

五子棋算法

摘要：人工智能是一门正在迅速发展新兴的综合性很强的边缘科学。博弈是人工智能的主要研究领域之一，他涉及人工智能中的推理技术、搜索方法和决策规划。本文将这些技术用于五子棋中。设计了一个智能五子棋系统，实现人和计算机两方进行博弈。

关键词：五子棋，人工智能，搜索

Gobang algorithm to research

CAI Jie Information and Computational Science, Grade 2003

Directed by WANG Jian (Associate Professor)

Abstract: Artificial intelligence is a newly-developed and highly comprehensive frontier science of rapid development. Gambling and chess is one of the major artificial intelligence research areas. It involves reasoning, decision—making and planning. These techniques are applied to the gobang. An intelligent gobang system is designed and realized in the game between human and computer.

Keywords: Gobang, Artificial intelligence, Search

1 绪论

在人类文明发展的初期，人们便开始进行棋类博弈的游戏了。在人工智能领域内，博弈是很重要的一个研究分支，很多实际问题可以在博弈的研究中得到解决，并且使计算机智能更加靠近人类智能。电脑博弈是人工智能研究的一个方向，到了近 50 年前，随着电子计算机的诞生，科学家们开始通过电脑模拟人的智能逐步向人类智能发起挑战，香农(1950)与图灵(1953)提出了对棋类博弈程序的描述，随着电脑硬件和软件的高速发展，从 1980 开始，电脑博弈便开始逐渐大规模地向人的智能发起了挑战，到了 1997 年，IBM 超

级电脑 Deeper Blue 击败了当时国际象棋世界冠军卡斯帕罗夫，成为了人工智能挑战人类智能发展的一个重要里程碑。

1.1 电脑五子棋简介

五子棋是起源于中国古代的传统黑白棋种之一。现代五子棋日文称之为“连珠”，英译为“Ren-ju”，英文称之为“Gobang”或“FIR”(Five in a Row 的缩写)，亦有“连五子”、“五子连”、“串珠”、“五目”、“五目碰”、“五格”等多种称谓。五子棋不仅能增强思维能力，提高智力，而且变化多端，非常富有趣味性和消遣性，因此为人民群众所喜闻乐见。本文在研究博弈机器人系统过程中，对五子棋博弈算法进行了一些有效的研究，设计和实现一个人机对弈的五子棋程序。五子棋属于黑白棋的一种，它的博弈树复杂度为 10^{70} ，状态空间复杂度为 10^{105} 。因此盘面预测的计算量是非常大的，比如对于五子棋的中盘走法中，如果要预测四步的局面数的话可以达到一百万。

1.2 设计思路总介绍

本文是对五子棋算法的设计原理和实现方法进行探讨和研究，主要包括数据结构、搜索算法和优劣评价函数组成，主要的特点包括快速的数据结构设计实现、以及高效率的搜索算法和尽可能的模拟人类的智能。

1.3 本文架构

第一章将会阐述计算机博弈发展以及对电脑五子棋的介绍、电脑五子棋的状态空间复杂度以及文章架构。

第二章首先描述了电脑象棋表示的一些基本数据结构。

第三章将重点讲述博弈树的搜索方法，包括博弈树构建、各算法介绍、以及五子棋规则的实现，最后对整体搜索框架进行总结。

最后，第四章是全文的总结及展望。

2 数据结构

2.1 棋局的表示

我们知道五子棋的走法中有优先和禁手，如连四肯定是没有三四优先，而如果是黑方出现三三（包括“四、三、三”）、四四（包括“四、四、三”），而黑方只能以四三取胜，如果黑方走出禁手则是输；五连与禁手同时形成，先五为胜，等等的规矩。但是电脑毕竟不是人类，可以类人但是却不可以自己思考，那么就需要给电脑一个它可以明白的评判标准。

首先介绍一个五子棋中的常用术语：

阳线：棋盘上可见的横纵直线。阴线：棋盘上无实线连接的隐形斜线。

活 3：由于一方走一着在无子交叉点上形成一个“活三”的局面，也就是在放一子就可行成 4 连了。

活 4：在棋盘某一条阳线或阴线上有同色 4 子不间隔地紧紧相连，且在此 4 子两端延长线上各有一个无子的交叉点与此 4 子紧密相连。

冲四：除“活四”外的，再下一着棋便可形成五连，并且存在五连的可能性的局面。

由于篇幅有限不能将所有的规则讲完，只是提出了对讲算法有用的几点加以叙述。

如何让电脑知道该落子在哪一点呢，在这方面。电脑要做得和人一样，判断棋盘上每一点的重要度。比如冲四比冲三强，冲三比造二强，遇到四三如果是对方的，堵死，如果是自己的，优先落子。遇到双三，如果是黑棋，黑方输，如果是白棋，优先等级仅次于四三。我们看到电脑在运算的时候，同种棋型在敌我双方的优先等级并不相同，由于禁手的缘故黑棋和白棋的处理也不相同。同样是四三，如果是玩家的，则电脑不会冲自己的，而是先堵玩家的。禁手的处理比较复杂，我们稍后讨论。

下面，我列出基本的棋型优先顺序：

造一个二<……<造四个二<冲三<……<冲两个二和一个三<冲双三<冲四<冲四三。

之所以这么设置是由于五子棋的下法所致，只要构成 5 颗一线就赢了，所以说一条线上构成的越多那么就有优势，当然就是棋数越多，分越多。

那么为了让电脑明确的知道是不是应该落子，就给它一个标准：分值。用程序语言表示：

```
enum Value {FAILED=-99999, SKIP=20, LENG=10, TWO =100, THREE =1000, IFOUR
=10000, FOUR =50000, SFOUR=70000, WIN=100000};
```

如果某一点导致失败，则分值为 FAILED，由于它足够小，所以无论任何棋型与它的组合都小于 0，SKIP 之所以给它 20 分，是为了避免电脑出现无棋可下的情况，LENG 是有可能发生长连的点，这有可能导致禁手，TWO、THREE、FOUR、WIN 观名知意，TFOUR 和 SFOUR 分别是弱四和强四。强四的点比普通的冲四重要的多，比如一个活四，意味着即将判出胜负。弱四是为了提供更大的灵活。也许有某种冲四并不一定比冲三重要，我在这里并没有使用，以后可以扩充。

3 涉及算法

3.1 五子棋搜索树种的极大极小搜索

在五子棋乃至所有博弈类游戏的研究中^[1-4]，最理想的方法是让电脑不论在对手走那种

棋路的时候都可以根据对手的走法来预测下一步，以至于使电脑立于不败之地，但是五子棋博弈树的复杂度为 10^{70} ，状态空间的复杂度为 10^{105} ，因此即使电脑的运算速度已经很快了，又有强有力的启发式搜索技术，但是要完成如此复杂的搜索也是不可取的。这种情况下搜索深度可根据实际情况进行调整，从局部搜索树中选取一步最好的走步，等对方走步后再寻找下一步好棋，通过结果来判断以后的走法。我们可以通过极大极小搜索方法来实现这种搜索策略。

为了使谈论更有条理性^[5]，将博弈的双方分别命名为 MAX 和 MIN。而搜索的任务是为 MAX 找最佳的移动。假设 MAX 先移动(此时暂时不考虑五子棋的禁手等规则)，然后 2 个博弈者轮流移动。因此，深度为偶数的节点，对应于 MAX 下一步移动的位置，称为 MAX 节点；深度为奇数的节点对应于 MIN 下一步移动的位置，称为 MIN 节点(博弈树的顶节点深度为 0)。k 层包括深度为 $2k$ 和 $2k+1$ 的节点。通常用层数表示博弈树的搜索深度。他可以表示出向前预测的 MAX 和 MIN 交替运动的回合数。对于复杂的博弈，博弈者必须认识到由于博弈树的复杂程度所以搜索到终点是不可能的(除了在博弈快结束时)。所以，常采用在有限范围搜索方法，这里使用启发式搜索。在启发式搜索的过程中，关键的一步是如何确定下一个要考察的节点，在确定节点时只要能充分利用与问题有关的信息，估计出节点的重要性，就能在搜索时选择重要性较高的节点，以利于博弈者以较快的速度求出最佳的棋步。

在这里我采用静态评估函数^[6-13]，用评估函数 $h(i)$ 衡量每一个叶节点位置的“值”。一个最佳首步可以由一个最小最大化过程产生。假设轮到 MAX 从搜索树的叶节点中选取，他肯定选择拥有最大值的节点。因此，MIN 叶节点的一个 MAX 节点双亲的倒推值就等于叶节点的静态评估值中的最大值。另一方面，MIN 从叶节点中选取时，必然选取最小的节点(即最负的值)。既然如此，MAX 叶节点的 MIN 双亲节点被分配一个倒推值，他等于叶节点静态评估值的最小值。在所有叶节点的父节点被赋予倒推值后，开始倒推另一层，假定 MAX 将选择有最大倒推值的 MIN 的后继节点，而 MIN 会选择有最小倒推值的 MAX 后继节点。继续逐层对节点评估，直到最后开始节点的后继者被赋予倒推值。MAX 将选择有最大倒推值的节点作为他的首步。整个过程的有效性基于这样的假设。用整个棋盘估值的函数 $h(n)$ 为静态估值函数。设想当前棋局 S 为轮到计算机方下棋(用方框表示)，任选一空点作为计算机方的下棋位置(可有若干种选择)，接着考虑在此情况下游戏者一方下棋的棋局(用圆圈表示)；从某一个圆圈棋局出发，任选一空点作为游戏者一方的落子处(又有若干种选择)。再次形成计算机方下棋的方框棋局；依此类推，这样可形成一棵以 S 为根结

点的博弈树，该树以圆圈棋局为第2层子结点，以方框棋局为第3层子结点等等。如果继续向前搜索，可形成多层子结点，现在以向前搜索3层子结点为例来说明极大极小值的过程。对第3层子结点的某一棋局n，求出其估计值 $h(n)$ ，假设有一博弈树已形成，如图1所示^[2]， $h(n)$ 的值由各结点旁的数值给出。

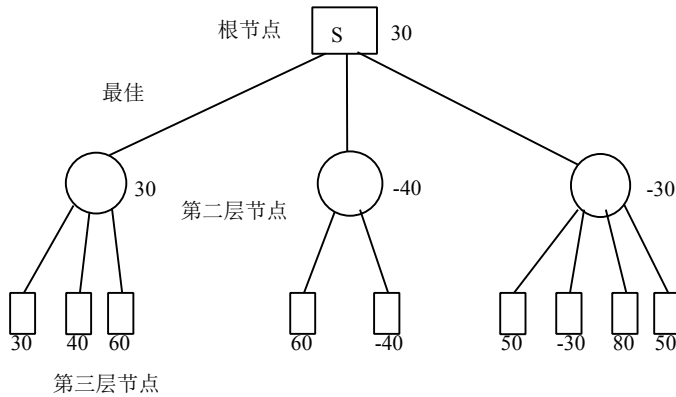


图1：博弈树

根据极小极大化分析法，先计算第3层子结点 $h(n)$ 值，然后第2层子结点的估计值取他的各后继子结点的极小值，根结点的估计值取他的各子结点的极大值。这个取得最大估计值的子结点即为从S出发的计算机方的最佳落子方案。棋盘上某一行、某一列或某一对角线为一路，这里使用的棋盘为19行19列，因此，行和列方向上共有 $19+19=38$ 路；从左下到右上方向的对角线有37路，同样，从左上到右下方向的对角线也有37路。但对于五子棋来说必须在一条直线上有连续五个棋子才能赢。因此，在对角线上就可以减少8路。所以，整个棋盘路的总数为： $38+(37-8)+(37-8)=96$ 路。

对某一棋局 $n^{[14-15]}$ ，第 i 路得分： $h(i)=h_c(i)-h_m(i)^{[1]}$ 。其中： $h_c(i)$ 为计算机方在第 i 路估计值得分， $h_m(i)$ 为玩家一方在第 i 路得分。对的得分规则作如下说明。规则中+代表一空点；O代表对方棋子；*代表有计算机方棋子；++表示连续2点为空点；+++*++表示连续3个空点接一个计算机方棋子，再接两空点。为了使电脑可以对棋局进行判断，那就需要告诉电脑什么时候的局势好过什么时候的局势，我们可以通过给不同局势时不同分值的办法来让电脑明白。我给出的局势评估如下：

+++*++: $h_c(i)=10$;

++**++: $h_c(i)=100$;

+****++: $h_c(i)=1000$;

O*****+: $h_c(i)=10000$;

+ * * * * +; $h_i(i)=50000$;

* * * * *的时候落子方都已经是胜利了： $h_i(i)=100000$ （之所以给分200是为了以后其他棋局的时候好判断）。之所以要给出如此大的分值差距，主要是考虑以后的全局判断的时候不会因为分数的累加使得电脑判断错误。而电脑在判断双方的分值的时候，就是在落子时对自己的落子点涉及分值和对方的分值进行对比来进行之后落子的判断。

3.2 α - β 剪枝

我们不用把每个节点都搜索一遍也可获得和极大极小搜索同样结果的走步，不搜索分支节点而舍弃该分支的过程叫剪枝。常用的剪枝方法是 α - β 剪枝算法。

在极大极小搜索的过程中，存在着一定程度的数据冗余。如下图 2 所示左半部的一棵极大极小树的片断，节点下面为该节点的值，设 A 为极大值节点，节点 B 的值为 18，节点 D 的值为 16，由此可以判断节点 C 的值将小于等于 16(取极小值)；而节点 A 的值为节点 $\text{Max}(B, C)$ ，是 18，也就是说不再需要估节点 C 的其他子节点如 E、F 的值就可以得出父节点 A 的值了。这样将节点 D 的后继兄弟节点减去称为 Alpha 剪枝(α 剪枝)。如图 2：

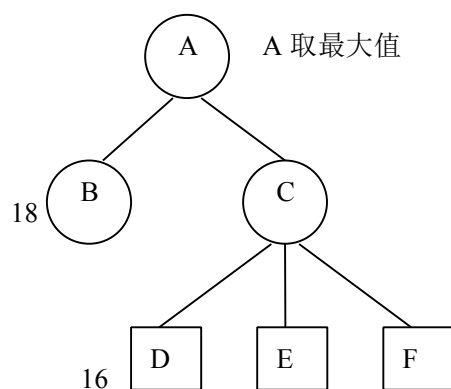


图 2: Alpha 剪枝(α 剪枝)

同样道理在图 3 右半部一棵极大极小树的片段中，设 A 为极小值节点，节点 B 的估值为 8，节点 D 的估值为 18，由此可以判断节点 C 的值将大于等于 18(取极大值)；而节点 A 的值为 $\text{Min}(B, c)$ ，是 8。也就是说不再需要求节点 C 的其他子节点如 E、F 值就可以得出父节点 A 的值了。这样将节点 D 的后继兄弟节点剪去称为 Beta 剪枝(β 剪枝)。如图 3：

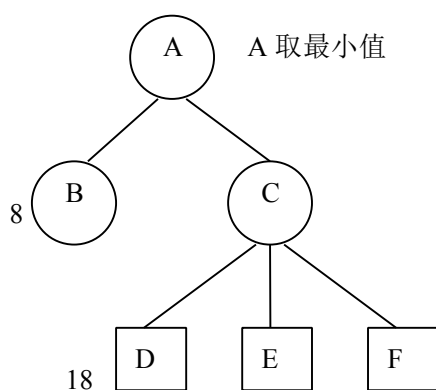


图 3: Beta 剪枝(β 剪枝)

将 Alpha 剪枝和 Beta 剪枝加入极大极小搜索，就得到 α - β 搜索算法。

经过了上面的介绍，我们再回到五子棋问题上来。

上边讲到了极大极小搜索，以及搜索时产生子结点的顺序。五子棋的静态估值函数有多种方法得到，具体如何实现除了要有一定的五子棋知识外，还要能运用这些知识正确地给出对一个五子棋当前局面的估值。可以进行全局搜索，对棋盘的每一个点判断棋形计算得分，最后综合得到全局得分，优先选择分值最高的走步。若对五子棋比较熟悉，可以在程序中规定几个下棋定式，根据定式判断当前局势得分。

根据极大极小搜索中给出的判断棋形所设的得分值，搜索整个棋盘得出共有多少个活一，死一，活二，……最后把各部分得分求和，得到当前局势的评价值。这个静态估值函数的好坏直接决定了程序智能的高低。可以在实践中不断调整得分值，使其智能更高。

而对于搜索时产生子结点的顺序，这关系到程序运行的快慢，倘若在 α - β 剪枝过程中，很早就得到一个较大的值，则在其后的搜索过程中，剪枝数目增加，程序效率也会提高。产生子结点的顺序，光靠说不行，还需要在实际中慢慢体会，得出一个较优方案。在这里我暂且假设它从(0, 0)点开始到(14, 14)为止产生结点，对于每一次预测走步，我们假设白棋从(0, 0)开始的第一个无子的点开始预测(假设电脑为白棋)，再轮到黑棋走，再轮到白棋走，一直向后预测 n 步后用静态估值函数对整个局面打分，再选择另一个位置，再为局面打分，重复此过程对深度为 n 的搜索树进行 α - β 剪枝搜索。

现在假设五子棋问题深度为 3 的搜索树给出 α - β 剪枝算法，用类 C 语言对其进行描述：

设电脑为 Max，人为 Min，初始时 α 为 $-\infty$ ， β 为 $+\infty$ ；函数 `Evalue()` 返回对当前局面的估值

```
int Max(point p,  $\alpha$ ,  $\beta$ )
```

```
{.....
```

```
在 p 处下白子；
```

```
if(已到达搜索的最后一层) return Evaluate();
```

```
for(int i=0; i< 15; i++ )
```

```
    for(int j=0; j< 15; j++ )
```

```
        {point pt=(i, j);
```

```
             $\alpha = \max(\alpha, \text{Min}(\text{pt}, \alpha, \beta))$ 
```

```
            if( $\alpha > \beta$ ) return  $\beta$ ;)
        }
```

```
    return  $\alpha$ ;)
}
```

MIN 结点的 β 剪枝函数:

```
int Min (point p,  $\alpha$ ,  $\beta$ )
```

```
{.....
```

在 p 处下黑子;

```
if(已到达搜索的最后一层) return Evaluate();
```

```
for(int i= 0; i< 15; i++ )
```

```
    for(int j=0; j< 15; j++ )
```

```
        {point pt=(i,j);
```

```
             $\beta = \min(\beta, \text{Max}(\text{pt}, \alpha, \beta))$ 
```

```
            if( $\beta \leq \alpha$ ) return  $\alpha$ ;)
        }
```

```
    return  $\beta$ ;)
}
```

如果 $\text{Max}(\text{point p}, \alpha, \beta)$ 函数, 则返回对 Max 结点最有利的走步的局势静态估值函数值。反之 $\text{Min}(\text{point p}, \alpha, \beta)$ 函数, 则返回对 Min 结点最有利的局势静态估值函数值。两种互为递归调用, 可以动态改变搜索深度。

3.3 优化估值函数

通过以上的研究可以看出, 在博弈的程序中电脑的行为都是以估值函数为依据的, 所以说估值函数在很大的程度上是决定了电脑的棋力高低。我们可以采用遗传算法来改进静态估值。遗传算法的估计值在很大的程度上也依赖于实施者的经验。但是可以利用一些高水平名局棋谱或同其他博弈程序对弈, 看使用某一组参数时取胜的机率有多大来进行较验。经过几次的试验一般就可以得到较好的静态估值。

传统的算法一般只能维护一组较好的参数。遗传算法同传统的算法相比可以同时维护几组较好的参数, 通过向其中添加一组新参数, 通常是将几组老参数中选出的值组合在一起做一点变化。然后同几组老的参数比较孰优孰劣, 将最差的一组从中去除。这里面较重

要的操作是交叉和变异操作。交叉通过随机交换父代个体的信息来继承已有参数中的优良内容；变异通过随机改变个体中的基因而增加种群的多样性，避免优化的因局部震荡而失败。

遗传算法的优点：

(1)由于搜索算法是从问题的解开始的，而不是一组参数。所以被局部震荡干扰导致求最优解失败的可能性大大减小。

(2)此算法能将搜索重点集中于性能高的部分，能较快地求出最佳的参数。

遗传算法的优化估值参数的过程：首先是①随机产生几组参数作为初始种群；接着②对种群的参数进行收敛判断，收敛则③输出优化结果并且评估过程结束，否则就④执行复制操作产生一组新参数；然后⑤取 0、1 之间的随机数，让随机数和交叉概率进行比较；若大于⑥则执行交叉操作改变新参数，若小于就不用执行这一步；⑦接着取 0、1 之间的随机数，随机数和变异概率进行比较，若⑧大于就执行变异操作改变新参数，小于就不执行这一步；接着是⑨把较差的一组参数从种群中去掉；接着跳回第二步判断是否已经收敛，如此循环就能将搜索重点集中于性能高的部分，能较快地求出最佳的参数。画图 2 表示如下：

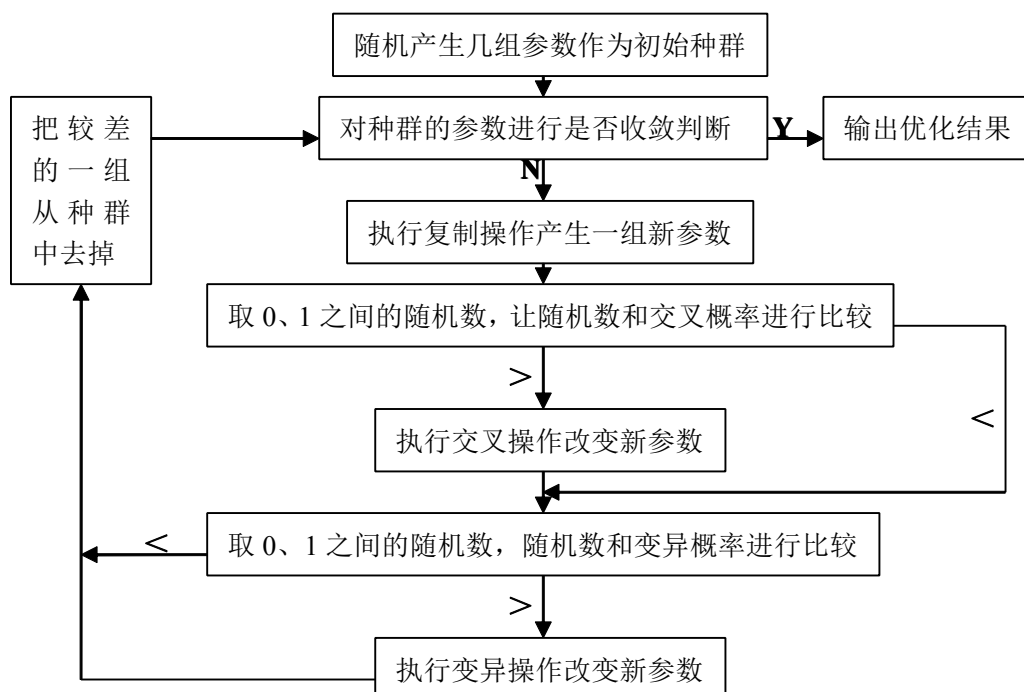


图 4：遗传算法评估过程图

3.4 禁手特征计算

五子棋中还有项规则就是是禁手的判断，在上边已经给出了 3 颗棋子的时候的分值为

1000。如果是双三则需要乘以 2，即 2000。电脑在分值表中将会判断双三点比一个三的点更为重要。这在电脑进行全局判断的时候也是正确的，可是如果电脑执黑子，这一点是不能落子的，否则将导致失败。但是电脑不能真的会自己去判断，只有认为的给他一个判断的标准，处理的方法是：在累加分值的时候增加一个判断语句，如果正好形成双三，那么这一点不加入分值表同时判断此点落棋将判为负，其他禁手按同样的方法处理。

4 结论及展望未来

4.1 总结

本文介绍了应用人工智能设计五子棋的总体方法。用极大极小值的过程以及启发式搜索的方法，采用静态估值函数对各节点的代价进行估计，应用遗传算法对估计的值进行了优化，克服了人为主观因素的片面性。对博弈的研究具有一定的借鉴意义。在实际五子棋系统的设计中。应用本方法只向前搜索了三步，就可以取得很好的效果。也可以增加搜索的深度来提高系统的判断能力，但是是以牺牲电脑的运算速度为前提的，毕竟加深搜索的话就需要更大的运算量。此外，也可以通过增加机器学习，比如电脑对棋局进行记忆，在对手落子之前对棋局和之前记忆的棋局进行比较，先判断出更优的走法，就可以进一步提高系统的智能。

4.2 未来展望

随着 Intel HP(超线程技术)的实现和将来多处理器 PC 机的普及，对于数据计算量大的人机对弈问题必然要求应用并行的思想去处理。超快搜索速度和必要的复盘“反思”必然带给下棋电脑更多智慧。

参考文献：

- [1] 蔡自兴. 人工智能及其应用[M]. 北京：清华大学出版社，1999
- [2] 阎平凡. 人工神经网络与模拟进化计算[M]. 北京：清华大学出版社，2002
- [3] 王小春. 游戏编程(人机博弈)[M]. 重庆：重庆大学出版社，2002
- [4] Nils J Nilsson, 郑扣根, 庄越挺译. Artificial Intelligence A New Synthesis[M]. 北京：机械工业出版社，2000
- [5] 陈尔绍. 传感器实用装置制作集锦[M]. 北京：人民邮电出版社，1999
- [6] 邓天炎, 李爱华, 尹正主编. 离散数学教程[M]. 徐州：中国矿业大学出版社，2002
- [7] 潘金贵, 顾铁成, 曾俭等编译. 现代计算机常用数据结构和算法[M]. 南京：南京大学出版社，1994

- [8] 王小平. 遗传算法—理论应用与软件实现[M]. 西安: 西安交通大学出版社, 1998
- [9] 王永庆. 人工智能原理与方法[M]. 西安: 西安交通大学出版社, 1998
- [10] 林尧瑞, 马少平. 人工智能导论[M]. 北京: 清华大学出版社, 1989
- [11] 田盛丰, 黄厚宽. 人工智能与知识工程[M]. 北京: 中国铁道出版社, 1999
- [12] 陆汝钤. 人工智能[M]. 北京: 科学出版社, 1995
- [13] 王鏊. 博弈树搜索的算法改进[J]. 福建电脑. 2004, (2)
- [14] 蒋加伏, 陈蔼样, 唐贤英. 基于知识推理的博弈树搜索算法[J]. 计算机工程与应用, 2004, (1)
- [15] Nilsson NJ. 人工智能[M]. 北京: 机械工业出版社, 2000

致谢

本论文是在汪建老师的悉心指导下完成的, 在论文写作过程中王老师深切关注, 在我题目选择上和文章写作上都得到了汪建老师的悉心指导和帮助, 在此对汪建老师表示衷心的感谢!