Determinization and Information Set Monte Carlo Tree Search for the Card Game Dou Di Zhu

纸牌游戏《斗朱迪》的确定性和信息集蒙特卡罗树搜索

Daniel Whitehouse, Edward J. Powley, *Member, IEEE*, and Peter I. Cowling, *Member, IEEE*

Daniel Whitehouse、IEEE 成员 Edward J. Powley 和 IEEE 成员 Peter I。Cowling

Abstract—Determinization **is a technique for making decisions in games with stochasticity and/or imperfect information by sampling instances of the equivalent deterministic game of perfect information.** ***Monte-Carlo Tree Search (MCTS)*** **is an AI technique that has recently proved successful in the domain of deterministic games of perfect information. This paper studies the strengths and weaknesses of determinization coupled with MCTS on a game of imperfect information, the popular Chinese card game Dou Di Zhu. We compare a "cheating" agent (with access to hidden information) to an agent using determinization with random deals. We investigate the fraction of knowledge that a non-cheating agent could possibly infer about opponents' hidden cards. Furthermore, we show that an important source of error in determinization arises since this approach searches a tree that does not truly resemble the game tree for a game with stochasticity and imperfect information. Hence we introduce a novel variant of MCTS that operates directly on trees of information sets and show that our algorithm performs well in precisely those situations where determinization using random deals performs poorly.**

摘要——确定性是一种通过对完全信息的等价确定性博弈的实例进行抽样，在具有随机性和/或不完全信息的博弈中做出决策的技术。蒙特卡罗树搜索（MCTS）是一种人工智能技术，最近在完美信息的确定性游戏领域被证明是成功的。本文结合 MCTS 对一个不完全信息游戏——中国流行的纸牌游戏斗进行了研究。我们将"作弊"代理（可以访问隐藏信息）与使用确定性随机交易的代理进行比较。我们调查了一个非作弊特工可能推断出的对手隐藏卡的知识比例。此外，我们表明，确定性中的一个重要错误来源出现了，因为这种方法搜索的树并不真正类似于具有随机性和不完善信息的博弈树。因此，我们引入了一种新的 MCTS 变体，它直接在信息集树上操作，并表明我们的算法在那些使用随机交易的确定性表现不佳的情况下表现良好。

I. INTRODUCTION

II. 介绍

Historically, the vast majority of research in game AI has focussed on games of *perfect information* (such as chess and checkers) in which the exact state of the game is visible to all players at all times. Work on AI for games of *imperfect information*, where the state of the game is only partially observable, is comparatively recent. One popular approach to such games is *determinization*. The game is reduced to several instances of a deterministic game of perfect information (called *determinizations*) compatible with the agent's observations of the game of imperfect information. Each of these determinizations is analysed by standard AI techniques (e.g. game tree search) and the results are combined to yield a decision for the original game. Determinization has proven successful in the domains of Bridge [1], Klondike solitaire [2] and probabilistic planning [3], among others, although there are situations where it performs poorly [4,5].

历史上，游戏人工智能的绝大多数研究都集中在完全信息游戏（如国际象棋和跳棋）上，在这种游戏中，所有玩家在任何时候都可以看到游戏的确切状态。针对不完全信息游戏（游戏状态只能部分观察到）的人工智能研究相对较新。这类游戏的一种流行方法是确定性。这个博弈被简化为与代理人对不完美信息博弈的观察相一致的完美信息的确定性博弈（称为确定性）的几个实例。通过标准的人工智能技术（例如游戏树搜索）来分析这些决定中的每一个，并将结果组合起来以产生原始游戏的决定。确定性在桥牌【1】、克朗代克纸牌游戏【2】和概率规划【3】等领域被证明是成功的，尽管也有表现不佳的情况【4，5】。

One recent innovation in AI for deterministic games of perfect information is *Monte-Carlo Tree Search (MCTS)* [6-

最近人工智能在完美信息的确定性博弈方面的一项创新是蒙特卡罗树搜索（MCTS）【6-

D. Whitehouse, E. J. Powley and P. I. Cowling are with the Artificial Intelligence Research Centre, School of Computing, Informatics and Media, University of Bradford, UK (e-mail: {d.whitehouse1, e.powley, p.i.cowling}@bradford.ac.uk).

D.Whitehouse、E. J. Powley 和 p . I。Cowling 就职于英国布拉德福德大学计算、信息学和媒体学院人工智能研究中心（电子邮件：{ d.whitehouse1，e.powley，p.i.cowling } @ bradford.ac.uk）。

This work is funded by grant EP/H049061/1 of the UK Engineering and
这项工作得到了英国工程和
Physical Sciences Research Council (EPSRC).
物理科学研究委员会。

978-1-4577-0011-8/11/$26.00 ©2011 IEEE

8]. The best known example of an MCTS algorithm is *upper confidence for trees (UCT)* [6]. MCTS has proven particularly successful in games where depth-limited minimax search performs poorly, with Go [9] as a prime example.

8].MCTS 算法最著名的例子是树的上置信度（UCT）【6】。事实证明，MCTS 在深度受限的极大极小搜索表现不佳的游戏中尤其成功，围棋【9】就是一个最好的例子。

Board and card games provide an interesting intermediate step in demonstrating the potential of MCTS for video games and real-time planning problems. Through studying games such as Dou Di Zhu, with hidden information and randomness as well as deep strategy for multiple cooperating and competing players, we aim to gain insights that may be used on still more complex domains.

棋盘和纸牌游戏提供了一个有趣的中间步骤，展示了 MCTS 在视频游戏和实时规划问题方面的潜力。通过研究像《斗》这样的游戏，我们旨在获得可能用于更复杂领域的见解。这些游戏包含隐藏信息和随机性，以及多个合作和竞争玩家的深层策略。

In this paper we compare (for the popular Chinese card game Dou Di Zhu) the playing strength of an agent using a determinized version of UCT with that of a UCT agent that is able to “cheat” and observe the actual state of the game. It is not surprising that the latter outperforms the former. However, by considering a simplified version of the game, we argue that the majority of this apparent advantage is not attributable to the difference in information available to the two agents, but is instead a consequence of problems with the method of determinization itself.

在本文中，我们比较了（在流行的中国纸牌游戏《斗朱迪》中）使用确定版 UCT 的代理人和能够“作弊”并观察游戏实际状态的 UCT 代理人的游戏实力。后者的表现优于前者并不奇怪。然而，通过考虑游戏的简化版本，我们认为这种明显的优势主要不是由于两个代理人可用信息的差异，而是确定性方法本身存在问题的结果。

There are three main advantages of cheating over determinization:

作弊相对于确定性有三个主要优势:

- Lack of *non-locality* and *strategy fusion* (two common problems with determinization that are discussed in Section II.B);
- 缺乏非局部性和策略融合（这是第二部分中讨论的确定性的两个常见问题。b）;
- *Inference* (knowledge of information that could conceivably be inferred from the opponent’s decisions);
- *推理（可以从对手的决策中推断出的信息知识）;*
- *Clairvoyance* (knowledge of information that could not possibly be inferred).
- *千里眼（无法推断的信息知识）。*

Clearly the third of these advantages is insurmountable for a player that does not cheat. Our experimental evidence suggests that the effect of the second is negligible for Dou Di Zhu (this may be different for other games), and so addressing the first is the most promising avenue to obtaining a strong imperfect information player.

显然，第三个优势对于一个不作弊的玩家来说是不可逾越的。我们的实验证据表明，第二个问题对斗朱迪的影响可以忽略不计（这可能与其他游戏不同），因此解

决第一个问题是获得强不完美信息玩家的最有希望的途径。

In light of this, we modify UCT to directly search trees for games of imperfect information, with the aim of overcoming the weaknesses of determinization. Our algorithm does not significantly outperform determinized UCT on average, but it succeeds in performing well for precisely those deals where the advantage of cheating UCT over determinized UCT is largest.

有鉴于此，我们修改UCT直接搜索树的不完美信息博弈，目的是克服确定性的弱点。平均而言，我们的算法并没有显著优于确定性UCT，但它成功地在欺骗UCT相对于确定性UCT的优势最大的交易中表现良好。

978-1-4577-0011-8/11/$26.00 ©2011 IEEE

III. Background and Literature Review

IV. 背景和文献综述

A. *Definitions and notation*

B. *定义和符号*

This section introduces briefly the notions of *uncertainty* (stochasticity and/or imperfect information) in games, as well as the notation we use throughout this paper. For more details on the concepts introduced here we refer the reader to a standard textbook on game theory, e.g. [10].

本节简要介绍游戏中不确定性（随机性和/或不完美信息）的概念，以及我们在本文中使用的符号。关于这里介绍的概念的更多细节，我们建议读者参考博弈论的标准教科书，例如【10】。

A *game* can be thought of as a Markov process where, at each state, one of several agents (or *players*) is allowed to make a decision which influences the transition to the next state. A game is said to be *deterministic* if taking a particular action from a given state always leads to the same state transition. If a game is not deterministic, i.e. there is some element of randomness (or *chance*) to the transitions, the game is said to be *stochastic*. A game has *perfect information* if all players are able to observe the current state of the game. If a game does not have perfect information, i.e. the underlying Markov process is partially observable, the game is said to have *imperfect information*. For example, many card games are stochastic (because they are played with a shuffled deck) with imperfect information (because players are unable to observe opponents' cards).

游戏可以被认为是一个马尔可夫过程，在每个状态下，几个代理人（或玩家）中的一个被允许做出决定，该决定影响到下一个状态的转换。如果从一个给定的状态采取一个特定的动作总是导致相同的状态转换，那么我们就说这个游戏是确定性的。如果一个游戏不是确定性的，也就是说，过渡存在一定的随机性（或偶然性），那么这个游戏就被称为随机的。如果所有玩家都能观察到游戏的当前状态，那么游戏就具有完美的信息。如果一个博弈不具有完美信息，即潜在的马尔可夫过程是部分可观测的，则称该博弈具有不完美信息。例如，许多纸牌游戏是随机的（因为它们是用一副洗牌的牌玩的），信息不完善（因为玩家无法观察对手的牌）。

In a game of imperfect information, the states as observed by each player are partitioned into *information sets*, defined as sets of states that are indistinguishable from the player's point of view. During the game, the player cannot observe the current state, but can observe the current information set.

In the notation of this paper, a game is defined by a set

在不完全信息游戏中，每个玩家观察到的状态被划分为信息集，这些信息集被定义为从玩家的角度无法区分的状态集。在游戏过程中，玩家无法观察到当前状态，但可以观察到当前的信息集。在本文的符号中，一个游戏由一个集合定义

$\mathbf{\Lambda} \subseteq S \times A$ of *state-action pairs* (for some sets S and A of *states* and *actions* respectively), along with a *state transition function* $f: \mathbf{\Lambda} \to S$. The set of *legal actions* for a state $s \in S$ is denoted $A(s)$, and is defined as the set of actions $a \in A$ for which $(s, a) \in \mathbf{\Lambda}$. The *player about to act* in a state s is denoted $\rho(s)$. If at time t the game is in state s, then player

状态–动作对的$\mathbf{\Lambda} \subseteq S \times A$（对于某些集合，分别为状态和动作的$S$和$A$），以及状态转移函数$f$：$\mathbf{\Lambda} \to S$. 一个州的法律行为集合$s \in S$表示为$A(s)$，并被定义为$a \in A$的行为集合，其中（$s$，$a$）$\in \mathbf{\Lambda}$. 将要在$s$州行动的玩家表示为$\rho(s)$. 如果在$t$时间游戏处于$s$状态，那么玩家

$\rho(s)$ chooses an action $a \in A(s)$, and the game transitions to state $f(s, a)$ at time $t + 1$. At a given moment, player j cannot observe the entire state s, but knows that the actual state must be in *information set* $[s]$j. If two states are in the same information set then the player about to act must be the same in both; if the player about to act is the observer j, then the legal actions must also be the same. Thus we define

$\rho(s)$选择动作$a \in A(s)$，并且游戏在时间$t + 1$过渡到状态f（s，a。在给定时刻，玩家 j 无法观察到整个状态s，但知道实际状态必须在信息集中【s]j. 】如果两个状态在同一信息集中，那么即将采取行动的玩家在两个信息集中必须相同；如果将要采取行动的玩家是观察者 j，那么法律行动也必须相同。因此我们定义

$\rho([s]$j), and $A([s]$j) in the case where $\rho([s]$j) $=$ j, in the natural way.

$\rho([s]$j），$A([s]$j）在$\rho([s]$j = j 的情况下，以自然的方式。

C. *Determinization*

D. *确定化*

One approach to designing AI for games with stochasticity and/or imperfect information is *determinization*. For a stochastic game with imperfect information, a *determinization* is an instance of the equivalent deterministic game of perfect information, in which the current state is chosen from the AI agent's current information set, and the outcomes of all future chance events are fixed and known. For example, a determinization of a card game is an instance of the game where all players' cards and the shuffled deck are visible to all players. We can

为具有随机性和/或不完美信息的游戏设计人工智能的一种方法是确定性。对于具有不完全信息的随机游戏，确定化是完全信息的等价确定性游戏的一个实例，其中当前状态是从 AI 智能体的当前信息集中选择的，并且所有未来机会事件的结果都是固定和已知的。例如，纸牌游戏的确定性是所有玩家的牌和洗牌对所有玩家可见的游戏实例。我们能

sample several determinizations from the current information set and for each one analyse the value of each action using AI techniques for deterministic games of perfect information. An example is Ginsberg's GIB system

从当前信息集中抽取几个确定性样本，并对每个样本使用AI技术分析每个行动的价值，以进行完美信息的确定性博弈。金斯伯格的GIB系统就是一个例子

[1] which applies determinization to create an AI player for the card game Bridge that plays at the level of human experts. Buro et al [23] apply determinization to the card game Skat, attaining the level of a human expert player. Bjarnason [2] applies a determinized variant of UCT to the single-player card game Klondike Solitaire, resulting in an agent that achieves more than twice the estimated win rate of a human player. Determinized MCTS also shows promise in games such as Phantom Go [11] and Phantom Chess (Kriegspiel) [12], among others.

【1】它应用确定性为纸牌游戏桥牌创建了一个AI玩家，该玩家可以在人类专家的水平上进行游戏。Buro等人【23】将确定性应用于纸牌游戏Skat，达到了人类专家玩家的水平。比雅纳松【2】将UCT的一种确定性变体应用于单人纸牌游戏克朗代克纸牌游戏，其结果是一个智能体达到了人类玩家预计胜率的两倍以上。意志坚定的 MCTS 在诸如幻影围棋【11】和幻影象棋（Kriegspiel）【12】等游戏中也表现出了希望。

Despite these successes, determinization is not without its critics. Russell and Norvig [13] point out that "averaging over clairvoyance" will never result in plays that gather or hide information: from the point of view of the decision-making process in each determinization, all information is visible so there is nothing to gather or hide. Ginsberg [1] makes the same observations about the behaviour of GIB. Frank and Basin [4] identify two key problems with determinization:

尽管取得了这些成功，但决定论并不是没有批评者。Russell和nor vig【13】指出"平均预测"永远不会导致收集或隐藏信息的行为:从每个确定性决策过程的角度来看，所有信息都是可见的，因此没有什么可收集或隐藏的。金斯伯格【1】对GIB的行为进行了相同的观察。Frank和Basin【4】指出了确定性的两个关键问题:

- *Strategy fusion.* An AI agent cannot make different decisions from different states in the same information set; however, the deterministic solvers can and do make different decisions in different determinizations.
- *策略融合。一个AI智能体不能从同一信息集中的不同状态做出不同的决策；然而，确定性求解器可以在不同的确定性中做出不同的决策。*
- *Non-locality.* Some determinizations may be vanishingly unlikely (rendering their solutions irrelevant to the overall decision process) due to the other players' abilities to direct play away from the corresponding states.
- *非本地性。由于其他玩家的能力使游戏远离相应的状态，一些确定性可能几乎不存在（使他们的解决方案与整个决策过程无关）。*

E. Dou Di Zhu

F. 窦

Dou Di Zhu [14] is a 3-player gambling card game, in the class of climbing games but also with bidding elements similar to trick taking games. Dou Di Zhu originated in

China, and has increased in popularity there in recent years, particularly with internet versions of the game. One website reports 1,450,000 players per hour, and a 2008 tournament attracted 200,000 players [15].

《斗朱迪》【14】是一款三人赌博卡牌游戏，属于攀爬类游戏，但也有类似于整蛊游戏的叫牌元素。《斗朱迪》起源于中国，近年来在中国越来越受欢迎，尤其是这款游戏的网络版。一个网站报道称每小时有 145 万名玩家，2008 年的一场锦标赛吸引了 20 万名玩家【15】。

The full rules of Dou Di Zhu are given in [14], and the variant we study is discussed in [16]. This section outlines the key features of the game from an AI perspective.

窦朱迪的完整规则在【14】中给出，我们研究的变体在【16】中讨论。本节从人工智能的角度概述了游戏的主要功能。

At the beginning of the game, cards are dealt to the players and one player is designated the *landlord*. Starting with the landlord, players take turns to play groups of cards from their hands. There are several *categories* of group: for example, one may play a single card, or a pair of cards of the same rank, or a sequence of five cards of consecutive rank, and so on. The first player can play any group available to them; subsequent players must play a group of the same category but of higher rank. Players also have the option of passing, regardless of whether they are able to play a group of cards. If two consecutive players pass, the third (i.e. the last player to play any cards) once again has a free choice of

游戏开始时，向玩家分发纸牌，一名玩家被指定为地主。从地主开始，玩家轮流从他们的手中打出几组牌。分组有几种类型:例如，一个人可以玩一张牌，或一对相同等级的牌，或一系列连续等级的五张牌，等等。第一个玩家可以玩他们可用的任何组；随后的玩家必须参加同一类别但等级更高的一组游戏。玩家也可以选择传球，不管他们是否能打出一组牌。如果连续两名玩家过关，第三名玩家（即最后一名出牌的玩家）可以再次自由选择

group from any category. Cards are discarded once they are played, and the goal of the game is to be the first to play all cards in hand. If one of the non-landlord players achieves this then both are declared equal winners; if the landlord achieves it then he is the sole winner. Thus the non-landlord players must cooperate to compete against the landlord.

来自任何类别的组。牌一旦打出就被丢弃，游戏的目标是第一个打出手中所有的牌。如果其中一个非地主玩家达到了这个目标，那么双方将被宣布为平等的赢家；如果房东成功了，那么他就是唯一的赢家。因此非地主玩家必须合作来对抗地主。

The branching factor in Dou Di Zhu varies based on the stage of the game, and particularly whether the current player must beat a previous group or has a free choice of category. For the first decision node in the game, a branching factor between 20 and 70 is typical; for subsequent nodes the branching factor is typically less than 10, and often 1 (when a player has no option but to pass).

《斗朱迪》中的分支因素因游戏阶段而异，特别是当前玩家是否必须击败前一组玩家或者是否可以自由选择类别。对于游戏中的第一个决策节点，20 到 70 之间的分支因子是典型的；对于后续节点，分支因子通常小于 10，通常为 1（当玩家别无选择只能传球时）。

In some categories, *kicker* cards are played along with the main cards in the group. This can result in a large branching factor: if there are n choices for the main group and m for the kickers then there are mn available moves in total. We address this with an approach similar to the move grouping approach of Childs et al [17], treating the choices of main group and kickers as two separate decisions at consecutive levels in the game tree. This increases the number of nodes and levels in the tree but decreases the branching factor at each node, and so is beneficial for tree search. Without this modification, the branching factor at the game's initial decision node can often exceed 300.

在某些类别中，踢球者卡与组中的主卡一起使用。这可能导致一个很大的分支因素：如果主组有n的选择，踢球者有m的选择，那么总共就有mn的选择。我们采用类似于 Childs 等人【17】的移动分组方法来解决这个问题，将主组和踢球者的选择视为博弈树中连续层次上的两个独立决策。这增加了树中的节点和级别的数量，但减少了每个节点的分支因子，因此有利于树搜索。如果没有这种修改，游戏初始决策节点的分支因子通常可以超过 300。

Dou Di Zhu is a game of imperfect information, as each player's cards in hand are hidden from the other two players (although the number of cards in hand is not hidden). A perfect information variant of Dou Di Zhu can be defined in the natural way, by allowing players to see each other's hands. It is worth noting that the perfect information variant retains some of the strategic depth of the original game; this is in contrast to games such as poker, which become trivial once the aspect of hidden information is removed.

斗朱迪是一个不完全信息游戏，因为每个玩家手里的牌对其他两个玩家都是隐藏的（尽管手里的牌数是不隐藏的）。斗的完美信息变体可以以自然的方式定义，即允许玩家看到彼此的手牌。值得注意的是，完美信息变体保留了原游戏的一些战略深度；这与扑克等游戏形成鲜明对比，一旦隐藏信息方面被移除，这些游戏就会变得微不足道。

G. *Mini Dou Di Zhu*

H. *迷你豆朱迪*

We introduce a simplified version of Dou Di Zhu that removes some of the complications of the full game and is small enough to be solved with exhaustive tree search techniques, while still retaining some of the strategic depth of the full game.

我们介绍了一个简化版的斗朱迪，它消除了整个游戏的一些复杂性，并且足够小，可以用穷尽树搜索技术解决，同时仍然保留了整个游戏的一些战略深度。

Mini Dou Di Zhu is a 2-player game, played with a reduced deck of 18 cards (four ranks and two jokers). Each player receives seven cards, and the four remaining cards are hidden from both players. There are four move categories, consisting of 1, 2, 3 or 4 cards of the same rank. As with the full game, the aim is to be the first player to play all cards, but unlike the full game there is no element of cooperation.

迷你斗朱迪是一款双人游戏，共 18 张牌（四个级别和两个小丑）。每个玩家得到七张牌，剩下的四张牌对两个玩家都是隐藏的。有四种移动类别，由 1、2、3 或 4 张相同等级的牌组成。与完整游戏一样，目标是成为第一个打出所有牌的玩家，但与完整游戏不同的是，这里没有合作元素。

The total number of distinct deals in Mini Dou Di Zhu is 8832. The game trees for the perfect information variant are small enough that minimax search can be used to exactly determine the game theoretic value of each perfect information deal; when the non-uniform probabilities of obtaining each deal by shuffling and dealing cards are taken into account, approximately 70.7% of games are wins for player 1.

迷你豆朱迪的不同交易总数为 8832 笔。完美信息变体的博弈树足够小，可以使用极大极小搜索来精确确定每个完美信息交易的博弈理论值；当考虑到通过洗牌和发牌获得每张牌的非均匀概率时，大约 70.7%的游戏是玩家 1 获胜。

V. DESIGN OF ALGORITHMS

VI. 算法设计

The game tree for full Dou Di Zhu has a single chance node corresponding to the dealing of cards to each player. However, even after fixing the root player's own cards, the branching factor at this node is of the order 10^6, so searching this tree directly is impractical. Thus we introduce two classes of algorithm: one that uses determinization and searches each determinized game tree individually, and one that constructs and searches a tree of information sets. For comparison, we also consider "cheating" algorithms, which are allowed to observe the true state of the game and thus search the actual game tree. Each of these classes contains two algorithms: one based on exact tree search techniques (i.e. minimax or expectimax), and one based on MCTS. The former can only be applied to small games (such as Mini Dou Di Zhu), but they give us insight into the strengths and weaknesses of the latter.

全斗朱迪的游戏树有一个对应于向每个玩家发牌的单一机会节点。然而，即使在修复了根玩家自己的卡之后，该节点处的分支因子的数量级为 106，因此直接搜索该树是不切实际的。因此，我们引入两类算法:一类使用确定性并单独搜索每个确定性博弈树，另一类构建并搜索信息集树。为了进行比较，我们还考虑了“作弊”算法，这些算法被允许观察游戏的真实状态，从而搜索实际的游戏树。每一类都包含两种算法:一种基于精确树搜索技术（即 minimax 或 expectimax），另一种基于 MCTS。前者只能应用于小型游戏（如迷你斗朱迪），但它们让我们深入了解了后者的优势和劣势。

A. *Cheating minimax*

B. *作弊最小最大*

The minimax algorithm can easily search the entire depth of the Mini Dou Di Zhu game tree. Minimax is optimal against a minimax opponent, but can occasionally make poor decisions against other types of opponent. For example, suppose that the minimax player has available two lines of play: one is a certain loss, while the other is a loss only if the opponent plays optimally from that point. Both lines have a minimax value of -1, and so minimax chooses between them arbitrarily. However, if there is any possibility that the opponent will make a "mistake" (i.e. deviate from the minimax policy, which is not unlikely if the opponent does not cheat), the second line is clearly the better choice.

极大极小算法可以很容易地搜索迷你斗朱迪游戏树的整个深度。极小极大是对付极小极大对手的最佳选择，但在对付其他类型的对手时偶尔会做出糟糕的决定。例如，假设最小最大玩家有两条路线可供选择:一条是一定会输，而另一条是只有当对手从该点开始以最佳方式玩时才会输。两条线的最大最小值都是 1，因此最大最小值会在它们之间任意选择。然而，如果对手有任何可能犯“错误”（即偏离最小最大政策，如果对手不作弊，这种可能性也不是没有），那么第二条线显然是更好的选择。

To solve this problem, we equip minimax with the following tie-breaking mechanism. Each state s is assigned a value $m_s(s)$ by

为了解决这个问题，我们为 minimax 配备了以下平局决胜机制。每个州s被分配一个ms（s值

$$m_s(s) = \max_{a \in A(s)} -m_s(f(s,a)) + \varepsilon \left(\frac{\sum_{a \in A(s)} -m_s(f(s,a))}{|A(s)|} \right) . \quad (1)$$

a的马克斯・$-ms$（f（s）

$a \in A\ s$

\+ ε（$\sum a \in A$（s） $-ms$（f（s，a））） .（1）

$|A(s)|$

The first term is the standard negamax formulation of the minimax value; the second term is proportional to the expected value of playing a random action from state s. If two moves have the same minimax value, the tie will be broken by choosing the move that gives the opponent more opportunities to make a mistake. The constant ε must be small enough that if $m_0(s) < m_0(s')$ (where m_0 denotes the standard minimax value) then $m_s(s) < m_s(s')$, so that actions that maximise m_s also maximise m; in other words, the set of actions identified as optimal by m_s is a subset of those identified as optimal by m.

第一项是 minimax 值的标准 negamax 公式；第二项与s.州随机行动的期望值成比例如果两个移动具有相同的最小最大值，则通过选择给对手更多犯错机会的移动来打破平局。常数ε必须足够小，如果$m0$（s）《$m0$（s'（其中$m0$ 表示标准最小最大值）则ms（s）《ms（s′），从而使ms最大化的动作也使m最大化；换句话说，ms确定为最佳的一系列行动是m.确定为最佳的行动的子集

C. *Cheating UCT*

D. *欺骗 UCT*

For consistency with determinized UCT (Section III.D), our cheating UCT agent uses multiple independent search trees: the root of each tree corresponds to the same (current)

为了与确定的 UCT 保持一致（第三节。我们的作弊 UCT 代理使用多个独立的搜索树：每棵树的根对应于相同的（当前的）

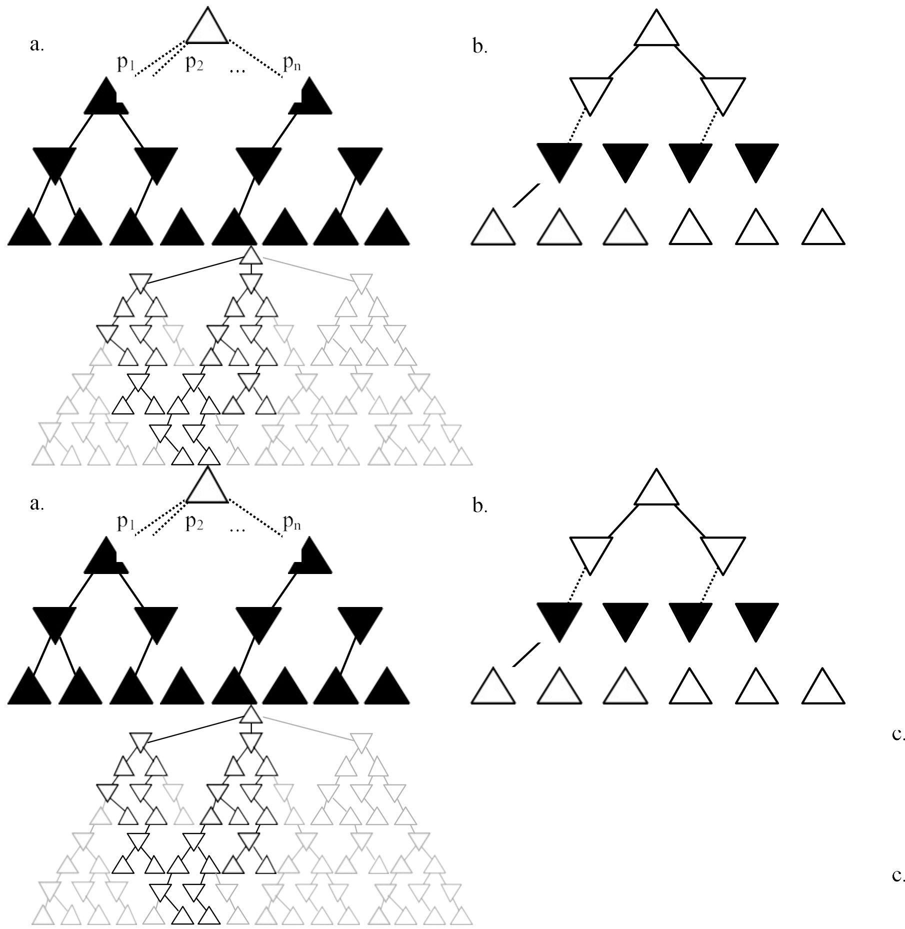


Fig. 1. Illustration of search trees for (a) determinization, (b) information set expectimax and (c) information set UCT. White triangles correspond to information sets, and black triangles to individual states. Triangles pointing upwards denote nodes belonging to the root (maximising) player, and triangles pointing downwards to the opponent (minimising) player. Solid lines denote decisions, whereas dashed lines denote decomposition of information sets into their constituent states. In (c), black nodes and edges are compatible with the current determinization and grey nodes and edges are not. All information sets are from the point of view of the root player.

图一。（a）确定性，（b）信息集 expectimax 和（c）信息集 UCT 的搜索树图示。白色三角形对应于信息集，黑色三角形对应于单个状态。向上的三角形表示属于根（最大化）玩家的节点，向下的三角形指向对手（最小化）玩家。实线表示决策，而虚线表示将信息集分解成其组成状态。在（c）中，黑色节点和边缘与当前确定性兼容，而灰色节点和边缘不兼容。所有信息集都是从根玩家的角度来看的。

state of the game, but each tree is searched by UCT independently. After the searches are completed, the numbers of visits for each action from the root are summed across all trees, and an action is chosen that maximises the total number of visits. This algorithm is equivalent to UCT with *root parallelization*, also known as *single-run parallelization* [18]. As Chaslot et al [19] observe, root parallelisation affects how UCT explores the game tree, and particularly the ability of UCT to escape local optima. For this reason, comparing determinized UCT (which uses multiple search trees) against UCT with root parallelisation is a fairer comparison than determinized UCT against UCT with a single search tree.

游戏状态，但每棵树都是由 UCT 独立搜索的。搜索完成后，从根开始的每个操作的访问次数在所有树中求和，并选择一个使总访问次数最大化的操作。该算法相当于根并行化的 UCT，也称为单运行并行化【18】。正如查斯洛等人【19】所观察到的，根并行化影响了 UCT 探索博弈树的方式，尤其是 UCT 摆脱局部最优的能力。出于这个原因，将确定的 UCT（使用多个搜索树）与具有根并行化的 UCT 进行比较，比将确定的 UCT 与具有单个搜索树的 UCT 进行比较更公平。

UCT does not appear to be as susceptible as minimax to the effects of poor tie-breaking. Exploration and random simulations in UCT are somewhat analogous to the second term in Equation (1), in that they cause suboptimal opponent actions to be factored into the evaluations of nodes.

UCT 似乎不像极小极大那样容易受到糟糕的平局决胜的影响。UCT 的探索和随机模拟有点类似于等式（1）中的第二项，因为它们导致次优对手行动被计入节点的评估中。

E. *Determinized minimax*

F. *确定的极小极大*

We apply a determinization approach similar to the approach Ginsberg [1] uses for Bridge, searching multiple determinizations (states in the current information set) at each decision step.

我们应用了一种确定性方法，类似于金斯伯格【1】用于桥接的方法，在每个决策步骤中搜索多个确定性（当前信息集中的状态）。

take the more conventional approach of sampling determinizations at random. So that the most likely determinizations have the greatest influence over the decision process, determinizations are sampled according to the probabilities of the states in the information set. Fixing the agent’s own cards and randomly dealing the unseen cards to the opponents achieves this distribution.

采取更传统的随机抽样确定方法。为了使最可能的确定对决策过程具有最大的影响，根据信息集中状态的概率对确定进行采样。固定代理人自己的牌，并随机将看不见的牌发给对手，从而实现这种分配。

As with cheating UCT, the results of the individual searches are combined by summing numbers of visits from the root.

与作弊 UCT 一样，个人搜索的结果是通过对根网站的访问量求和而得到的。

E. Information set expectimax

E. 信息集预期最大值

Instead of searching determinized trees of states, the two algorithms below search trees of information sets. For Mini Dou Di Zhu, we use the *expectimax* search algorithm [13], which modifies the minimax algorithm to game trees

In Mini Dou Di Zhu, the maximum number of states in an information set is 66, so it is feasible to iterate over all

在迷你 Dou 朱迪中，信息集中状态的最大数量是 66，因此对所有状态进行迭代是可行的

containing chance nodes: the value of a chance node is the expected value of choosing one of its children at random. For trees of information sets, we treat the opponent's decision nodes as chance nodes (branching for the states in the current information set) followed by perfect information decision nodes. Each nonterminal information set $[s]_j$ is

下面的两个算法不是搜索确定的状态树，而是搜索信息集的树。对于迷你斗朱迪，我们使用 expectimax 搜索算法【13】，该算法将 minimax 算法修改为包含机会节点的博弈树:机会节点的值是随机选择其子节点之一的期望值。对于信息集的树，我们将对手的决策节点视为机会节点（当前信息集中状态的分支），后面是完美信息决策节点。每个非终结信息集【s]j 是

assigned a value $v_j([s]_j)$ recursively by

被递归赋值vj([s]j)

$$\max$$

$$v([s]) = \{$$

$$v([s] \{$$

最大
$a \in A([s]_j)$
$a \in A([s]_j)$

$v_j([f(s, a)]_j)$ if $\rho([s]_j) = j$ $a)]_j v_j([f(s)$ 如果 $\rho([s]_j = j$

possible determinizations. We combine the results of the j j $\mathbb{E}$ $\mathbb{E}$
可能的决定。我们结合了

$\min \ v([f(s, a)])$ if $\rho([s]) \text{ G } j$,

闵 v 】( )【 f 如 (s, a) 果 $\rho([s】)$ g j,

$s \in [s]_j$ $a \in A(s)$ j j j

individual minimax searches by weighted majority voting: the “number of votes” for an action is the sum of probabilities of determinizations in which that action is chosen by minimax, and the action selected by the agent is one for which this sum is maximal.

通过加权多数投票进行单个极小极大搜索:一个动作的“投票数”是极小极大选择该动作的概率之和，代理选择的动作是该和最大的动作。

The determinization process is illustrated in Fig. 1a. There is an initial branch for each possible determinization for the current information set, each labelled with its associated probability. Each determinization fixes the structure of the tree that is then searched by minimax.

图 1a 中示出了确定过程。当前信息集的每个可能确定性都有一个初始分支，每个分支都标有其相关概率。每一种确定性都固定了树的结构，然后用极大极小搜索树。

D. Determinized UCT

D. 确定的 UCT

Information sets in the full game of Dou Di Zhu are generally several orders of magnitude larger than those for Mini Dou Di Zhu: over the 1000 initial deals used in this paper (see Section IV.A), the maximum number of states in an information set is just over 2.6 million. Thus we must

在斗的整个游戏中，信息集通常比迷你斗的信息集大几个数量级:本文使用了 1000 多个初始交易（见第四节）。a），一个信息集中的最大状态数刚刚超过 260 万。因此我们必须

$s \in [s]_j \quad a \in A(s)$

with terminal information sets assigned values of ± 1 for wins and losses in the usual way. The agent selects a move to maximise the value of the resulting information set. The search tree is illustrated in Fig. 1b.

终端信息集以通常的方式为赢和输分配值 1。代理选择一个移动来最大化结果信息集的价值。图 1b 中示出了搜索树。

For the notion of a tree of information sets to make sense, we must have a well-defined mapping from (information set, action) pairs to information sets. In other words, if states s and s' are in the same information set then we must have

为了使信息集树的概念有意义，我们必须有一个定义良好的从（信息集、动作）对到信息集的映射。换句话说，如果s和s' 在同一个信息集中，那么我们必须

$f(s, a)$ in the same information set as $f(s', a)$ for all legal actions a, so that the extension of f to information sets given by $f([s]_j, a) = [f(s, a)]_j$ is well-defined. This is true for Dou Di Zhu, but is not true for games in general; extending the idea of information set search to games where this property does not hold is a subject for future work.

$f(s, a)$ 与$f(s', a)$ 在同一个信息集中，以便将f扩展到$a f([s]_j$ 给出的信息集。这对于斗来说是正确的，但对于一般的游戏来说并不正确；将信息集搜索的思想扩展到不具备这一特性的游戏是未来工作的主题。

Define:
$\mu(N)$ = {All information sets reachable from the information set at node N}
$\mu(N, D) = \{n \in \mu(N) \mid n$ contains a state reachable from N in determinization $D\}$
$\mu T(N) = \{n \in \mu(N) \mid n$ is in the search tree $T\}$
$\mu T(N, D) = \{n \in \mu(N, D) \mid n$ is in the search tree $T\}$
$V(N)$ = number of visits to node N
$R(N)$ = total reward at node N

Repeat for a large number of iterations
N = root node of tree T
D = a random determinization of N
// Selection
Repeat until $\mu T(N, D) = \emptyset$ or $\mu(N, D) \setminus \mu T(N, D)$ G $\emptyset$

Choose a child M of N for which $\frac{R(M)}{V(M)} + k\sqrt{\frac{\ln V(N)}{V(M)}}$ is maximal
Let $N = M$
// Expansion
If $\mu(N, D) \setminus \mu T(N, D)$ G $\emptyset$
Add any $M \in \mu(N, D) \setminus \mu T(N, D)$ to T
Let $N = M$
// Simulation
r = result of performing a simulation restricted to D from N
// Backpropagation
For each ancestor M of N, including N itself and the root node Increment $V(M)$ by 1
Increase $R(M)$ by r


```
Define:
μ(N) = {All information sets reachable from the information set at node N}
μ(N, D) = {n ∈ μ(N) | n contains a state reachable from N in determinization D}
μT(N) = {n ∈ μ(N) | n is in the search tree T}
μT(N, D) = {n ∈ μ(N, D) | n is in the search tree T}
V(N) = number of visits to node N
R(N) = total reward at node N

Repeat for a large number of iterations
N = root node of tree T
D = a random determinization of N
// Selection
Repeat until μT(N, D) = Ø or μ(N, D) \ μT(N, D) G Ø

    Choose a child M of N for which R(M)/V(M) + k √(lnV(N)/V(M)) is maximal
  Let N = M
  // Expansion
  If μ(N, D) \ μT(N, D) G Ø
  Add any M ∈ μ(N, D) \ μT(N, D) to T
  Let N = M
  // Simulation
  r = result of performing a simulation restricted to D from N
  // Backpropagation
  For each ancestor M of N, including N itself and the root node Increment V(M) by 1
  Increase R(M) by r
```


Fig. 2. Pseudocode demonstrating how information set UCT constructs and searches a decision tree.

图二。演示信息集 UCT 如何构建和搜索决策树的伪代码。

F. *Information set UCT*

G. *UCT 信息集*

Just as determinized (or cheating) UCT can be thought of as an analogue to determinized (or cheating) minimax, we introduce information set UCT as an analogue to information set expectimax.

正如确定的（或欺骗的）UCT 可以被认为是确定的（或欺骗的）极小极大的类似物一样，我们引入了信息集 UCT 作为信息集 expectimax 的类似物。

In general the set of legal opponent actions from the current information set is not known, as it is a function of the (hidden) state of the game. What is known however is the set of actions that are legal in at least one state in the information set, i.e. the union of legal action sets for all states in the information set. Applying one of these actions leads to a new information set. Thus a tree of information sets can be constructed; the construction is similar to that of the information set expectimax tree, except that states are not explicitly considered at any point.

一般来说，来自当前信息集的合法对手行动集是未知的，因为它是游戏的（隐藏）状态的函数。然而，已知的是在信息集中至少一个州合法的动作集合，即信息集中所有州的合法动作集合的并集。应用这些操作之一会产生一个新的信息集。因此，可以构建信息集树；该结构类似于信息集 expectimax 树的结构，只是在任何时候都没有显式地考虑状态。

Some lines of play may be more likely than others, and some lines of play are more relevant to the decision than others. The search must strike a balance between the two when modelling decisions at opponent nodes. To gain a more realistic estimate of the outcome of the game, UCT's selection phase should generally favour a move that is likely to be available and has a good expected return over a highly unlikely move with an extremely high expected reward, although the latter cannot be ignored entirely. We achieve this by searching only one information set tree, but restricting each UCT iteration to a subset of the tree corresponding to a random determinization. Unlikely moves will be available in relatively few determinizations and so will be selected only rarely, but moves with high rewards

一些行动路线可能比其他路线更有可能，一些行动路线比其他路线与决策更相关。当在对手节点模拟决策时，搜索必须在两者之间取得平衡。为了获得对比赛结果的更现实的估计，UCT 的选择阶段通常应该倾向于可能出现且有良好预期回报的举动，而不是预期回报极高但可能性极低的举动，尽管后者不能被完全忽视。我们通过仅搜索一个信息集树来实现这一点，但是将每个 UCT 迭代限制到与随机确定性相对应的树的子集。不太可能的移动将在相对较少的决定中可用，因此只会很少被选择，但移动具有高回报

will be exploited when they are available. This algorithm can be thought of as an extreme case of determinized UCT, where each determinization is allocated only a single UCT iteration, but these iterations operate on a single tree, rather than each on a separate tree. Hence information about the quality of actions given from different determinizations is accumulated at information set nodes so that the exploration/exploitation ideas of UCT continue to work.

当它们可用时将被利用。该算法可以被认为是确定性UCT的一个极端情况，其中每个确定性仅分配给一个UCT迭代，但这些迭代在单棵树上操作，而不是在单独的树上操作。因此，由不同决策给出的关于行动质量的信息在信息集节点积累，以便UCT的勘探/开发思想继续发挥作用。

The construction of the information set tree and its restriction to a determinization D are illustrated in Fig. 1c. Pseudocode for the information set UCT algorithm is given in Fig. 2.

图1c示出了信息集树的构造及其对确定性D的限制。图2给出了信息集UCT算法的伪代码。

H. Inference

I. 推理

In games of imperfect information, it is often possible to infer hidden information by observing the actions of the other players, according to some model of the other players' decision processes. This type of inference has frequently been applied to the game of poker [20,21], but also to other games such as Scrabble [22] and the card game Skat [23] which has similarities to Dou Di Zhu.

在不完全信息的游戏中，根据其他玩家决策过程的模型，通过观察其他玩家的行为通常可以推断出隐藏的信息。这种类型的推理经常被应用于扑克游戏【20，21】，但也用于其他游戏，如拼字游戏【22】和纸牌游戏Skat【23】，这与斗朱迪有相似之处。

In Mini Dou Di Zhu, we can perform inference by applying an opponent model to all states in the current information set and comparing the observed move from the current information set with the opponent model's choice from each state: if the moves for a particular state do not match, we conclude that that state is not the true state of the game. This is Bayesian inference in the special case of pure policy opponents, where the probability of playing a move given a state is 0 or 1, similar to [22].

在迷你斗朱迪中，我们可以通过将对手模型应用于当前信息集中的所有状态，并将从当前信息集中观察到的移动与对手模型从每个状态中选择的移动进行比较来进行推断:如果特定状态的移动不匹配，我们可以得出该状态不是游戏的真实状态的结论。这是纯政策对手特殊情况下的贝叶斯推断，在给定状态下出招的概率为0或1，类似于【22】。

This type of inference requires consideration of all states in the current information set, which is infeasible for the full game of Dou Di Zhu. Developing an inference model for the full game is a subject for future work; the feature-based approach of Buro et al [23] is one possibility.

这种类型的推断需要考虑当前信息集中的所有状态，这对于斗的完整游戏是不可行的。开发完整游戏的推理模型是未来工作的主题；Buro等人【23】的基于特征的方法是一种可能性。

VII. EXPERIMENTAL RESULTS

VIII. 实验结果

A. Determinization for Dou Di Zhu

B. 窦的决定论

In [16] we study determinized UCT for Dou Di Zhu. Specifically, we investigate the effect of changing the number of randomly sampled determinizations and the number of UCT iterations performed for each determinization on the overall playing strength. We find that as long as both parameters are sufficiently large, their precise values do not have a significant impact on playing strength: 40 determinizations with 250 UCT iterations each is adequate, and so these are the parameters we use throughout this paper.

在【16】中，我们研究了对于窦的决定性。具体来说，我们研究了改变随机抽样确定性的数量和为每个确定性执行的UCT迭代的数量对整体播放强度的影响。我们发现，只要这两个参数足够大，它们的精确值就不会对播放强度产生重大影响:40次确定，每次250次UCT迭代就足够了，因此这些是我们在本文中使用的参数。

Although players’ decisions have a significant effect on the outcome of a game of Dou Di Zhu, the effect is larger in some random deals than others. In an effort to reduce the variance of subsequent results and thus allow them to be compared more easily, we use a common set of 1000 Dou Di Zhu deals for all experiments in this paper. The practice of specifying deck ordering in advance is common in Bridge

虽然玩家的决定对斗朱迪游戏的结果有很大的影响，但这种影响在一些随机交易中比其他交易中更大。为了减少后续结果的差异，从而使它们更容易比较，我们在本文的所有实验中使用一组1000个Dou朱迪交易。提前指定桥面顺序的做法在桥梁中很常见

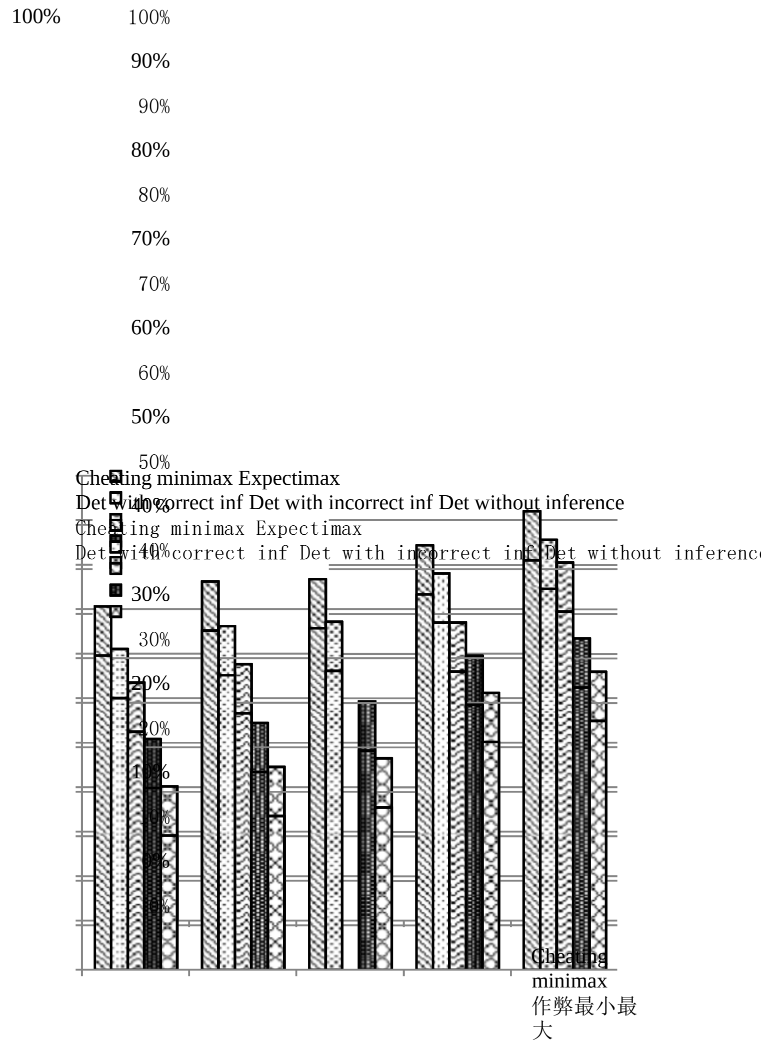
TABLE 1

表 1

Playing strength of players with perfect versus imperfect information. Each row shows the win rate for the specified player(s) when they use cheating UCT or determinized UCT and all other players use determinized UCT.

完美信息与不完美信息下玩家的博弈强度。每行显示了特定玩家使用作弊 UCT 或确定性 UCT 以及所有其他玩家使用确定性 UCT 时的胜率。

	Cheating	**Determinized**	**Difference**
Player 1	49.9%	43.0%	6.9%
Player 2	65.6%	57.0%	8.6%
Player 3	68.9%	57.0%	11.9%
Players 2 & 3	78.0%	57.0%	21.0%
	欺骗的	**确定的**	**差异**
玩家 1	49.9%	43.0%	6.9%
玩家 2	65.6%	57.0%	8.6%
玩家 3	68.9%	57.0%	11.9%
玩家 2 和 3	78.0%	57.0%	21.0%



and Whist tournaments between human players, to minimise the effect of luck when comparing players. Details of how these deals are chosen are given in [16].

和人类玩家之间的惠斯特锦标赛，以最小化比较玩家时运气的影响。如何选择这些交易的细节在【16】中给出。

In this experiment (which also appears in [16]) we tested cheating UCT and determinized UCT against each other in various combinations. We take as a baseline the number of

在这个实验中（也出现在【16】中），我们测试了作弊的 UCT，并确定了不同组合中的 UCT。我们将以下数量作为基线

with
Expecti
max Det
with
Det
with

corr
ect
inf
inco
rrec
t
正
确
的
信
息

**Op
pon
ent
(pla
yer
2)
type**
**对
手
（
玩
家
2）
类
型**

Expect
imax
Det
with
Det

Det without
inference
无推理检
测

wins (playing one game from each of the 1000 deals identified above) when each player uses determinized UCT and measure the increase in numbers of wins when various players instead use cheating UCT. Table 1 shows the results of this experiment. These figures give us upper bounds on the performance of a non-cheating agent for Dou Di Zhu.

当每个玩家使用确定的 UCT 时获胜（从上面确定的 1000 笔交易中各玩一局），并测量当不同玩家使用作弊 UCT 时获胜次数的增加。表 1 显示了该实验的结果。这些数据为我们提供了一个不作弊的窦经纪人的表现上限。

C. *Exact algorithms for Mini Dou Di Zhu*

D. *迷你斗朱迪的精确算法*

1) *Effects of cheating*

2) *作弊的影响*

To investigate the gap between cheating and determinization in Dou Di Zhu, we shift our attention to the simpler game of Mini Dou Di Zhu. This section describes an experiment to compare the playing strength of several AI agents for Mini Dou Di Zhu. Specifically, the agents are cheating minimax, determinized minimax and information set expectimax; for the latter two, we test variants with no inference model, Bayesian inference with an incorrect opponent model, and Bayesian inference with the correct opponent model. Here the “correct” opponent model uses exactly the same algorithm as the opponent, whereas the “incorrect” model uses a different, but still sensible, algorithm. The former can be considered a best case for the effectiveness of inference, whereas the latter is a more realistic test of its effectiveness against an unknown opponent.

为了研究《斗朱迪》中欺骗和确定性之间的差距，我们将注意力转移到更简单的迷你游戏《斗朱迪》上。本节描述了一个实验，以比较迷你斗朱迪的几个人工智能代理的游戏强度。具体来说，代理是欺骗极大极小值、确定极大极小值和信息集期望极大；对于后两者，我们测试了没有推理模型的变体，使用不正确的对手模型的贝叶斯推理和使用正确的对手模型的贝叶斯推理。这里“正确的”对手模型使用与对手完全相同的算法，而“不正确的”模型使用不同的但仍然合理的算法。前者可以被认为是推理有效性的最佳案例，而后者是对未知对手的有效性的更现实的测试。

In each case we iterate through all 8832 deals, playing a single game for each combination of agents; a single game suffices as all agents are deterministic. The measure of a particular agent’s strength against some opponent is the probability that it wins a randomly dealt game, which is obtained by summing the probabilities of those deals from the 8832 that it wins. The results of this experiment are shown in Fig. 3. We see that the win rate for cheating minimax is approximately 40% greater than that of determinized minimax, although the exact value depends on the opponent type. It should come as no surprise that cheating outperforms determinization. The former has access to information that the latter does not, and so can make more

在每种情况下，我们迭代所有 8832 笔交易，为每个代理组合玩一个游戏；单个游戏就足够了，因为所有的代理都是确定性的。衡量某个特定代理对某个对手的实力的标准是它赢得随机发牌游戏的概率，这是通过将它赢得的 8832 中那些发牌的概率相加而获得的。该实验的结果如图 3 所示。我们看到作弊的极小极大的胜率比确定的极小极大的胜率大约高 40%，尽管确切的数值取决于对手的类型。作弊胜过确定性并不奇怪。前者可以获得后者没有的信息，因此可以赚更多

Fig. 3. Performance of several agents in Mini Dou Di Zhu. For each group of bars, player 2 uses the algorithm specified in the x-axis label; for each bar within a group, player 1 uses the algorithm specified by the legend. The bars themselves represent the win rate over all deals for player 1. We did not test determinization with correct inference against itself: while possible, it is not straightforward to implement this in a way that does not require infinite nesting of opponent models.

图3。迷你斗中几位特工的表现。对于每组条形，玩家2使用x轴标签中指定的算法；对于组中的每个条形，玩家1使用图例指定的算法。条形本身代表玩家1在所有交易中的胜率。我们没有用针对自身的正确推断来测试确定化:尽管可能，但以不需要对手模型的无限嵌套的方式实现这一点并不简单。

informed decisions and better anticipate its opponent’s responses.

明智的决策并更好地预测对手的反应。

Expectimax is a significant improvement over determinized minimax, outperforming it by around 30% and achieving a win rate around 10% lower than that of cheating minimax. This suggests that, contrary to intuition, approximately three quarters of the apparent benefit of cheating in Mini Dou Di Zhu can actually be attributed to the effects of strategy fusion and non-locality, from which expectimax does not suffer.

Expectimax是对确定性极小极大的重大改进，比确定性极小极大高出约30%，胜率比作弊极小极大低约10%。这表明，与直觉相反，在迷你豆朱迪中作弊的大约四分之三的明显好处实际上可以归因于策略融合和非局部性的影响，而expectimax并没有受到这种影响。

3) Effects of inference

4) 推理的效果

Inference improves the strength of determinized minimax, increasing the win rate by around 10%, or 23% with a perfect opponent model. However, the performance of determinized minimax with inference still falls short of expectimax without inference. The determinized minimax agent with inference and the correct opponent model correctly identifies the actual state of the game (i.e. assigns the actual state probability 1 and all other states probability

推理提高了确定性极小极大的强度，增加了大约10%的胜率，或者在完美对手模型下增加了23%。然而，带推断的确定性极大极小值的性能仍然低于不带推断的期望极大值。具有推理和正确对手模型的确定的最小最大代理正确地识别游戏的实际状态（即分配实际状态概率1和所有其他状态概率

0) by the fourth turn in 48% of deals against an expectimax opponent; by the end of the game, this proportion of deals increases to 74%.

0）在与expectimax对手的交易中，48%的交易发生在第四轮；到游戏结束时，这一交易比例增加到74%。

Inference significantly improves the performance of determinized minimax; however, the same cannot be said for expectimax. We tested a variant of our expectimax player in which the probabilities used in calculating expected values at opponent nodes are determined by Bayesian inference. The resulting increase in playing strength is negligible: less than 0.2%. (Since these results are almost identical to those for expectimax without inference, they are omitted from Fig. 3.) One explanation could be that inferred information is

推理显著提高了确定性极大极小的性能；然而，对于expectimax来说，情况并非如此。我们测试了expectimax播放器的一个变体，其中用于计算对手节点期望值的概率由贝叶斯推理确定。由此带来的游戏强度提升微乎其微:不到0.2%。（由于这些结果几乎与没有推断的expectimax的结果相同，因此在图3中省略了它们。）一种解释可能是推断的信息

generally not helpful in Mini Dou Di Zhu, and the apparent benefit of inference in determinized minimax arises by reducing the number of determinizations to be searched, thus reducing the variance among the minimax results and ameliorating the effects of strategy fusion and non-locality.

在迷你杜朱迪中通常没有帮助，而在确定性极大极小中推断的明显好处是通过减少要搜索的确定性的数量而产生的，从而减少极大极小结果之间的方差并改善策略融合和非局部性的影响。

E. Information set UCT for Mini Dou Di Zhu

F. 迷你斗资料集

To test the effectiveness of information set UCT, it was played against determinized UCT for Mini Dou Di Zhu. In all cases player 2 used determinized UCT with 20 trees and 200 UCT iterations per tree. Each of the 8832 deals was played 10 times with player 1 using determinized UCT with 20 trees and 200 UCT iterations per tree, and 10 times with player 1 using information set UCT with 4000 iterations. Information set UCT performed better with a 67.2% win rate versus 62.7% for determinized UCT. The next section investigates whether information set UCT outperforms determinized UCT in the full version of the game.

为了测试信息集 UCT 的有效性，它与迷你斗朱迪的确定性 UCT 进行了比赛。在所有案例中，2 号玩家使用了 20 棵树和 200 次 UCT 迭代。8832 笔交易中的每笔交易都被玩家 1 使用确定性 UCT（20 棵树，每棵树 200 次 UCT 迭代）玩了 10 次，并被玩家 1 使用信息集 UCT 玩了 10 次，每次迭代 4000 次。UCT 的信息集表现更好，胜率为 67.2%，而 UCT 的胜率为 62.7%。下一节调查在游戏的完整版本中，信息集 UCT 是否优于确定性 UCT。

G. Information set UCT for Dou Di Zhu

H. 窦资料集

Experiments were run to determine the amount of exploration that should be performed (i.e. the value of k in Fig. 2) and the number of iterations required for good performance. Each of the 1000 selected deals for Dou Di Zhu was played 5 times with the landlord player as information set UCT, an exploration constant of 0.44, and varying numbers of iterations (between 500 and 30000) against determinized UCT opponents with 40 trees and 250 iterations per tree. The results indicated that the playing strength of information set UCT increased up to 10000 iterations where it achieved a win rate of 40.8%. Increasing the number of iterations further had no significant effect on playing strength. Similarly each deal was played 5 times with the landlord player as information set UCT using 10000 iterations and varying values for the UCT exploration constant (between 0.1 and 3). The results indicated that the algorithm performs poorly with exploration less than 0.5 and achieves best performance with exploration greater than 1. Increasing the exploration beyond 1 had little effect on playing strength.

进行实验以确定应该执行的探索量（即图 2 中的 k 值）和良好性能所需的迭代次数。窦朱迪的 1000 个选定交易中的每一个都与地主玩家作为信息集 UCT 玩了 5 次，探索常数为 0.44，与确定的 UCT 对手的迭代次数不同（在 500 和 30000 之间），每棵树有 40 次迭代和 250 次迭代。结果表明，信息集 UCT 的播放强度提高到 10000 次迭代，其胜率为 40.8%。进一步增加迭代次数对游戏强度没有显著影响。类似地，作为信息集 UCT，每笔交易与地主玩家玩 5 次，使用 10000 次迭代和不同的 UCT 探索常数值（在 0.1 和 3 之间）。结果表明，当探索度小于 0.5 时，该算法性能较差；当探索度大于 1 时，该算法性能最佳。将探索增加到 1 以上对游戏强度几乎没有影响。

To measure the difference between cheating UCT, determinized UCT and information set UCT, each of our 1000 deals of Dou Di Zhu was played 40 times with the landlord as cheating UCT, determinized UCT and information set UCT. In all cases 40 trees with 250 iterations per tree or 10000 iterations in total were used and the exploration constant for information set UCT was chosen to be 1.0. The opponents used determinized UCT. First of all this data was used to calculate the average win rate for each player as the landlord. The results indicate that there is no significant difference in playing strength between information set UCT (42.0%) and determinized UCT (42.4%). Cheating UCT was much stronger, achieving a win rate of 54.0%.

为了衡量欺骗、确定性的和信息集之间的区别，我们在斗地主欺骗、确定性的和信息集的情况下，将斗地主的 1000 笔交易各玩 40 次。在所有情况下，使用 40 棵树，每棵树 250 次迭代或总共 10000 次迭代，并且信息集 UCT 的探索常数被选择为 1.0。对手使用了决定性的 UCT。首先，这些数据被用来计算作为地主的每个玩家的平均胜率。结果表明，信息集 UCT（42.0%）和确定化 UCT（42.4%）的游戏强度没有显著差异。作弊的 UCT 强得多，胜率为 54.0%。

The original aim of developing information set UCT was to overcome strategy fusion difficulties in the deals where

开发信息集 UCT 的最初目的是克服交易中的战略融合困难，其中

TABLE 2

表 2

Win rates for determinized UCT and information set UCT. Each row shows win rates for a subset of the 1000 initial Dou Di Zhu deals, filtered by the difference in win rate between cheating UCT and determinized UCT (“Threshold” column).

确定性 UCT 和信息集 UCT 的胜率。每行显示了最初 1000 笔斗朱迪交易的一个子集的胜率，通过作弊的 UCT 和确定的 UCT 之间的胜率差异进行过滤（“阈值”列）。

Threshold	**Determinized UCT win rate**	**Information set UCT win rate**	**Number of deals**
None	42.4%	42.0%	1000
< 0%	53.0%	45.2%	133
= 0%	44.9%	43.3%	138
> 0%	40.0%	41.1%	729
> 25%	31.6%	38.4%	166
> 50%	20.4%	35.0%	12
> 75%	15.0%	95.0%	1
阈值	**确定的 UCT 胜率**	**信息集 UCT 胜率**	**数字 交易数量**
没有人	42.4%	42.0%	1000
< 0%	53.0%	45.2%	133
= 0%	44.9%	43.3%	138
> 0%	40.0%	41.1%	729
> 25%	31.6%	38.4%	166
> 50%	20.4%	35.0%	12
> 75%	15.0%	95.0%	一

the cheating player’s advantage is largest. To investigate whether this has been achieved, the average win rate for each deal was calculated for each player type. Then for each deal the difference in win rate between cheating UCT and determinized UCT was calculated. By looking only at deals in which this difference is above a certain threshold, it is possible to compare the performance of information set UCT and determinized UCT in the deals where knowing the hidden information is most beneficial to the cheating player. Similarly the difference between information set UCT and determinized UCT can be compared for the deals in which knowing the hidden information is not beneficial. The results of this experiment are presented in Table 2.

作弊玩家的优势最大。为了调查是否实现了这一目标，计算了每种玩家类型的每笔交易的平均胜率。然后计算每笔交易中作弊的 UCT 和意志坚定的 UCT 之间的胜率差异。通过只观察这种差异高于某个阈值的交易，可以比较信息集 UCT 和确定 UCT 在交易中的表现，其中知道隐藏信息对作弊玩家最有利。类似地，信息集 UCT 和确定性 UCT 之间的差异可以用于了解隐藏信息没有好处的交易。该实验的结果如表 2 所示。

For deals where the threshold in Table 2 is less than or equal to zero, cheating UCT does not outperform determinized UCT, and the advantage of knowing the opponent’s cards is not significant. In this situation, information set UCT offers no advantage and performs slightly worse than determinized UCT. When the threshold is greater than zero there is an advantage to knowing opponent cards, and information set UCT is stronger than determinized UCT. Furthermore, as the gap between cheating UCT and determinized UCT increases, the gap between information set UCT and determinized UCT increases also.

对于表 2 中阈值小于或等于零的交易，作弊的 UCT 并不比确定的 UCT 表现更好，并且知道对手底牌的优势并不显著。在这种情况下，信息集 UCT 没有提供任何优势，其表现略逊于确定性 UCT。当阈值大于零时，知道对手的牌是有优势的，信息集 UCT 比确定性 UCT 更强。此外，随着欺骗的 UCT 和确定的 UCT 之间的差距增加，信息集 UCT 和确定的 UCT 之间的差距也增加。

IX. CONCLUSION

X. 结论

In this paper we compared the performance of a cheating UCT agent with that of a determinized UCT agent for Dou Di Zhu. By studying the performance of exact algorithms (cheating minimax, determinized minimax and expectimax) on a simplified version of the game, we provided evidence that a large proportion of the apparent benefit of cheating has less to do with gaining access to hidden information and more to do with overcoming the inherent shortcomings of determinization. Furthermore, the most obvious approach to closing the gap between perfect and imperfect information, namely inference, has little effect on the strength of an agent that does not use determinization. (Note that these results are specific to Mini Dou Di Zhu: inference is demonstrably beneficial in many other games.)

在本文中，我们比较了一个欺骗的 UCT 代理人和一个确定的 UCT 代理人对杜・朱迪的表现。通过研究精确算法（作弊最小最大值、确定性最小最大值和预期最大值）在简化版游戏中的表现，我们提供了证据表明，作弊的很大一部分明显好处与获取隐藏信息关系不大，而与克服确定性的固有缺点关系更大。此外，缩小完美信息和不完美信息之间差距的最明显的方法，即推理，对不使用确定性的代理的强度几乎没有影响。（请注意，这些结果是针对迷你斗朱迪的:推理在许多其他游戏中显然是有益的。）

In light of this, we introduced a version of UCT that

有鉴于此，我们推出了一个版本的 UCT

operates on trees of information sets rather than trees of game states. This algorithm still uses determinization, but only as a mechanism for guiding the search during each iteration: rather than combining statistics from the root of each determinized tree after the search has ended, information set UCT combines statistics over the full depth of a single tree at every iteration of the search. Since much of the advantage of MCTS over non-tree-based Monte Carlo approaches appears to be through combining search information at all levels of the tree, information set UCT has the potential to exploit this property.

对信息集的树而不是博弈状态的树进行操作。该算法仍然使用确定性，但仅作为在每次迭代期间引导搜索的机制:信息集 UCT 在每次迭代搜索时组合单棵树的完整深度的统计信息，而不是在搜索结束后组合来自每个确定性树的根的统计信息。由于 MCTS 相对于非基于树的蒙特卡罗方法的优势似乎是通过组合树的所有级别的搜索信息，UCT 信息集有潜力利用这一特性。

One potential problem with the algorithms presented here is that they assume the opponents have access to the player's hidden information: determinization does not randomise the player's own cards, and information set trees are built solely from the point of view of the root player. In a sense this is a worst case assumption, but it does mean that these algorithms can never exploit the opponents' lack of information. However, the solution is not as simple as merely randomising one's own cards during determinization: this assumes that the agent will not know its own cards on its next turn, making it impossible to plan more than one move ahead. Addressing this issue is particularly important for games where information hiding is a significant part of successful play.

这里提出的算法的一个潜在问题是，它们假设对手可以访问玩家的隐藏信息:确定化不会使玩家自己的牌随机化，信息集树仅从根玩家的角度构建。从某种意义上说，这是一个最坏的假设，但这确实意味着这些算法永远无法利用对手缺乏信息的情况。然而，解决方案并不像在确定性过程中仅仅随机分配自己的牌那么简单:这假设代理人在下一轮不会知道自己的牌，因此不可能提前计划多个步骤。对于信息隐藏是成功游戏的重要组成部分的游戏来说，解决这个问题尤为重要。

Our experimental results show that information set UCT performs well in precisely those cases where the advantage of cheating UCT over determinized UCT is larger; unfortunately it seems that there are equally many cases where determinized UCT performs as well as cheating UCT, and slightly outperforms information set UCT, resulting in no significant increase in playing strength on average. Investigating why information set UCT fails in these cases and thus rectifying this failure is a subject for future work. If this problem can be solved, the next step is to generalise and apply the algorithm to wider classes of games with stochasticity, imperfect information and/or incomplete information.

我们的实验结果表明，信息集 UCT 在欺骗 UCT 比确定 UCT 的优势更大的情况下表现良好；不幸的是，在很多情况下，意志坚定的 UCT 表现得和作弊的 UCT 一样好，略胜于信息集 UCT，导致平均游戏强度没有显著提高。调查 UCT 信息集在这些情况下失败的原因并纠正这一失败是未来工作的主题。如果这个问题可以解决，下一步就是将该算法推广并应用到更广泛的具有随机性、不完全信息和/或不完全信息的游戏类别中。

ACKNOWLEDGEMENTS

承认

We thank Jeff Rollason of AI Factory (www.aifactory.co.uk) for introducing us to Dou Di Zhu and for several useful and interesting conversations. We also thank the anonymous reviewers for their helpful comments.

我们感谢艾工厂（）的杰夫·罗拉森向我们介绍了窦，并与我们进行了几次有益而有趣的交谈。我们也感谢匿名评论者的有益评论。

REFERENCES

参考

[1] M.L. Ginsberg, "GIB: Imperfect information in a computationally challenging game," *Journal of Artificial Intelligence Research*, vol. 14, 2001, pp. 303-358.

[2] M.L .金斯伯格，“GIB:计算挑战性游戏中的不完美信息”，《人工智能研究杂志》，第 14 卷，2001 年，第 303-358 页。

[3] R. Bjarnason, A. Fern, and P. Tadepalli, "Lower bounding Klondike solitaire with Monte-Carlo planning," *Proc. ICAPS-2009*, 2009, p. 26–33.

[4] R.比雅纳松，a .弗恩和 p .塔德帕利，“蒙特卡洛计划的克朗代克纸牌下限”，Proc。2009 年国际会计师联合会，2009 年，第 26-33 页。

[5] S. Yoon, A. Fern, and R. Givan, "FF-Replan: A Baseline for Probabilistic Planning," *Proc. ICAPS-2007*, 2007, p. 352–359.

[6] 南 Yoon，A. Fern 和 R. Givan，“FF-Replan:概率规划的基线”，Proc。2007 年国际会计师联合会，2007 年，第 352-359 页。

[7] I. Frank and D. Basin, "Search in games with incomplete

[8] I。Frank 和 d。Basin，“不完全博弈中的搜索

information: a case study using Bridge card play,”
信息:使用桥牌玩法的案例研究
Artificial Intelligence, 1998, pp. 87-123.
人工智能，1998 年，第 87-123 页。

[9] J. Long, N. Sturtevant, and M. Buro, “Understanding the Success of Perfect Information Monte Carlo Sampling in Game Tree Search,” *Proc. AAAI-10*, 2010, pp. 134-140.

[10] J.龙，n．斯特蒂文特和 m．布鲁，“理解博弈树搜索中完美信息蒙特卡罗抽样的成功”，Proc。AAAI-10，2010 年，第 134-140 页。

[11] L. Kocsis and C. Szepesvári, “Bandit based Monte-Carlo planning,” *Machine Learning: ECML 2006*, 2006, p. 282–293.

[12] 长度 Kocsis 和 C. Szepesvári，“基于 Bandit 的蒙特卡罗规划”，《机器学习:ECML 2006 年》，2006 年，第 282–293 页。

[13] R. Coulom, “Efficient selectivity and backup operators in Monte-Carlo tree search,” *Proc. 5th International Conference on Computers and Games*, Springer-Verlag, 2006, p. 72–83.

[14] R.库仑，“蒙特卡罗树搜索中的有效选择性和备份算子”，Proc。第五届计算机和游戏国际会议，施普林格出版社，2006 年，第 72-83 页。

[15] G. Chaslot, J.T. Saito, B. Bouzy, J. Uiterwijk, and H.J. Van Den Herik, “Monte-Carlo strategies for Computer Go,” *Proc. 18th BeNeLux Conference on Artificial Intelligence*, 2006, p. 83–91.

[16] G.Chaslot，J.T. Saito，B. Bouzy，J. Uiterwijk 和 H.J. Van Den Herik，“计算机围棋的蒙特卡罗策略”，Proc。第 18 届比荷卢人工智能会议，2006 年，第 83–91 页。

[17] C.S. Lee, M.H. Wang, G. Chaslot, J.B. Hoock, A. Rimmel, O. Teytaud, S.R. Tsai, S.C. Hsu, and T.P. Hong, “The computational intelligence of MoGo revealed in Taiwan s computer Go tournaments,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 1, 2009, p. 73–89.

[18] 、M.H. Wang、G. Chaslot、J.B. Hoock、A. Rimmel、O. Teytaud、S.R. Tsai、S.C. Hsu 和 T.P. Hong，“台湾计算机围棋锦标赛揭示的围棋计算智能”，《IEEE 游戏中的计算智能和人工智能汇刊》，第 1 卷，2009 年，第 73–89 页。

[19] R.B. Myerson, *Game Theory: Analysis of Conflict*, Harvard University Press, 1997.

[20] 迈尔森，《博弈论:冲突分析》，哈佛大学出版社，1997 年。

[21] J. Borsboom, J.-takeshi Saito, G. Chaslot, and J. Uiterwijk, “A comparison of Monte-Carlo methods for Phantom Go,” *Proc. 19th Belgian–Dutch Conference on Artificial Intelligence–BNAIC*, 2007.

[22] J.Borsboom，j-Takeshi Saito，G. Chaslot 和 J. Uiterwijk，“幻影围棋的蒙特卡罗方法比较”，Proc。第 19 届比利时-荷兰人工智能会议 BNAIC，2007 年。

[23] P. Ciancarini and G.P. Favini, “Monte Carlo tree search in Kriegspiel,” *Artificial Intelligence*, vol. 174, Jul. 2010, pp. 670-684.

[24] 页（page 的缩写）钱卡里尼和 G.P．法维尼，“克里格施皮尔的蒙特卡洛树搜索”，《人工智能》第 174 卷，2010 年 7 月，第 670-684 页。

[25] S.J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2009.

[26] S.J．罗素和 p．诺维格，《人工智能:现代方法》，普伦蒂斯霍尔出版社，2009 年。

[27] J. McLeod, “Dou Dizhu,” 2010. http://www.pagat.com/

climbing/doudizhu.html, accessed 28th March 2011.

[28] J.麦克劳德，《斗朱迪》，2010 年。http://www.pagat.com/climbing/doudizhu.html，2011 年 3 月 28 日访问。

[29] Wikipedia, “Dou Di Zhu,” 2011. http://en.wikipedia.org/wiki/Dou_Di_Zhu, accessed 28th March 2011.

[30] 维基百科，“斗朱迪”，2011 年。http://en.wikipedia.org/wiki/Dou_Di_Zhu，于 2011 年 3 月 28 日访问。

[31] E. Powley, D. Whitehouse, and P. Cowling, “Determinization in Monte-Carlo Tree Search for the card game Dou Di Zhu,” *Proc. AISB 2011*, 2011.

[32] E.鲍威尔、d．怀特豪斯和 p．考林，《纸牌游戏《斗朱迪》的蒙特卡罗树搜索中的确定性》。AISB 2011，2011。

[33] B.E. Childs, J.H. Brodeur, and L. Kocsis, “Transpositions and move groups in Monte Carlo tree search,” *Proc. CIG 08*, 2008, p. 389–395.

[34] B.E. Childs、J.H. Brodeur 和 L. Kocsis，“蒙特卡洛树搜索中的换位和移动组”，Proc。CIG 08，2008 年，第 389-395 页。

[35] T. Cazenave and N. Jouandeau, “On the parallelization of UCT,” *Proc. CGW07*, 2007, p. 93–101.

[36] T.卡泽纳夫和 n．朱安迪奥，“关于 UCT 的并行化”，Proc。CGW07，2007 年，第 93-101 页。

[37] G. Chaslot, M. Winands, and H. van Den Herik, “Parallel Monte-Carlo Tree Search,” *Proc. CG 2008*, 2008, p. 60–71.

[38] G.Chaslot，M. Winands 和 H. van Den Herik，“并行蒙特卡罗树搜索”，Proc。CG 2008，2008，第 60-71 页。

[39] M. Ponsen, J. Ramon, T. Croonenborghs, K. Driessens, and K. Tuyls, “Bayes-Relational Learning of Opponent Models from Incomplete Information in No-Limit Poker,” *Proc. AAAI-08*, 2008, pp. 1485-1487.

[40] 米（meter 的缩写））Ponsen，J. Ramon，T. Croonenborghs，K. Driessens 和 K. Tuyls，“从无限扑克中的不完整信息对对手模型进行贝叶斯关系学习”，Proc。AAAI 第 08 期，2008 年，第 1485-1487 页。

[41] M. Ponsen, G. Gerritsen, and G. Chaslot, “Integrating Opponent Models with Monte-Carlo Tree Search in Poker,” *Proc. Interactive Decision Theory and Game Theory Workshop at AAAI-10*, 2010, pp. 37-42.

[42] 米（meter 的缩写））Ponsen，G. Gerritsen 和 G. Chaslot，“在扑克游戏中整合对手模型与蒙特卡罗树搜索”，Proc。互动决策理论和博弈论研讨会，AAAI-10，2010 年，第 37-42 页。

[43] M. Richards and E. Amir, “Opponent modeling in Scrabble,” *Proc. IJCAI 2007*, 2007, p. 1482–1487.

[44] 米（meter 的缩写））理查兹和 e．阿米尔，“拼字游戏中的对手建模”，Proc。2007 年国际法学家委员会，2007 年，第 1482-1487 页。

[45] M. Buro, J.R. Long, T. Furtak, and N. Sturtevant, “Improving state evaluation, inference, and search in trick- based card games,” *Proc. IJCAI 2009*, 2009, pp. 1407- 1413.

[46] 米（meter 的缩写））布鲁，J.R．龙，t．福塔克和 n．斯特蒂文特，“改进基于技巧的纸牌游戏中的状态评估、推理和搜索”，Proc。2009 年国际法学家委员会，2009 年，第 1407- 1413 页。