arXiv:2305.00308v1 [cs.GT] 29 Apr 2023

奇偶博弈拟多项式算法的改进复杂度分析

Paweł Parys[0000−0001−7247−1408]⋆ 和 Aleksander Wiącek

波兰华沙大学信息学研究所

摘要。对于 $d = \omega(\log n)$，我们将求解（顶点带有优先级的）奇偶博弈的复杂度降低了 $\Theta(d^2)$ 倍：迄今已知的最佳复杂度为 $\mathcal{O}(mdn^{1.45+\log_2(d/\log_2 n)})$，而我们得到的复杂度为 $\mathcal{O}(mn^{1.45+\log_2(d/\log_2 n)}/d)$，其中 n 是顶点数，m 是边数，d 是优先级数。

我们的工作以使用通用树的现有算法为基础，并改进了这些算法的复杂度。我们提出了两项相互独立的改进。首先，通过更仔细地分析通用树的宽度，复杂度降低了 $\Theta(d)$ 倍。其次，我们对通用树的要求进行了更精细的分析（或者说，回顾了这项分析）：求解边上带有优先级的博弈需要一棵 n-通用树，而对于顶点带有优先级的博弈，使用一棵 $n/2$-通用树就足够了。由此，我们可以在此前求解规模为 n 的博弈所需的时间内，求解规模为 $2n$ 的博弈；这一变化使拟多项式复杂度再次降低了 $\Theta(d)$ 倍。

关键词： 奇偶博弈 · 通用树 · 准多项式时间

1 引言

自20世纪90年代初以来，奇偶博弈一直在自动机理论、逻辑及其在验证与综合中的应用中发挥着基础性作用。求解奇偶博弈中的胜者这一算法问题，可视为自动化验证与控制器综合问题的算法后端。它在多项式时间内等价于带奇偶接受条件的无限树上非确定性自动机的空性问题，也等价于模态 μ-演算 [14]的模型检验问题。它 [29]也处于 Church 综合问题算法求解方案的核心位置。此外，模态逻辑中的有效性或公式可满足性等判定问题，都可以归约为奇偶博弈求解。源自奇偶博弈的一些思想，不仅帮助人们在 ω-字 [5,10], 的自动机模型转换方面取得了新成果，也推动了计算机科学中相距较远领域（如马尔可夫决策过程 [15] 和线性规划 [19,18]）的发展。

⋆作者获波兰国家科学中心资助（资助编号：2021/41/B/ ST6/03914）。

求解奇偶博弈的复杂度，也就是判定哪位玩家拥有获胜策略，长期以来一直是一个悬而未决的问题。已知该问题属于 UP ∩coUP [20]（NP ∩ coNP 的子类），而其搜索变体（即寻找获胜策略）属于 PLS、PPAD，甚至属于它们的子类 CLS [9]。二十多年来，奇偶博弈求解算法的研究一直由这样一类算法主导：其运行时间随不同优先级 [33,6,31,21,32,30,2],的数量呈指数增长，或者在优先级数量较多时呈轻微的次指数增长 [4,25]。2017年，Calude、Jain、Khoussainov、Li 和 Stephan [7] 取得突破，提出了首个准多项式时间算法，并采用了对局摘要这一新颖思想。此后不久，又有若干准多项式时间算法问世 [22,16,26,28,27,11,24,3]。

Fijalkow [17] 明确提出了通用树 这一概念，而这一概念隐含在 Jurdziński and Lazić [22]的简洁树编码 结果中。随后人们发现，通用树不仅存在于所有现有的拟多项式时间奇偶博弈求解方法中，而且实际上是这些方法所必需的 [8,23,1]。具体而言，研究表明，任何采用现有方法求解具有 n 个顶点的奇偶博弈的算法，都需要在一棵 n-通用树（定义见下文）上运行。然而，这里有一个问题：这一必要性证明针对的是边上带有优先级的奇偶博弈，而人们通常考虑的是更不简洁的、顶点上带有优先级的博弈。（将边上的优先级转换为顶点上的优先级的一种通用方法是：将每个顶点替换为 d 个顶点，每个优先级对应一个，并将每条边重定向到目标顶点中优先级恰当的副本；这样，顶点数就会从 n 变为 nd。）

本文的主要贡献在于对n-通用树的宽度进行了更精细的分析。我们将这一宽度的上界（从而也将使用此类树的算法的运行时间上界）降低了$\Theta\left(\frac{d}{\log n}\right)$倍，其中 d是优先级的数量。

其次，回顾 Jurdziński and Lazić [22] 的结果可知，要解决顶点带有优先级的奇偶博弈，利用上文提到的“诀窍”，使用 $\lfloor n/2 \rfloor$-通用树就足够了，无需使用 n-通用树。这使得我们能够在原先解决规模为 n 的博弈所需的时间内，解决规模为 $2n$ 的博弈；因此，求解（顶点带有优先级的）奇偶博弈的复杂度再次降低了 $\Theta\left(\frac{d}{\log n}\right)$倍。

结合这两项改进，我们将求解奇偶博弈的复杂度上界降低了 $d = \omega(\log n)$ 倍的 $\Theta\left(\frac{d^2}{\text{对数}^2 n}\right)$：迄今已知的最佳界 [17,22], 即 $\mathcal{O}(mdn^{1.45+\log_2(d/\text{对数}_2 n)})$，降低为界 $\mathcal{O}(mn^{1.45+\log_2(d/\text{对数}_2 n)}/d)$。值得注意的是，这两个界都没有显示多对数因子；它们均由 $n^{\mathcal{O}(1)}$ 主导，其中 $\mathcal{O}(1)$ 来自 1.45 与实际应出现在指数中的常数 $\log_2 e$ 之间的差异（以这种形式书写界的方式沿用了先前的研究）。因此，虽然界以这种形式书写，改进幅度为 $\Theta(d^2)$ 倍。与此同时，这两项观察在 $d = \mathcal{O}(\log n)$ 的情形下都不足以改善复杂度的渐近性。

2 预备知识

奇偶博弈。 奇偶博弈 是两个玩家 Even 和 Odd 在游戏图 上进行的一种双人博弈。游戏图定义为一个元组 $\mathcal{G} = (V, V_{\mathsf{Even}}, E, d, \pi)$，其中 (V, E) 是一个非空有限有向图，每个顶点至少有一个后继顶点；其顶点标记有正整数优先级 $\pi\colon V \to \{1, 2, \ldots, d\}$（对于某个偶数 数 $d \in \mathbb{N}$），并划分为属于 Even 的顶点 V_{Even} 和属于 Odd 的顶点 $V_{\mathsf{Odd}} = V \setminus V_{\mathsf{Even}}$。我们通常将 $|V|$ 记作n，将 $|E|$记作 m。

直观地说，博弈的动态过程定义如下。对局从一个指定的起始顶点开始。接着，当前顶点所属的玩家选择该顶点的一个后继顶点，博弈便在该处继续。经过无限长的时间后，我们检查无限次出现的最大优先级；其奇偶性决定哪位玩家获胜。

形式上，我们使用位置策略（即无记忆策略）来定义博弈的胜者。*Even* 的位置策略 是一组边 $\sigma \subseteq E$，满足：对于 Even 的每个顶点 v， σ 中恰有一条以 v为起点的边；对于 Odd 的每个顶点 v， σ 中包含所有以 v为起点的边。*Odd* 的位置策略 通过交换 Even 和 Odd 的角色来定义。如果对于子图 (V, σ) 中的每条无限路径，沿该路径无限次出现的最大优先级为偶数（分别为奇数），则 Even（Odd）的位置策略 σ 从顶点 v 出发是获胜策略。如果 Even/Odd 拥有一个从 v出发的获胜位置策略，我们就说 Even/Odd 从 v 获胜 。

上述定义准确描述了奇偶博弈中的位置确定性 所保证的获胜者：对于奇偶博弈的每个顶点 v，Even 或 Odd 中必有一方从 v [13]获胜。因此，如果某位玩家能够通过一般策略（此处未定义）从某个顶点获胜，那么他也能够通过位置策略获胜。

树与通用树。 在进度测度算法中 [22,17,11]，策略通过从博弈图到有序树的映射来描述，定义如下：高度为 h 的有序树 （或简称树） T是一个有限的连通无环图，并为其中每个节点指定一个线性序 $\leq_x$，使得

– **</style}恰有一个入度为 0 的节点，称为根节点**；其他每个节点的入度均为 1；– 对于每个叶节点 （即出度为 0 的节点），从根节点到该叶节点的唯一路径包含 h 条边；– $\leq_x$ 是子节点 上的一个线性序，其中子节点指从 x出发有边指向的节点；该线性序描述了这些子节点从左到右的排列顺序。

树 T 的宽度 定义为其叶子的数量，记作 $|T|$。

设 T_1 和 T_2 是两棵高度相同的有序树。若存在一个保持子关系及$\leq_x$ 关系的单射映射 f，我们称 T_1 嵌入到 T_2 中：

– 如果 y 是 x 中 T_1的子节点，那么 $f(y)$ 是 T_2中 $f(x)$ 的子节点；并且– 如果 $y \leq_x z$ 在 T_1中，那么 $f(y) \leq_{f(x)} f(z)$ 在 T_2中。

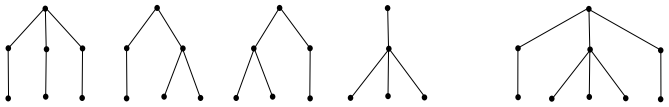


图 1. 所有四棵高度为 2、宽度为 3 的树（左）；一棵高度为 2 的 3-通用树（右）

高度为 h 的树 $\mathcal{T}$，如果每棵高度为 h 且宽度至多为 n 的有序树都嵌入到 $\mathcal{T}$ 中，则称其为 n-通用。示例请参见图 1。

3 关于通用树的宽度

本节将证明通用树宽度的一个改进上界，如下述定理所述：

定理 3.1. 对于所有 $n, h \in \mathbb{N}$，其中 $n \geq 1$，都存在一棵高度为 h、宽度至多为 $f(n, h)$ 的 n-通用树，其中

$$f(n,h) \leq n \cdot \binom{h-1+\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor} \leq n^{2.45-\varepsilon+\log_2\left(1+\frac{h-1}{\log_2 n}\right)}$$

对于某个 $\varepsilon > 0$。此外，$f(n, h) = \mathcal{O}(n^{2.45-\varepsilon+\log_2(h/\log_2 n))}$，若 $h = \omega(\log n)$。

我们指出，对于高度 h，通常取（接近）$d/2$，其中 d 是奇偶博弈中的优先级数量。指数中的值 $2.45-\varepsilon$ 表示实际常数略小于 2.45。

在给出证明之前，让我们将上述界与迄今已知的界进行比较：Fijalkow [17, 定理 4] 得出了一个 $2^{\lceil \log_2 n \rceil}\binom{h-1+\lceil \log_2 n \rceil}{\lceil \log_2 n \rceil}_{\log}$ 的上界。因此，差异在于将 $\lceil \log_2 n \rceil$ 替换为 $\lfloor \log_2 n \rfloor$。对于 n这两个值中的“大多数”情况，它们相差 1；此时两个二项式系数之比为 $\frac{h-1+\lceil \log_2 n \rceil}{h-1+\lceil \log_2 n \rceil}_{\log}$；当 $h = \omega(\log n)$ 时，该比值为 $\Theta\left(\frac{h}{\log n}\right)$。从渐近性来看，Fijalkow 的界为 $\mathcal{O}(hn^{2.45-\varepsilon+\log_2(h/\log_2 n)})$（证明见 Jurdziński and Lazić [22, 引理 6]），这确实比我们的界大一个 $\Theta(h)$ 的因子。

为了证明定理 3.1，我们采用 Fijalkow [17, Theorem4] 的一种构造（这本质上与 Jurdziński and Lazić [22] 的构造相同）。他证明存在一棵高度为 h、宽度为 $f(n, h)$ 的 n-通用树，其中函数 f（通过令 $f(0, h) = 0$ 将其扩展到 $n = 0$，并通过令 $f(n, 0) = 1$ 将其扩展到 $h = 0$、$n \geq 1$）由以下递归公式定义：

$$\begin{aligned}
&f(0,h) = 0 && \text{for } h \geq 0,\\
&f(n,0) = 1 && \text{for } n \geq 1,\\
&f(n,h) = f(n,h-1) + f(\lfloor n/2 \rfloor, h) + f(n-1-\lfloor n/2 \rfloor, h) && \text{for } n,h \geq 1. \quad (1)
\end{aligned}$$

与 Fijalkow 相比，[17], 我们因此对上述递归公式进行了更细致的分析。首先，我们给出 $f(n, h)$ 的显式公式：

引理 *3.2. </style}*函数 f 可以用以下显式公式表示，对所有 n、$h \geq 1$ 均成立：

$$f(n,h) = \sum_{i=0}^{\lfloor \log_2 n \rfloor - 1} 2^i \cdot \binom{h-1+i}{h-1} + (n - 2^{\lfloor \log_2 n \rfloor} + 1) \cdot \binom{h-1+\lfloor \log_2 n \rfloor}{h-1}.$$

虽然可以通过归纳证明来验证引理 3.2 的正确性，但这里我们给出一个基于生成函数的证明，因为这种证明更有启发性。

证明（引理 *3.2*）。令 $F(x,y) = \sum_{n,h\geq 1} f(n,h) x^n y^h$ 为二维数列 $f(n,h)$ 的生成函数，不包括 $n=0$ 和 $h=0$ 对应的值。将方程（1）的两边乘以 $x^n y^h$，然后对所有 $n,h \geq 1$ 求和；我们得到

$$F(x,y) = \sum_{\substack{n\geq 1\\ h\geq 1}} f(n,h-1)x^n y^h + \sum_{\substack{k\geq 1\\ h\geq 1}} (f(k,h)+f(k-1,h))x^{2k}y^h + \sum_{\substack{k\geq 0\\ h\geq 1}} 2f(k,h)x^{2k+1}y^h.$$

接下来，在上面的第一个求和中，我们将 h 加 1，并把包含 $f(n,0)$ 的项单独列为一个和；在上面的第二个求和中，我们将 $f(k,h) + f(k-1,h)$ 拆分为两个独立的和，其中在后一个和中，我们将 k 加 1，并把包含 $f(0,h)$ 的项单独列为一个和；在上面的最后一个求和中，我们把 $k=0$ 对应的项单独列为一个和。于是得到

$$\begin{aligned} F(x,y) &= \sum_{n\geq 1} f(n,0)x^n y + \sum_{\substack{n\geq 1\\ h\geq 1}} f(n,h)x^n y^{h+1} \\ &\quad + \sum_{\substack{k\geq 1\\ h\geq 1}} f(k,h)(x^2)^k y^h + \sum_{h\geq 1} f(0,h)x^2 y^h + \sum_{\substack{k\geq 1\\ h\geq 1}} f(k,h)(x^2)^k y^h x^2 \\ &\quad + \sum_{h\geq 1} 2f(0,h)xy^h + \sum_{\substack{k\geq 1\\ h\geq 1}} 2f(k,h)(x^2)^k y^h x \\ &= \frac{xy}{1-x} + yF(x,y) + F(x^2,y) + 0 + x^2F(x^2,y) + 0 + 2xF(x^2,y). \end{aligned}$$

由此得到关于生成函数的下列方程：

$$F(x,y) = yF(x,y) + F(x^2,y)(1+2x+x^2) + \frac{xy}{1-x}. \tag{2}$$

令 $H(x,\ y) = \frac{1-x}{y} F(x,\ y)$；则 $F(x,\ y) = \frac{y}{1-x} H(x,\ y)$。注意，$H$ 是表示 f 连续取值时各值之差的生成函数

n 的取值，其中 h 向后移位 1。将其代入方程（2），得到：

$$\frac{y}{1-x}H(x,y)=\frac{y^2}{1-x}H(x,y)+\frac{y}{1-x^2}H(x^2,y)(1+x)^2+\frac{xy}{1-x},$$
$$\frac{y}{1-x}H(x,y)=\frac{y^2}{1-x}H(x,y)+\frac{y}{1-x}H(x^2,y)(1+x)+\frac{xy}{1-x},$$
$$H(x,y)=yH(x,y)+H(x^2,y)(1+x)+x.$$

接下来，记 $H(x,\ y)=\sum_{n\geq 1}x^n h_n(y)$。将其代入上面的方程：

$$\sum_{n\geq 1}x^n h_n(y)=\sum_{n\geq 1}x^n y h_n(y)+\sum_{n\geq 1}(x^{2n}+x^{2n+1})h_n(y)+x.$$

为了求出 h_n，我们比较方程两边 x^n 前面的系数。对于 $n=1$，有

$$h_1(y)=yh_1(y)+1,\qquad \text{so}\qquad h_1(y)=\frac{1}{1-y},$$

对于 $n\geq 2$，有

$$h_n(y)=yh_n(y)+h_{\lfloor n/2\rfloor}(y),\qquad \text{so}\qquad h_n(y)=\frac{h_{\lfloor n/2\rfloor}(y)}{1-y}.$$

由此可知，对所有 $n\geq 1$，都有如下公式

$$h_n(y)=\frac{1}{(1-y)^{\lfloor \log_2 n\rfloor+1}}.$$

下文用记号 $[x^n]A(x)$ 表示函数 $A(x)$ 中 x^n 前面的系数。我们还要用到以下公式，其中 $k\geq 1$:

$$\frac{1}{(1-x)^k}=\sum_{n=0}^{\infty}\binom{n+k-1}{n}x^n. \tag{3}$$

现在我们结束证明（这里假设 $n,\ h\geq 1$）：

$$\begin{aligned}
f(n,h)&=[x^n y^h]F(x,y)=[x^n y^h]\frac{y}{1-x}H(x,y)\\
&=[y^{h-1}]\sum_{j=0}^{n}[x^j]H(x,y)=[y^{h-1}]\sum_{j=1}^{n}h_j(y)\\
&=[y^{h-1}]\left(\sum_{i=0}^{\lfloor \log_2 n\rfloor-1}2^i h_{2^i}(y)+(n-2^{\lfloor \log_2 n\rfloor}+1)h_{2^{\lfloor \log_2 n\rfloor}}(y)\right)\\
&=\sum_{i=0}^{\lfloor \log_2 n\rfloor-1}2^i[y^{h-1}]\frac{1}{(1-y)^{i+1}}+(n-2^{\lfloor \log_2 n\rfloor}+1)[y^{h-1}]\frac{1}{(1-y)^{\lfloor \log_2 n\rfloor+1}}\\
&=\sum_{i=0}^{\lfloor \log_2 n\rfloor-1}2^i\cdot\binom{h-1+i}{h-1}+(n-2^{\lfloor \log_2 n\rfloor}+1)\cdot\binom{h-1+\lfloor \log_2 n\rfloor}{h-1}.
\end{aligned}$$

上面，第三行是根据 $h_{2^i} = h_{2^i+1} = \ldots = h_{2^{i+1}-1}$ 得出的，最后一行是根据公式（3）得出的。引理 3.2 的证明至此完成。

接下来，我们给出两个辅助引理，它们在求渐近界时很有用：

引理 3.3. 对于所有 $x \geq 0$，都有 $\ln(1+x) \cdot (1+x) \geq x$。

证明。设 $h(x) = \ln(1+x) \cdot (1+x) - x$。由于 $h(0) = 0$，只需证明函数 h 是递增的。为此，我们计算其导数，得到 $h'(x) = \ln(1+x)$，因此对于 $x > 0$，$h'(x) > 0$，函数 h 确实是递增的。

引理 3.4. 对于每个 $c \geq 0$，函数

$$\alpha_c(x) = \left(1 + \frac{c}{x}\right)^x$$

在 $x > 0$ 上非递减。

证明。我们计算 α_c 的导数：

$$\alpha_c'(x) = \alpha_c(x) \cdot \left(\ln\left(1 + \frac{c}{x}\right) - \frac{c}{x \cdot \left(1 + \frac{c}{x}\right)}\right).$$

由于对于 $x > 0$，$\alpha_c(x) \geq 0$，要确认 $\alpha_c'(x) \geq 0$，只需检查

$$\ln\left(1 + \frac{c}{x}\right) \geq \frac{c}{x \cdot \left(1 + \frac{c}{x}\right)}.$$

为证明这个不等式，我们将两边都乘以 $1 + \frac{c}{x}$，并使用引理 3.3。

现在我们可以完成定理 3.1 的证明：

证明（定理 *3.1*）。为了得到 $f(n, h) \leq n \cdot \binom{h-1+\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor}$对数，我们将引理 3.2 公式中的所有二项式系数替换为 $\binom{h-1+\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor}$对数；显然 $\binom{h-1+i}{h-1} \leq \binom{h-1+\lfloor \log_2 n \rfloor}{h-1} = \binom{h-1+\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor}$对数对于 $i \leq \lfloor$对数$_2\, n \rfloor$。

回忆一下，对于任意 $x,\ y > 0,\ x^{\log_2 y} = y^{\log_2 x}$。定理陈述中的第二个不等式依次由估计式 $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$、不等式 $\log_2 e < 1.45$ 和引理 3.4 得出：

$$\begin{aligned} f(n,h) &\leq n \cdot \binom{h-1+\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor} \leq n \cdot \left(e\left(1 + \frac{h-1}{\lfloor \log_2 n \rfloor}\right)\right)^{\lfloor \log_2 n \rfloor} \\ &\leq n \cdot e^{\log_2 n} \cdot \left(1 + \frac{h-1}{\lfloor \log_2 n \rfloor}\right)^{\lfloor \log_2 n \rfloor} = n^{1+\log_2 e} \cdot \left(1 + \frac{h-1}{\lfloor \log_2 n \rfloor}\right)^{\lfloor \log_2 n \rfloor} \\ &\leq n^{2.45-\varepsilon} \cdot \left(1 + \frac{h-1}{\log_2 n}\right)^{\log_2 n} = n^{2.45-\varepsilon+\log_2\left(1+\frac{h-1}{\log_2 n}\right)}. \end{aligned}$$

现在假设 $h = \omega(\log n)$。那么

$$\log_2\left(1 + \frac{h-1}{\log_2 n}\right) = \log_2\left(\frac{h}{\log_2 n} \cdot (1 + o(1))\right)$$
$$= \log_2\left(\frac{h}{\log_2 n}\right) + \log_2(1 + o(1)).$$

分量 $\log_2(1 + o(1)) = o(1)$ 可以移除，代价是降低常数 ε，因此我们得到

$$f(n,h) \le n^{2.45-\varepsilon+\log_2\left(1+\frac{h-1}{\log_2 n}\right)} = \mathcal{O}\left(n^{2.45-\varepsilon+\log_2\left(\frac{h}{\log_2 n}\right)}\right). \qquad \square$$

4 使用更小的树

展示如何使用通用树来求解奇偶博弈的论文 [17,11,23,24]假设，要想求解含有 n 个顶点的博弈，就需要 n- 通用树。对于边上的优先级的博弈，可以证明确实需要 n-通用性 [8,1]。然而，大多数论文（包括我们的论文）定义的奇偶博弈是在顶点上设置优先级的；在这种情况下，必要性证明并不适用。

让我们回到 Jurdziński and Lazić 的论文 [22]。尽管该论文没有明确提及通用树，却使用了一种非常特殊的通用树，称为简洁树编码。重要的是，这棵树并非n-通用，而是 η-通用，其中 η 可以是奇优先级顶点的数量，也可以是偶优先级顶点的数量，取两者中较小者即可，因此显然有 $\eta \le \lfloor n/2 \rfloor$ （严格来说，在他们的论文中， η 表示偶优先级顶点的数量，但他们在第 4 节开头解释说，可以将优先级平移 1，从而确保 $\eta \le \lfloor n/2 \rfloor$）。此外，阅读他们的论文可以发现，第 2 节定义简洁树编码，与后续使用该编码的章节之间唯一的“接口”就是引理 1，其中证明了简洁树编码是一棵 η-通用树。由此很容易看出，Jurdziński and Lazić 的算法同样适用于用任何其他η-通用树（高度适当，即 $d/2$）替代简洁树编码。特别是，我们可以使用定理 3.1 中宽度更小的通用树。

现在，我们来界定算法所需通用树的宽度，并将其与以往方法进行比较。为避免引入额外参数 η，下文用 $\lfloor n/2 \rfloor$ 替代它，并利用不等式 $\eta \le \lfloor n/2 \rfloor$。将定理 3.1 中公式里的 $d/2$ 替换为 h，并将 $\lfloor n/2 \rfloor$ 替换为 n，可得高度为 $d/2$ 的 $\lfloor n/2 \rfloor$-通用树 $\mathcal{T}$ 的宽度可以满足

$$|\mathcal{T}| \le \left\lfloor \frac{n}{2} \right\rfloor \cdot \binom{d/2 - 1 + \lfloor \log_2 \lfloor n/2 \rfloor \rfloor}{\lfloor \log_2 \lfloor n/2 \rfloor \rfloor}.$$

作为比较，根据定理 3.1，对于 n-通用树 $\mathcal{T}'$，我们有

$$|\mathcal{T}'| \leq n \cdot \binom{d/2 - 1 + \lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor}.$$

对于自然数集 n，我们总有 $\lfloor \log_2 \lfloor n/2 \rfloor \rfloor = \lfloor \log_2 (n/2) \rfloor = \lfloor \log_2 n \rfloor - 1$，因此两个二项式系数的商为 $\binom{d/2-1+\lfloor \log_2 \lfloor n/2 \rfloor \rfloor}{\lfloor \log_2 n \rfloor}$，当 $d = \omega(\log n)$ 时，该商属于 $\Theta\left(d^{\log n}\right)$。

现在让我们确定 $d = \omega(\log n)$ 的渐近性。首先，简化 n-通用树 $\mathcal{T}'$ 的公式：

$$|\mathcal{T}'| = \mathcal{O}(n^{2.45-\varepsilon+\log_2(d/2/\log_2 n)}) = \mathcal{O}(n^{1.45-\varepsilon+\log_2(d/\log_2 n)}).$$

因此，对于 $\lfloor n/2 \rfloor$-通用树 $\mathcal{T}$，我们有

$$|\mathcal{T}| = \mathcal{O}\left(\left\lfloor \frac{n}{2} \right\rfloor^{1.45-\varepsilon+\log_2(d/\log_2 \lfloor n/2 \rfloor)}\right) = \mathcal{O}\left(\left(\frac{n}{2}\right)^{1.45-\varepsilon+\log_2(d/\log_2 n)}\right). \quad (4)$$

上面的第二个等式是通过将 $\lfloor \frac{n}{2} \rfloor$ 替换为稍大的值 $\frac{n}{2}$，并注意到以下事实得到的：

$$\begin{aligned}\log_2\left(\frac{d}{\log_2 \lfloor n/2 \rfloor}\right) &= \log_2\left(\frac{d}{\log_2 n} \cdot (1 + o(1))\right) \\ &= \log_2\left(\frac{d}{\log_2 n}\right) + \log_2(1 + o(1));\end{aligned}$$

通过减小常数 ε，可以去掉分量 $\log_2(1 + o(1)) = o(1)$。接下来我们分析界的对数：

$$\begin{aligned}\log_2\left(\frac{n}{2}\right) \cdot \left(1.45 - \varepsilon + \log_2 \frac{d}{\log_2 n}\right) &= (\log_2 n - 1) \cdot \left(1.45 - \varepsilon + \log_2 \frac{d}{\log_2 n}\right) \\ &\leq \log_2 n \cdot \left(1.45 - \varepsilon + \log_2 \frac{d}{\log_2 n}\right) - \log_2 \frac{d}{\log_2 n} + \varepsilon \\ &\leq \log_2 n \cdot \left(1.45 - \varepsilon + o(1) + \log_2 \frac{d}{\log_2 n}\right) - \log_2 d + \varepsilon; \quad (5)\end{aligned}$$

上面的最后一个等式是通过观察以下事实得到的：

$$-\log_2 \frac{d}{\log_2 n} = -\log_2 d + \log_2 \log_2 n = -\log_2 d + \log_2 n \cdot o(1).$$

结合方程（4）和（5），我们得到关于 $|\mathcal{T}|$ 的以下界：

$$|\mathcal{T}| = \mathcal{O}\left(n^{1.45-\varepsilon+o(1)+\log_2(d/\log_2 n)} \cdot \frac{1}{d}\right).$$

通过再次减小常数 ε，可以移除 $o(1)$分量；因此我们可以写成

$$|\mathcal{T}| = \mathcal{O}\left(n^{1.45-\varepsilon+\log_2(d/\log_2 n)} \cdot \frac{1}{d}\right).$$

现在我们来讨论算法本身的复杂度。正如 Jurdziński 和 Lazić [22, 定理 7], 所指出的，他们的算法在使用通用树 $\widehat{\mathcal{T}}$ 时，运行时间为 $\mathcal{O}(m \cdot \log n \cdot \log d \cdot |\widehat{\mathcal{T}}|)$。利用 $\mathcal{T}$ 作为 $\widehat{\mathcal{T}}$，并注意到多对数函数 $O(\log n \cdot \log d)$ 可以再次被指数中的 $-\varepsilon$ 项“吸收”，我们得到以下复杂度界，这也是我们的主要定理：

定理 4.1. 对于 $d = \omega(\log n)$，可以在运行时间为以下表达式的算法中，找出一个含有 n个顶点、 m 条边和 d 个优先级的奇偶博弈的赢家：

$$\mathcal{O}\left(m \cdot n^{1.45+\log_2(d/\log_2 n)} \cdot \frac{1}{d}\right).$$

注 *4.2.* 与先前的界 $\mathcal{O}\left(m \cdot d \cdot n^{1.45+\log_2(d/\log_2 n)}\right)$（见 Fijalkow [17]）相比，我们将其改进了 $\Theta(d^2)$ 倍。通过改进通用树大小的界，在定理 3.1 中获得了一个 $\Theta(d)$ 倍的改进。通过使用 $\lfloor n/2 \rfloor$-通用树代替 n-通用树，又获得了第二个 $\Theta(d)$ 倍的改进。

注 *4.3*。 Fijalkow [17]得到的复杂度与 Jurdziński and Lazić [22]相同：通过采用更好的通用树构造，而非“简洁树编码”，他将复杂度降低了一个 $\Theta(d)$ 因子；但与此同时，由于使用 n-通用树而不是 $\lfloor n/2 \rfloor$-通用树，复杂度又增加了一个 $\Theta(d)$ 因子。

我们也借此指出，Jurdziński and Lazić [22]的论文中有一个小错误。具体来说，在引理 6 的证明中，他们改为分析一个更简单的表达式 $\binom{\lceil \log_2 \eta \rceil + d/2}{\lceil \log_2 \eta \rceil}_{\text{对数}}$，而不是 $\binom{\lceil \log_2 \eta \rceil + d/2 + 1}{\lceil \log_2 \eta \rceil + 1}_{\text{对数}}$，并称这个较简单的表达式与后者相差一个常数因子。不过，这一说法是错误的：两个表达式的比值为 $\frac{\lceil \text{对数} \binom{\log_2 \eta \rceil + d/2+1}{\log_2 \eta \rceil + 1} \text{对数}}{\omega}$，当 $d = \omega(\log \eta)$ 时，它属于 $\Theta\left(\frac{d}{\text{对数}\ \eta}\right)$。因此，根据他们对 $d = \omega(\log \eta)$ 算法的分析，实际得到的复杂度上界不应如他们所称为 $\mathcal{O}\left(m \cdot d \cdot \eta^{1.45+\log_2(\log)}\right)$，而应为 $\mathcal{O}\left(m \cdot d^2 \cdot \eta^{1.45+\log_2(d/\text{对数}_2\ \eta)}\right)$（并且取 $n/2$ 作为 η，可得复杂度 $\mathcal{O}\left(m \cdot d \cdot n^{1.45+\log_2(d/\text{对数}_2\ n)}\right)$，与 Fijalkow [17]的结果相同）。

注 *4.4*。 本论文准备期间，Dell’Erba 和 Schewe [12]发表了对 Jurdziński 和 Lazić 算法 [22]的另一项改进。由于作者没有给出精确的界，因此难以进行完整讨论。

参考文献

1. Arnold, A., Niwiński, D., Parys, P.：用于不动点评估的准多项式黑盒算法。载于：CSL。LIPIcs，第 183 卷，第 9:1–9:23 页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik（2021） 2. Benerecetti, M., Dell'Erba, D., Mogavero, F.：通过优先级提升求解奇偶博弈。Formal Methods Syst. Des. **52**(2), 193–226（2018） 3. Benerecetti, M., Dell'Erba, D., Mogavero, F., Schewe, S., Wojtczak, D.：具有 Parysian 风格的优先级提升。CoRR **abs/2105.01738**（2021） 4. Björklund, H., Vorobyov, S.G.：用于均值收益博弈的组合型强次指数策略改进算法。Discret. Appl. Math. **155**(2), 210–229（2007） 5. Boker, U., Lehtinen, K.：迈向交替弱自动机。载于：FSTTCS。LIPIcs，第 122 卷，第 21:1–21:22 页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik（2018） 6. Browne, A., Clarke, E.M., Jha, S., Long, D.E., Marrero, W.R.：一种改进的不动点表达式求值算法。Theor. Comput. Sci. **178**(1-2), 237–255（1997） 7. Calude, C.S., Jain, S., Khoussainov, B., Li, W., Stephan, F.：在准多项式时间内判定奇偶博弈。载于：STOC。第 252–263 页。ACM（2017） 8. Czerwiński, W., Daviaud, L., Fijalkow, N., Jurdziński, M., Lazić, R., Parys, P.：通用树在分离自动机中生长：奇偶博弈的准多项式下界。载于：SODA。第 2333–2349 页。SIAM（2019） 9. Daskalakis, C., Papadimitriou, C.H.：连续局部搜索。载于：SODA。第 790–804 页。SIAM（2011） 10. Daviaud, L., Jurdziński, M., Lehtinen, K.：由通用树构造交替弱自动机。载于：CONCUR。LIPIcs，第 140 卷，第 18:1–18:14 页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik（2019） 11. Daviaud, L., Jurdziński, M., Thejaswini, K.S.：奇偶博弈的 Strahler 数。载于：ICALP。LIPIcs，第 168 卷，第 123:1–123:19 页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik（2020） 12. Dell'Erba, D., Schewe, S.：更小的进度度量与奇偶博弈的分离自动机。CoRR **abs/2205.00744**（2022） 13. Emerson, E.A., Jutla, C.S.：树自动机、μ演算与确定性（扩展摘要）。载于：FOCS。第 368–377 页。IEEE Computer Society（1991） 14. Emerson, E.A., Jutla, C.S., Sistla, A.P.：关于 μ 演算及其片段的模型检测。Theor. Comput. Sci. **258**(1-2), 491–522（2001） 15. Fearnley, J.：策略迭代的指数下界。载于：ICALP (2)。Lecture Notes in Computer Science，第 6199 卷，第 551–562 页。Springer（2010） 16. Fearnley, J., Jain, S., de Keijzer, B., Schewe, S., Stephan, F., Wojtczak, D.：一种在准多项式时间和准线性空间内求解奇偶博弈的有序方法。Int. J. Softw. Tools Technol. Transf. **21**(3), 325–349（2019） 17. Fijalkow, N.：用于奇偶博弈的最优值迭代算法。CoRR **abs/1801.09618**（2018） 18. Friedmann, O.：求解线性规划和博弈的 Zadeh 的枢轴规则的次指数下界。载于：IPCO。Lecture Notes in Computer Science，第 6655 卷，第 192–206 页。Springer（2011） 19. Friedmann, O., Hansen, T.D., Zwick, U.：单纯形算法随机枢轴规则的次指数下界。载于：STOC。第 283–292 页。ACM（2011）

20. Jurdziński, M.：判定奇偶博弈的胜者属于 UP ∩ co-UP。信息处理快报 **68**(3)，119–124（1998年）21. Jurdziński, M.：用于求解奇偶博弈的小型进展度量。载于：STACS。《Lecture Notes in Computer Science》，第1770卷，第290–301页。Springer（2000年）22. Jurdziński, M., Lazić, R.：用于求解奇偶博弈的简洁进展度量。载于：LICS。第1–9页。IEEE Computer Society（2017年）23. Jurdziński, M., Morvan, R.：用于奇偶博弈的通用吸引子分解算法。CoRR **abs/2001.04333**（2020）24. Jurdziński, M., Morvan, R., Ohlmann, P., Thejaswini, K.S.：用于奇偶博弈的对称吸引子分解提升算法。CoRR **abs/2010.08288**（2020）25. Jurdziński, M., Paterson, M., Zwick, U.：求解奇偶博弈的确定性次指数算法。SIAM计算杂志 **38**(4)，1519–1532（2008年）26. Lehtinen, K.：关于准多项式时间求解奇偶博弈的模态视角。载于：LICS。第639–648页。ACM（2018）27. Lehtinen, K., Parys, P., Schewe, S., Wojtczak, D.：一种以准多项式时间求解奇偶博弈的递归方法。计算机科学逻辑方法 **18**(1)（2022）28. Parys, P.：奇偶博弈：准多项式时间的Zielonka算法。载于：MFCS。LIPIcs，第138卷，第10:1–10:13页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik（2019）29. Rabin, M.O.：《无限对象上的自动机与Church问题》。美国数学学会，美国马萨诸塞州波士顿（1972年）30. Schewe, S.：以大步求解奇偶博弈。计算机与系统科学杂志 **84**，243–262（2017年）31. Seidl, H.：快速而简单的嵌套不动点。信息处理快报 **59**(6)，303–308（1996年）32. Vöge, J., Jurdziński, M.：一种用于求解奇偶博弈的离散策略改进算法。载于：CAV。《Lecture Notes in Computer Science》，第1855卷，第202–215页。Springer（2000年）33. Zielonka, W.：有限着色图上的无限博弈及其在无限树上自动机中的应用。Theor. Comput. Sci. **200**(1-2)，135–183（1998年）