

斗地主AI算法实现 - typ0520的专栏 - 博客频道

分类:

算法 (1)



斗地主AI算法实现 一（拆牌）

by wojiushi3344

转载请说明出处

[源代码下载](#)

ps: 前面已经写了几篇地主游戏的基本算法实现，今天来讲讲单机地主中最重要的，也是开发中最难得AI算法实现。在此声明，本博文只适合像我一样的菜鸟阅读，高手们请别喷呀，欢迎你们留点宝贵的建议和方案，谢谢！如果你还对斗地主游戏的其他算法感兴趣欢迎你访问这个地址：[棋牌游戏开发专栏](#)

首先我们来看一下AI的定义：

“人工智能”（Artificial Intelligence）简称AI。它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。人工智能研究如何用计算机去模拟、延伸和扩展人的智能；如何把计算机用得更聪明；如何设计和建造具有高智能水平的计算机应用系统；如何设计和制造更聪明的计算机以及智能水平更高的智能计算机等 -----摘自百度百科

在这里我们主要的谈论如何电脑来模拟人的思维出牌，并且实现一种具体的算法。

名词解释：

出牌手数：在假设别人都不要并且满足出牌规则的情况下，最多几次能把所有的牌出完，这几次就叫做出牌手数，你可千万别理解成这个手术哟。

首先我们可以将地主AI算法分成2部分来讨论。

今天我们来讨论第一部分：

拆牌

拆牌是地主AI中比较重要的一部分，拆牌的好坏直接影响着地主AI算法的高效问题。

我们首先还是上一张图片，这样比较直观，有利于我们以后的讨论。



为每一种牌型定义权值的大小：

单张 1

对子 2

三带 3

连牌 4（每多一张牌权值+1）

连对 5（每多一对牌，权值+2）

飞机 6（每对以飞机，权值在基础上+3）

炸弹 7（包括对王在内）

看到这副牌，我们先按照人的正常思路来拆牌看是怎么样子的。

QQQ 222

45678910J

AA

小鬼 6, 4

我们现在就来讨论一种方案，看最终能达到这样的拆牌方案吗？

我们把拆牌的过程分几个步骤来讨论。

第一步：找牌

定义对应的牌型vector

```
vector<three> //3条
```

```
vector<lianzi> //连子
```

```
vector<duizi> //对子
```

```
vector<danzhang> //单张
```

```
vector<fly> //飞机
```

首先 找出一副牌中只能组成一种牌型的牌（3条，对子，单张为一种牌型。）意思就是有一张牌和剩余牌中的任何一张牌没有联系。当然是否和剩余的牌有联系是需要定义一个规则的：是否和剩余的牌能组成连子，连队。如果这2个条件都不满足那么我们称这张牌和剩余的牌没有任何联系。一般情况下通过这个步骤就能找出几张牌，这样能为以后的拆牌减轻任务量。

比如上面我们就能找出：小鬼，2，A，接下来找出牌值和他们一样的牌。即小鬼，222，AA，剩下的牌就是：QQQJ1098766544，可能有人会问为什么没有找出QQQ了，这是因为Q除了能组成3条，对子以外，还能和剩余的牌组合成连牌。这里找出来是对子最终的拆牌方案中就为对子，是连3条就是3条。3条>对>单张于是有了我们接下来的结论。

```
vector<three> 222
```

```
vector<duizi> AA
```

```
vector<danzhang> 小鬼
```

接下来我们就来拆分剩余的牌：QQQJ1098766544

找出牌中所有的炸弹

找出牌中所有的3个

找出牌中所有的对子

根据剩余的牌我们首先提出 QQQ 66 44

我们这里可以定义对应的vector来存入我们临时找出的牌，注意里定义的vector和上面的vector是不一样的。

```
vector<three> QQQ
```

```
vector<two>66, 44
```

第二步：计算每一种牌的手数和权值问题（拆牌中的难点）

分几种情况来讨论, 每次只找出剩余牌中的5张连牌，然后拿这5张连牌和剩余牌依次拼接看能否组合成更长的连牌

提出牌之后，剩余的全部牌全部看做为单牌，不管是炸弹，3条，还是对子。

1 在剩余牌中提出QQQ，J1098766544 首先找出5张单牌，从最小还是最大由你来决定。我们这里就以最小来讨论。

第一次：45678 剩余牌为 4 6 9 10 J

第二次：45678910J 46

最后拆出来的2组牌都没有联系，拆牌结束。

方案为：45678910J 4 6 QQQ

对应的权值和手数： $7+1+1+3=12$ 3（因为3个可以带一张或者是1对，当有对子和单张张的时候手数要减去1）

以下的几种方案也是同样的步骤，我就不写过程了，直接写最终的方案。

2在剩余牌中提出 QQQ 66

方案为：QQQ 66 78910j 5 44

对应的权值和手数： $3+2+4+1+2=12$ 4

3. 在剩余牌中提出 QQQ 44

方案为：QQQ 44 678910j 6

对应的权值和手数： $3+2+5+1=11$ 3

4 . 在剩余排中提出 66

方案为：66 78910jQ QQ 5 44

对应的权值和手数： $2+5+2+1+2=10$ 5

5 . 在剩余牌中提出 44

方案为： 44 5678910JQ 6 QQ

对应的权值和手数： $2+7+1+2=12$ 4

6 . 什么牌也不提出

45678910JQ 4 6 Q

对应的权值和手数： $9+1+1+1=12$ 4

第三步：选出最优的一组牌作为最后的拆牌方案

通过以上步骤我们确定出了几种出牌方案，我们最后找出出牌手数最小的方案。

如果手数相同的情况下，找权值最大的一组拆牌方案。

如果2者都相同，就随机选一种就是了。（这种情况很小很小）

通过比较我们确定了下面的这一种拆牌方案。

方案为：45678910J 4 6 QQQ

对应的权值和手数： $7+1+1+3=12$ 3（因为3个可以带一张或者是1对，当有对子和单张张的时候手数要减去1）

最后加入我们最先找出的牌，最终的拆牌方案为：

vector<three> 222 , QQQ

vector<duizi> AA

vector<danzhang> 小鬼 , 4, 6

vector<lianzi> 45678910J

呵呵，通过这样拆出来的牌是不是很符合正常人的思维呀。

我们拆这样一种拆牌方案出来是当电脑出牌的时候用。

当然我们还需要定义一个结构体来存入每张牌能组成的所有牌型。和每种牌型的最大值，这样供以后的接牌最准备。

顶

0