

华中科技大学

本科生毕业设计（论文）参考文献译文本

Science; Studies from University of Alberta Further Understanding
of General Science (Heads-up limit hold'em poker is solved)[J].

Science Letter,2015.

院 系 人工智能与自动化学院

专业班级 自卓 1601

姓 名 薛博阳

学 号 U201614481

指导教师 罗云峰

2020 年 2 月

译文要求

- 一、译文内容须与课题（或专业内容）联系，并需在封面注明详细出处。
- 二、出处格式为
图书：作者. 书名. 版本（第×版）. 译者. 出版地：出版者，出版年. 起页～止页
期刊：作者. 文章名称. 期刊名称，年号，卷号（期号）：起页～止页
- 三、译文不少于 5000 汉字（或 2 万印刷符）。
- 四、翻译内容用五号宋体字编辑，采用 A4 号纸双面打印，封面与封底采用浅蓝色封面纸（卡纸）打印。要求内容明确，语句通顺。
- 五、译文及其相应参考文献一起装订，顺序依次为封面、译文、文献。
- 六、翻译应在第七学期完成。

译文评阅

导师评语

应根据学校“译文要求”，对学生译文翻译的准确性、翻译数量以及译文的文字表述情况等做具体的评价后，再评分。

评分：_____ (百分制)

指导教师(签名)：_____

年 月 日

两人有限注德州扑克

1. 前言

扑克是一类非完备信息的游戏，其中玩家对场上现有的游戏状态并不完全了解。尽管许多完备信息游戏已被解决（例如，“四子棋”和“跳棋”），但是很长时间内还没有人类玩家参与的非确定性非完备信息游戏被解决。而在本文中，两人有限注德州扑克博弈问题已基本解决。此外，本文的方法证明了游戏庄家具有较大优势这一基本常识。这一结果由改进的虚拟遗憾最小化（CFR+）实现，该算法能够解决比以前更大数量级的扩展式博弈游戏。

在计算机早期发展中，游戏便与博弈论和人工智能紧密相连。在计算方面，巴贝奇（Babbage）制定了一个非常详细，能够玩井字棋的“自动机”的计划，该计划可以用于分析机器下棋。人工智能之父图灵设计了一个下棋的程序，信息论创始人香农分别为程序实现提供了硬件基础，这一程序的诞生验证了计算机科学与人工智能早期的学术观点。半个多世纪以来，机器博弈一直是新算法思想的试验平台，在这方面取得的成就是人工智能发展的重要里程碑。比如，1994 年计算机程序 Chinook 在跳棋上击败人类顶级玩家，成为第一个赢得人类冠军的机器算法；1997 年 IBM 开发的深蓝（Deep Blue）在国际象棋上击败 Kasparov，Watson 在 Jeopardy（一种博弈游戏）上击败 Jennings 和 Rutter。但是，击败顶尖的人类玩家并不等于“解决”游戏，而是计算出一种理论上可行的最优方案，该方案在所任何情况下都可以保持不败。彻底解决诸如“四子棋”和“跳棋”之类的游戏，实现了 AI 进步的里程碑。

迄今为止，人类解决的每一个游戏都是完备信息游戏。在完备信息游戏中，所有玩家在做出决策之前都会知晓当前场上的一切信息。象棋，西洋跳棋和西洋双陆棋等都是典型的完备信息游戏。在非完备信息游戏中，玩家并不总是知道场上已发生的全部信息（例如，在桥牌游戏中玩家并不知道其他玩家的手牌，再比如拍卖时卖方并不能预测拍卖品价值）。这些游戏更具挑战性，其理论知识，计算方法和已解决的实例均落后于完备信息游戏。尽管大

部分室内游戏都是完备信息，但完备信息在现实世界的决策环境中却很少见。在与博罗诺夫斯基叙述的一次谈话中，现代博弈论的创始人冯·诺伊曼（von Neumann）得出了相同的看法：“现实生活中充满了虚张声势的欺骗和小战术，我不可能知道别人下一步的行动，如果想用某种理论解释日常生活的内在规律，那博弈论再恰当不过了。”

冯·诺依曼（Von Neumann）的看法与扑克游戏这一典型非完备信息游戏的策略不谋而合。扑克游戏中每位玩家在游戏开始获得发给自己的手牌，然后轮流进行结构化下注，押注者可能拥有较大手牌（也可能是诈唬），然后跟对手下注或弃牌以放弃跟注。扑克在博弈论领域的早期发展中发挥了重要作用。鲍罗（Borel）和冯·诺依曼（Von Neumann）的基础著作是通过发展扑克游戏中诈骗手段的数学原理而来的，小型扑克游戏在许多早期论文中很普遍。扑克也是世界上最受欢迎的纸牌游戏，全球有超过 1.5 亿玩家。

今天最受欢迎的扑克游戏是德州扑克。如果只有两名玩家，同时固定下注大小和加注次数（即有限注）进行比赛时，称为两人有限注德州扑克（HULHE，下文皆使用简称）。HULHE 被《教授，银行家和自杀之王》中记载的一系列高赌注游戏所普及。它也是人类竞争性扑克游戏的最小形式。HULHE 具有 3.16×10^{17} 个游戏可以达到的可能状态，大于四子棋而小于跳棋。但是，由于 HULHE 是一种非完备信息游戏，因此在许多状态下玩家需要做出猜测，因为涉及到场上的未知信息（即对手的手牌）。于是，游戏具有 3.19×10^{14} 个需要玩家自己决策的决策点。

尽管 HULHE 的信息量小于跳棋，但它的非完备特性使它成为对于计算机而言更具挑战性的游戏。Chinook 对抗世界跳棋冠军 Tinsley 赢得第一场比赛过去 17 年后，电脑程序北极星赢得了与职业扑克玩家的第一场比赛。尽管 Schaeffer 等人在 2007 年利用计算机解决了跳棋问题，然而两人有限注德州扑克问题仍未解决。这种缓慢的进展并非因为缺乏这方面的探索研究。扑克一直是人工智能，运筹学和心理学三大领域的难题，其研究历程可以追溯到 40

年前。17 年前，Koller 和 Pfeffer 就宣称：“我们离解决大型扑克之类的游戏还差得很远，而且我们目前也不太可能做到这一点。”以 HULHE 为例“大型扑克”的诞生始于 10 多年前，自 2006 年起，世界年度扑克机器博弈大赛每年举办一次，并由人工智能国际协会（AAAI）举行，成立之初就成为数十个研究小组和业余爱好者的关注焦点。本文展示了为“全面解决”扑克游戏的持续研究成果。

Allis 对于游戏解决的程度给出了三个不同的定义。首先明确定义，我们称这种解决游戏的方式为“拆解”（solve）。对于二人（回合制、完全信息、不存在运气成分、有限）游戏，拆解分为以下几个层次：极弱拆解（Ultra-weakly Solved）：证明在初始局面下，双方的最大收益值是否可以确定存在 d ，只需证明存在性即。弱拆解（Weakly Solved）：给出一个算法，能从初始局面开始保证先行方（或者后行方）取得最大收益，无论其对手采取何种策略。强拆解（Strongly Solved）：给出一个算法，能从任何局面开始给出双方的最强应对，即当前状态的任意一方都能获得最大收益。在非完备信息的游戏，从一开始策略就不是唯一时，Allis 的“强拆解”方法就无法很好的实现。此外，由于玩家策略或游戏本身具有随机性，非完备信息游戏通常具有不同的实际收益值，而不是单一的结果（例如在国际象棋和跳棋中“胜局”，“败局”和“和局”）。最后，游戏的理论收益值通常是近似估计的，因此解决这类问题的一个关键因素是衡量解决方案的近似程度。弱拆解游戏近似估计的自然水平并不能得到精确的解，除非人类的在一局游戏中的时长足以建立一个基于统计学的方案。

我们在本文中已经取得了两人有限注德州扑克的弱拆解。此外，我们联系实际的收益值，证明了该游戏中庄家具有较大优势。

2. 解决非完备信息博弈游戏

扩展式博弈是典型的非完备信息博弈。这里的“博弈”泛指获益者和损失者之间的互动模型，在某些休闲游戏和工作场景十分常见，例如拍卖，谈判等。图 1 简单表述了 Kuhn 扑

克的起始步骤。扩展式博弈大都可以用树状图表示，它的分支表示游戏中可能存在的状态，即玩家采取的策略或执行后产生的结果。该树状图根据采取策略产生不同分支，每个分支都表示场上一名玩家采取的策略或当前游戏状态可能产生的结果。树状图的每个叶子节点表示特定的游戏过程直到结局，并且每个玩家都有各自的收益或亏损。与玩家相关的状态被划分为不同的信息集，若同一信息集包含多个游戏状态且这些游戏状态是无法分辨的，则是由于扩展式非完备信息博弈游戏中玩家无法获得确定的信息导致的[2]，扑克游戏中的信息集通常是玩家们无法区分的（例如发给对手的手牌）。信息集中各状态的分支是玩家的可采取行为，玩家为了最大化收益则针对每种信息状态建立模型，使得可采取策略符合某种概率分布。如果游戏中恰好有两个玩家，并且每个叶子节点处的玩家的收益值的和为零，则游戏称为零和游戏。

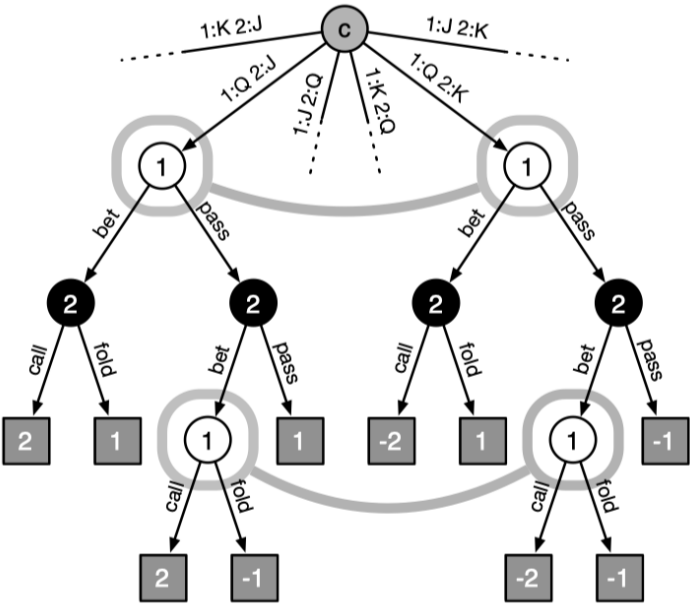


图 1：两人 Kuhn 扑克的树状图的一部分。玩家 1 被发牌 Q，而对手则被分配为牌 J 或牌 K。游戏状态指上图中，在每个信息集状态下，执行策略的玩家所标记的圆框（其中“c”表示信息集，并随机选择初始游戏状态）。箭头表示玩家可以选择的策略，并标注了游戏中的

含义。叶子节点是正方形，方框中是玩家 1 的最终游戏收益值（玩家 2 的收益值是玩家 1 取反）。用粗灰线连接的状态是同一信息集的一部分，也就意味着，玩家 1 无法区分这一对状态，因为它们各自代表发给对手的另一张未观察到的卡牌。玩家 2 的状态也在信息集中，其中包含了此树状图未显示的状态。

博弈游戏的经典解决方法是达到纳什均衡，在两人博弈中，无论对方的策略选择如何，另一方都会选择某个确定的策略，则该策略被称作支配性策略。如果博弈双方的策略组合分别构成各自的支配性策略，那么这个组合就被定义为纳什均衡。达到纳什均衡时，每个博弈者的均衡策略都是为了达到自己期望收益的最大值，同时，其他玩家也遵循这样的原则，任何玩家都无法通过单方面违背该策略来增加其预期收益。所有有限的扩展式博弈都至少有一个纳什均衡，在零和博弈游戏中，玩家在所有均衡状态下的期望收益值都相同。一个 ϵ -纳什均衡对每个玩家而言，是一种可以使得任何玩家都不能通过选择其他策略来增加自己的收益值的策略。按照 Allis 的分类，如果能计算出零和博弈的收益值，那么这种情况属于极弱拆解；而如果能计算出纳什均衡策略，则是弱拆解。如果一个游戏计算出的纳什均衡值极小，甚至一个人终其一生玩过的无数次后都从统计上无法区分纳什均衡解的话，我们称该解为究极弱拆解。对于完备信息博弈游戏，求解纳什均衡通常涉及对游戏决策树的（部分）遍历。但是，相同的处理方法不能用于非完备信息博弈。我们简述了解决非完备信息博弈的发展现状，并通过在解决复杂度越来越大的扑克游戏方面所取得的进步来衡量算法，如图 2 所示。

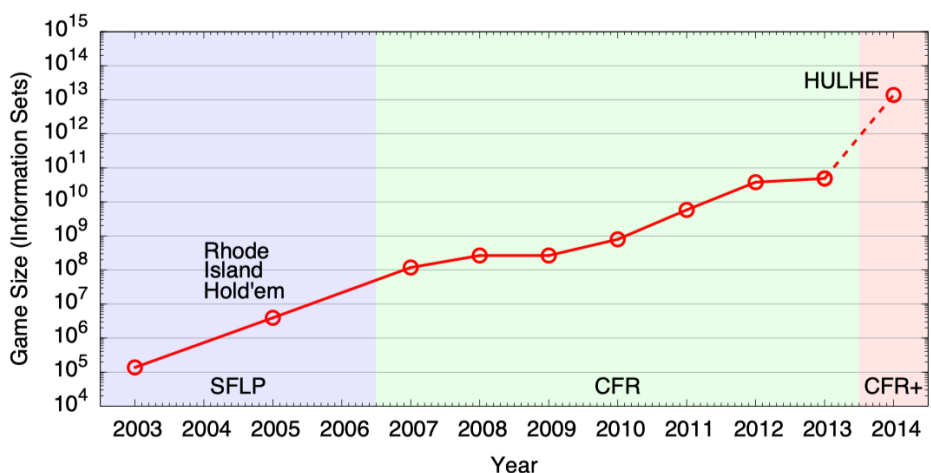


图 2：随着时间的推移，以独特信息集（即去除对称性后）衡量的非完备信息游戏的复杂度不断增加。不同颜色的区域代表所用的算法。虚线部分是本文获得的结果。

正则式线性规划：解决扩展式博弈游戏的最早方法是将其转换为正则博弈，将原始扩展式博弈中每对可能的策略值用矩阵表示，然后使用线性代数求解。然而，可确定的策略数量与游戏的信息集数量成指数关系。因此，尽管线性规划可以用数千种策略来处理正则式博弈，但即使只有几十个决策点也是不可行的。两人 Kuhn 扑克是一种有三张牌，一个下注回合和最大信息集数量为 12 的扑克游戏，这种情况可以用线性规划解决。但是即使是 Leduc Hold'em，有 6 张牌，2 轮下注回合以及 2 次最大注（总共有 288 个信息集），也具有超过 10^{86} 种确定性策略。

序列式线性规划：Romanovskii 和 Koller 等人建立了解决非完备信息游戏的现代理论，提出一种用序列形式表示策略的方法。通过简单更改变量，他们得出扩展式博弈可以用线性形式表示并直接解决，而无需将指数形式转换为一般形式。序列式线性规划是解决扩展式非完备信息博弈的第一个算法，其计算时间随着表示游戏策略的多项式的增加而增加。2003 年，Billings 等人将 SFLP 首次应用于扑克，解决了 HULHE 一系列简化问题，并构建了第一个扑克竞赛的程序。2005 年，Gilpin 和 Sandholm 将 SFLP 与一种寻找游戏对称

性的自动化方法一起解决了 Rhode Island Hold'em，这是一款合成的扑克游戏，在去除对称性后有 3.94×10^6 个信息集。

虚拟遗憾最小化：2006 年，第一届年度计算机扑克大赛举办后，比赛中算法在解决复杂度较大的游戏方面不断取得进步，提出了多种改进方式。虚拟遗憾最小化（CFR）是目前最具有竞争性应用最广泛的算法。CFR 是一种通过两种遗憾最小化算法之间反复训练的过程来逼近扩展式博弈纳什均衡的迭代方法。通过多次迭代，将整体遗憾分解到各个独立的信息集中计算局部最小后悔值，后悔值是算法没有选择唯一的最佳策略而损失的游戏收益，只有在执行策略后才知道。遗憾最小化算法可以使后悔值随着游戏进行呈亚线性增长，最终实现与采用最优策略相同的效果。CFR 的关键点在于不会存储和最小化呈指数增长的决策数量的后悔值，而是存储和最小化在每个信息集和相应后续决策中的不断修改调整的后悔值，这样就可得出任何决策下的后悔值。通过在每一次迭代中对每个玩家的策略取平均值，可以近似求出纳什均衡，并且随着迭代次数的增加，近似程度也会提高。该算法所需的内存与信息集的数量是线性关系，而不是二次关系，高效的 LP 方法就应该是这样。由于之前解决大型博弈游戏通常会受到可用内存的限制，CFR 的出现使得可解决的游戏复杂度较之前大幅提升，比如 Koller 等人近期取得的成就。自 2007 年被提出后，CFR 已用于解决更为复杂的 HULHE 简化问题，在 2012 年可以处理具有 3.8×10^{10} 个信息集的博弈游戏。

3. 解决两人有限注德州扑克

两人有限注德州扑克完整的博弈树包含 3.19×10^{14} 个信息集。即使在消除游戏对称性之后，仍具有 3.19×10^{14} 个信息集（比以前解决的游戏大三个数量级），处理如此规模的博弈树结合 CFR 算法本身的特点，在实现过程中就面临两个挑战：存储和计算。在计算

过程中，CFR 算法必须存储每个信息集的最终解决方案和累加的后悔值。即使使用单精度（4 字节）浮点数，也需要 262 TB 的存储空间。此外，以往的经验表明，增加三个数量级的信息集，需要增加至少三个数量级以上的计算。为了解决这两个挑战，我们使用了本文共同作者最近提出的两个想法。

为了处理 CFR 算法在存储空间上面临的挑战，我们使用压缩的方法存储平均策略和累积的后悔值，同时使用定点计算的方法，设定定标因数，将其与所有的值首次相乘，结果截断为整数。生成整数进行排序，以达到压缩效率最大化，后悔值的压缩比约为 13:1，策略值的压缩比约为 28:1。这样，计算过程中只需要 11 TB 的存储空间来存储后悔值，6 TB 的存储空间来存储平均策略值，分布遍及全部的计算节点。这样的规模存储在主存中是不切实际的，所以我们把值存储在每一个节点的本地磁盘中。每个节点负责一个子博弈集，是游戏决策树的一部分，子博弈树是根据公共可见的动作和自己的手牌以及底牌划分的，这样每个信息集都与一个子博弈相关。子博弈的后悔值和策略值使用流压缩技术从磁盘载入，更新，再存储回磁盘的，子博弈的部分需要的话，使用流媒体压缩技术进行解压缩和再压缩。通过使子博弈树变得足够大，更新时间主导子博弈过程的总时间。由于磁盘预缓存，会使效率下降的 5%。

为了解决计算方面的困难，来自 UACPRG 团队的 Michael Bowling 等人发明了一个叫做 CFR+ 的算法，这种算法是 CFR 的一个变体。CFR 算法每次迭代执行时仅仅抽样部分子博弈树进行更新，对每个信息集执行后悔匹配 regret matching 算法，为每个策略维持后悔值，选择后悔值为正数的策略，且动作的选择概率与后悔值成正比。相比之下，CFR+ 的迭代次数遍及整个博弈树，同时使用了后悔匹配 regret matching 算法的变形，即限定后悔值为非负数。后悔值为暂时负数的策略在证明有效后会被立即选择，而不是等多次迭代后其后悔值变成正数的时候才被选择。与 CFR 算法不同的是，策略值计算接近于 0 的期间，可

以凭借经验观察到玩家当前的策略利用情况，所以我们可以跳过计算和存储平均策略值得步骤，而是使用玩家当前的策略作为 CFR+ 的解决方案。根据以往的经验可以知道，即使计算平均策略值，这样计算方式也比蒙特卡洛 CFR 算法少了很多次计算，在计算过程中逐渐逼近纳什均衡，同时也非常适合大型博弈游戏的并行化处理。

与 CFR 算法一样，CFR+ 也是迭代算法，在计算过程中逐次逼近纳什均衡。近似程度通过其可用性衡量：与最坏情况下对手所采取的策略时相比，收益值小了多少。计算策略可用性需要计算最坏情况的收益值，通常需要遍历整个博弈树，因此长期以来，这对于 HULHE 这样复杂度很高的游戏来说都是不现实的。近期研究表明，可通过利用博弈游戏的非完备信息结构和收益规则来加速计算。这就是我们衡量纳什均衡策略值的近似程度的方法，并已在小型博弈游戏中进行了验证，而且针对 HULHE 中简单策略的可用性进行了独立计算。

我们通常根据预期值选择策略，但由于游戏具有随机性，即使全部考虑最坏情况，仍然不能保证在几次后可以获胜。对于一个游戏，如果适当的选择策略，从统计学上可以达到 95% 的置信度，我们就称其基本解决。假设一个人每天玩 12 小时，每小时玩 200 次扑克游戏，每天如此玩 70 年，然后该玩家考虑最坏情况下，对手所采取的收益最大化的策略并且自己零失误。玩家的总收入，作为数百万个单局游戏收益的总和，将服从正态分布。因此，在此扑克玩家的职业生涯中，收益总会有 1.64 的标准差或在 20 次游戏中至少有一次低于期望值（即策略值）。据报道，在 HULHE 中游戏收益的一般偏差约为 5 bb/g （每场比赛的大盲注，其中大盲注是 HULHE 的赌注单位），我们得出的阈值为

$(1.64 \times 5) / \sqrt{(200 \times 12 \times 365 \times 70)} \approx 0.00105$ 。因此，期望值在 1 M bb/g （每场游戏百万个大盲注）以下时的精确方案有很高的置信度，并且确实有 $1/20$ 的机会可以在最坏情况下胜过对手。于是， 1 M bb/g 这一阈值宣布 HULHE 已基本解决。

4. 解决方案

我们的 CFR + 算法是在 200 个计算节点的群集上实现的，每个计算节点具有 24 个 2.1 GHz AMD 内核，32GB RAM 和 1TB 本地磁盘。我们将游戏分为 110,565 个子游戏（根据底注和翻牌时的押注划分）。子游戏又分为 199 个工作节点，工作节点中的父节点负责博弈树的起始部分。工作节点采用并行方式更新，计算子节点的效用值然后回溯递归到父节点执行更新，平均花费 61 分钟才能完成一次迭代。该计算执行了 1579 次迭代，耗时 68.5 天，总共使用了 900 个核心计算量和 10.9 TB 的磁盘空间，其中包括大量文件的文件系统开销。

图 3 显示了随着计算量的增加，策略值的变化。该策略值可达到 $0.986 \text{ M } bb/g$ ，意味着 HULHE 基本取得弱拆解。根据每个位置（庄家和非庄家）各自的策略值，我们可以得到游戏实际收益的界限：庄家的价值值在 87.7 和 $89.7 \text{ M } bb/g$ 之间，符合庄家在两人有限注德州扑克游戏中具备优势这一常识。

最终的策略会非常接近纳什均衡，也可以解决有关 HULHE 比赛中一些基本且长期存在的理论问题。图 4 展示了两个早期决策中的最终策略。人类玩家不同意用双手“limp”

（即呼唤是第一个动作而不是举手）。一般认为，该行为会迷惑对手使其放弃弃牌的机会，然后选择跟注。我们的解决方案证实了这一观点（请参见图 4a 中不存在蓝色）。该策略只拖延 0.06 % 的时间，而手牌下注时间不会超过 0.5 %。在其他情况下，该策略展示了超出传统观点的一面，表明了人类可能会改善这一情况。该策略几乎不会在作为庄家的第一回合就梭哈（即赌上全部），而一些玩命的人类玩家则会以各种方式梭哈。即使握住最强的手牌（一对 A），该策略也会在 0.01 % 的时间进程内限制下注，最有可能加盖的手牌是一对 2，概率为 0.06 %。也许更重要的是，该策略会选择（而不是弃牌）作为非庄家时比大多数人类玩家都会选择更激进的策略（参见图 4b 中相对较少的红色）。持有低排名的一对牌（例如三四）时，也有可能加注。

尽管以上这些观察仅是博弈论上最佳结果的一个示例（不同的纳什均衡可能产生不同效果），但它们已得到证实并且与人类当前对均衡博弈的认知相违背，说明人类可以从这种大规模博弈中学到很多关于游戏理论推理的知识。

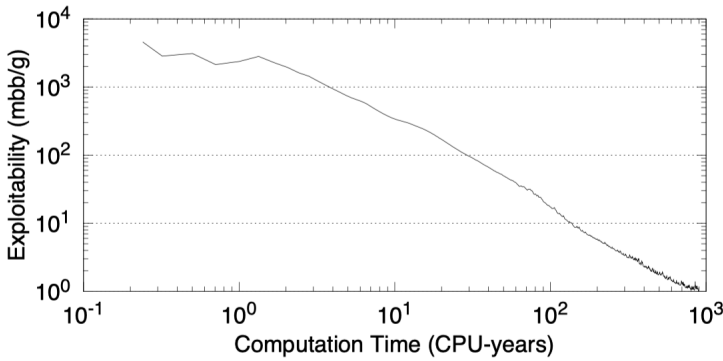


图 3: 随着复杂度的增加，近似的策略值。以每百万个大盲注点 (Mbb/g) 衡量收益，是每次 CFR + 迭代后采取当前策略收益值的衡量标准。经过 1579 次迭代或 900 核心计算后，其策略值达到 $0.986 Mbb/g$ 。

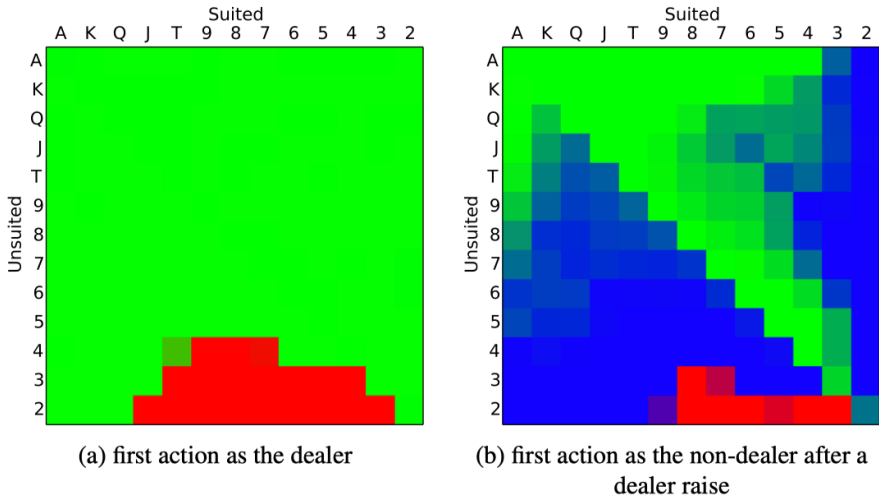


图 4: 两个早期决策方案的策略概率。(A) 发牌者在游戏中第一个动作的概率。(B) 非庄家在庄家 limp 时的第一项行动的概率。每个单元代表 169 张牌中的一对手牌（即玩家两张手牌），右上对角线由具有相同花色的牌组成，左下对角线由不同花色的牌组成。单元格的顏色表示所采取的操作：红色表示弃牌，蓝色表示跟注，绿色表示加注，顏色的混合表示随机决策。

5. 总结

解决扑克游戏最重要的意义是什么？是为了实现算法上的突破，这些突破可以使任何大规模博弈游戏的推理都变得更加容易。而且，尽管大部分时候博弈论看似只运用在棋牌、博弈和游戏方面，但在一些重要场合仍扮演着重要的角色[例如，它对冷战时期的政治格局的影响]。最近，涉及安全和隐私的的博弈论应用程序需求量越来越大，包括部署用于机场检查站，空中指挥调度中心和海上警卫队巡逻的系统。基于上述应用场景的 CFR 算法在没有意外干扰时已能够独立进行可靠的决策，并有可能进一步用于医疗卫生的决策。现实生活中的决策环境总会伴随着信息的不确定和丢失，因此算法的进步是投入到未来应用程序的关键。但是，我们也对图灵为自己的游戏作品的辩护作出回应：“我们不应该掩饰这样一事实：完成一项工作的主要动机完全是寻找乐趣”。

参考文献

- [1] Science; Studies from University of Alberta Further Understanding of General Science (Heads-up limit hold'em poker is solved) [J]. Science Letter, 2015.
- [2] 代佳宁. 基于虚拟遗憾最小化算法的非完备信息机器博弈研究[D]. 哈尔滨工业大学, 2017.
- [3] 滕雯娟. 基于虚拟遗憾最小化算法的德州扑克机器博弈研究[D]. 哈尔滨工业大学, 2015.
- [4] 胡裕靖, 高阳, 安波. 不完美信息扩展式博弈中在线虚拟遗憾最小化[J]. 计算机研究与发展, 2014, 51(10): 2160-2170.

参考文献原文

Heads-up Limit Hold'em Poker is Solved

Michael Bowling,^{1*} Neil Burch,¹ Michael Johanson,¹ Oskari Tammelin²

¹Department of Computing Science, University of Alberta,
Edmonton, Alberta, T6G2E8, Canada

²Unaffiliated, <http://jeskola.net>

*To whom correspondence should be addressed; E-mail: bowling@cs.ualberta.ca

Poker is a family of games that exhibit imperfect information, where players do not have full knowledge of past events. Whereas many perfect information games have been solved (e.g., Connect Four and checkers), no nontrivial imperfect information game played competitively by humans has previously been solved. Here, we announce that heads-up limit Texas hold'em is now essentially weakly solved. Furthermore, this computation formally proves the common wisdom that the dealer in the game holds a substantial advantage. This result was enabled by a new algorithm, CFR⁺, which is capable of solving extensive-form games orders of magnitude larger than previously possible.

Games have been intertwined with the earliest developments in computation, game theory, and artificial intelligence (AI). At the very conception of computing, Babbage had detailed plans for an “automaton” capable of playing tic-tac-toe and dreamt of his Analytical Engine playing chess (1). Both Turing (2) and Shannon (3) — on paper and in hardware, respectively — developed programs to play chess as a means of validating early ideas in computation and AI. For more than a half century, games have continued to act as testbeds for new ideas and the resulting successes have marked important milestones in the progress of AI. Examples include the checkers-playing computer program Chinook becoming the first to win a world championship title against humans (4), Deep Blue defeating Kasparov in chess (5), and Watson defeating Jennings and Rutter on Jeopardy! (6). However, defeating top human players is not the same as “solving” a game — that is, computing a game-theoretically optimal solution that is incapable of losing against any opponent in a fair game. Notable milestones in the advancement of AI have been achieved through solving games such as Connect Four (7) and checkers (8).

Every nontrivial game played competitively by humans that has been solved to date is a perfect-information game (9). In perfect-information games, all players are informed of everything that has occurred in the game before making a decision. Chess, checkers, and backgammon are examples of perfect-information games. In imperfect-information games, players do

not always have full knowledge of past events (e.g., cards dealt to other players in bridge and poker, or a seller’s knowledge of the value of an item in an auction). These games are more challenging, with theory, computational algorithms, and instances of solved games lagging behind results in the perfect-information setting (10). And although perfect-information may be a common property of parlor games, it is far less common in real-world decision-making settings. In a conversation recounted by Bronowski, von Neumann, the founder of modern game theory, made the same observation: “Real life is not like that. Real life consists of bluffing, of little tactics of deception, of asking yourself what is the other man going to think I mean to do. And that is what games are about in my theory” (11).

Von Neumann’s statement hints at the quintessential game of imperfect-information: the game of poker. Poker involves each player being dealt private cards, with players taking structured turns making bets on having the strongest hand (possibly bluffing), calling opponents’ bets, or folding to give up the hand. Poker played an important role in early developments in the field of game theory. Borel’s (12) and von Neumann’s (13, 14) foundational works were motivated by developing a mathematical rationale for bluffing in poker, and small synthetic poker games (15) were commonplace in many early papers (12, 14, 16, 17). Poker is also arguably the most popular card game in the world, with more than 150 million players worldwide (18).

The most popular variant of poker today is Texas hold’em. When it is played with just two players (heads-up) and with fixed bet sizes and a fixed number of raises (limit), it is called heads-up limit hold’em or HULHE (19). HULHE was popularized by a series of high-stakes games chronicled in the book *The Professor, the Banker, and the Suicide King* (20). It is also the smallest variant of poker played competitively by humans. HULHE has 3.16×10^{17} possible states the game can reach, making it larger than Connect Four and smaller than checkers. However, because HULHE is an imperfect-information game, many of these states cannot be distinguished by the acting player, as they involve information about unseen past events (i.e., private cards dealt to the opponent). As a result, the game has 3.19×10^{14} decision points where a player is required to make a decision.

Although smaller than checkers, the imperfect-information nature of HULHE makes it a far more challenging game for computers to play or solve. It was 17 years after Chinook won its first game against world champion Tinsley in checkers that the computer program Polaris won the first meaningful match against professional poker players (21). Whereas Schaeffer et al. solved checkers in 2007 (8), heads-up limit Texas hold’em poker had remained unsolved. This slow progress is not for lack of effort. Poker has been a challenge problem for artificial intelligence, operations research, and psychology, with work going back more than 40 years (22). 17 years ago, Koller and Pfeffer (23) declared, “We are nowhere close to being able to solve huge games such as full-scale poker, and it is unlikely that we will ever be able to do so.” The focus on HULHE as one example of “full-scale poker” began in earnest more than 10 years ago (24) and became the focus of dozens of research groups and hobbyists after 2006, when it became the inaugural event in the Annual Computer Poker Competition (25), held in conjunction with the main conference of the Association for the Advancement of Artificial Intelligence (AAAI). This paper is the culmination of this sustained research effort toward solving a “full-scale”

poker game (19).

Allis (26) gave three different definitions for solving a game. A game is said to be ultra-weakly solved if, for the initial position(s), the game-theoretic value has been determined; weakly solved if, for the initial position(s), a strategy has been determined to obtain at least the game-theoretic value, for both players, under reasonable resources; and strongly solved if, for all legal positions, a strategy has been determined to obtain the game-theoretic value of the position, for both players, under reasonable resources. In an imperfect-information game, where the game-theoretic value of a position beyond the initial position is not unique, Allis's notion of "strongly solved" is not well defined. Furthermore, imperfect-information games, because of stochasticity in the players' strategies or the game itself, typically have game-theoretic values that are real-valued rather than discretely valued (such as "win," "loss," and "draw" in chess and checkers) and are achieved in expectation over many playings of the game. As a result, game-theoretic values are often approximated, and so an additional consideration in solving a game is the degree of approximation in a solution. A natural level of approximation under which a game is essentially weakly solved is if a human lifetime of play is not sufficient to establish with statistical significance that the strategy is not an exact solution.

In this paper, we announce that heads-up limit Texas hold'em poker is essentially weakly solved. Furthermore, we bound the game-theoretic value of the game, proving that the game is a winning game for the dealer.

Solving Imperfect-Information Games

The classical representation for an imperfect-information setting is the extensive-form game. Here the word "game" refers to a formal model of interaction between self-interested agents and applies to both recreational games and serious endeavours such as auctions, negotiation, and security. See Figure 1 for a graphical depiction of a portion of a simple poker game in extensive-form. The core of an extensive-form game is a game tree specifying branches of possible events, namely player actions or chance outcomes. The branches of the tree split at game states and each is associated with one of the players (or chance) who is responsible for determining the result of that event. The leaves of the tree signify the end of the game, and have an associated utility for each player. The states associated with a player are partitioned into information sets, which are sets of states among which the acting player cannot distinguish (e.g., corresponding to states where the opponent was dealt different private cards). The branches from states within an information set are the player's available actions. A strategy for a player specifies for each information set a probability distribution over the available actions. If the game has exactly two players and the utilities at every leaf sum to zero, the game is called zero-sum.

The classical solution concept for games is a Nash equilibrium, a strategy for each player such that no player can increase his or her expected utility by unilaterally choosing a different strategy. All finite extensive-form games have at least one Nash equilibrium. In zero-sum games, all equilibria have the same expected utilities for the players, and this value is called

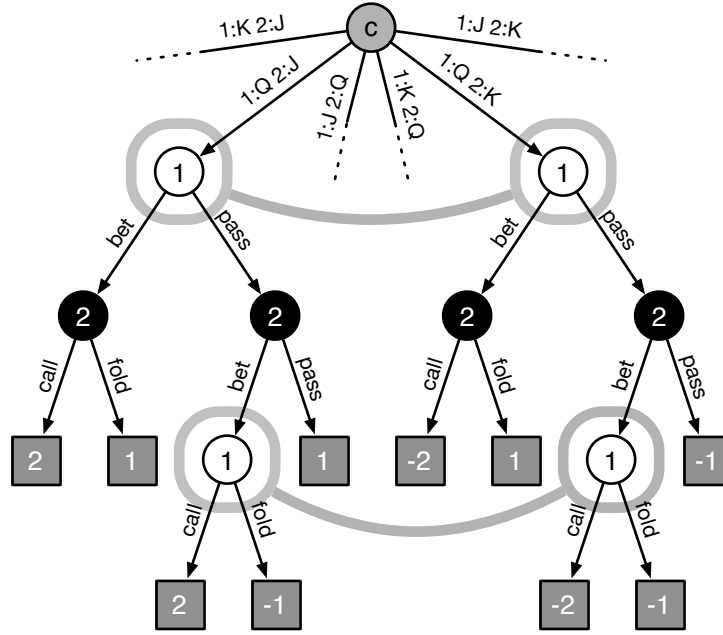


Figure 1: **Portion of the extensive-form game representation of three-card Kuhn poker (16).** Player 1 is dealt a queen (Q) and the opponent is given either the jack (J) or king (K). Game states are circles labeled by the player acting at each state (“c” refers to chance, which randomly chooses the initial deal). The arrows show the events the acting player can choose from, labeled with their in-game meaning. The leaves are square vertices labeled with the associated utility for player 1 (player 2’s utility is the negation of player 1’s). The states connected by thick gray lines are part of the same information set; that is, player 1 cannot distinguish between the states in each pair because they each represent a different unobserved card being dealt to the opponent. Player 2’s states are also in information sets, containing other states not pictured in this diagram.

the game-theoretic value of the game. An ϵ -Nash equilibrium is a strategy for each player where no player can increase his or her utility by more than ϵ by choosing a different strategy. By Allis’s categories, a zero-sum game is ultra-weakly solved if its game-theoretic value is computed, and weakly solved if a Nash equilibrium strategy is computed. We call a game essentially weakly solved if an ϵ -Nash equilibrium is computed for a sufficiently small ϵ to be statistically indistinguishable from zero in a human lifetime of played games. For perfect-information games, solving typically involves a (partial) traversal of the game tree. However, the same techniques cannot apply to imperfect-information settings. We briefly review the advances in solving imperfect-information games, benchmarking the algorithms by their progress in solving increasingly larger synthetic poker games as summarized in Figure 2.

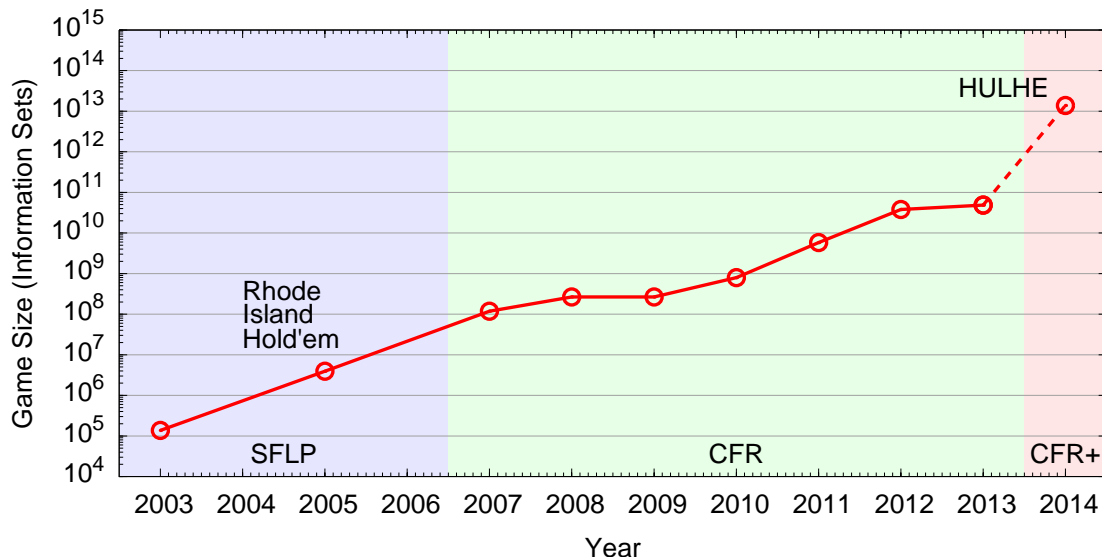


Figure 2: **Increasing sizes of imperfect-information games solved over time measured in unique information sets (i.e., after symmetries are removed).** The shaded regions refer to the technique used to achieve the result; the dashed line shows the result established in this paper.

Normal-Form Linear Programming The earliest method for solving an extensive-form game involved converting it into a normal-form game, represented as a matrix of values for every pair of possible deterministic strategies in the original extensive-form game, and then solving it with a linear program (LP). Unfortunately, the number of possible deterministic strategies is exponential in the number of information sets of the game. So, although LPs can handle normal-form games with many thousands of strategies, even just a few dozen decision points makes this method impractical. Kuhn poker, a poker game with three cards, one betting round, and a one-bet maximum having a total of 12 information sets (see Figure 1), can be solved with this approach. But even Leduc hold'em (27), with six cards, two betting rounds, and a two-bet maximum having a total of 288 information sets, is intractable having more than 10^{86} possible deterministic strategies.

Sequence-Form Linear Programming Romanovskii (28) and later Koller et al. (29, 30) established the modern era of solving imperfect-information games, introducing the sequence-form representation of a strategy. With this simple change of variables, they showed that the extensive-form game could be solved directly as an LP, without the need for an exponential conversion to normal-form. A sequence-form linear program (SFLP) was the first algorithm to solve imperfect-information extensive-form games with computation time that grows as a polynomial of the size of the game representation. In 2003, Billings et al. (24) applied this technique to poker, solving a set of simplifications of HULHE to build the first competitive poker-playing program. In 2005, Gilpin and Sandholm (31) used the approach along with an automated tech-

nique for finding game symmetries to solve Rhode Island hold'em (32), a synthetic poker game with 3.94×10^6 information sets after symmetries are removed.

Counterfactual Regret Minimization In 2006, the Annual Computer Poker Competition was started (25). The competition drove advancements in solving larger and larger games, with multiple techniques and refinements being proposed in the years that followed (33, 34). One of the techniques to emerge, and currently the most widely adopted in the competition, is counterfactual regret minimization (CFR) (35). CFR is an iterative method for approximating a Nash equilibrium of an extensive-form game through the process of repeated self-play between two regret-minimizing algorithms (19, 36). Regret is the loss in utility an algorithm suffers for not having selected the single best deterministic strategy, which can only be known in hindsight. A regret-minimizing algorithm is one that guarantees that its regret grows sub-linearly over time, and so eventually achieves the same utility as the best deterministic strategy. The key insight of CFR is that instead of storing and minimizing regret for the exponential number of deterministic strategies, CFR stores and minimizes a modified regret for each information set and subsequent action, which can be used to form an upper bound on the regret for any deterministic strategy. An approximate Nash equilibrium is retrieved by averaging each player's strategies over all of the iterations, and the approximation improves as the number of iterations increases. The memory needed for the algorithm is linear in the number of information sets, rather than quadratic, which is the case for efficient LP methods (37). Because solving large games is usually bounded by available memory, CFR has resulted in an increase in the size of solved games similar to that of Koller et al.'s advance. Since its introduction in 2007, CFR has been used to solve increasingly complex simplifications of HULHE, reaching as many as 3.8×10^{10} information sets in 2012 (38).

Solving Heads-Up Limit Hold'em

The full game of HULHE has 3.19×10^{14} information sets. Even after removing game symmetries, it has 1.38×10^{13} information sets (i.e., three orders of magnitude larger than previously solved games). There are two challenges for established CFR variants to handle games at this scale: memory and computation. During computation, CFR must store the resulting solution and the accumulated regret values for each information set. Even with single-precision (4-byte) floating-point numbers, this requires 262 TB of storage. Furthermore, past experience has shown that an increase in the number of information sets by three orders of magnitude requires at least three orders of magnitude more computation. To tackle these two challenges we use two ideas recently proposed by a co-author of this paper (39).

To address the memory challenge we store the average strategy and accumulated regrets using compression. We use fixed-point arithmetic by first multiplying all values by a scaling factor and truncating them to integers. The resulting integers are then ordered to maximize compression efficiency, with compression ratios around 13-to-1 on the regrets and 28-to-1 on

the strategy. Overall, we require less than 11 TB of storage to store the regrets and 6 TB to store the average strategy during the computation, which is distributed across a cluster of computation nodes. This amount is infeasible to store in main memory, and so we store the values on each node’s local disk. Each node is responsible for a set of subgames; that is portions of the game tree are partitioned on the basis of publicly observed actions and cards such that each information set is associated with one subgame. The regrets and strategy for a subgame are loaded from disk, updated, and saved back to disk, using a streaming compression technique that decompresses and recompresses portions of the subgame as needed. By making the subgames large enough, the update time dominates the total time to process a subgame. With disk pre-caching, the inefficiency incurred by disk storage is approximately 5% of the total time.

To address the computation challenge we use a variant of CFR called CFR^+ (19, 39). CFR implementations typically sample only portions of the game tree to update on each iteration. They also use regret matching at each information set, which maintains regrets for each action and chooses among actions with positive regret with probability proportional to that regret. By contrast, CFR^+ does exhaustive iterations over the entire game tree and uses a variant of regret matching (regret matching⁺) where regrets are constrained to be non-negative. Actions that have appeared poor (with less than zero regret for not having been played) will be chosen again immediately after proving useful (rather than waiting many iterations for the regret to become positive). Finally, unlike with CFR, we have empirically observed that the exploitability of the players’ current strategies during the computation regularly approaches zero. Therefore, we can skip the step of computing and storing the average strategy, instead using the players’ current strategies as the CFR^+ solution. We have empirically observed CFR^+ to require considerably less computation, even when computing the average strategy, than state-of-the-art sampling CFR (40), while also being highly suitable for massive parallelization.

Like CFR, CFR^+ is an iterative algorithm that computes successive approximations to a Nash equilibrium solution. The quality of the approximation can be measured by its exploitability: the amount less than the game value that the strategy achieves against the worst-case opponent strategy in expectation (19). Computing the exploitability of a strategy involves computing this worst-case value, which traditionally requires a traversal of the entire game tree. This was long thought to be intractable for games the size of HULHE. Recently it was shown that this calculation could be accelerated by exploiting the imperfect-information structure of the game and regularities in the utilities (41). This is the technique we use to confirm the approximation quality of our resulting strategy. The technique and implementation has been verified on small games and against independent calculations of the exploitability of simple strategies in HULHE.

A strategy can be exploitable in expectation and yet, because of chance elements in the game and randomization in the strategy, its worst-case opponent still is not guaranteed to be winning after any finite number of hands. We define a game to be essentially solved if a lifetime of play is unable to statistically differentiate it from being solved at 95% confidence. Imagine someone playing 200 games of poker an hour for 12 hours a day without missing a day for 70 years. Furthermore imagine that player using the worst-case, maximally exploitive, op-

ponent strategy and never making a mistake. The player’s total winnings, as a sum of many millions of independent outcomes, would be normally distributed. Hence, the observed winnings in this lifetime of poker would be 1.64 standard deviations or more below its expected value (i.e., the strategy’s exploitability) at least 1 time out of 20. Using the standard deviation of a single game of HULHE, which has been reported to be around 5 bb/g (big-blinds per game, where the big-blind is the unit of stakes in HULHE) (42), we arrive at a threshold of $(1.64 * 5) / \sqrt{200 * 12 * 365 * 70} \approx 0.00105$. Therefore, an approximate solution with an exploitability less than 1mbb/g (milli-big-blinds per game) cannot be distinguished with high confidence from an exact solution, and indeed has a 1-in-20 chance of winning against its worst-case adversary even after a human lifetime of games. Hence, 1mbb/g is the threshold for declaring HULHE essentially solved.

The Solution

Our CFR⁺ implementation was executed on a cluster of 200 computation nodes each with 24 2.1-GHz AMD cores, 32GB of RAM, and a 1-TB local disk. We divided the game into 110,565 subgames (partitioned according to preflop betting, flop cards, and flop betting). The subgames were split among 199 worker nodes, with one parent node responsible for the initial portion of the game tree. The worker nodes performed their updates in parallel, passing values back to the parent node for it to perform its update, taking 61 minutes on average to complete one iteration. The computation was then run for 1579 iterations, taking 68.5 days, and using a total of 900 core-years of computation (43) and 10.9 TB of disk space, including filesystem overhead from the large number of files.

Figure 3 shows the exploitability of the computed strategy with increasing computation. The strategy reaches an exploitability of 0.986 mbb/g, making HULHE essentially weakly solved. Using the separate exploitability values for each position (as the dealer and non-dealer) we get exact bounds on the game-theoretic value of the game: between 87.7 and 89.7 mbb/g for the dealer, proving the common wisdom that the dealer holds a substantial advantage in HULHE.

The final strategy, as a close approximation to a Nash equilibrium, can also answer some fundamental and long-debated questions about game-theoretically optimal play in HULHE. Figure 4 gives a glimpse of the final strategy in two early decisions of the game. Human players have disagreed about whether it may be desirable to “limp” (i.e., call as the very first action rather than raise) with certain hands. Conventional wisdom is that limping forgoes the opportunity to provoke an immediate fold by the opponent, and so raising is preferred. Our solution emphatically agrees (see the absence of blue in Figure 4a). The strategy limps just 0.06% of the time and with no hand more than 0.5%. In other situations, the strategy gives insights beyond conventional wisdom, indicating areas where humans might improve. The strategy almost never “caps” (i.e., makes the final allowed raise) in the first round as the dealer, whereas some strong human players cap the betting with a wide range of hands. Even when holding the strongest hand — a pair of aces — the strategy caps the betting less than 0.01% of

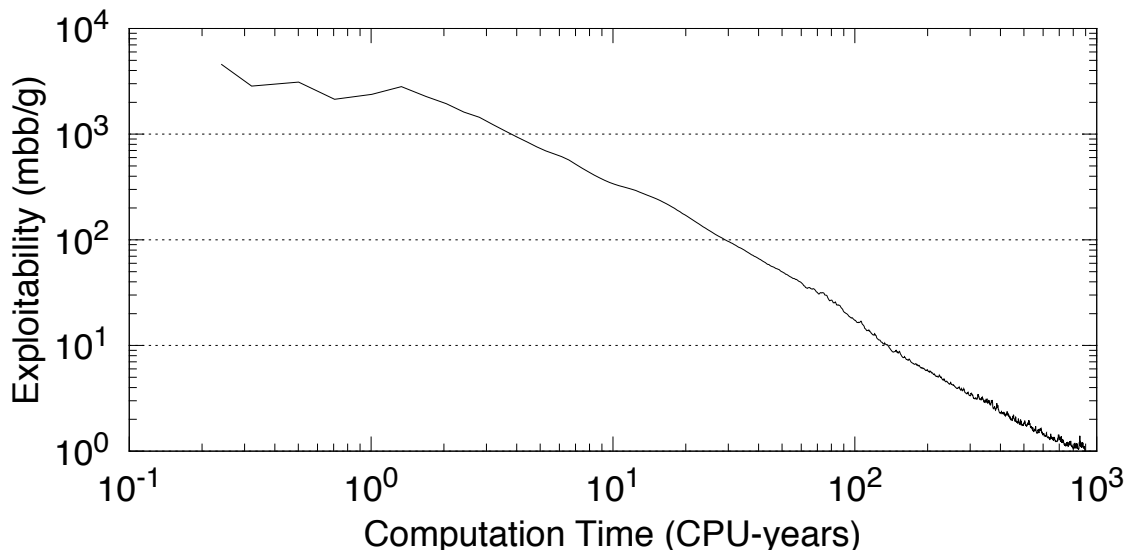


Figure 3: **Exploitability of the approximate solution with increasing computation.** The exploitability, measured in milli-big-blinds per game (mbb/g), is that of the current strategy measured after each iteration of CFR^+ . After 1579 iterations or 900 core-years of computation, it reaches an exploitability of 0.986 mbb/g.

the time, and the hand most likely to cap is a pair of twos, with probability 0.06%. Perhaps more important, the strategy chooses to play (i.e., not fold) a broader range of hands as the non-dealer than most human players (see the relatively small amount of red in Figure 4b). It is also much more likely to re-raise when holding a low-rank pair (such as threes or fours) (44).

Although these observations are for only one example of game-theoretically optimal play (different Nash equilibria may play differently), they confirm as well as contradict current human beliefs about equilibria play, and illustrate that humans can learn considerably from such large-scale game-theoretic reasoning.

Conclusion

What is the ultimate importance of solving poker? The breakthroughs behind our result are general algorithmic advances that make game-theoretic reasoning in large-scale models of any sort more tractable. And, although seemingly playful, game theory has always been envisioned to have serious implications [e.g., its early impact on Cold War politics (45)]. More recently, there has been a surge in game-theoretic applications involving security, including systems being deployed for airport checkpoints, air marshal scheduling, and coast guard patrolling (46). CFR algorithms based on those described above, have been used for robust decision-making in settings where there is no apparent adversary, with potential application to medical decision

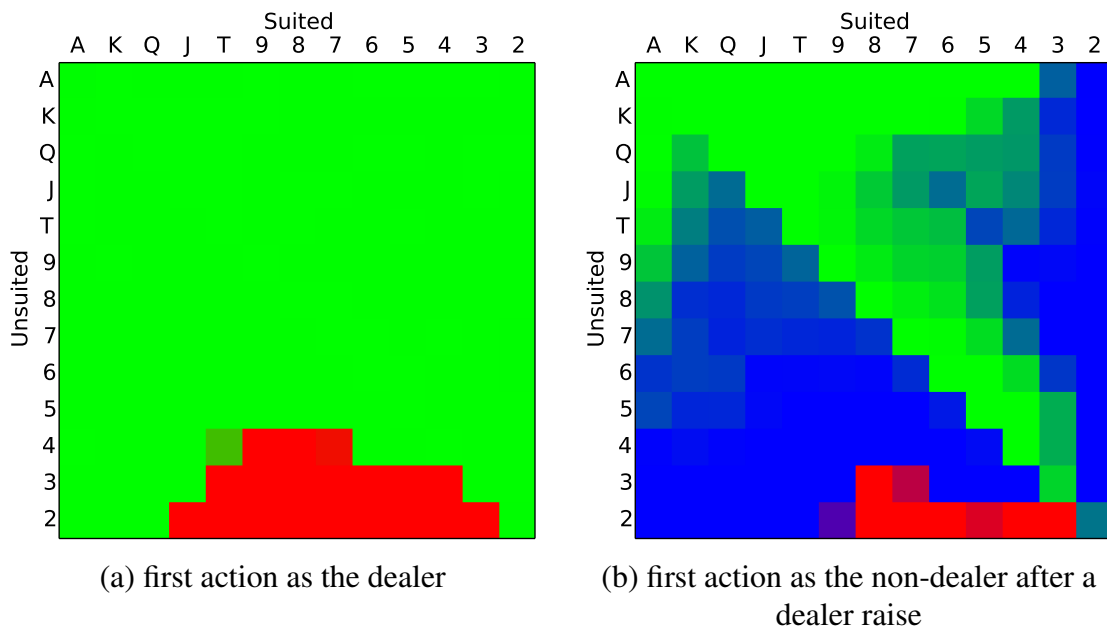


Figure 4: **Action probabilities in the solution strategy for two early decisions.** (A) The action probabilities for the dealers first action of the game. (B) The action probabilities for the nondealers first action in the event that the dealer raises. Each cell represents one of the possible 169 hands (i.e., two private cards), with the upper right diagonal consisting of cards with the same suit and the lower left diagonal consisting of cards of different suits. The color of the cell represents the action taken: red for fold, blue for call, and green for raise, with mixtures of colors representing a stochastic decision.

support (47). With real-life decision-making settings almost always involving uncertainty and missing information, algorithmic advances such as those needed to solve poker the key to future applications. However, we also echo a response attributed to Turing in defense of his own work in games: “It would be disingenuous of us to disguise the fact that the principal motive which prompted the work was the sheer fun of the thing” (48).

References and Notes

1. C. Babbage, *Passages from the Life of a Philosopher* (Longman, Green, Longman, Roberts, and Green, London, 1864). Chapter 34.
2. A. Turing, *Faster Than Thought*, B. V. Bowden, ed. (Pitman, London, 1976), chap. 25.
3. C. E. Shannon, *Philosophical Magazine, Series 7* **41**, 256 (1950).
4. J. Schaeffer, R. Lake, P. Lu, M. Bryant, *AI Magazine* **17**, 21 (1996).

5. M. Campbell, A. J. Hoane Jr., F. Hsu, *Artificial Intelligence* **134**, 57 (2002).
6. D. Ferrucci, *IBM Journal of Research and Development* **56**, 1:1 (2012).
7. V. Allis, A knowledge-based approach to connect-four. the game is solved: White wins, Master's thesis, Vrije Universiteit (1988).
8. J. Schaeffer, *et al.*, *Science* **317**, 1518 (2007).
9. We use the word trivial to describe a game that can be solved without the use of a machine. The one near-exception to this claim is oshi-zumo, but it is not played competitively by humans and is a simultaneous-move game that otherwise has perfect information (49). Furthermore, almost all nontrivial games played by humans that have been solved to-date also have no chance elements. The one notable exception is hypergammon, a three-checker variant of backgammon invented by Sconyers in 1993, which he then strongly solved (i.e., the game-theoretic value is known for all board positions). It has seen play in human competitions (see <http://www.bkgm.com/variants/HyperBackgammon.html>).
10. For example, Zermelo proved the solvability of finite, two-player, zero-sum, perfect information games in 1913 (50), whereas von Neumann's more general minimax theorem appeared in 1928 (13). Minimax and alpha-beta pruning, the fundamental computational algorithm for perfect-information games, were developed in the 1950s; the first polynomial-time technique for imperfect-information games was introduced in the 1960s but was not well known until the 1990s (29).
11. J. Bronowski, *The ascent of man*, Documentary (1973). Episode 13.
12. É. Borel, J. Ville, *Applications de la théorie des probabilités aux jeux de hasard* (Gauthier-Villars, Paris, 1938).
13. J. von Neumann, *Mathematische Annalen* **100**, 295 (1928).
14. J. von Neumann, O. Morgenstern, *Theory of Games and Economic Behavior* (Princeton University Press, Princeton, Second Edition, 1947).
15. We use the word synthetic to describe a game that was invented for the purpose of being studied or solved rather than played by humans. A synthetic game may be trivial, such as Kuhn poker (16), or nontrivial such as Rhode Island hold'em (32).
16. H. Kuhn, *Contributions to the Theory of Games I*, H. Kuhn, A. Tucker, eds., *Annals of mathematics studies* (Princeton University Press, 1950), pp. 97–103.
17. J. F. Nash, L. S. Shapley, *Contributions to the Theory of Games I*, H. Kuhn, A. Tucker, eds. (Princeton University Press, 1950), pp. 105–116.

18. *The Economist* **December 22**, 31 (2007).
19. See Supplementary Materials.
20. M. Craig, *The Professor, the Banker, and the Suicide King: Inside the Richest Poker Game of All Time* (Grand Central Publishing, New York City, 2006).
21. J. Rehmeyer, N. Fox, R. Rico, *Wired* **16.12**, 186 (2008).
22. D. Billings, A. Davidson, J. Schaeffer, D. Szafron, *Artificial Intelligence* **134**, 201 (2002).
23. D. Koller, A. Pfeffer, *Artificial Intelligence* **94**, 167 (1997).
24. D. Billings, *et al.*, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence* (2003), pp. 661–668.
25. M. Zinkevich, M. Littman, *Journal of the International Computer Games Association* **29**, 166 (2006). News item.
26. V. L. Allis, Searching for solutions in games and artificial intelligence, Ph.D. thesis, University of Limburg (1994).
27. F. Southey, *et al.*, *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence* (2005), pp. 550–558.
28. I. V. Romanovskii, *Soviet Mathematics* **3**, 678 (1962).
29. D. Koller, N. Megiddo, *Games and Economic Behavior* **4**, 528 (1992).
30. D. Koller, N. Megiddo, B. von Stengel, *Games and Economic Behavior* **14**, 247 (1996).
31. A. Gilpin, T. Sandholm, *Journal of the ACM* **54** (2007).
32. J. Shi, M. L. Littman, *Revised Papers from the Second International Conference on Computers and Games* (2000), pp. 333–345.
33. T. Sandholm, *AI Magazine* **31**, 13 (2010).
34. J. Rubin, I. Watson, *Artificial Intelligence* **175**, 958 (2011).
35. Another notable algorithm to emerge from the Annual Computer Poker Competition is an application of Nesterov’s excessive gap technique (51) to solving extensive-form games (52). The technique has some desirable properties, including better asymptotic time complexity than what is known for CFR. However, it has not seen widespread use among competition participants because of its lack of flexibility in incorporating sampling schemes and its inability to be used with powerful (but unsound) abstractions that make use of imperfect recall. Recently, Waugh and Bagnell (53) have shown that CFR and the excessive

gap technique are more alike than different, which suggests that the individual advantages of each approach may be attainable in the other.

36. M. Zinkevich, M. Johanson, M. Bowling, C. Piccione, *Advances in Neural Information Processing Systems 20* (2008), pp. 905–912.
37. N. Karmarkar, *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing* (ACM, 1984), pp. 302–311.
38. E. Jackson, *Proceedings of the 2012 Computer Poker Symposium* (2012). Jackson reports a higher number of information sets, which counts terminal information sets rather than only those where a player is to act.
39. O. Tammelin, *CoRR* **abs/1407.5042** (2014).
40. M. Johanson, N. Bard, M. Lanctot, R. Gibson, M. Bowling, *Proceedings of the Eleventh International Conference on Autonomous Agents and Multi-Agent Systems* (2012), pp. 837–846.
41. M. Johanson, K. Waugh, M. Bowling, M. Zinkevich, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence* (2011), pp. 258–265.
42. M. Bowling, M. Johanson, N. Burch, D. Szafron, *Proceedings of the Twenty-Fifth International Conference on Machine Learning* (2008), pp. 72–79.
43. The total time and number of core-years is larger than was strictly necessary as it includes computation of an average strategy that was later measured to be more exploitable than the current strategy and so was discarded. The total space noted, on the other hand, is without storing the average strategy.
44. These insights were the result of discussions with Bryce Paradis, previously a professional poker player who specialized in HULHE.
45. O. Morgenstern, *New York Times Magazine* pp. 21–22 (1961).
46. M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned* (Cambridge University Press, 2011).
47. K. Chen, M. Bowling, *Advances in Neural Information Processing Systems 25* (2012), pp. 2078–2086.
48. P. Mirowski, *Toward a History of Game Theory*, E. R. Weintraub, ed. (Duke University Press, Durham, North Carolina, 1992), pp. 113–147. Mirowski cites Turing as author of the paragraph containing this remark. The paragraph appeared in (2), in a chapter with Turing listed as one of three contributors. Which parts of the chapter are the work of which

contributor, particularly the introductory material containing this quote, is not made explicit.

- 49. M. Buro, *Advances in Computer Games* **135**, 361 (2004).
- 50. E. Zermelo, *Proceedings of the Fifth International Congress of Mathematicians* (Cambridge University Press, Cambridge, 1913), pp. 501–504.
- 51. Y. Nesterov, *SIAM Journal on Optimization* **16**, 233 (2005).
- 52. A. Gilpin, S. Hoda, J. Peña, T. Sandholm, *Proceedings of the Third International Workshop On Internet And Network Economics* (2007), pp. 57–69.
- 53. K. Waugh, J. A. Bagnell, *AAAI Workshop on Computer Poker and Imperfect Information* (2015).

Acknowledgments

The author order is alphabetical reflecting equal contribution by the authors. The idea of CFR⁺ and compressing the regrets and strategy originated with O. T. (39). This research was supported by Natural Sciences and Engineering Research Council of Canada and Alberta Innovates Technology Futures through the Alberta Innovates Centre for Machine Learning, and was only made possible by the computing resources of Compute Canada and Calcul Québec. We thank all of the current and past members of the University of Alberta Computer Poker Research Group, where the idea to solve heads-up limit Texas hold'em was first discussed; J. Schaeffer, R. Holte, D. Szafron, and A. Brown for comments on early drafts of this article; and B. Paradis for insights into the conventional wisdom of top human poker players.

Supplementary Materials

Source code used to compute the solution strategy

Supplementary Text

Fig. S1

References (54–62)