基于蒙特卡洛搜索的炉石传说人工智能

彭翌

中山大学数据科学与计算机学院

2016 年 4 月

目录

摘要

本论文将讨论如何使用蒙特卡洛搜索方法为《炉石传说：魔兽英雄传》开发人工智能，进而指出作为博弈游戏，《炉石传说》十分适合用于人工智能研究。《炉石传说》目前已包含超过 700 张不同的卡牌，所有的这些卡牌都能够在不同程度上改变游戏进行的规则，这使得要为其开发一个完备的人工智能变得十分困难（尽管本文中只考虑《炉石传说》的一小部分卡牌）。除此之外，《炉石传说》的复杂度以及其作为卡牌游戏固有的随机性也为人工智能的开发增加了难度，这其中包括了不完全信息的搜索以及对手行为的预测。

蒙特卡洛搜索一直以来便是人工智能领域的著名算法，其延伸算法蒙特卡洛树搜索更是被用于构建了著名的围棋人工智能 AlphaGo。作为一个搜索算法，它保证其能在任何时候结束搜索并返回当前已知的最优解，而且它是非启发的 – 在不加入任何领域特定知识的情况下它也能有很好的表现。在能够进行足够多数量的游戏模拟的情况下，蒙特卡洛搜索保证能够找出当前的最优步骤。

本论文构建了完全随机、基于规则以及基于蒙特卡洛树搜索的三个炉石传说人工智能，并通过实验对比了它们的决策强度差异。同时，通过添加启发式的优化与剪枝策略，展示了向蒙特卡洛搜索中加入领域特定知识是否能显著提高搜索算法的表现。

关键词：蒙特卡洛搜索、炉石传说、人工智能

1 引言

1.1 背景

《炉石传说：魔兽英雄传》（Hearthstone: Heroes of Warcraft，下简称“炉石传说”）是由暴雪娱乐推出的线上集换式卡牌游戏。2014 年 3 月 11 日，炉石传说登陆 Microsoft Windows 平台并在全世界发行。此后，炉石传说相继登陆了 OS X、iOS 和安卓平台，吸引了大批的忠实玩家。截止至 2015 年的 11 月，炉石传说已拥有超过 4 千万的注册用户，并为暴雪娱乐带来了每月超过 2 千万美金的利润 [1]。与此同时，暴雪娱乐也为炉石传说举办了世界级的比赛以吸引人气。两届炉石传说世界锦标赛分别于 2014 年和 2015 年的暴雪嘉年华上举行，两届的冠军选手均获得了 10 万美元的奖金，奖金池总额更是达到了 25 万美元 [2]。炉石传说在 2014 年游戏大奖（the Game Awards）上被评选为年度最佳手机游戏 [3]，并在 2015 年游戏大奖上被提名为年度电子竞技游戏 [4]。

作为卡牌游戏，炉石传说也有着像桥牌那样的随机性和不完全信息性。与桥牌等传统扑克牌游戏不同的是，玩家在游戏中使用的卡组不会是固定的标准卡组，而是由他根据情况自行构建的卡组。炉石传说中的卡牌都能够在不同程度上改变游戏的规则，不同卡牌之间的相互影响更是大大提高了游戏的多样性。在本论文所讨论的炉石传说构筑模式中，玩家需要从炉石传说的所有可用的卡牌中选取出 30 张构成自己的卡组。而炉石传说中所有可用的所有卡牌种数已经达到了 743，且每年均会有 200300 张新的卡牌被加入到游戏中。游戏本身有着极高的多样性，不同的卡组可能意味着完全不同的打法，因此要一个相对有竞争力的炉石传说人工智能更类似于开发一个普适游戏人工智能（General Game Playing）[5]。本质上，炉石传说的核心规则其实极为简单，其规则的多样性主要来自于卡组的变化。由此，炉石传说十分适合作为不完全信息、普适游戏以及对手模拟的人工智能的研究对象。

这篇论文将讨论如何用蒙特卡洛方法搜索炉石传说的最优策略。基于蒙特卡洛搜索的炉石传说智能体将与基于规则的智能体进行对弈，以判明何种数量的对弈模拟可以使得蒙特卡洛智能体拥有与规则智能体同等的对弈能力。在实验中，蒙特卡洛智能体会先使用较弱的（随机）智能体进行模拟，以验证在模拟数量足够大的情况下，蒙特卡洛智能体是否能打败较强的规则智能体；同时，规则智能体也会在后面的实验中被用于蒙特卡洛智能体的模拟策略，以验证在为蒙特卡洛智能体的模拟策略加入更多的启发知识的情况下，蒙特卡洛智能体的对弈能力是否能有所提高。鉴于炉石传说本身的特性，本文中将使用基于 UCB 算法的蒙特卡洛搜索，而不使用更为强大的基于 UCT 的蒙特卡洛搜索。基于 UCT 的蒙特卡洛搜索算法在炉石传说的应用将属于本文可选的后续工作。

1.2 研究现状

卡牌游戏是典型的不完全信息随机游戏，包括了由对手手牌的不可见带来的信息隐藏，以及由下一张抽到的牌的不可知带来的随机性。信息隐藏与随机性两者相结合，使得卡牌游戏对于人类玩家或是人工智能开发来说都是十分有趣的领域 [6]。其中，被大量用于人工智能研究的卡牌游戏包括扑克牌和桥牌 [7]。

扑克牌是一款多人卡牌游戏，通常一局对弈最多可同时包含 8 名玩家。扑克牌的对弈包含了概率分析以及对手行为预测等方面 [8,9]。要成为一个扑克牌高手通常需要根据自己现有的手牌来正确估计自己的当前形势，并据此来决定自己跟或不跟。相关的人工智能研究多数集中在研究如何通过基于现有的手牌进行多次模拟来判断当前有多大几率获胜 [10]。除此之外，也有相关研究在探索如何判断对手的决策强度，并以此对智能体进行调整。研究显示，贝叶斯分析可被用于根据玩家已有的出牌方式来判断他正在使用的策略，甚至判断出他何时会改变自己的策略 [11,12]。

桥牌则是另一款卡牌游戏，一局对弈包含 4 名被分为两队的玩家。在考虑到应用蒙特卡洛搜索算法可能可以使桥牌人工智能拥有更强的决策强度的情况下，Ginsberg 运用了包含蒙特卡洛搜索算法、分部搜索 [13] 以及各类优化算法在内的相关技术开发出了一款名为 GIB 的桥牌人工智能。GIB 也是首款在决策强度上能与人类桥牌大师相提并论的桥牌人工智能。

事实证明，部分传统最小最大搜索算法所不能应付的游戏，使用蒙特卡洛搜索算法时可以有不错的效果。围棋本身也属于完全信息游戏，但它与国际象棋 [14] 等不同的地方在于，其过大的分支系数使得运用暴力搜索的人工智能难以获得良好的表现。在无法使用简单的搜索算法来开发围棋人工智能的情况下，人们提出了蒙特卡洛方法 [15]。自此，各种不同的基于蒙特卡洛方法的搜索算法被相继提出 [16]，其中包括了将蒙特卡洛方法与策略搜索相结合的搜索算法 [17] 以及后来的蒙特卡洛树搜索算法 [18]。所有的这些算法都基于蒙特卡洛方法的基本思想：与其在决策时考虑未来每一步所有可能的走法（进而产生一棵巨大的对弈树），倒不如只考虑第一步所有可能的走法，然后使用一个随机或是基于规则的策略产生器将对弈模拟到结束状态，并根据模拟的对弈结果来更新该走法的收益值。这背后的本质在于，当我们无法验证未来所有可能的对弈状态变化时，为数众多的模拟对弈的结果可以反映该走法的期望收益。

最近，研究人员也提出了一种名为赌博机决策（bandit based planning）的有趣方法 [19]。方法的名称来源于概率论中著名的多臂赌博机问题（multi-armed bandit problem）。该问题的内容大致如下：给定一排若干数量的赌博机，所有的这些赌博机都有着各自不同的出奖几率，那么为了获取最高的期望收益，你应该以何种顺序和次数去尝试这些不同的赌博机呢？Kocsis et al [19] 利用了一个可以用于在多次尝试赌博机后获取最大收益的算法并将其利用于对弈树搜索中，以找到当前的最优策略。由此，一个名为 UCT（Upper Conficence Bounds for Trees）的算法被提出。算法首先会对当前所有可用的策略进行一次采样，然后在一个概率模型的指导下再在所有这些策略中选取某个策略

来再次采样。如此一来，算法便能很好地在穷尽某个策略（exploitation）和探索其他策略（exploration）之间获得良好的平衡，因为拥有高收益的策略将会被更多地采样，其他收益相对较低的策略也不会被完全抛弃，也会被不时地进行采样。该决策方法也在围棋上获得了巨大的成功 [20,21]。

与此同时，UCT 算法还被应用于普适游戏领域并获得了不错的成果 [22]。一款游戏人工智能的核心在于搜索和估价，其中搜索功能可用于预测游戏未来的走向而估价功能则可被运用于评估搜索功能发现的策略的收益。在普适游戏（general game playing）中，人工智能无法使用任何先验的与游戏有关的特定知识来对对弈状态进行估价，所有可用的知识只可以通过对弈的过程自行推断 [23]。由于蒙特卡洛搜索算法和 UCT 算法均无需依赖于任何启发式的估价函数，它们最终都能够在普适游戏领域获得巨大的成功 [24]。

在炉石传说中进行决策，最大的难点仍然在于隐藏的信息（对手的手牌不可见）以及随机性（对手和自己抽到的下一张牌不可知）。我们需要考虑到对手的手牌可能是炉石传说 743 种卡牌中的任何一张，所有暴力的枚举算法最终都必须应对无比巨大的分支系数（考虑到正是分支系数的巨大正是无法为围棋开发基于暴力搜索算法的人工智能的主要原因）。除此之外，在没有将对弈模拟至游戏结束的情况下，也难以对炉石传说的某个中间状态进行估价。由此，考虑到蒙特卡洛搜索在其他游戏，尤其是普适游戏领域获得的优秀表现，我们有理由相信蒙特卡洛搜索应用于炉石传说同样可以有不错的成果。

1.3 论文结构

本文余下部分结构如下：第 2 章将用于讲述本文将使用的蒙特卡洛搜索算法的基本概念；第 3 章讲述本文实验的基本方法，包括了对实验所使用的不同人工智能的描述以及将蒙特卡洛搜索算法应用至炉石传说的基本方法描述；第 4 章将给出实验的具体结果；第 5 章给出实验的结论以及未来的工作方向。

2 蒙特卡洛搜索

2.1 蒙特卡洛方法

蒙特卡洛方法（Monte Carlo methods）实际上为一类计算算法，这些算法依赖于重复的随机采样来求取数值结果。蒙特卡洛方法主要起源于统计物理领域，被用于求出某些不可解的积分方程的近似值，现多被用于求解那些无法通过数学方法求解的物理或数学问题，而其重复采样求取近似值的思想更是被延伸到了其他各种领域。

Abramson [25] 首次展示了蒙特卡洛方法的随机采样可以被用于近似某个对弈决策的理论收益。使用 Gelly 和 Silver 所提出的代数符号 [26]，那么某个对弈操作的期望收益可以表示为 Q 值如下：

$$Q(s,a) = \frac{1}{N(s,a)} \sum_{i=1}^{N(s)} \mathbb{I}_i(s,a) z_i \tag{1}$$

其中 $N(s,a)$ 为操作 a 在状态 s 下被选择的次数，$N(s)$ 为游戏从状态 s 被模拟的总次数，z_i 为从状态 s 开始的第 i 次模拟的结果；当操作 a 在第 i 次模拟被从状态 s 选中时，$\mathbb{I}_i(s,a)$ 为 1，否则为 0。

有一些蒙特卡洛方法在给定的对弈状态下会对所有的可以进行的操作进行均匀采样，这样的蒙特卡洛方法被称为平蒙特卡洛方法（flat Monte Carlo）。Ginsberg 曾利用这样的蒙特卡洛方法开发出了能与世界级选手相竞的桥牌人工智能 [27]，可见这样的蒙特卡洛方法已有一定的实力。这样的方法的不足之处也是显而易见的：它完全无法对对手的行为进行任何预测 [28]。然而，在模拟时根据以往的经验偏向某些操作是完全可以提高所得的操作期望收益的准确度的。根据已有的操作期望收益，我们完全可以选择偏向那些拥有较高收益的操作。

2.2 赌博机方法

赌博机问题（bandit problems）是概率论中的经典问题。问题给定 K 个操作（例如，K 个不同的赌博机），要求参与者在这些操作中进行选择，以使得在重复选取最优收益后获取最高的累计收益。每个操作所对应的收益分布是未知的，这使得参与者难以在这些操作中进行选择，而他只能通过曾经进行的尝试的结果来估算每个操作背后可能对应的收益。这就产生了经典的穷尽 -探索困境（exploitation-exploration dillemma）：参与者必须在“穷尽（exploitation）当前已知的最优操作”与“探索（exploration）其他现在不是最优但有可能在多次模拟后成为最优操作的操作”之间进行取舍。

一个 K 臂赌博机可被定义为随机变量 $X_{i,n}$，其中 $i \in [1,K]$ 代表第 i 个赌博机（赌博机的第 i 个“臂”）[19,29,30]，$X_{i,1}, X_{i,2}, \ldots$ 依次为尝试第 i 个赌博机时得到的结果，它们之间相互独立且一致地遵循着某个未知的分布法则，且有着未知的期望值 μ_i。通过定义一个可以根据赌博机过往给出的奖励决定该尝试哪个赌博机的策略（policy）可以解决 K

臂赌博机问题。

2.2.1 后悔程度

给定的策略应能使得玩家的后悔程度（regret）最小。在 n 次尝试后，玩家的后悔程度可定义如下：

$$R_N = \mu^* n - \mu_j \sum_{j=1}^{K} \mathbb{E}[T_j(n)] \tag{2}$$

其中 μ^* 为期望收益最高的赌博机的期望收益，$\mathbb{E}[T_j(n)]$ 代表在这 n 次尝试中尝试了第 j 个赌博机的期望次数。换句话说，玩家的后悔程度可以被理解为因没能尝试收益最高的赌博机所带来的损失期望。值得注意是，无论为任何时候，任何一个赌博机被选中的几率都不能为零，否则收益最高的赌博机可能会因为其他暂时拥有较高收益的赌博机而被搜索算法所忽略。为了保证这一点，我们应为每一个赌博机所观察到的收益值加上置信上界（upper confidence bound）。

2.2.2 置信上界

为了解决 K 臂赌博机问题，搜索算法有必要引入置信上界，因为在任何时候，任何一个赌博机都可能是最优的赌博机。由 Auer et al 提出的名为 UCB1 的策略 [29] 可以在未预先设置任何与收益分布有关的启发知识的情况下使玩家的后悔程度随尝试次数 n 呈对数级增长（$O(\ln n)$）。该策略在每次选取赌博机时都会按照给定的公式为每个赌博机计算其对应的值，并选取所对应的值最大的赌博机。该公式定义如下：

$$UCB1 = \bar{X}_j + \sqrt{\frac{2 \ln n}{n_j}} \tag{3}$$

其中 $\bar{X}_j$ 为第 j 个赌博机的平均收益，n_j 为第 j 个赌博机被尝试的次数，n 为尝试的总次数。不难看出，收益项 $\bar{X}_j$ 鼓励搜索算法穷尽（exploitation）目前平均收益最高的赌博机，而 $\sqrt{\frac{2 \ln n}{n_j}}$ 项则鼓励搜索算法探索（exploration）其他尝试次数较少的赌博机。

参考文献

[1] GameSpot, "Hearthstone reaches 40 million players." http://www.gamespot.com/articles/hearthstone-reaches-40-million-players-up-10-milli/1100-6432063/. November 6, 2015.

[2] B. Entertainment, "The hearthstone world championship is here!." http://us.battle.net/hearthstone/en/blog/19920718/the-hearthstone-world-championship-is-here-10-22-2015. October 22, 2015.

[3] GameSpot, "Dragon age: Inquisition wins goty at game awards." http://www.gamespot.com/articles/dragon-age-inquisition-wins-goty-at-game-awards/1100-6424005/. December 5, 2014.

[4] T. G. Awards, "Nominees | the game awards 2015." http://thegameawards.com/nominees/. November 12, 2015.

[5] Y. Björnsson and H. Finnsson, "Cadiaplayer: A simulation-based general game player," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 1, no. 1, pp. 4–15, 2009.

[6] M. Bowling, M. Johanson, N. Burch, and D. Szafron, "Strategy evaluation in extensive games with importance sampling," in *Proceedings of the 25th international conference on Machine learning*, pp. 72–79, ACM, 2008.

[7] J. Schaeffer, "A gamut of games," *AI Magazine*, vol. 22, no. 3, p. 29, 2001.

[8] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron, "Approximating game-theoretic optimal strategies for full-scale poker," in *IJCAI*, pp. 661–668, 2003.

[9] D. Billings, A. Davidson, J. Schaeffer, and D. Szafron, "The challenge of poker," *Artificial Intelligence*, vol. 134, no. 1, pp. 201–240, 2002.

[10] D. Billings, A. Davidson, T. Schauenberg, N. Burch, M. Bowling, R. Holte, J. Schaeffer, and D. Szafron, "Game-tree search with adaptation in stochastic imperfect-information games," in *Computers and Games*, pp. 21–34, Springer, 2004.

[11] R. J. Baker and P. I. Cowling, "Bayesian opponent modeling in a simple poker environment," in *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, pp. 125–131, IEEE, 2007.

[12] R. J. Baker, P. I. Cowling, T. W. Randall, and P. Jiang, "Can opponent models aid poker player evolution?," in *Computational Intelligence and Games, 2008. CIG'08. IEEE Symposium On*, pp. 23–30, Ieee, 2008.

[13] M. L. Ginsberg, “Partition search,” in *AAAI/IAAI, Vol. 1*, pp. 228–233, 1996.

[14] M. Campbell, A. J. Hoane, and F.-h. Hsu, “Deep blue,” *Artificial intelligence*, vol. 134, no. 1, pp. 57–83, 2002.

[15] B. Brügmann, “Monte carlo go,” tech. rep., Citeseer, 1993.

[16] G. Chaslot, J.-T. Saito, B. Bouzy, J. Uiterwijk, and H. J. Van Den Herik, “Monte-carlo strategies for computer go,” in *Proceedings of the 18th BeNeLux Conference on Artificial Intelligence, Namur, Belgium*, pp. 83–91, 2006.

[17] T. Cazenave and B. Helmstetter, “Combining tactical search and monte-carlo in the game of go.,” *CIG*, vol. 5, pp. 171–175, 2005.

[18] G. Chaslot, M. Winands, J. Uiterwijk, H. van den Herik, and B. Bouzy, “Progressive strategies for monte-carlo tree search,” in *Proceedings of the 10th Joint Conference on Information Sciences (JCIS 2007)*, pp. 655–661, 2007.

[19] L. Kocsis and C. Szepesvári, “Bandit based monte-carlo planning,” in *Machine Learning: ECML 2006*, pp. 282–293, Springer, 2006.

[20] Y. Wang and S. Gelly, “Modifications of uct and sequence-like simulations for monte-carlo go.,” *CIG*, vol. 7, pp. 175–182, 2007.

[21] 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,” *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 1, no. 1, pp. 73–89, 2009.

[22] H. Finnsson and Y. Björnsson, “Simulation-based approach to general game playing.,” in *AAAI*, vol. 8, pp. 259–264, 2008.

[23] J. Clune, “Heuristic evaluation functions for general game playing,” in *AAAI*, vol. 7, pp. 1134–1139, 2007.

[24] S. Sharma, Z. Kobti, and S. Goodwin, “Knowledge generation for improving simulations in uct for general game playing,” in *AI 2008: Advances in Artificial Intelligence*, pp. 49–55, Springer, 2008.

[25] B. Abramson, “Expected-outcome: A general model of static evaluation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 2, pp. 182–193, 1990.

[26] S. Gelly and D. Silver, “Monte-carlo tree search and rapid action value estimation in computer go,” *Artificial Intelligence*, vol. 175, no. 11, pp. 1856–1875, 2011.

[27] M. L. Ginsberg, “Gib: Imperfect information in a computationally challenging game,” *Journal of Artificial Intelligence Research*, pp. 303–358, 2001.

[28] C. Browne, “The dangers of random playouts,” *ICGA Journal*, vol. 34, no. 1, pp. 25–26, 2011.

[29] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[30] L. Kocsis, C. Szepesvári, and J. Willemson, “Improved monte-carlo search,” *Univ. Tartu, Estonia, Tech. Rep*, vol. 1, 2006.