

文章编号: 2095-2163(2022)04-0151-08

中图分类号: TP3-05

文献标志码: A

## 二打一游戏残局模式下的对弈策略研究

郭荣城, 李淑琴, 龚元函, 黄韶华, 衡鑫

(北京信息科技大学 计算机学院, 北京 100101)

**摘要:** 机器博弈作为人工智能学科中的一个重要领域,是检测人工智能发展水平的一个重要平台。在机器博弈领域中,斗地主游戏具有典型的代表性。本文针对斗地主游戏中的残局策略进行研究,实现了基于 Alpha-Beta 剪枝算法的残局智能 AI。在微信平台小程序“欢乐斗地主”中进行了大量双人明牌斗地主残局测试,均成功获胜,实验表明本文所实现的斗地主残局策略是可行的。

**关键词:** 二打一游戏; 残局模式; Alpha-Beta 剪枝

### Research on the game strategy in the endgame mode of two-on-one game

GUO Rongcheng, LI Shuqin, GONG Yuanhan, HUANG Shaohua, HENG Xin

(Computer School, Beijing Information Science and Technology University, Beijing 100101, China)

**[Abstract]** As an important field in the subject of artificial intelligence, machine game is an important platform for testing the development level of artificial intelligence. In the field of machine games, the Doudizhu game has a typical representative. This paper studies the endgame strategy in Doudizhu game, and realizes the endgame intelligent AI based on the Alpha-Beta pruning algorithm. A large number of two-person Mingpai Fighting Doudizhu endgame tests are carried out in the WeChat platform small program "Happy Fighting Landlords", and are all succeeded. The experiment shows that the landlord endgame strategy implemented in this article is feasible.

**[Key words]** two-on-one game; endgame mode; Alpha-Beta pruning

## 0 引言

计算机博弈作为人工智能的重要发展方向之一,被称为是人工智能学科的“果蝇”<sup>[1]</sup>。二打一智力游戏(也称“斗地主”)是一种玩法简单、娱乐性强的扑克游戏。经典的玩法是扑克牌一共 54 张,斗地主智力游戏需由 3 个玩家进行,每人首先有 17 张手牌,底牌有 3 张。其中一名玩家根据手牌选择要底牌,该玩家成为地主,其余 2 名玩家则为农民。以某一玩家率先出尽手中牌来结束牌局判定胜负。

随着斗地主智力游戏作为应用软件在各类电子平台的日渐普及,游戏运营平台逐渐对游戏玩法进行拓展,除经典玩法外又陆续推出了多样化的各种玩法,如残局玩法、癞子玩法、换三张挑战赛等,这些玩法基于斗地主智力游戏的基础规则衍生出其他新的规则或针对传统牌局的局部进行独立,其中“残局玩法”(如图 1)便是其中的一个代表。



图 1 一残局模式开局

Fig. 1 One endgame start

“残局玩法”的规则为:牌局中有 2 个角色:一位地主和一位农民,双方分别有部分手牌(张数不固定),玩家将固定当地主首先出牌,农民方为运营平台的智能 AI,双方均为明牌,整个牌局至少存在一种地主胜利的解法,玩家需要尝试赢下牌局,出牌规则与经典玩法一致。

**基金项目:** 北京信息科技大学 2021 年大学生创新创业资助项目(5102110805)。

**作者简介:** 郭荣城(2001-),男,本科生,主要研究方向:网络工程;李淑琴(1963-),女,博士,教授,主要研究方向:人工智能、计算机博弈;龚元函(2000-),男,本科生,主要研究方向:计算机科学与技术;黄韶华(2000-),男,本科生,主要研究方向:计算机科学与技术;衡鑫(2001-),男,本科生,主要研究方向:计算机科学与技术。

**通讯作者:** 李淑琴 Email:lishuqin\_de@126.com

**收稿日期:** 2021-11-18

二打一智力游戏作为一种经典的博弈类游戏<sup>[2]</sup>,受到机器博弈领域<sup>[3-4]</sup>的广泛研究与关注,由于这是一种非完全信息类游戏,每个玩家看到的信息是不对称的,就使得玩家只能靠“猜”来做决策,增加了很多变数,并且二打一智力游戏中合作与竞争并存的设定,空间动作的巨大使得自身与其他机器博弈类游戏有很大差距,研究可知针对围棋的著名 AI 算法 AlphaZero<sup>[5]</sup>中的算法 MCTS 就无法直接应用在二打一中。所以二打一 AI 算法很多都采用了蒙特卡洛、深度学习、神经网络等,例如 2021 年快手推出的 DouZero<sup>[6]</sup>作为一匹黑马击败 344 个 AI 登顶二打一 AI 排行榜便是应用了蒙特卡洛、深度神经网络<sup>[7]</sup>、动作编码等技术。

而二打一残局模式相较于经典二打一模式,针对于少量手牌将局面做了很大程度简化,由三人非完全信息博弈转变为双人完全信息博弈,残局模式旨在锻炼玩家对拆牌压牌的理解,以达到在普通玩法中合理运用己方手牌达到“以弱胜强”的效果。本文拟采用基于 Alpha-Beta 剪枝<sup>[8]</sup>的方法,对二打一残局模式进行研究。

## 1 二打一游戏残局对弈策略设计与实现

本文二打一游戏残局对弈整体策略由手牌拆分、招法过滤和招法应对三个主要部分组成。

### 1.1 手牌拆分的设计与实现

本文首先将二打一手中手牌定义为火箭、炸弹、单牌、对子、三张、三带一、三带二、单顺、连对、飞机、三带一飞机、三带二飞机、四带二和四带两对,总共 14 种牌型。由二打一游戏对弈规则,火箭最大,可以打任意其他的牌。炸弹比火箭小,比其他牌大,都是炸弹时按牌的分值比大小。除火箭和炸弹外,其他牌必须要牌型相同且总张数相同才能比大小。对牌、三张牌都按分值比大小。单顺牌按最大的一张牌的分值来比大小。单牌按分值比大小,依次是:大王>小王>2>A>K>Q>J>10>9>8>7>6>5>4>3,不分花色。为此,本文根据对弈规则为每种牌型赋予了牌型内分值和牌型价。其中,牌型内分值指各牌型中的招法之间存在分值大小关系。牌型价值是指除炸弹与火箭外所有牌型价值为 0,炸弹价值为 1,火箭价值为 2。具体见表 1。

表 1 牌型定义及分值表

Tab. 1 Card type definition and score table

牌型名	牌型组合	牌型内分值
火箭	双王(小王大王)	由于一局游戏中只能存在一个火箭,不存在牌型内分值比较,即无牌型内分值
炸弹	4 张相同点数的牌(型如 AAAA)	牌型内分值随点数升序增加,3333 为最小分值 0,其余一次加 1
单牌	单张牌(型如 A)	3 为最小分值 0,其余按点数依次加 1
对子	2 张点数相同的牌(型如 AA)	33 为最小分值 0,其余按点数依次加 1
三张	3 张点数相同的牌(型如 AAA)	333 为最小分值 0,其余按点数依次加 1
三带一	3 张同点牌和一张牌(型如 AAAB)	333 * 为最小分值 0,其余按点数依次加 1
三带二	3 张同点牌和一对牌(型如 AAABB)	333 * * 为最小分值 0,其余按点数依次加 1
单顺	5 张或更多连续单牌(型如 ABCDE...)	牌型内分值按第一张牌的点数升序增加,3 为最小分值,其余按点数依次加 1
连对	3 个或更多的连续对子不包括 2 和王(型如 AABBC...)	牌型内分值按第一张牌的点数升序增加,3 为最小分值,其余按点数依次加 1
飞机	2 个或更多的连续三张(型如 AAABBB...)	牌型内分值按第一张牌的点数升序增加,3 为最小分值,其余按点数依次加 1
三带一飞机	2 个或更多的连续三带一(型如 AAABBB...)	牌型内分值按第一张牌的点数升序增加,3 为最小分值,其余按点数依次加 1
三带二飞机	2 个或更多的连续三带二(型如 AAABBB...)	牌型内分值按第一张牌的点数升序增加,3 为最小分值,其余按点数依次加 1
四带二	4 张同点牌和 2 张单牌(型如 AAAABC)	3333 * * 为最小分值 0,其余按点数依次加 1
四带两对	4 张同点牌和 2 个对子(型如 AAAABBC)	3333 * * * 为最小分值 0,其余按点数依次加 1

手牌拆分就是当轮到己方出牌时将手牌进行循环遍历, 将手牌依据表 1 中的牌型组合出所有对应的招法, 招法指根据 14 种牌型组出的具体手牌组合。然后将招法分别存入到对应的牌型 *List* 中, 并

根据牌型类型与点数大小赋予对应的分值。程序针对图 1 返回的己方所有可行招法的实际演示见图 2, 共有 11 种组合招法。

```

地主的牌: ['2', '4', '6', '6', '7', '7', '8', '8', 'K', 'Q']
农民的牌: ['2', '3', '3', '4', '4', '5', '5', 'K', 'Q']
当前出牌者: 地主
-----
11 Moves totally: [['6', '6', '7', '7', '8', '8'], ['6', '6',
['7', '7'], ['8', '8'], ['4'], ['6'], ['7'], ['8'], ['Q'], ['K'], ['2']]
  
```

图 2 针对图 1 的己方所有招法

Fig. 2 All the moves against Fig. 1 in self side

### 1.2 招法过滤设计与实现

拆分好手牌后进行招法过滤, 招法过滤分 2 种情况:

(1) 当牌局开始或对方 Pass 轮到己方出牌时, 遍历存储己方招法的 *List*, 按表 1 中牌型顺序返回当前手牌可组成的所有招法。

(2) 当对方出牌后, 将对方的招法与 14 种牌型做对比, 判断对手招法是 14 种牌型类型的哪一种, 依据前文中提到的招法价值大小关系将对方招法与己方所有招法进行牌型内分值与牌型价值比较, 返回所有牌型内分值或牌型价值大于对方招法的己方招法。

### 1.3 招法应对设计与实现

招法应对对应招法过滤也分 2 种情况:

(1) 当牌局开始或对方 Pass 轮到己方出牌时, 招法选择分值为 100 的招式作为出牌策略。

(2) 当对方出牌后、己方跟牌时, 从招法过滤获取到己方所有可组成的招法中, 通过 Alpha-Beta 剪枝算法, 选择出一种必胜的出牌策略。Alpha-Beta 剪枝搜索大体可以分为 3 个步骤: 局面标定、传递倒推、剪枝。3 个步骤依次进行, 最终获取到一条最优路径, 即选出最佳出牌策略。

#### 1.3.1 深度搜索局面标定

Alpha-Beta 剪枝搜索经常用于计算机零和博弈游戏中。在建立博弈树的过程中, 地主状态层, 农民状态层交替建立, 两者博弈树关系如图 3 所示。图 3 中, 方形代表地主, 圆形代表农民。由于斗地主残局模式属于双人完备信息游戏, 所以获得胜利的决策近乎唯一。在这种情况下进行 Alpha-Beta 剪枝搜索, 可以找出最佳出牌路径, 通过搜索到底得到精确的输赢反馈值, 从而对局面进行标定。

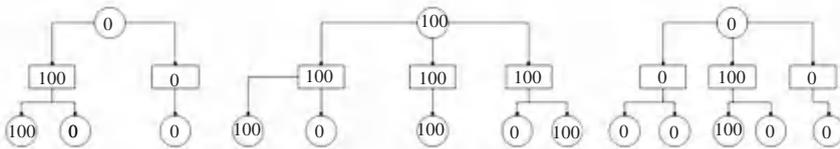


图 3 完全搜索倒推的预计结果

Fig. 3 Expected results of full search backwards

深度搜索局面标定法步骤如下:

- (1) 将双方剩余手牌作为当前局面。
- (2) 对当前局面执行 Alpha-Beta 深度搜索, 每执行一步为一节点。
- (3) 每走一步便进行手牌拆分, 并将新拆分出的招法作为下一步的分支。
- (4) 将一条分支延伸到一方手牌出尽, 若地主胜利则将当前局面 *value* 值标定为 100, 若农民胜利则将当前局面 *value* 值标定为 0。

#### 1.3.2 向上传递

(1) 在偶数层节点的招法、即轮到农民出牌的节点时, 农民应考虑最好的情况, 选择其全部子节点中评估值最大的一个, 即:

$$F(v) = \max \{ F(v_1), F(v_2), \dots, F(v_n) \} \quad (1)$$

其中,  $F(\cdot)$  为节点估值,  $v_1, v_2, \dots, v_n$  为节点  $v$  的子节点。

(2) 在奇数层节点的招法、即轮到地主出牌的节点时, 地主应考虑最坏的情况, 选择其全部子节点

中评估值最小的一个,即:

$$F(v) = \min \{ F(v_1), F(v_2), \dots, F(v_n) \} \quad (2)$$

其中,  $F(\cdot)$  为节点估值,  $v_1, v_2, \dots, v_n$  为节点  $v$  的子节点。

(3)评价往回倒推时,相当于2个局中人的对抗策略,交替使用(1)、(2)两种方法传递倒推值。具体实例参见图3。

### 1.3.3 Alpha-Beta 剪枝

本程序中 Alpha-Beta 剪枝与普通的 Alpha-Beta 剪枝算法稍有不同,由于局面标定中只会出现0和100两个  $value$  值,分别代表农民胜利和地主胜利,所以当 min 层中出现0,便意味着此条分支的父节点所选择的招法已无法获得地主胜利的情况,所

以当前父节点还未搜索的分支已无搜索的必要,即减去后续未搜索的分支。max 层中出现100时与之同理。

剪枝实例如图4所示。图4中,①处分支为 min 层,由于遍历搜索已经搜索到了  $value$  值为100的分支,所以①分支便停止搜索即被剪枝,②处同理,③、④分支处于 max 层,由于已搜索到了  $value$  值为0的分支,所以③、④分支被剪枝,但由于已搜索到一条分值为100的树,程序已经停止搜索,故实际上③、④所处子树并未被搜索。

上述算法的实现流程如图5所示。图5中,  $\alpha$  为 min 层  $value$  值,  $\beta$  为 max 层  $value$  值。

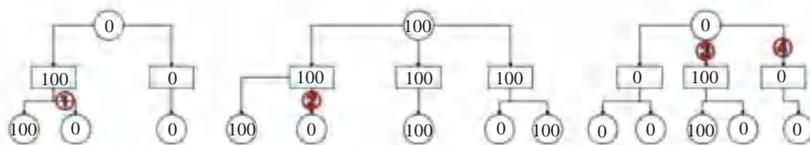


图4 剪枝实例

Fig. 4 Pruning example

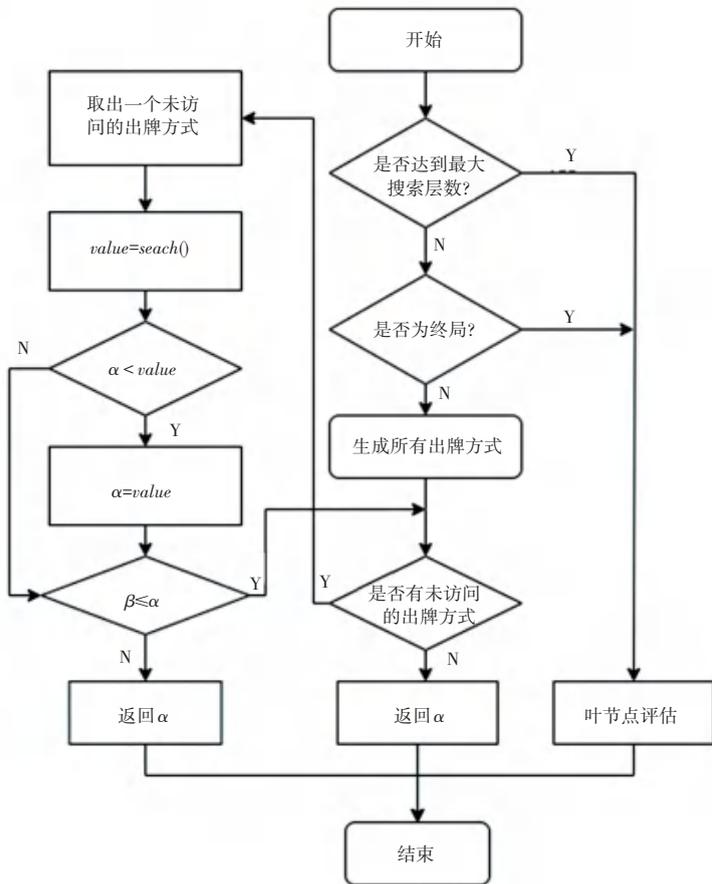


图5 算法流程图

Fig. 5 Algorithm flowchart

## 2 实验结果与分析

本文通过微信平台小程序“欢乐斗地主”中的残局闯关模式对算法程序进行测试,以下为其中一残局的完整对弈局,并将己方出牌执行情况附在图的右侧。

己方出牌为[‘4’]时的牌局如图6所示,将残局双方手牌导入进程序,第一步为己方出牌,根据双方手牌运行 Alpha-Beta 剪枝程序,通过多线程搜索分别基于不同出牌方式构建多个博弈树,由图6可见,当前手牌程序返回了11种出牌方式,即基于这11种出牌方式构建11棵博弈树,由图6可见程序依次返回基于出牌方式[‘k’,’k’,’k’]、[‘8’]、[‘6’]、[‘7’]、[‘K’]所构建的博弈树的返回值为0,即这些出牌方式己方无法胜利,继续返回基于出牌方式[‘4’]所构建的博弈树,返回值为100,即此种出牌方式己方必胜,搜索到必胜的出牌方法后停止对其余出牌方式的博弈树的构建,但考虑到搜索为多线程进行,在发出停止命令之前其他线程又返回

回了基于出牌方式[‘Q’]、[‘6’,’6’]、[‘7’,’7’]、[‘8’,’8’]、[‘2’]所构建的博弈树,由于已经搜索到必胜的出牌方式[‘4’],所以下一步的手牌出法为[‘4’]。

对方出 Q,如图7所示。

当前本程序返回了3种出牌方式,即基于这3种出牌方式构建3棵博弈树,由图6、图7可见程序首先返回基于出牌方式[‘K’]所构建的博弈树,返回值为100,即此种出牌方式己方必胜,搜索到必胜的出牌方法后停止对其余出牌方式的博弈树的构建,但也考虑到搜索为多线程进行,在发出停止命令之前其他线程又返回了基于出牌方式[‘2’]所构建的博弈树,由于已经搜索到必胜的出牌方式[‘K’],所以下一步的手牌出法为[‘K’]。己方出牌为[‘K’]时的牌局如图8所示。

对方出[‘2’],如图9所示。

对方出牌为[‘2’],己方无可出牌型,即只有一种手牌出法:“pass”。如图10所示。



图6 己方出牌为[‘4’]

Fig. 6 The card of the self side is [‘4’]



图7 对方出牌为[‘Q’]

Fig. 7 The card of the opponent side is [‘Q’]

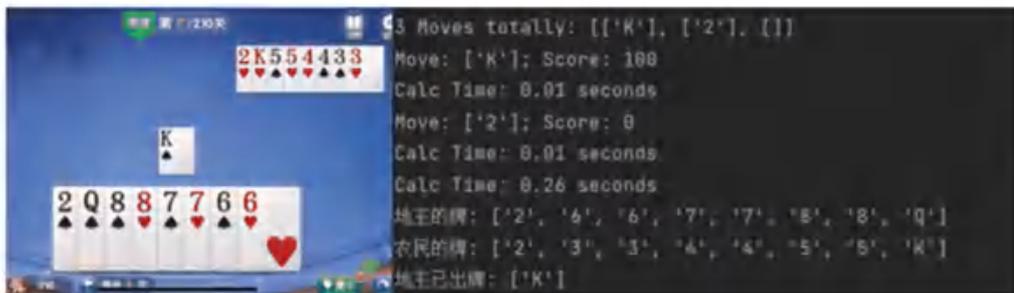


图 8 己方出牌为[ 'K' ]

Fig. 8 The card of the self side is [ 'K' ]



图 9 对方出牌为[ '2' ]

Fig. 9 The card of the opponent side is [ '2' ]

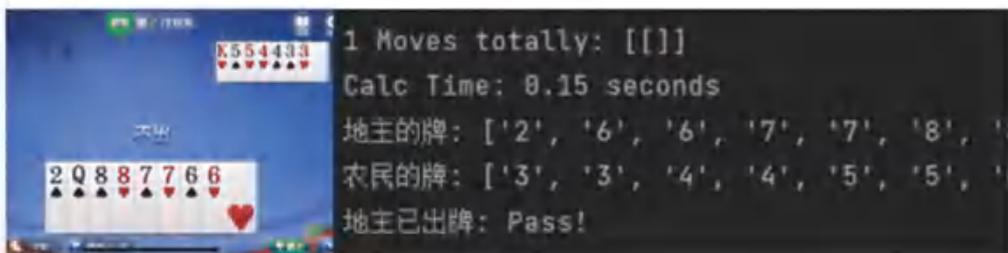


图 10 己方 pass

Fig. 10 The self is pass

对方出“334455”,如图 11 所示。

己方当前手牌程序返回了 2 种出牌方式,即基于这 2 种出牌方式构建 2 棵博弈树,由图 11 可见程序首先返回基于出牌方式[ '6' '6' '7' '7' '8' '8' ]所构建的博弈树,返回值为 100,即此种出牌方式己方必胜,搜索到必胜的出牌方法后停止对其余出牌方式的博弈树的构建,但由于搜索为多线程进行看,在发出停止命令之前其他线程又返回了基于出牌方式 pass 所构建的博弈树,由于已经搜索到必胜的

出牌方式[ '6' '6' '7' '7' '8' '8' ],所以下一步的手牌出法为[ '6' '6' '7' '7' '8' '8' ]。如图 12 所示。

对方出“pass”,如图 13 所示。

己方当前手牌程序返回了 2 种出牌方式,即基于这 2 种出牌方式构建 2 棵博弈树,由图可见程序首先返回基于出牌方式[ '2' ]所构建的博弈树,返回值为 100,即此种出牌方式己方必胜,搜索到必胜的出牌方法后停止对其余出牌方式的博弈树的构建,下一步的手牌出法为[ '2' ]。如图 14 所示。



图 11 对方出牌为 [ '3' '3' '4' '4' '5' '5' ]

Fig. 11 The opponent's card is [ '3' '3' '4' '4' '5' '5' ]

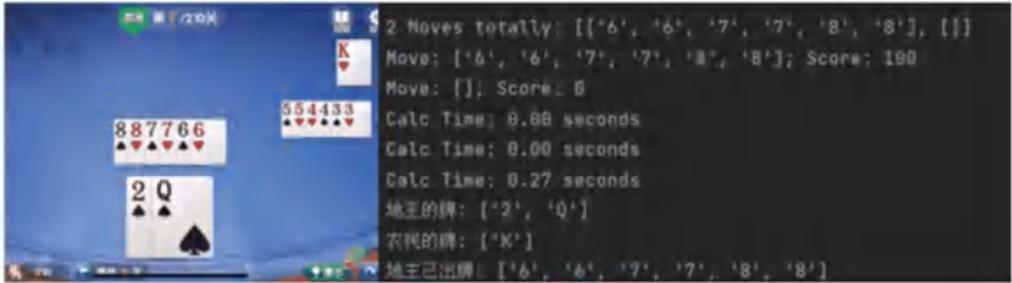


图 12 己方出牌为 [ '6' '6' '7' '7' '8' '8' ]

Fig. 12 The card of the self side is [ '6' '6' '7' '7' '8' '8' ]

对方继续“pass”,如图 15 所示。



图 13 对方 pass

Fig. 13 The opponent is pass



图 15 对方 pass

Fig. 15 The opponent is pass

己方只剩一张牌 [ 'Q' ], 下一步的手牌出法为 [ 'Q' ], 手牌出完, 游戏胜利。如图 16 和图 17 所示。

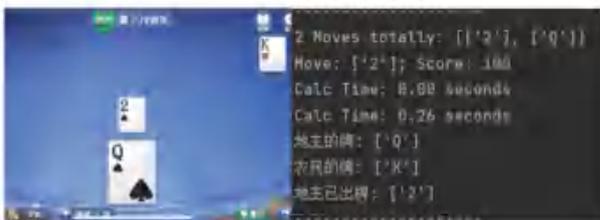


图 14 己方出牌为 [ '2' ]

Fig. 14 The card of the self side is [ '2' ]



图 16 己方出牌为 [ 'Q' ]

Fig. 16 The card of the self side is [ 'Q' ]



图 17 游戏胜利

Fig. 17 Game victory

本文通过微信端欢乐斗地主小程序中的餐具闯关模式对程序进行实际测试,共进行了 140 关残局闯关测试,均成功取得胜利,实验结果如图 18 所示。



图 18 残局闯关通过关数截图

Fig. 18 Screenshot of the number of passes through the endgame

### 3 结束语

在斗地主游戏中,残局策略非常关键,对博弈胜负有着至关重要的作用。本文基于 Alpha-Beta 剪枝算法对斗地主残局模式出牌策略展开了研究分析,提出了一种新的残局策略,并加以实现,在与双人明牌斗地主残局中有着较好的表现。在接下来的

工作中还需进一步提炼判定手牌优劣的因子加入完善算法,有利于后续研究的深入开展。

### 参考文献

- [1] WANG Yajie, QIU Hongkun. Computer game competition and reform of innovative talent training mode [J]. *Experimental Technology and Management*, 2016, 33 (10): 10-14.
- [2] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of Go with deep neural networks and tree search [J]. *Nature*, 2016, 529(7587): 484-489.
- [3] SILVER D, SCHRITTWIESER J, SIMONYAN K, et al. Mastering the game of Go without human knowledge. [J]. *Nature*, 2017, 550(7676): 354.
- [4] SILVER D, HUBERT T, SCHRITTWIESER J, et al. Mastering chess and Shogi by self-play with a general reinforcement learning algorithm [J]. *arXiv preprint arXiv:1712.01815*, 2017.
- [5] WHEELER T. AlphaGo zero-how and why it works [EB/OL]. [2017-11-02]. <http://tim.hibal.org/blog/alpha-zero-how-and-why-it-works/>.
- [6] ZHA Daochen, XIE Jingru, MA Wenye, et al. DouZero: Mastering DouDizhu with self-play deep reinforcement learning [C]//*Proceedings of the 38<sup>th</sup> International Conference on Machine Learning*. PMLR, 2021:12333-12344.
- [7] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning [J]. *Nature*, 2015, 518(7540): 529.
- [8] PEARL J. The solution for the branching factor of the alpha-beta pruning algorithm and its optimality [J]. *Communications of the ACM*, 1982, 25(8): 559-564.
- [9] 备协会 (China Association of Medical Equipment), 《中国医学装备》杂志社, 2021: 137-144.
- [10] LIU S, AN Z, WANG N, et al. Research on elevator passenger fall detection based on machine vision [J]. *IOP Conference Series: Earth and Environmental Science*, 2021, 791(1): 012108.
- [11] Cao Xianghong, Zhang Hua. Falling detection research based on elderly behavior infrared video image contours ellipse fitting [J]. *International Journal of Pattern Recognition and Artificial Intelligence*, 2021, 35(4): 2154004.
- [12] 张璐, 方春, 祝铭. 基于 Res2Net-YOLACT 和融合特征的室内跌倒检测算法 [J/OL]. *计算机应用*; 1-7 [2021-07-21]. <http://kns.cnki.net/kcms/detail/51.1307.TP.20210720.1454.006.html>.
- [13] 陶帅, 常蒙月, 汪祖民, 等. 基于可穿戴设备的跌倒检测方法研究进展 [J]. *中国老年保健医学*, 2021, 19(04): 119-125.
- [14] 徐攀峰, 卢媛, 王丽敏. 基于足底压力的跌倒检测定位系统设计 [J]. *辽宁大学学报(自然科学版)*, 2020, 47(04): 296-300.
- [15] GONZÁLEZCAÑETE F J, CASILARI E. A feasibility study of the use of smartwatches in wearable fall detection systems [J]. *Sensors (Basel, Switzerland)*, 2021, 21(6): 2254.
- [16] ZHANG Jing, LI Jia, WANG Weibing. A class-imbalanced deep learning fall detection algorithm using wearable sensors [J]. *Sensors*, 2021, 21(19): 6511.
- [17] 周旭虎, 张华伟. 基于 STM32 芯片与 MPU6050 运动传感器对人体姿态检测与跌倒判定的研究与实现. 中国医学装备协会 (China Association of Medical Equipment) [C]//中国医学装备大会暨 2021 医学装备展览会论文集汇编. 江苏苏州: 中国医学装备协会 (China Association of Medical Equipment), 《中国医学装备》杂志社, 2021: 137-144.
- [18] RAHME M, FOLKEARD P, SCOLLIE S. Evaluating the accuracy of step tracking and fall detection in the starkey livio artificial intelligence hearing aids: A pilot study [J]. *American Journal of Audiology*, 2021, 30(1): 1-8.
- [19] 中华人民共和国卫生部. 老年人跌倒干预技术指南 [Z]. 北京: 中华人民共和国卫生部, 2011.
- [20] 项云, 王晶晶. 社区老年人跌倒风险评估工具的研究及应用 [J]. *体育科研*, 2021, 42(05): 39-48.
- [21] 杨丽, 吴雨茜, 王俊丽, 等. 循环神经网络研究综述 [J]. *计算机应用*, 2018, 38(S2): 1-6, 26.
- [22] ZHANG Jing, WANG Yongfeng, MA Yingnan, et al. Fall detection with the optimal feature vectors based on support vector machine [M] // ZHANG Y, YAO G, HE J, et al. *Health Information Science. HIS 2014. Lecture Notes in Computer Science*. Cham: Springer, 2014, 8423: 160-166.
- [23] SHI Guangyi, ZHANG Jiye, DONG Chao, et al. Fall detection system based on inertial mems sensors: Analysis design and realization [C] // *Proceedings of International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*. Shenyang, China: IEEE, 2015: 1834-1839.
- [24] NOURY N, FLEURY A, RUMEAU P, et al. Fall detection - principles and methods [C] // 2007 29<sup>th</sup> Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Lyon, France: IEEE, 2007: 1663-1666.

(上接第 150 页)