儿子节点便是假设再走一步棋后的各种棋局, 孙子节点就是从儿子节点的棋局再走一步棋后的各种棋局, 以此类推, 便可构造出一棵完整的博弈树。整棵博弈树的节点数目非常庞大, 并且不一样的棋有不同的情况。

博弈程序要回答的就是要怎么才能在当前的博弈树中找到最好的一步棋。其实, 程序不能也没有必要去搜索整棵构造出的博弈树的所有节点,

对于那些明显是不好的走法就可以不必考虑, 将以它为根的子树剪去。而且, 搜索也没有必要进行到可以确定胜负的棋局, 只要让搜索进行到我们预先要求的深度就行了。若要使搜索能真正地进行下去, 就必须把搜索空间缩到足够小才行。当我们搜索到一定深度时, 就要用局面评价机制对当前的局面进行评价, 给出参考值, 并依据极大极小的原则选出具有最好走法的节点, 然后将该节点返回给上级, 进而对当前局面的父亲节点的价值给予一个比较客观的评价, 然后再继续将所得的节点返回给上级, 直到根节点为止, 这样我们就可以得到最好的走法了。

在搜索过程中, 搜索算法是最重要的, 效率高的搜索算法可以很大程度上为我们节省时间和空间, 从而可以获得更好的走法。但真的想要提高博弈程序的棋力, 还需要一个好的局面评价机制, 就是估值算法。利用估值算法可以比较客观地、正确地评价局面的好坏及其程度。

3. 机器博弈系统的构成

从机器博弈的基本思想出发, 我们能比较容易地确定一个机器博弈系统的一般构成。

首先是如何表示的问题, 也就是要找到一种较为合适的方法来记录棋局。棋盘的表示方法是一个重要问题, 一般方法用二维数组来表示棋盘, 一个位置往往用一个 byte 来表示, 但是一般棋类每一个位置的状态远远小于 256 个。对于很多棋类来说, 比特棋盘都是一种节省空间, 提高性能的有效方法。

其次, 针对不同的棋类, 依据其规则要有一个与之对应的走法产生函数。我们可以用它来构造整棵博弈树, 即我们能通过该函数生成任意节点的所有儿孙子节点, 也就是接下来的所有符合规

则的走法。

然后,就是估值函数。估值函数是基于某些既定规则来评估一个棋局价值的技术,其中涉及棋子的价值评估、棋子的灵活性、棋盘控制能力、棋子间关系的评估等等,还要与所选的搜索算法相结合。该部分与设计者的棋类知识关系很大,而且不同棋类差别很大。

除了以上三个模块以外,就是本文要重点讨论的搜索算法。这四个部分相互配合运转起来,就可以实现性能良好的机器博弈程序。

4. 机器博弈中的搜索算法

博弈搜索的基本思想从提出到现在已经有半个多世纪,目前受到广泛研究的是二人、零和、完备信息的博弈搜索。二人、零和、完备信息的博弈搜索理论已经很系统,最为基础的就是极大极小算法。在此基础上是作为主流的深度优先的 alpha-beta 搜索及其一系列增强算法。

4.1 基本搜索算法

(1) 极大极小值算法 (Minimax Algorithm)

始终站在两个博弈玩家其中某一方的立场上给当前的棋局进行估值,对我方有利的棋局赋予一个较高的分数,不利于我方的给予一个较低的分数,对于不能较为明确地判断出双方优劣的局面则赋予一个中等分数。在我方走棋的时候,选择分数值极大的儿子节点来走,另一方行棋则选择价值极小的儿子节点走。

(2) 负极大值搜索 (Negamax Algorithm)

在进行极大极小搜索时,每一次在极大极小节点的选择时,都要判断此节点是要取极大值或极小值,为了消除两方面的差别使程序变得简单易懂,经过半个多世纪后,Knuth 和 Moore 在 1979 年提出了负极大值算法,该算法只需博弈双方都取极大值。其核心思想在于:父节点的值取得是各子节点的值负数中的最大值。

4.2 深度优先的 alpha-beta 搜索及其增强算法

在进行极大极小搜索时,一个明显的问题就是数据冗余,冗余的数据是必然要删除的,以减小搜索的空间,1975 年 Monroe Newborn 提出的 alpha-beta 剪枝就是我们所要用的方法。在极大极小搜索或者负极大值搜索中加入 alpha-beta 剪枝就形成了重要的 alpha-beta 搜索。

alpha-beta 剪枝的示例如下,如图 1 左半部所示极大极小树的某个片断,节点 B 的值为 10,节点 D 的值为 8,由此容易判断节点 C 的值肯定

小于等于 8(取极小值);而节点 A 的值为节点 B 与 C 的较大值,即为 10,也就是说不用对节点 C 的其他子节点如 E、F 的值进行估计就可以得到父节点 A 的值。这样剪去节点 D 的后继兄弟节点的方法称为 alpha 剪枝。

图 1 右半部所示极大极小树的某个片断,节点 B 的估值为 8,节点 D 的估值为 10,由此容易判断节点 C 的值肯定大于等于 10(取极大值);而节点 A 的值取节点 B 和 C 的较小值,即为 8。也就是说不用再求节点 C 的其他子节点如 E、F 的值便能得出父节点 A 的值。这样剪去节点 D 的后继兄弟节点的方法称为 beta 剪枝。

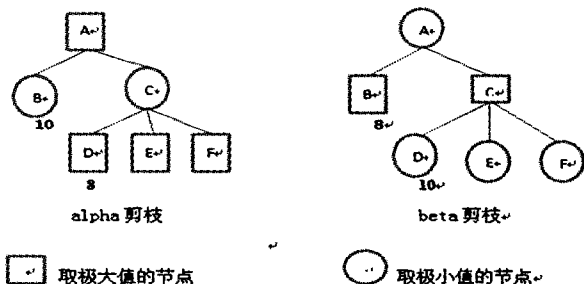


图 1

因为过程简单、理解容易、并且空间占用小等特点,alpha-beta 搜索被广泛应用实现,成为当前主流搜索算法的基础。但其有个缺点就是对于节点的排列顺序非常敏感。对于同一层上的节点,假如排列合理的话,剪枝效率就可以达到很高,使实际搜索的博弈树接近甚至达到最小树。最小树的定义就是一经向下搜索,首个节点就是要查找的走法。可是,一旦节点的排列顺序不是理想中的那样从最差到最好,就有可能无法产生剪枝,实际上就等于搜索了整棵的博弈树。

正常情况下建立的一棵博弈树其节点的排列顺序是没有什么特殊规则的。因为 alpha-beta 剪枝与节点的排列顺序关系密切,所以如果能使用其他方法将节点的排列顺序调整为剪枝效率较高的顺序就显得很重要了。通过对 alpha-beta 搜索特点的研究,很多种增强算法被陆续提出来,其中一些已经被证明是非常高效的,并成了该领域的标准搜索算法。

(1) 渴望搜索 (Aspiration Search)

在用 alpha-beta 剪枝时,初始的搜索窗口的范围一般是从 $-\infty$ (即初始的 alpha 值) 到 $+\infty$ (初始的 beta 值),在搜索中再根据某些条件不断减小窗口,从而提高剪枝效率。

渴望搜索顾名思义就是希望在开始的时候就得到尽量小的窗口,以便在搜索开始的阶段,就能

剪去许多不必要的枝。显然,怎么选定初始窗口是一个很重要的问题,假如选择得好,即最好的走法就落在这个窗口中,这样搜索就可以继续进行下去。假如第一步搜索所返回的值被证明出最好的走法应该大于 β 值,则需要重新定义窗口,以之前的 β 值为新的 α 值,以 $+\infty$ 为新的 β 值,然后重新进行搜索。反之,假如第一步搜索所返回的值被证明出最好的走法应该小于 α 值,那么重新定义的窗口,以 $-\infty$ 为新的 α 值,以原来的 α 值为新的 β 值,然后重新进行搜索。作为基本的搜索方法,渴望搜索和后面将会叙述的遍历深化结合时,就能依据上一次搜索的结果来猜测当前搜索的窗口了。

(2) 极小窗口搜索 (Minimal Window Search / PVS)

极小窗口搜索,也被称为是主变量导向搜索 (Principal Variation Search), 简称 PVS 算法或者 NegaScout 算法,其原理比较类似于渴望搜索,即用尽可能小的窗口来产生尽可能多的剪枝。过程如下:在根节点处,假设第一个儿子节点为主变量(最好走法),然后对该节点进行完整窗口 (a,b) 的搜索并返回值 v , 对该节点的兄弟节点依次用极小窗口(又称零窗口) $(v,v+1)$ 来进行搜索,若搜索过程中返回的值大于零窗口,那么就证明这一分支也是主变量,对它进行搜索的窗口就为 $(v+1,b)$ 。但是若返回值小于零窗口,则证明最好走法比已经得到的走法还差,因此可以忽略。

极小窗口搜索所使用的是极小的窗口,使得剪枝具有最高的效率,而且只是针对主变量进行大窗口的搜索,因此所进行的大部分的搜索都是有用的,搜索所产生的博弈树也是比较小的。某种程度上说小窗口也是 PVS 的一个缺点,由于窗口宽度太小,许多时候最好走法的值会落在窗口外面,等于先前进行的搜索没有一点用处,浪费了很多时间。但总的来看,在效率上极小窗口搜索是比 α - β 搜索高的,最差也是相等,许多研究者的实践也证实了这一点。因此 PVS 算法在实际的应用中相当普遍,被许多博弈系统的设计者所熟知采用。

(3) 迭代加深 (Iterative Deepening)

一般来说,程序搜索得越深,程序的棋力就越强。如果搜索深度足以算到终盘,就能得到真正的最佳棋步和精确比分。但是随着搜索深度的增加,搜索所花费的时间也将急剧上升。而在现实下棋或比赛中,时间往往是有限的。由于受到时间的制

约,程序的搜索深度无法事先确定,而需要在下棋过程中动态地进行调整。为此,程序在下每步棋时,可以根据己方剩余时间的多少,合理分配当前这一步棋的思考时间。但由于程序软硬件环境的不同和局面的不断变化,搜索深度和搜索时间之间没有一成不变的关系,搜索深度仍然无法确定。固定深度的搜索已无法适用,这时可以采用迭代加深搜索。

迭代加深搜索的思路十分简单,先从很浅的搜索深度(比如初始搜索深度为 1)开始进行搜索,然后逐渐加深搜索深度进行下一轮搜索,多次对博弈树进行遍历,并且深度不断增加,直至时间用完为止,因此又名蛮力搜索。这样就可以做到有多少时间,就搜索多少深度。当然,在实际应用中,算法子函数内部也应进行超时判断。如果在某一深度的搜索过程中时间用完,搜索应立即中止,以免因等待本轮搜索结束而造成时间耗尽。

该算法的遍历每次都是从根节点开始,看似蛮力,但实际上由于每次都能从前一次的结果中得到启发,优先搜索相对好的节点,提高了剪枝的效率,使得算法的效率也很高,目前该算法的性能也得到了广泛的认可。顶级的博弈程序几乎都把迭代深化作为调整节点顺序、改进搜索效率的重要手段。

(4) 置换表 (Transposition Table)

当我们搜索查询所有的走法时,往往会遇到同样的棋局,这时就没有必要对子树进行重复的搜索,将子树根节点的一些关键信息,如最好走法、估值及取得这个值的深度等等,保存在某种数据结构中,以便下次遇到的时候可以直接运用,这种数据结构我们称之为置换表。

当然,对一些不同的情况做法也有所区别。进行深度固定的搜索时,若上一次保存的子树的深度小于当前新节点所要搜索的深度,那么为了保证所取得数据的准确性,就必须进行重新的搜索。反之,若由置换表可知,子树的深度大于或者等于当前的新节点所要搜索的深度,那么我们就可以直接利用其信息。

若能将置换表增强和有效的 α - β 搜索相结合,那么实际搜索的博弈树就会小于最小树。在图论中,可将博弈树的搜索看成是对有向图的搜索。其中在置换表中被选中的节点,就可看成是有向图中几条路径的交叉点。因此,若要使搜索过程中产生的搜索树小于最小树,就要尽量避免对这些交点进行重复搜索。

这个算法存在的问题是,置换表的构造,必须使对置换表的查找和插入操作不成为整个搜索的负担,才能提高效率。一般使用散列表来实现。这边需要指出的是对冲突的解决。与一般想法不同,设计中我们通常不会用链表或重定位的方法来解决冲突,而是在发生冲突时直接覆盖已有的数据。理由如下:

1. 哈希表很小(相对于可能的局面数量),如果解决则很快表就被占满了。
2. 如果用动态申请,同上,则内存不够。
3. 开局时的哈希数据对中盘及残局几乎没有用,导致大量数据冗余,降低效率。
4. 依赖高散列度的哈希方法配合足够大小的哈希表,可以取得极低的冲突率。
5. 解决冲突带来的命中率上的提高微不足道。

(5) 历史启发搜索(History Heuristic)

博弈树中除了许多相同局面之外,还存在着很多相似局面。相似局面之间只有无关大局的一些棋子的落位不同。而好的走法一般是指能对局势产生深远影响的走法,因此有理由相信相似局面之间可能存在相同的“好的走法”。所谓好的走法是指,该走法引发了大量剪枝,或者是经搜索找到的最佳走法。所以,好的走法不一定是最佳走法,仍然要通过搜索才能得到最佳走法。

就像人们利用经验知识进行社会实践一样,好的走法也有很好的应用价值。在基于 alpha-beta 思想的搜索算法中,走法展开顺序对算法的效率有较大的影响。若走法展开顺序不好,假设能够产生剪枝的节点刚好被留到最后才展开,那么在这种情况下 alpha-beta 算法就和蛮力完全搜索一样是最差的。所以走法顺序是对搜索效率有着很大影响的因素。因为当我们搜索得到好的走法时,这很可能是某些比较相似的局面的好的走法,那么如果下次搜索到比较相似的局面时,为了可以产生更多的剪枝,我们就可以先考虑这些走法。

遇到好的走法时,都将其存在历史表中,同时赋予相应的值。进行搜索的时候,如果我们好几次都认为一个走法是好的,那么就给予该走法一个

比较高的历史值。当要处理一个新局面时,首先我们对该局面下的所有走法的历史值进行排序,接着依据排序的顺序对走法进行展开。通过实验,证明了历史表可以产生大量的剪枝,提高了效率。

历史启发搜索中有个特例其效果也不错,就是杀手启发。该算法把同一层节点中产生剪枝最多的节点成为杀手,当进行下一次搜索时,如果杀手的走法是合法的就优先搜索杀手节点。

5. 结束语

以上简明地阐述了机器博弈理论的原理,机器博弈系统的构成,对现存的二人、零和、完备信息搜索方法给出了比较全面的讲述。二人、零和、完备信息博弈的研究已经有半个多世纪的历史,其知识结构系统,层次清晰,已经取得了许多惊人的成果。本文重点地介绍并深入地讨论了机器博弈中的各种经典的搜索算法,对个别算法的性能和特点进行了比较深入的分析,为以后的应用提供了有意义的参考。

另一个方向是非完备信息游戏的博弈研究,本文没有涉及。这方面的研究也取得了一些成果,如拼字游戏 Scrabble,桥牌和扑克,都可以实现人机对战的过程,有一些程序还相当强大,但比之完备信息的博弈,它的研究还远远不够广泛。

参考文献:

- [1] 王小春. PC 游戏编程. 重庆大学出版社, 2002.
- [2] 王赠凯, 吕维先. 机器博弈搜索技术分析. 软件导刊, 2007, 2: 26-27.
- [3] 王镌. 博弈树搜索的算法改进. 福建电脑, 2004, 2: 26-27.
- [4] 危春波, 王海瑞, 文乔农. 博弈树搜索算法的分析与实现. 科技广场, 2007, 5: 15-17.
- [5] 黎德玲. 中国象棋的机器下棋. 北京大学应用数学系硕士学位论文, 2001.
- [6] 莫建文. 机器自学习博弈策略研究与实现. 广西: 广西师范大学硕士学位论文, 2002.
- [7] 危春波. 中国象棋博弈系统的研究与实现. 昆明: 昆明理工大学硕士学位论文, 2008.
- [8] 肖齐英, 王正志. 博弈树搜索与静态估值函数. 计算机应用研究, 1997, 4: 74-76.

《福建电脑》投稿信箱

《福建电脑》(国际标准刊号:ISSN 1673-2782;国内统一刊号 CN 35-1115/TP)是福建省科技厅主管,福建省计算机学会主办的计算机学科综合月刊。欢迎投稿。编辑部投稿邮箱:fjdn@vip.163.com

万方数据