DOI:10.16208/j.issn1000-7024.2012.03.075

UCT-RAVE 算法在多人非完备信息博弈中的应用

芮雄星，王一莉

（南京工业大学 电子与信息工程学院，江苏 南京 210009）

摘 要：针对传统博弈搜索算法无法适用于多人非完备信息博弈，通过分析 UCT-RAVE 算法的原理和特性，提出了运用 UCT-RAVE 算法与蒙特卡罗抽样技术相结合的方法。通过蒙特卡罗抽样技术将非完备信息提取为有一定可信度的完备信息，运用 UCT-RAVE 算法基于此完备信息进行搜索，结合多次蒙特卡罗抽样下的最佳收益，选择最适行动。实例结果表明了该方法的可行性和有效性。

关键词：博弈搜索；UCT-RAVE 算法；多人非完备信息博弈；蒙特卡罗抽样；牌类博弈

中图法分类号：TP181 **文献标识号**：A **文章编号**：1000-7024（2012）03-1136-04

Application of UCT-RAVE algorithm in multi-player games with imperfect information

RUI Xiong-xing, WANG Yi-li

(College of Electronic and Information Engineering, Nanjing University of Technology, Nanjing 210009, China)

Abstract: Aimed at the problems that traditional gaming search algorithms do not suit to multi-palyer games with imperfect information, a method of combining UCT-RAVE and Monte-Carlo sampling is proposed, after analyszing the principle and characteristic of UCT-RAVE algorithm. First, the imperfect information is replaced by simulating perfect information with Monte-Carlo sampling, then UCT-RAVE is used based on perfect information for searching, at last most suitable action is selected after considering the best profits of many Monte-Carlo samples. Simulation demonstrated the feasibility and the effectiveness of the method.

Key words: gaming search; UCT-RAVE algorithm; multi-player games; Monte-Carlo sampling; card gaming

0 引 言

完备信息博弈和非完备信息博弈是机器博弈的两个分支，对于完备信息博弈，国内外已经取得了的很多较好的研究成果。而非完备信息博弈领域的相关研究还不十分成熟，目前为止非完备信息下很成功的人工智能博弈程序还很少。传统的基于最小最大（minimax）搜索的算法很难适用于多人非完备信息博弈，由于每个博弈者可能使用不同的博弈策略，很难找到一个静态评价函数能够很好的应对每种博弈策略；而非完备信息的存在导致不确定博弈行为大量增加，博弈搜索空间将变得庞大[1]。而且，alpha-beta 剪枝在多人博弈中效率很低，虽然在二人完备信息博弈中 alpha-beta 剪枝能使空间复杂度从 $O(b^d)$ 降为 $O(b^{d/2})$，但在多人博弈中，最好的情况只能使空间复杂度降为 $O(b^{\frac{n-1}{n}d})$，其中 n 为玩家人数[2]。本文将介绍一种新的博弈搜索算法 UCT-RAVE[3-4]。并通过与蒙特卡罗抽样（Monte-Carlo sampling）技术[5]相结合，将其应用于多人非完备信息博弈中。通过简单的三人争上游牌类博弈实例，验证此方法的可行性和有效性；并与 UCT 算法比较，测试其性能。

1 UCT-RAVE 介绍

UCT-RAVE 是应用于树搜索的上限置信区间（upper confidence bound applied to tree search）方法和快速动作值估计（rapid action value estimation）方法的结合，是结合蒙特卡罗（Monte-Carlo）搜索方法和强化学习（reinforcement learning）方法为一体的一种博弈搜索算法，由 Sylvain 应用在计算机围棋上获得了巨大的成功[6]。

1.1 UCT 介绍

UCT（UCB applied to TRee）是利用 UCB（upper confidence bound）公式和蒙特卡罗模拟的结果来增量扩展搜索状态的一种算法。UCB 是为了解决 K 臂赌博机问题[7]而产生的，K 臂赌博机是一种假想的具有 K 只手柄的老虎机，可做的动作是选择并拉下其中的一只手柄，而由此所赢取

收稿日期：2011-03-09；**修订日期**：2011-05-10

作者简介：芮雄星（1983-），男，浙江浦江人，硕士研究生，研究方向为模式识别与智能计算；王一莉（1964-），女，江苏盐城人，副教授，研究方向为搜索引擎与智能计算。E-mail: ruixiongxing@163.com

的一定数量的钱就是和这个手柄（动作）相关联的收益（reward）。问题是如何根据当前已经掌握的每只手柄的收益情况决定下次拉下哪知手柄，一般来说，玩家会根据当前所积累的知识来做决定，这称之为开发（exploitation）。如果一味的选择已开发过的手柄，而不尝试其它的手柄，则可能会错过收益率更高的手柄，因此应当适度地尝试未开发过的手柄，这称之为探索（exploration）。UCB 试图解决开发与探索之间的矛盾，寻找开发与探索之间的平衡点[8]。

UCB 利用当前掌握的知识加上一个调整项来平衡开发与探索之间的矛盾。在每次做出选择时，UCB 根据每只手柄到目前为止的平均收益值，加上调整项的值，得出本次选择此手柄的 UCB 值，挑选拥有最大 UCB 值的手柄作为本次所要选择的手柄。UCB 公式表示如下

$$UCB_i = X_i + c\sqrt{\frac{\ln N}{T_i}}$$

式中：$UCBi$——第 i 只手柄经由 UCB 公式运算后所得到的值，X_i——第 i 只手柄到目前为止的平均收益值，T_i——第 i 只手柄被选择的次数，N——到目前为止所有手柄选择次数的总和。公式中前项即为此手柄的过去表现，即开发项，后项则是调整值，即探索项。其中调整项值会随这只手柄被选择次数的增加而减少，以便选择手柄时，不过分拘泥于旧有的表现，适当探索其它手柄。从而在开发和探索之间进行平衡。

UCT 把每一个节点都当作是一个 K 臂赌博机问题[9-10]，而此节点的每一个分支，都是 K 臂赌博机的一只手柄。选择分支，就会获得相对应的收益，对于博弈而言，UCB 公式的收益值就等于该状态下的胜率，而该胜率是按照蒙特卡罗抽样的概念，用模拟博弈的结果来决定。所谓的模拟博弈，就是给定一个局面（在此给的是叶节点所代表的局面），由计算机接手博弈，直到终局，然后判定并回传胜负结果。UCT 会据此结果，更新叶节点到根节点路径上所有节点的收益值。可以用状态动作对（s，a）的方式来表示 UCT 收益公式

$$Q_{UCT}^{\oplus}(s,a) = Q_{UCT}(s,a) + c\sqrt{\frac{\log n(s)}{n(s,a)}}$$

式中：n（s，a）——s 状态下 a 动作被选择的次数，n（s）——s 状态被访问的次数，而选择动作的策略 π_{UCT}（s）就是使平均收益最大化：$\pi_{UCT}(s) = \underset{a}{\operatorname{argmax}} Q_{UCT}(s,a)$。

1.2 RAVE 介绍

RAVE（rapid action value estimation）[11]是基于值（value-based）函数的强化学习思想在 UCT 方法中的应用。RAVE 收集并评价 UCT 搜索中产生的状态动作对，并在下一次 UCT 搜索时加以引导，使 UCT 能够更多的搜索更好的分支。

强化学习是一种无监督的机器学习方法，它被称之为“和批评者一起学习”。批评者（critic）并不反馈应该做什么，而仅仅反馈之前所做的怎么样[12]。最典型的强化学习算法是 Q 学习算法，可以看作是马尔可夫决策过程（Markov decision processes）的一种变化形式。

马尔可夫决策过程是强化学习的数学模型，它是由四元组组成：$<S, A, R, T>$，其中 S 是离散的状态集，A 是离散的动作集，R：$S\times A\rightarrow R$ 是奖励函数，T：$S\times A\rightarrow PD(S)$ 是状态转移函数，$PD(S)$ 是状态集 S 上的概率分布函数。

典型的基于折扣报酬的强化学习问题通常可以描述为给定$<S, A, R, T>$，寻找策略 π 使得期望折扣报酬总和最大

$$\pi(s) = \underset{\pi}{\operatorname{argmax}} V^{\pi}(s)$$

式中：$V^{\pi}(s)$——折算累积回报，上式可以改写为

$$\pi(s) = \underset{\pi}{\operatorname{argmax}}[r(s,a) + \gamma V^{\pi}(\delta(s,a))]$$

式中：r（s，a）——s 状态下执行 a 所得的报酬值，γ 是折扣因子。

定义 Q（s，a）为从状态 s 开始并使用 a 作为第一个动作时的最大折算累积回报，换言之，Q 的值为从状态 s 执行动作 a 的立即回报加上以后遵循最优策略的值

$$Q(s,a) = r(s,a) + \gamma V^{\pi}(\delta(s,a))$$

则

$$\pi(s) = \underset{\pi}{\operatorname{argmax}} Q(s,a)$$

动态规划理论保证至少存在一个策略 π^* 使得对任意 $s\in S$ 有

$$\pi^*(s) = \underset{\pi}{\operatorname{argmax}} Q(s,a)$$

值函数 Q（s，a）的估计有很多种算法，比如 $TD(\lambda)$[13]。如果环境模型是已知的或是可学习的，那么基于值函数的强化学习算法可用于基于样本的搜索。可从模型中抽样来获得模拟场景，通过模拟经验来更新值函数。

RAVE 是基于值函数 Q（s，a）的强化学习方法，通过基于样本的搜索树来动态更新值函数。为了与 UCT 相结合，RAVE 的收益公式定义为

$$Q_{RAVE}(s,a) = Q_{RAVE}(s,a) + c\sqrt{\frac{\log m(s)}{m(s,a)}}$$

式中：m（s，a）——s 状态下 a 动作被选择的次数，m（s）——s 状态被访问的次数。

1.3 UCT 和 RAVE 结合

UCT 需要对每个 $s\in S$ 状态下可供选择的动作进行抽样，以便比较各分支的收益情况并做出路径选择。如果动作空间巨大，可供选择的分支数就很多，要用足够多的模拟次数来区分分支的好坏[14-15]，而巨大的模拟次数将影响算法的性能。为减少模拟次数，UCT 中加入在线学习知识 RAVE，将 RAVE 值作为分支选择的另一参考，以提高分支选择的准确性。引入线性因子 β（s，a）把 Q_{UCT} 和 Q_{RAVE} 线性组合到一起

$$\beta(s,a) = \sqrt{\frac{k}{3n(s)+k}}$$

$$Q_{UR}(s,a)=\beta(s,a)Q_{RAVE}(s,a)+(1-\beta(s,a))Q_{UCT}(s,a)$$

式中：k——平等因子，控制了 Q_{UCT} 和 Q_{RAVE} 所占的比重。同样选择动作的策略 π_{UR}（s）就是使平均收益最大化

$$\pi_{UR}(s)=\arg\max_{a}Q_{UR}(s,a)$$

2 UCT-RAVE 在非完备信息博弈中的应用

在非完备信息博弈中，博弈的真实状态是不可知的，比如牌类游戏中对手的牌是未知的，进行博弈的玩家所掌握的信息是不对称的和不完备的，这使得非完备信息博弈的研究更具有挑战性。为了应用 UCT-RAVE 算法，必须把这部分未知信息确定下来，可以通过猜测或计算等多种方法，而应用最广泛的是蒙特卡罗抽样方法。

2.1 蒙特卡罗抽样算法

蒙特卡罗方法又称为计算机随机模拟方法，是一种基于随机数的计算方法。它通过随机抽样将非完备信息博弈问题转换为完备信息博弈问题，同时通过大规模的抽样次数来逼近真实的情况。该方法在一些非完备信息博弈游戏中，例如 Alberta 的桥牌程序，已经取得了较好的效果。

2.2 UCT-RAVE 与蒙特卡罗抽样相结合

UCT-RAVE 算法运行过程中的两个重要因素在于节点的动态扩展和节点值的回溯运算。在非完备信息条件下，这两点是无法实现的。因此 UCT-RAVE 算法必须与可以将非完备信息条件转换为完备信息条件的蒙特卡罗抽样算法相结合。

UCT-RAVE 与蒙特卡罗抽样算法的结合体现在搜索算法初始化过程中完备信息局面的生成。在 UCT-RAVE 算法进行一次搜索时，首先使用蒙特卡罗抽样算法对非完备信息局面进行抽样生成完备信息局面。然后 UCT-RAVE 算法依据这个完备信息局面进行一次搜索和节点的扩展。下次搜索将基于另一个蒙特卡罗抽样生成的完备信息局面，每次搜索所生成的节点都保存于同一棵搜索树中，树中的每一个节点的胜率将代表综合各种可能的局面下的平均表现。

图 1 为应用于非完备信息博弈的 UCT-RAVE 算法伪代码，与蒙特卡罗抽样技术的结合使得 UCT-RAVE 算法在非完备信息博弈树的搜索问题中可以有效的运行并发挥自己的优势。

3 实例分析

为了验证本文方法在多人非完备信息博弈中的效果，选择了一个简单的三人争上游牌类博弈模型，争上游又称拱猪、跑得快等，游戏主要流行于江浙一带，游戏规则决定了玩家需要尽快把自己手中的牌尽量多的打出去，先把手中的牌出完的玩家获得胜利。失败的玩家，根据手中所剩的牌的数量计算，剩余的牌越多扣的分数越多，如图 2 所示。

用不同的算法作为玩家出牌的策略进行游戏，比较不同算法的性能表现。为更好的做出比较，限制每次用两种

```
Create root node //根据当前局面建立根节点
While simulation<max simulation //未达最大次数
  node←Monte-Carlo sample //局面确定化
  While node has children //未达叶节点
    node←Max QUR child //根据 QUR 选择子节点
  End While
  While node not terminal //叶节点不是最终状态
    node←Monte-Carlo simulate //模拟博弈至结束
  End While
  While node not root node //更新路径上节点 QUR 值
    Update node QUR
    node←node's parent
  End while
End while
Select max QUR child //选择最大 QUR 值节点
```

图 1 应用于非完备信息博弈的 UCT-RAVE 算法伪代码

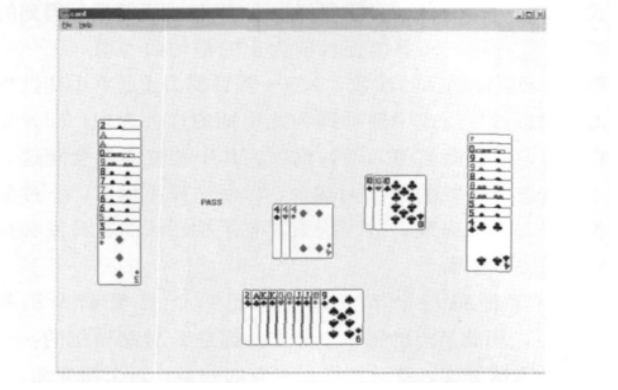

图 2 三人争上游牌类博弈画面

算法控制 3 个玩家进行博弈，即只有两种类型的玩家，用 Type A 和 Type B 表示，同时为了消除位置对算法胜率的影响，选择表 1 所示的 6 种不同的位置排列，并平均各种位置排列下算法的表现。

表 1 两种类型玩家的不同位置排列

Player 1	Player 2	Player 3
Type A	Type A	Type B
Type A	Type B	Type A
Type A	Type B	Type B
Type B	Type A	Type A
Type B	Type A	Type B
Type B	Type B	Type A

选取 UCT-RAVE 算法参数 C 值为 0.44，k 值为 100，模拟次数取 5000，分别与 UCT 算法、随机（Random）策略方法进行比较，每种位置排列进行 1000 次博弈，取平均计算胜率和失败时剩余的牌的数量，结果如表 2 所示。

表 2 UCT-RAVE 算法对战结果

UCT-RAVE	Random	UCT
Win rate	96.23%	74.49%
Lost cards	1.84	2.9

由表 2 可以看出，在上述参数设置情况下，本文算法对随机策略时胜率在 95%以上，说明了本文算法适用于多人非完备信息博弈模型，表现出一定的智能，和随机的胡乱出牌有质的区别。对 UCT 算法的胜率在 75%左右，说明了本文算法的智能水平。

为研究模拟次数对本文 UCT-RAVE 算法的智能影响，选取 10 至 10000 区间内的若干模拟次数，并与此模拟次数下的 UCT 算法进行博弈比较，结果如图 3 所示。

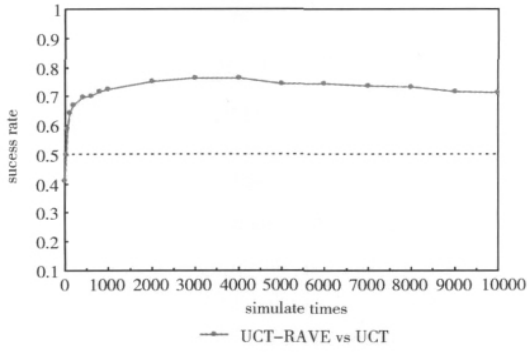

图 3 不同模拟次数下 UCT-RAVE 算法的性能

图 3 横坐标为两种算法所用的蒙特卡罗模拟次数，纵坐标为本文 UCT-RAVE 算法对 UCT 算法的胜率，图中曲线表示随着模拟次数变化 UCT-RAVE 对 UCT 算法的胜率的变化情况。

从图 3 中可以看出，本文 UCT-RAVE 算法对战 UCT 算法能够取得 70%以上的胜率，而从模拟次数的角度分析，本文 UCT-RAVE 算法可以在比较少的模拟次数下取得较好表现。在模拟次数极少（50 次以下）的情况下本文 UCT-RAVE 算法的胜率在 50%以下，原因在于过少的模拟次数不能较准确的积累强化学习的数据，使得到的 RAVE 值很不准确，从而干扰了 UCT 的选择。随着模拟次数的不断增加，胜率呈下降趋势，是因为大量的模拟次数足以获得准确的胜率值，对 RAVE 的依赖逐渐减少。

4 结束语

本文详细介绍了 UCT-RAVE 算法的原理和特性，提出了将其与蒙特卡罗抽样技术相结合应用于多人非完备信息博弈中，首先通过蒙特卡罗抽样技术将非完备信息转化为有一定可信度的完备信息，然后基于此完备信息运用 UCT-RAVE 算法进行搜索，最后综合多次蒙特卡罗抽样下的最佳平均收益，选择最适行动。通过简单的三人争上游牌类博弈实验证明此方法可行有效，并且同样模拟次数下能够获得比 UCT 算法更好的性能表现。但是如何选择本文 UCT-RAVE 算法的参数以获得更好的性能表现有待进一步研究。

参考文献：

[1] Sturtevant N R, Bowling M H. Robust game play against unknown opponents [C]. Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. USA: ACM, 2006: 713-719.

[2] Sturtevant N R. An analysis of UCT in multi-player games [C]. Proceedings of the 6th International Conference on Computers and Games. Berlin: Springer, 2008: 37-49.

[3] Chaslot G, Saito J T, Bouzy B, et al. Monte-Carlo strategies for computer go [C]. Proceedings of the 18th BeNeLux Conference on Artificial Intelligence. Park: AAAI Press, 2006: 83-91.

[4] Coulom R. Efficient selectivity and backup operators in Monte-Carlo tree search [C]. Proceedings of the 5th International Conference on Computers and Games. Berlin: Springer, 2006: 72-83.

[5] Krauth W. Algorithms and computations [M]. England: Oxford University Press, 2006: 264-275.

[6] Sylvian Gelly, WANG Yizao, Remi Munos, et al. Modification of UCT with patterns in Monte-Carlo go [R]. France: INRIA, 2006: 221-224.

[7] Kocsis L, Szepesvari C. Bandit based Monte-Carlo planning [C]. Proceedings of the 17th European Conference on Machine Learning. Berlin: Springer, 2006.

[8] Gelly S, WANG Yizao. Exploration exploitation in Go: UCT for Monte-Carlo go [C]. Twentieth Annual Conference on Neural Information Processing Systems. Canada: Citeseer, 2006: 225-236.

[9] WANG Y, Gelly S. Modifications of UCT and sequence-like simulations for Monte-Carlo go [C]. IEEE Symposium on Computational Intelligence and Games. USA: IEEE, 2007: 175-182.

[10] Winand M H M, Bjornsson S Y, Saito J T. monte-carlo tree search solver [C]. Proceedings of the 6th International Conference on Computers and Games. Berlin: Springer, 2008: 25-36.

[11] Gelly S, Silver D. Combining online and offline knowledge in UCT [C]. Proceedings of the 24th International Conference on Machine Learning. USA: ACM, 2007: 273-280.

[12] Silver D, Sutton R, Muller M. Reinforcement learning of local shape in the game of go [C]. Proceedings of the 20th International Joint Conference on Artifical Intelligence. India: Hyderabad, 2007: 1053-1058.

[13] Nathan Sturtevant, Adam White. Feature construction for reinforcement learning in hearts [C]. Proceedings of the 5th International Conference on Computers and Games. Berlin: Springer, 2006: 1305-1310.

[14] Buro M, LONG J R, Furtak T, et al. improving state evaluation, inference, and search in trick-based card games [C]. Proceedings of the 21st International Jont Conference on Artifical Intelligence. USA: ACM, 2009: 1407-1413.

[15] Arpad Rimmel, Fabien Teytaud, Olivier Teytaud. Biasing Monte-Carlo simulations through RAVE values [C]. The International Conference on Computers and Games. Berlin: Springer, 2010: 59-68.