

# 蒙特卡洛树搜索

维基百科，自由的百科全书

**蒙特卡洛树搜索**（英語：Monte Carlo tree search；简称：**MCTS**）是一种用于某些决策过程的启发式搜索算法，最引人注目的是在游戏中的使用。一个主要例子是**电脑围棋程序**<sup>[1]</sup>，它也用于其他**棋盘游戏**、即时电子游戏以及不确定性游戏。

## 目录

历史

原理

探索与利用

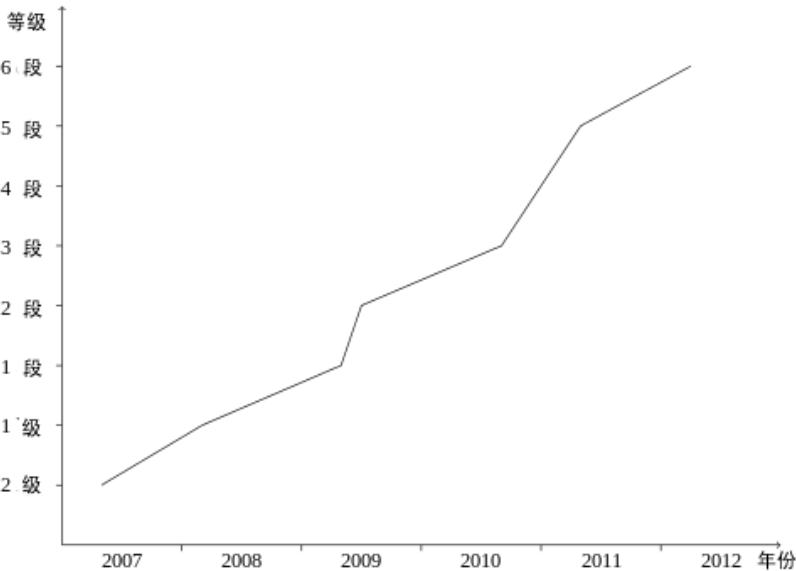
参见

参考来源

延伸阅读

## 历史

基于随机抽样的蒙特卡洛方法可以追溯到20世纪40年代。布鲁斯·艾布拉姆森（Bruce Abramson）在他1987年的博士论文中探索了这一想法，称它“展示出了准确、精密、易估、有效可计算以及域独立的特性“<sup>[2]</sup>。他深入试验了井字棋，然后试验了黑白棋和国际象棋的机器生成的评估函数。1992年，B·布鲁格曼（B. Brügmann）首次将其应用于对弈程序<sup>[3]</sup>，但他的想法未获得重视。2006年堪称围棋领域蒙特卡洛革命的一年<sup>[4]</sup>，雷米·库洛姆（Remi Coulom）描述了蒙特卡洛方法在游戏树搜索的应用并命名为蒙特卡洛树搜索<sup>[5]</sup>。列文特·科奇什（Levente Kocsis）和乔鲍·塞派什瓦里（Csaba Szepesvári）开发了UCT算法<sup>[6]</sup>，西尔万·热利（Sylvain Gelly）等人在他们的程序MoGo中实现了UCT<sup>[7]</sup>。2008年，MoGo在九路围棋中达到段位水平<sup>[8]</sup>，Fuego程序开始在九路围棋中战胜实力强劲的业余棋手<sup>[9]</sup>。2012年1月，Zen程序在19路围棋上以3：1击败二段棋手约翰·特朗普（John Tromp）<sup>[10]</sup>。



自2007年以来KGS围棋服务器上最优秀围棋程序的评级。自2006年以来，最优秀的程序全部使用蒙特卡洛树搜索。<sup>[11]</sup>

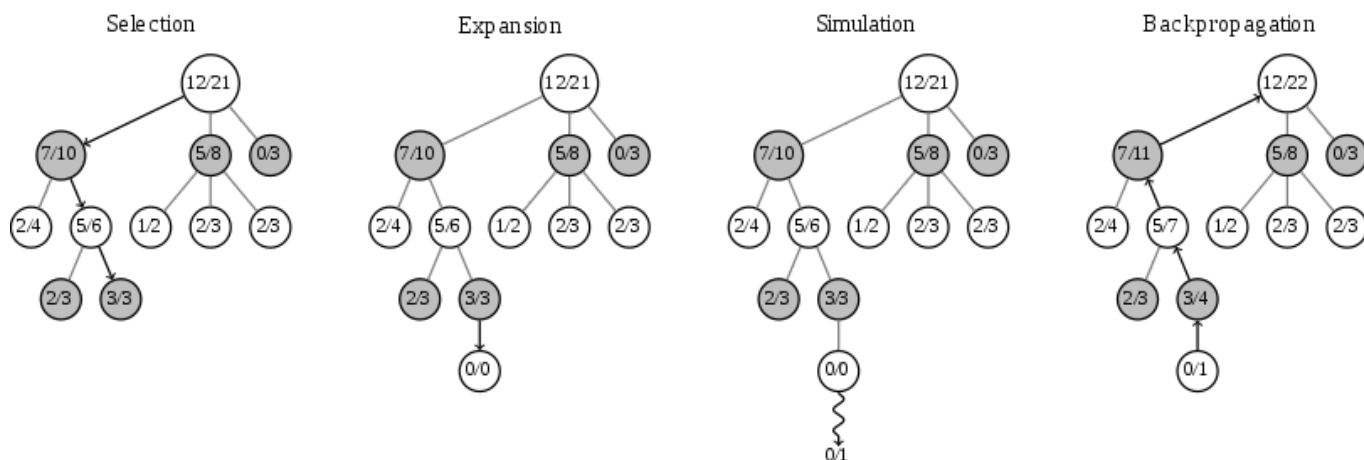
蒙特卡洛树搜索也被用于其他棋盘游戏程序，如六贯棋<sup>[12]</sup>、三宝棋<sup>[13]</sup>、亚马逊棋<sup>[14]</sup>和印度斗兽棋<sup>[15]</sup>；即时电子游戏，如《吃豆小姐》<sup>[16]</sup><sup>[17]</sup>、《神鬼寓言:传奇》<sup>[18]</sup>、《罗马II：全面战争》<sup>[19]</sup>；不确定性游戏，如斯卡特<sup>[20]</sup>、扑克<sup>[21]</sup>、万智牌<sup>[22]</sup>、卡坦岛<sup>[23]</sup>。

## 原理

蒙特卡洛树搜索的每个循环包括四个步骤：[24]

- 选择 (Selection)：从根节点R开始，选择连续的子节点向下至叶子节点L。后面给出了一种选择子节点的方法，让游戏树向最优的方向扩展，这是蒙特卡洛树搜索的精华所在。
- 扩展 (Expansion)：除非任意一方的输赢使得游戏在L结束，否则创建一个或多个子节点并选取其中一个节点C。
- 仿真 (Simulation)：在从节点C开始，用随机策略进行游戏，又称为playout或者rollout。
- 反向传播 (Backpropagation)：使用随机游戏的结果，更新从C到R的路径上的节点信息。

每一个节点的内容代表胜利次数/游戏次数



蒙特卡洛树搜索的步骤

## 探索与利用

选择子节点的主要困难是：在较高平均胜率的移动后，在对深层次变型的利用和对少数模拟移动的探索，这二者中保持某种平衡。第一个在游戏中平衡利用与探索的公式被称为UCT (Upper Confidence Bounds to Trees, 上限置信区间算法)，由匈牙利国家科学院计算机与自动化研究所高级研究员列文特·科奇什与阿尔伯塔大学全职教授乔鲍·塞派什瓦里提出<sup>[6]</sup>。UCT基于奥尔 (Auer)、西萨-比安奇 (Cesa-Bianchi) 和费舍尔 (Fischer) 提出的UCB1公式<sup>[25]</sup>，并首次由马库斯等人应用于多级决策模型（具体为马尔可夫决策过程）<sup>[26]</sup>。科奇什和塞派什瓦里建议选择

游戏树中的每个结点移动，从而使表达式  $\frac{w_i}{n_i} + c\sqrt{\frac{\ln t}{n_i}}$  具有最大值。在该式中：

- $w_i$  代表第  $i$  次移动后取胜的次数；
- $n_i$  代表第  $i$  次移动后仿真的次数；
- $c$  为探索参数—理论上等于  $\sqrt{2}$ ；在实际中通常可凭经验选择；
- $t$  代表仿真总次数，等于所有  $n_i$  的和。

大多数当代蒙特卡洛树搜索的实现都是基于UCT的一些变形。

## 参见

- AlphaGo，一个同时使用蒙特卡洛树搜索和深度学习的相当于人类的围棋程序。

## 参考来源

1. MCTS.ai: Everything Monte Carlo Tree Search. [2012-02-19].
2. Abramson, Bruce. The Expected-Outcome Model of Two-Player Games (PDF). Technical report, Department of Computer Science, Columbia University. 1987 [23 December 2013].
3. Brügmann, Bernd. Monte Carlo Go (PDF). Technical report, Department of Physics, Syracuse University. 1993.
4. Rémi Coulom. The Monte-Carlo Revolution in Go. Japanese-French Frontiers of Science Symposium (PDF). 2008.
5. Rémi Coulom. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. Computers and Games, 5th International Conference, CG 2006, Turin, Italy, May 29–31, 2006. Revised Papers. H. Jaap van den Herik, Paolo Ciancarini, H. H. L. M. Donkers (eds.). Springer. 2007: 72–83. ISBN 978-3-540-75537-1.
6. Kocsis, Levente; Szepesvári, Csaba. Bandit based Monte-Carlo Planning. Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, September 18–22, 2006, Proceedings. Lecture Notes in Computer Science 4212. Johannes Fürnkranz, Tobias Scheffer, Myra Spiliopoulou (eds.). Springer. 2006: 282–293. ISBN 3-540-45375-X.
7. Sylvain Gelly, Yizao Wang, Rémi Munos, Olivier Teytaud. Modification of UCT with Patterns in Monte-Carlo Go (PDF). Technical report, INRIA. November 2006.
8. Chang-Shing Lee, Mei-Hui Wang, Guillaume Chaslot, Jean-Baptiste Hoock, Arpad Rimmel, Olivier Teytaud, Shang-Rong Tsai, Shun-Chin Hsu, Tzung-Pei Hong. The Computational Intelligence of MoGo Revealed in Taiwan's Computer Go Tournaments (PDF). IEEE Transactions on Computational Intelligence and AI in Games. 2009, 1 (1): 73–89. doi:10.1109/tciaig.2009.2018703.

9. Markus Enzenberger, Martin Müller. Fuego – An Open-Source Framework for Board Games and Go Engine Based on Monte Carlo Tree Search (PDF). Technical report, University of Alberta. 2008. ([原始内容](#) (PDF)存档于2012-05-29) .
10. The Shodan Go Bet. [2012-05-02].
11. Sensei's Library: KGSBotRatings. [2012-05-03].
12. Broderick Arneson, Ryan Hayward, Philip Henderson. MoHex Wins Hex Tournament (PDF). ICGA Journal. June 2009, **32** (2): 114–116.
13. Timo Ewalds. Playing and Solving Havannah (PDF). Master's thesis, University of Alberta. 2011.
14. Richard J. Lorentz. Amazons Discover Monte-Carlo. Computers and Games, 6th International Conference, CG 2008, Beijing, China, September 29 – October 1, 2008. Proceedings. H. Jaap van den Herik, Xinhe Xu, Zongmin Ma, Mark H. M. Winands (eds.). Springer. 2008: 13–24. ISBN 978-3-540-87607-6.
15. Tomáš Kozelek. Methods of MCTS and the game Arimaa (PDF). Master's thesis, Charles University in Prague. 2009.
16. Xiaocong Gan, Yun Bao, Zhangang Han. Real-Time Search Method in Nondeterministic Game – Ms. Pac-Man. ICGA Journal. December 2011, **34** (4): 209–222.
17. Tom Pepels, Mark H. M. Winands, Marc Lanctot. Real-Time Monte Carlo Tree Search in Ms Pac-Man. IEEE Transactions on Computational Intelligence and AI in Games. September 2014, **6** (3): 245–257. doi:10.1109/tciaig.2013.2291577.
18. Tactical Planning and Real-time MCTS in Fable Legends. [2015-08-27].
19. Monte-Carlo Tree Search in TOTAL WAR: ROME II's Campaign AI. [2014-08-13].
20. Michael Buro, Jeffrey Richard Long, Timothy Furtak, Nathan R. Sturtevant. Improving State Evaluation, Inference, and Search in Trick-Based Card Games. IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11–17, 2009. Craig Boutilier (ed.). 2009: 1407–1413.
21. Jonathan Rubin, Ian Watson. Computer poker: A review (PDF). Artificial Intelligence. April 2011, **175** (5–6): 958–987. doi:10.1016/j.artint.2010.12.005. ([原始内容](#) (PDF)存档于2013-10-01) .
22. C.D. Ward, P.I. Cowling. Monte Carlo Search Applied to Card Selection in Magic: The Gathering. CIG'09 Proceedings of the 5th international conference on Computational Intelligence and Games (PDF). IEEE Press. 2009. ([原始内容](#) (PDF)存档于2016-05-28) .
23. István Szita, Guillaume Chaslot, Pieter Spronck. Monte-Carlo Tree Search in Settlers of Catan. Advances in Computer Games, 12th International Conference, ACG 2009, Pamplona, Spain, May 11–13, 2009. Revised Papers (PDF). H. Jaap van den Herik, Pieter Spronck (eds.). Springer. 2010: 21–32. ISBN 978-3-642-12992-6.
24. G.M.J.B. Chaslot, M.H.M. Winands, J.W.H.M. Uiterwijk, H.J. van den Herik, B. Bouzy. Progressive Strategies for Monte-Carlo Tree Search (PDF). New Mathematics and Natural Computation. 2008, **4** (3): 343–359. doi:10.1142/s1793005708001094.
25. Auer, Peter; Cesa-Bianchi, Nicolò; Fischer, Paul. Finite-time Analysis of the Multiarmed Bandit Problem (PDF). Machine Learning. 2002, **47**: 235–256. doi:10.1023/a:1013689704352.
26. Chang, Hyeong Soo; Fu, Michael C.; Hu, Jiaqiao; Marcus, Steven I. An Adaptive Sampling Algorithm for Solving Markov Decision Processes (PDF). Operations Research. 2005, **53**: 126–139.

## 延伸阅读

- Cameron Browne, Edward Powley, Daniel Whitehouse, Simon Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, Simon Colton. A Survey of Monte Carlo Tree Search Methods (蒙特卡洛树搜索方法综述) (PDF). IEEE Transactions on Computational Intelligence and AI in Games. March 2012, **4** (1). ([原始内容](#) (PDF)存档于2013-03-09) .

取自“<https://zh.wikipedia.org/w/index.php?title=蒙特卡洛树搜索&oldid=55501173>”

本页面最后修订于2019年8月3日 (星期六) 13:42。

本站的全部文字在知识共享 署名-相同方式共享 3.0协议之条款下提供，附加条款亦可能应用。（请参阅使用条款）

Wikipedia®和维基百科标志是维基媒体基金会的注册商标；维基™是维基媒体基金会的商标。

维基媒体基金会是按美国国内稅收法501(c)(3)登记的非营利慈善机构。