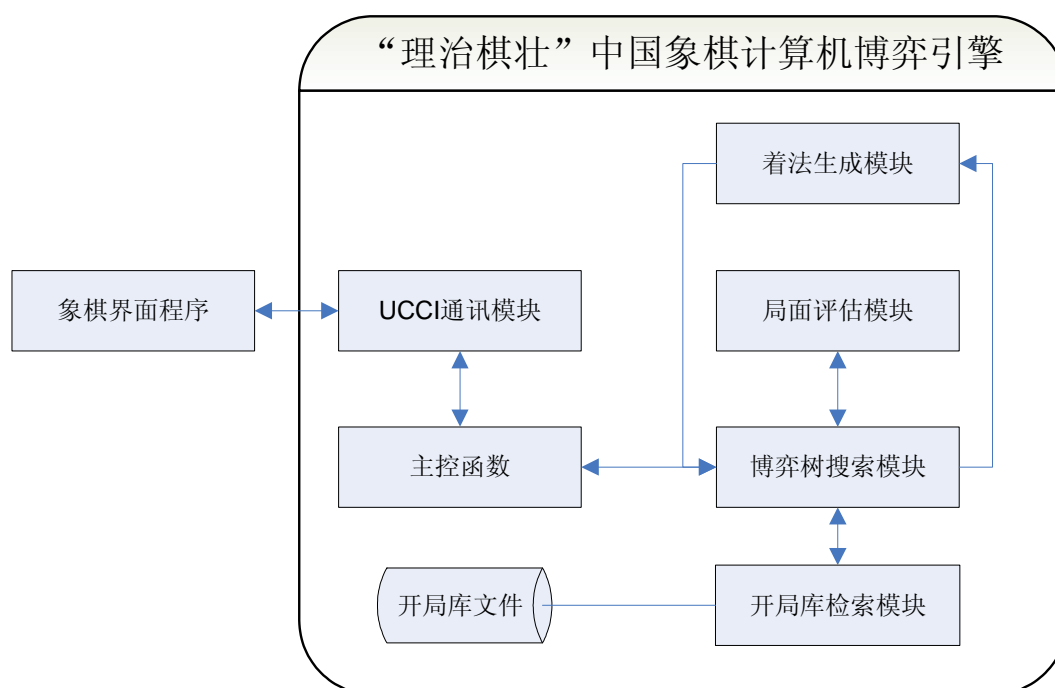


程序架构设计与主要算法概述

在各种棋牌博弈当中，中国象棋是一种完全知识博弈，即参与双方在任何时候都完全清楚每一个棋子是否存在，位于何处。中国象棋规则简单明确，成功与失败的判定标准简单，不包含任何机会或偶然性，问题的状态在数学意义上是有限的，因此从理论上讲易于用计算机人工智能方法实现。但象棋“有限”的数学状态事实上是一个天文数字，无法全部计算；象棋的经典开局与残局形式往往是人类经验的总结，无法用简单的算法泛泛表述，这又造成了计算机实现的不完备性和“教条”性。总的来说，象棋对弈程序是个系统工程，它是以下几个系统的有机结合：棋盘表示、着法生成、局面评估、博弈树搜索、其它（如开局库等）。

一、引擎架构设计



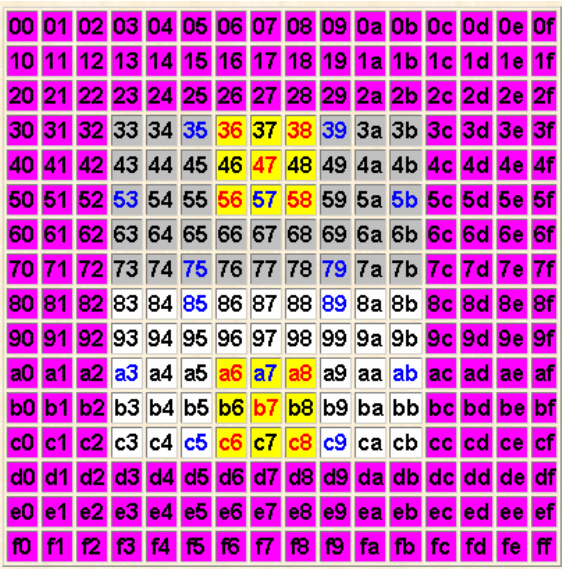
1

二、主要算法概述

1、棋盘表示

中国象棋的棋盘是一个 9×10 的矩阵，最容易转化为一个 9×10 的二维数组，用数组元素表示棋盘每一个交叉点有无棋子，是什么棋子。这种表示法清楚明了，相关算法与正常象棋思路一致，简单易写，方便在用户界面呈现，但算法实现效率比较低。为提高算法效率，一些领域专家提出了比特棋盘表示法，用 32 （棋子种类） $\times 90$ （棋盘交叉点数）个 64 位哈希数表示每个棋子在出现在每个交叉点上的情形，将所有棋子的哈希数求异或得到表示一个局面的哈希数。这种表示法算法充分利用位运算的高效性，有利于着法生成和局面评估，但实现的困难度较大，占用空间较大，同时不利于局面向用户界面呈现。

兼顾算法的效率与实现的方便，我们采用的是目前流行的另一种“ 16×16 棋盘表示法”（在实现时为运算方便，用长度为 256 的一维数组表示）。这种方法的存储原理与 9×10 的二维数组一致，但采用 16×16 矩阵的中间一块表示棋盘。这样做的好处，一方面行、列坐标增量分别为 $0x10$ 、 $0x01$ ，可以用各类位运算表示棋盘的变化，比在 9×10 的二维数组中做乘除 9 、 10 的运算快得多；另一方面周围一圈“哨兵”可以避免使用大量边界条件检测，自动防止棋子走出棋盘边界。



在这种棋盘数组中，用无符号单字节的整数元素表示有无棋子，是什么棋子。例如：

红帅	红仕	红相	红马	红车	红炮	红兵	棋盘中无子的位置
1	2	3	4	5	6	7	0
黑将	黑士	黑象	黑马	黑车	黑炮	黑卒	棋盘外的点
255	254	253	252	251	250	249	99

用 $255 \sim 249$ 表示黑方棋子，原因是这些数恰为 $-1 \sim -7$ 的 8 位二进制补码，与红方棋子一一对应，方便程序对棋子的判断。

2、着法生成

我们需要通过一个局面，得到指定一方所有可行的着法，以便分析评估其形成的局面，得到一个最佳着法。着法的生成过程是遍历整个棋盘，对所有棋子列出其符合棋规的着法（有必要排除“主动送将”和“不应将”等违规着法），记入一个着法表示类的数组。所谓着法表示类，记录了一种着法的起点、终点、吃了什么子、是否构成将军。

在寻找符合棋规的着法方面，我们充分利用 16×16 棋盘表示法在计算方面的优势，对不同的棋子，分别采用预置表、模板匹配、奇偶校验等多种位运算的方法，提高了算法效率。例如下图为马的匹配模板，我们用两个数组存储马的落子位置（“○”所示）和蹩马腿位置（“×”所示）相对于起始位置的偏移量，将其与起始位置相加，再与棋盘相应位置通过位运算进行匹配，即可找到其中的可行着法。

	○		○	
○		×		○
	×	马	×	
○		×		○
	○		○	

```
const POINT cnKnightMoveTab[8] =  
{0xdf, 0xe1, 0xee, 0xf2, 0x0e,  
0x12, 0x1f, 0x21};
```

```
const POINT cnHorseLegTab[8] =  
{0xf0, 0xf0, 0xff, 0x01, 0xff,  
0x01, 0x10, 0x10};
```

3、局面评估

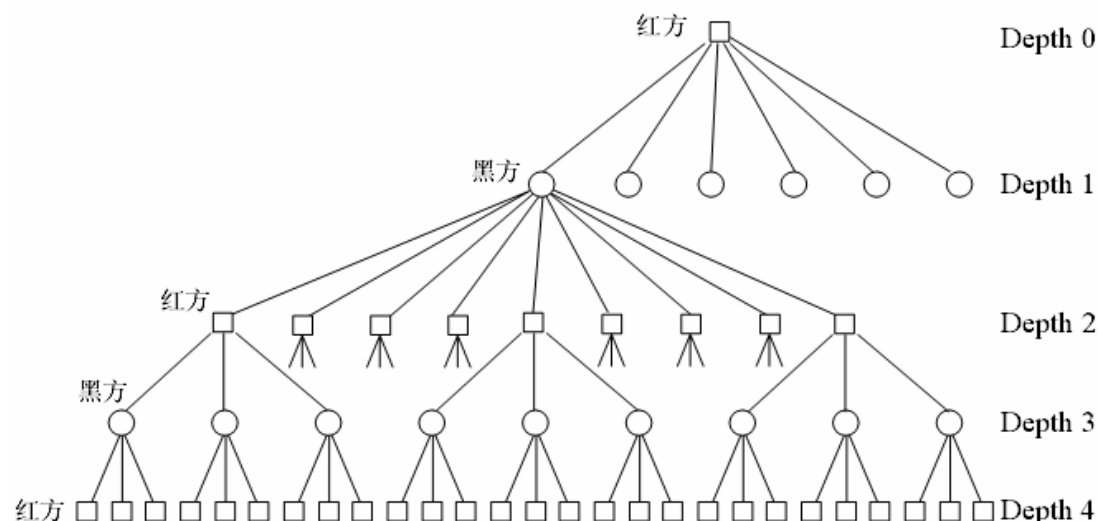
局面评估就是给棋局打分，识别棋局的好坏，以便在博弈树搜索时作为路径选择条件寻找最佳着法。可行的办法就是对局面进行量化，通过数值评判棋局的好坏。由于评估需要大量的象棋知识，仁者见仁，智者见智，评估函数的设计便成为计算机博弈中最为人性化的部分。一般评估函数要综合考虑以下几个方面：固定子力价值、子力位置价值、棋子灵活度、棋子威胁 / 保护、棋子配合作战、将帅安全。

固定子力价值、子力位置价值属于静态局面评估，我们将其放在同一个函数中完成，通过遍历棋盘，将每一棋子的评估值累加，一并计算我方与敌方的评估值，返回其差值。棋子灵活度、棋子威胁 / 保护属于未来局势发展评估；棋子配合作战、将帅安全属于知识型评估。它们也可通过静态局面评估获得，但这样一来计算量过大，因此我们将这一块放在着法生成模块动态计算，只计算走棋一方所走棋子的影响。由于局面变化的连续性，这种局部动态计算很大程度上可以反映某一着法对局面整体的影响。

4、博弈树搜索

博弈树上的每一个节点都代表一个局面，每一层的节点都是由其父节点的所有可行着法形成的。奇数层与偶数层节点分别表示双方的局面。引擎要在众多的叶子节点上挑选一个最

佳的局面，从而反推到当前的着法作为自己的选择。中国象棋博弈树的庞大是可以想象的。如果按照每一步平均有 45 种可行着法，每局棋平均走 90 步，那从开始局面展开到分出胜负，则要考虑的局面种数比地球上的原子数目还要多，无法完全计算，因此只能对有限层次展开分析。



对由对弈双方共同产生的博弈树搜索问题，香农(Claude Shannon)教授早在 1950 年提出了极大-极小算法 (Minimax Algorithm)。即将红方局面评估值计为正，黑方局面评估值取其负值，红黑双方在每一节点分别选择其子节点中评估值绝对值最大的一个。递归深度优先遍历有限层次的树，找到使起始节点评估值绝对值最大的一个局面，然后通知引擎走形成这一局面的着法。

极大-极小算法的改进形式有 Aspiration Search、Principal Variable Search、Tolerance Search 等，目前大多数国际象棋与中国象棋的算法核心青睐速度快而不会出现错误剪枝的 Principal Variable Search，我们的搜索模块也采用这一算法。同时加入迭代深化搜索功能，依局搜索用时的变化尽量深化博弈树搜索层数。

此外，对于博弈树中的重复节点，可以利用置换表和历史启发等技术避免重复计算，直接使用第一次的计算结果代入之后的重复节点，使用浅层次的评估结果为深层次待搜索节点排序，以节约运算时间。

5、其它（如开局库）

中国象棋开局的几个回合内，双方各自展开子力，占据棋盘有利位置。如果采用上述通用的博弈树搜索方法往往会因为局部很小的利益而忽视全局的发展，走出贻笑大方的开局。象棋有着经过千锤百炼的经典开局着法，如果将它们存储在数据文件中，在开局时用查询取代局面评估和博弈树搜索，就可以使我们快速出动子力，占据有利位置，同时防止被对手开局的“怪着”整昏。

我们通过编写小程序，将从网上收集到的专人整理好的经典开局棋谱转化格式作为我们的开局库。在博弈树搜索模块上挂接开局库检索模块，在开局的几个回合内首先将局面交由

开局库检索模块判断。如果是库中已存局面，则直接返回库中指定的着法，否则才调用博弈树搜索。

此外在残局阶段，人类的象棋知识和判断力往往强于计算机固化的搜索方式。我们的目标是优势求胜，下风谋和，均势求机。考虑到残局变化的复杂性，基于目前国内象棋残局库研究开发基本还是空白的实现，我们就不做残局库了，而是采用迭代深化的搜索方式加强对残局的分析能力。

参考文献：

- [1] 徐心和、王骄，中国象棋计算机博弈关键技术分析，小型微型计算机系统，Vol. 27, No. 6, 2006, pp961-969
- [2] 王小春，PC 游戏编程（人机博弈），重庆大学出版社，2002
- [3] 黄晨，解剖大象的眼睛——中国象棋程序设计探索，象棋百科全书网站 (<http://www.elephantbase.net>)，2005

北京理工大学“理治棋壮”中国象棋计算机博弈引擎开发小组版权所有（2006.7~2006.8）

项目主管：	林 健（北京理工大学计算机科学技术学院）
指导教师：	黄 鸿（北京理工大学信息科学技术学院自动控制系统工程研究所）
技术顾问：	赵陈翔（北京理工大学软件学院）
开发人员：	林 健（北京理工大学计算机科学技术学院）
	高 然（北京理工大学计算机科学技术学院）
	应张彬（北京理工大学软件学院）
	武 斌（北京理工大学软件学院）

联系方式：	lj@linjian.cn （林健）
	honghuang@bit.edu.cn （黄鸿）