均值收益博弈的对称递归算法

PierreOhlmann ✉CNRS，信息学与系统实验室（LIS），马赛，法国

摘要

我们提出一种用于求解均值收益博弈的新的确定性对称递归算法。

2012年 ACM 主题分类 计算理论 → 逻辑与验证

关键词与短语 均值收益博弈，算法

致谢 我们感谢 Pierre Gaillard、Antonio Casares 和 Aditya Prakash 就该算法展开讨论。

arXiv:2603.07555v1 [cs.GT] 8 Mar 2026

1 引言

在平均收益博弈中，Min 和 Max 轮流沿着无汇点有向图的整数权边移动一个令牌。对局无限持续，其值定义为长期权重的平均值。Ehrenfeucht 和 Mycielski [7]提出了这类博弈，并证明了其位置确定性：为了采取最优策略，玩家无需记忆，并且在每个给定顶点上始终可以作出相同的选择。后来，Gurvich、Karzanov 和 Khachyian [9], 对其进行了更深入的研究，随后 Zwick 和 Paterson [14] 推广了判断哪些顶点具有正值这一问题，指出该问题属于 **NP** ∩**coNP**（这是位置确定性的一个直接推论），提出了一种伪多项式算法，并提出了能否在多项式时间内求解这类博弈的问题。

迄今为止，人们提出了多种算法，其中最重要的包括 Brim 等人的值迭代算法 [3], 、GKK 算法 [9], ，以及 Björklund 和 Vorobiov [1] 、Schewe [12] 提出的策略改进算法；关于多种算法的统一介绍，另请参阅 [4]。就运行时间界而言，若 n, m 和 W分别表示顶点数、边数和权重绝对值的最大值，那么目前已知最佳的伪多项式上界是 $O(nmW \log(W))$, 由 [3], 给出；目前已知最佳的组合界为 $O(m2^{\frac{n}{2}} \log(W))$，适用于 GKK 算法 [6, 11][1],；而 [1], 的策略改进算法类似于线性规划中的随机面算法，其运行时间为随机亚指数时间 $O(2^{O(\sqrt{n \log n})} \log(W))$。Loff 和 Skomra 最近取得了一项值得关注的进展：他们提出了一种算法，对于一般的平均收益博弈，其运行时间为多项式时间（按平滑分析的意义而言） [10]。然而，尽管学界数十年来付出了大量努力，目前仍未发现能够求解平均收益博弈的确定性亚指数时间算法。

我们的贡献是一种求解平均收益博弈的新型确定性算法。该算法完全对称（即对两名玩家一视同仁）；在上述讨论的算法中，只有 GKK 算法也具有这一性质。该算法以递归形式自然呈现，这也是一项创新。我们的算法与所有已知平均收益算法的另一项重要区别（Zwick 和 Paterson 的算法除外） [14],在于，我们不计算能量值（定义见预备知识）。尽管我们依赖于

[1] Dorfman、Kaplan 和 Zwick 最初针对值迭代算法的一个复杂变体证明了这一点。我们认为，由此得到的算法实际上与（概念上更简单的）GKK 算法在思路上相似，并在 [11] 中证明了相同的界。

Gurvich、Karzanov 和 Khachyian [9] 提出的势函数约简，而这种约简也是 Cadilhac、Casares 与作者 [4],提出的统一框架的基础，但我们的算法并不适用于该框架。我们认为，该算法有望达到次指数运行时间，但将这一上界留待未来工作证明。

在介绍完预备知识（第 2 节）后，我们将描述算法（第 3 节）、证明其正确性（第 4 节）、给出伪代码（第 5 节）、提出优化与变体（第 6 节），最后以简短的讨论作结（第 7 节）。

2 预备知识与背景

现在我们来介绍一些关于平均收益博弈的背景知识。我们的目的不仅是给出阅读本文其余部分所必需的定义，也希望提供一些背景信息并阐述几点直观理解，让阅读更加轻松。在不太麻烦的情况下，我们尽量避免使用“正”或“非负”之类容易引起混淆的术语，而改用 >0 或 ≥ 0。

博弈是一个无汇点[2]有向图 $G=(V,\ E,\ w)$，其边的权重由 $w\colon E\to\mathbb{Z}$ 给出，且其顶点划分为 V_{Min} 和 V_{Max}，分别称为 Min顶点和 Max顶点。直观地说，当我们处于一个 Min顶点时，Min 会选择要沿哪条出边前进；Max 也是如此。子博弈是一个无汇点子图。Min 的陷阱集是一个子博弈 G'，满足所有来自 G' 的 Min 边都指向 G'；Max 的情形类似。

与引言中一样，我们写作 $n=|V|$、$m=|E|$ 和 $W=\max_{vv'\in E}|w(vv')|$。有时我们将边及其权重写作 $v\xrightarrow{w}v'$，将（有限和无限）路径写作 $v_0\xrightarrow{w_0}v_1\xrightarrow{w_1}\dots$。路径各前缀的部分和序列，即 0，$w_0$，$w_0+w_1$，……，称为该路径的剖面。

一个 Min策略[3] 由对每个 Min 顶点选择其一条出边组成。如果每当访问一个 Min 顶点时，下一条边都是策略指定的边，则称路径与该 Min策略一致。我们将用 $\pi\models\sigma$ 表示路径 π 与策略 σ 一致。Max 策略也类似。赋值是从无限路径到 $\mathbb{R}$ 的映射；本文涉及的三种赋值如下：

$$\begin{aligned}
\mathrm{MP}(v_0\xrightarrow{w_0}v_1\xrightarrow{w_1}\dots) &= \limsup_k \tfrac{1}{k}\textstyle\sum_{i=0}^{k-1} w_i,\\
\sup\Sigma^X(v_0\xrightarrow{w_0}v_1\xrightarrow{w_1}\dots) &= \sup_{k\leq n_X}\textstyle\sum_{i=0}^{k-1} w_i,\\
\inf\Sigma^X(v_0\xrightarrow{w_0}v_1\xrightarrow{w_1}\dots) &= \inf_{k\leq n_X}\textstyle\sum_{i=0}^{k-1} w_i,
\end{aligned}$$

其中 X 是一个顶点集，且 n_X 是使得 $v_i\in X$ 成立的最小 i（若不存在这样的顶点，则为 $n_X=\infty$，此时 sup 或 inf 遍历所有自然数 k）。换言之，在进行平均收益博弈（即赋值为 MP 的博弈）时，参与者力求优化长期平均值。[4] 在进行赋值为 $\sup\Sigma^X$ 的博弈时，参与者力求在到达 X 之前优化剖面的峰值（此时可以认为博弈停止）。若未到达 X，则博弈永远持续，特别地，我们可能得到值 ∞。对 $\inf\Sigma^X$ 亦然。由于权重为 0 的空路径也被考虑在内，注意 $\sup\Sigma^X$ 只取 ≥ 0 值，而 $\inf\Sigma^X$ 取 ≤ 0 值。当 X 为空时，我们只写作 sup 或 inf。在文献中，此类博弈（X为空）通常称为能量博弈，由 [5] 提出，并由 [2, 3]进一步研究并推广。我们现在正式陈述位置确定性。

[2] 汇点是没有出边的顶点。[3] 本文只考虑位置策略，因为它们对于所研究的博弈已经足够。[4] 由于位置确定性以及博弈的有限性，使用上极限而非下极限并不重要，因为它不影响值。

▶**定理 1** ([8, 2])**.** 对任意博弈、顶点的任意子集 X、任意 赋值 $\in\{\mathrm{MP}, \sup\Sigma^X, \inf\Sigma^X\}$，以及任意顶点 v，均成立

$$\inf_{\sigma} \sup_{\pi \models \sigma} \mathrm{val}(\pi) = \sup_{\tau} \inf_{\pi \models \tau} \mathrm{val}(\pi),$$

其中，σ 和 τ 分别遍历 *Min* 和 *Max* 的策略，而 π 遍历从 v开始的路径。

定理中的值称为顶点 v 的 val 值；如果根据上下文可以明确 val 的含义，也简称为其值。此外，最优策略（由于策略数量有限，显然存在）可以独立于起始顶点进行选择。得益于该定理，我们可以将这些博弈视为循环博弈；例如，在平均收益博弈中，Min 和 Max 固定最优策略，此时某个顶点的值就是沿着这两种策略从 v 出发所能到达的唯一环的平均权重。

在平均收益阈值问题中，我们询问：给定一个博弈，哪些顶点的值为≤ 0，哪些顶点的值为 > 0。这些顶点称为该博弈的获胜区域。更一般的值问题要求确定各顶点的确切值（在 $[-W, W]$ 中，该值是一个分母为 $\leq n$ 的有理数），它可以通过二分法归约为阈值问题：要检验值是否为 $\leq x$，只需在所有权重均平移了 $-x$ 的博弈中求解阈值问题。

正如 [3], 首次正式指出的那样，该定理的一个直接推论是：平均收益值为 < 0 的顶点具有有限的上确界值和负无穷的下确界值；同样地，平均收益值为 > 0 的顶点具有无穷的上确界值和有限的下确界值。平均收益值为 $= 0$ 的顶点的上确界值和下确界值均有限，处理它们可能会很繁琐，因为它们可能破坏对称性（例如，在上述平均收益阈值问题的定义中）。幸运的是，我们可以通过将每个权重 w 替换为 $nw + 1$来消除它们，而不会造成一般性损失：这会保留所有非零环的符号（从而保留非零值的符号），并将所有权重为零的环变为正环（因此，值为零的顶点现在会获得正值）。在全文中，我们都假设不存在权重为零的环。[5]

总而言之，我们有两类顶点：平均收益值为 < 0、上确界值有限且下确界值为负无穷的顶点，以及平均收益值为 > 0、上确界值为无穷且下确界值有限的顶点。我们的目标是确定每个顶点属于哪一类。我们将这种划分称为博弈中的获胜区域。正如引言中所述，几乎所有已知的算法（据我们所知，除了 [14]之外的全部算法）实际上都会计算上确界值（有时称为能量值），以确定它们是否有限。

接下来给出一个关于 supΣ 的实用技术性结果。（自然，对于 infΣ 也有类似结论。）

▶**引理 2**。 在一个博弈中，我们有

$$\sup\Sigma^X(v_0 \xrightarrow{w_0} v_1 \xrightarrow{w_1} v_2 \xrightarrow{w_2} \dots) = w_0 + \sup\Sigma^X(v_1 \xrightarrow{w_1} v_2 \xrightarrow{w_2} \dots)$$

只要 $v_0 \notin X$ 且右侧为 ≥ 0。

[5] 注意，将问题归约为这类博弈会把 W 替换为 $nW + 1$，但这不会影响组合界（即不依赖于 W 的界）。

证明。 由于 $v_0 \notin X$ 我们有 $n_X > 0$ 因此

$$\begin{aligned}\sup\Sigma^X(v_0 \xrightarrow{w_0} v_1 \xrightarrow{w_1} \dots) &= \sup_{k\le n_X}\sum_{i=0}^{k-1} w_i \\ &= \max(0, \sup_{1\le k\le n_X}\sum_{i=0}^{k-1} w_i) \\ &= \max(0, w_0 + \sup\Sigma(v_1 \xrightarrow{w_0} v_2 \xrightarrow{w_2} \dots)) \\ &= w_0 + \sup\Sigma(v_1 \xrightarrow{w_0} v_2 \xrightarrow{w_2} \dots)\end{aligned}$$

◀

现在，我们引入一些并非通常背景知识的一部分、但对本文至关重要的额外概念和术语。若 v 是 Min 顶点，且 vv' 是从 v出发的边中权重最小的边；或者 v 是 Max 顶点，且 vv'是从 v出发的边中权重最大的边，则边 vv' 是立即最优的。给定一个博弈 G，令 N、Z、P分别表示立即最优边的权重为 < 0、$= 0$、> 0的顶点集。令 ZN（分别为 ZP）表示这样一组顶点：Min（分别为 Max）能够迫使对局在遇到 > 0（分别为 < 0）边之前先遇到一条 < 0（分别为 > 0）边。（名称 ZN 源自"先零后 N"。）注意，$N \subseteq ZN$和 $P \subseteq ZP$，且 ZN和 ZP分别构成顶点的划分（因为不存在零权环）。如果上下文中无法明确博弈是哪一个，我们会将其写作下标，例如 P_G或 ZN_G。我们称 N、Z、P、NZ和 ZP为该博弈的区域；通过标准回溯，可以在 $O(m)$中计算出它们（用标准奇偶博弈术语来说，ZN是关于 ≤ 0边、指向 < 0 边的 Min 吸引集）。

如果 Min 能够迫使对手立即看到一条通向 ZN 的 ≤ 0 边，那么我们称 ZN 中的顶点是约简的；类似地，如果 Max 能够迫使对手立即看到一条通向 ZP 的 ≥ 0 边，那么我们称 ZP 中的顶点是约简的。注意，Z 中的顶点总是约简的，但 N 和 P 中的顶点则未必如此。如果所有顶点都是约简的，我们就称这个博弈是约简的。注意，在约简的博弈中，从 ZN 出发，Min 有一种策略可以确保只经过 ≤ 0 边，同时永远停留在 ZN 中（特别是，ZN 是 Max 的陷阱集）。因此，在约简的博弈中，ZN 中顶点的平均收益值为 < 0；同理，ZP 中顶点的值为 > 0。注意，只要 ZN 或 ZP 为空（等价地，N 或 P 为空），博弈就是约简的；如果 ZN 为空，我们称该博弈是正向约简的，否则称其为负向约简的。给定一个顶点子集 X，如果 $G \cap X$ 是约简的（分别为正向约简的或负向约简的），我们就称该博弈在 X 上是约简的（分别为正向约简的或负向约简的）。

势函数是一个映射 $\phi : V \to \mathbb{Z}$。相应的势函数约简将博弈G 变为 G_ϕ，方法是将每个权重 $w(vv')$ 替换为其修正权重 $w_\phi(vv')$，定义如下：$w_\phi(vv') = w(vv') + w(v') - w(v)$。请注意，环的权重不会改变（特别是，(1) 平均收益值和获胜区域保持不变，且 (2) 在 G_ϕ 中仍不存在权重为零的环）；并且请注意，当且仅当路径剖面在修正权重下也趋于无穷（或负无穷）时，它才会趋于无穷（或负无穷）。当 G 的含义可从上下文中明确时，我们将 N_ϕ 简写为 N_{G_ϕ}（同理，P、Z、ZN、ZP 也是如此）。若势函数 ϕ 使 G_ϕ 约简，则称其为约化势；若它在顶点子集 X 上使 G_ϕ 约简，则称其为在 X 上的约化势；正向约简和负向约简的情形也类似。

▶**定理 3** ([9, 11])。 每个博弈都存在一个约化势 ϕ。此外，得到的划分为 ZN_ϕ 和 ZP_ϕ，且与约化势 ϕ 无关。

根据上述讨论，所得划分显然与均值收益获胜区域一致（尤其是，它与 ϕ 无关）。约化势的存在性也很容易证明：例如，可以将平均收益值为 < 0 的顶点的（正的、有限的）上确界值，以及平均收益值为 > 0 的顶点的（负的、有限的）下确界值，分别赋给这些顶点，从而定义一个约化势。正如前面所提到的，我们的算法不一定会计算出这个势函数。

3 算法介绍

我们首先尝试解释算法背后的一些直觉；对读者来说，跳过接下来的部分，直接看算法更形式化的描述或许反而更容易理解。该描述大约只有一页长（或者，也许有些读者直接阅读第 5 节中的伪代码会更容易理解）。

算法概述

让我们把注意力转向顶点集 N。这些顶点 v 是 Min 能够确保第一条被访问的边权重为 < 0 的顶点。在最好的情况下（对 Min 而言），她甚至可以确保从 v 开始，剖面先经过一条负权重的边，此后再也不会取到 > 0 值；此时 v 的上确界值为 0（因此其平均收益值为 < 0，但从现在起我们不再提及平均收益值）。为便于说明，我们称 N 中这样的顶点为强负顶点。

现在反过来观察，只要 Min 的获胜区域非空，就存在一个强负顶点。这是因为，对于一个 Min获胜的顶点 v，如果双方都采取最优策略，那么达到剖面峰值的顶点 v' 必定是强负的，因为剖面此后永远不会超过该峰值。

当然， N 中并非所有顶点都是强负的（甚至可能一个也没有）。不过，作为初步近似（这对 Min 有利），算法会先假设 N 中的所有顶点都是强负的。在此假设下，我们可以合理地认为，或许能通过从N回溯，以某种方式计算出上确界值；事实上，这相当于计算上确界Σ^N 值，这将是算法首先关注的内容。（请注意，此处我们不对称地选择偏向 N 而非 P，但暂且忽略这一点。）因此，我们维护一个顶点集合 F，其中的顶点已计算出上确界Σ^N值；初始时，我们令 F为 N，其上确界Σ^N值为 0。

(∗) 回溯的第一步相当直接：很容易计算所有边都指向 F的顶点的上确界Σ^N值。递归地执行这一过程，我们可以求出所有路径都会经过 F的顶点的上确界Σ^N值，并将它们加入 F。

接下来的部分更棘手。注意，移除 F 后会留下一个子博弈，我们称之为 H。我们在 H 上递归调用算法，它会将 H 划分为其获胜区域 H^- 和 H^+，并为 H 给出一个约化势 ϕ_H （见图1）。现在，目标是从 H 中找到一个顶点，并确定其上确界值Σ^N 。

为找到这样的顶点，我们关注那些有办法逃离 H的顶点，也就是 H^+ 中的 Min顶点和 H^- 中的 Max顶点，且这些顶点都有边通向 F。因此，目标是找到一个对上确界Σ^N而言最优的逃逸。这里的关键洞见是，约化势 ϕ_H 以某种方式拉平了 H中的顶点，从而可以比较不同的逃逸选项，并确定对于某个顶点 v，控制该顶点的玩家最好逃逸到 F。

第一种情况是 H^+ 非空。如果不存在从$H^+ \cap V_{\text{Min}}$ 到 F 的边（图1中的红色边），那么 H^+ 中的顶点在G 中的上确界值$_N$为 ∞，因此其上确界值也为 ∞；此外， ϕ_H 为这些顶点提供了一个约化势。因此，通过标准回溯步骤，我们计算出集合 $A \supseteq H^+$，它包含 G 中那些 Max 能确保对局从 A到达 H^+ 的顶点，并同时得到 A上的一个约化势。随后，我们可以安全地从博弈中移除 A （引理4对此给出了证明），并在 $G \setminus A$上递归处理。

现在假设存在从 $H^+ \cap V_{\text{Min}}$ 指向 F的边。在这种情况下，显然 Min 应当从 H^+逃逸，因为这样会得到有限的 sup$_N$-值，而留在 H^+ 则会为她带来

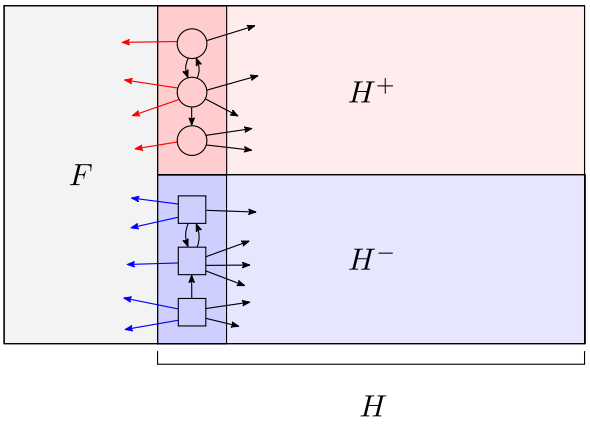


图1 情形示意图。圆圈为Min顶点，方块为Max顶点。红色边是从 H^+ 出发的逃逸边，蓝色边是从 H^- 出发的逃逸边。

无限的 $\sup_N$-值。利用约化势 ϕ_H，我们可以确定最佳的逃逸边 vv'（其中 $v \in H^+ \cap V_{\text{Min}}$ 和 $v' \in F$），结果表明，该边使 $\text{minimising} w(vv') + \sup\Sigma^N(v') - \phi_H(v)$ 最小（见引理5a）。因此，我们可以将 v添加到 F，并继续回溯，也就是说，返回到（$*$）。

在 H^+ 为空（而 H^- 不为空）的情形下，我们将看到必然存在从 $H^- \cap V_{\text{Max}}$ 到 F的若干边（图1中的蓝色边），并且从 $H^- \cap V_{\text{Max}}$到 F的最优逃逸边 vv'是使 $w(vv') + \sup\Sigma^N(v') - \phi_H(v)$最大化的那一条（见引理 5b）。因此我们按先前的方式进行，将 v 加入 F 并返回到(*)。

按此方式，在对更小的博弈总共进行 $\leq n$ 次递归调用之后，我们要么在(**)处退出循环，此时我们已经约简了博弈中Max获胜的一部分 A，并可安全地对 $G \setminus A$进行递归，要么我们已经成功计算出 G上的$\sup\Sigma^N$-值，且它们是有限的。

在此阶段，由于 $\sup\Sigma^N$-值为 sup-值提供了一个下界，我们可以重新审视关于强负顶点的假设。例如，如果 N 中的一个 Max 顶点 v 有一条 -1 边指向某个 $\sup\Sigma^N$ 值为 2 的顶点，那么显然 v 不可能是强负的。一个简便的方法是直接考虑这样一个势函数：为每个顶点赋予其 $\sup\Sigma^N$-值。这样做可能会使 N 中的一些顶点变为 P 或 Z，同时使 N 外的所有顶点变为 Z。（更形式化地说，如果 ϕ 表示该势函数，不难看出，N_ϕ 和 P_ϕ 包含于 N_G 中（引理7）。）因此，要么 $|N|+|P|$缩小，要么 N 和 P 中有一个变为空集；在后一种情况下，我们就完成了，因为此时博弈已经被约简。

回顾一下，我们任意选择了关注 N 并计算 $\sup\Sigma^N$，但也可以改为关注 P，并使用对偶算法计算 $\inf\Sigma^P$。自然地，这将得到对称的结果：要么我们找到一个对 Min获胜的部分（以及该部分的约化势），要么我们找到一个势函数 ϕ（即 $\phi = \inf\Sigma^P$），使得 N_ϕ 中的 P_ϕ 包含于 P 中。

我们将选择在 $|N| \leq |P|$ 时计算 $\sup\Sigma^N$，否则计算 $\inf\Sigma^P$。这一条件分支应用于每次递归调用，这正是该算法具有对称性的原因。

算法的形式化描述

现在，我们以更简洁的风格重复上述介绍。第 5 节给出了更为正式的伪代码。

1. 计算区域 N、Z、P、ZN 和 ZP，并检查博弈是否为约简的。若是，则停止。 2. 选择 N 和 P 中较小的一个；这里为明确起见，假设 $|N| \leq |P|$（否则，将所有内容对偶化）。接下来的步骤旨在计算 G 上的 $\sup\Sigma^N$，以及在 $\sup\Sigma^N$ 值为无穷大的顶点上构造一个约化势。在以下步骤中，我们将不断更新顶点集合 F，使得对于 F 中的顶点，$\sup\Sigma^N$ 的值已知。 3. 我们可以先将 N 加入 F，因为在 N 上 $\sup\Sigma^N$ 为零。 4. 接下来处理集合 S，其中的顶点不属于 F，且所有路径都通向 F：通过回溯即可轻松计算它们的 $\sup\Sigma^N$ 值，然后将它们加入 F。 5. 令 H 为从 G 中移除 F 内所有顶点后得到的子博弈（注意，由于上一步，这确实是一个子博弈）。在 H 上递归调用算法，从而得到 H 的约化势 ϕ_H，以及相应的获胜区域 H^- 和 H^+，分别属于 Min 和 Max。这里分几种情况。 a. 若 H^+ 非空。 i. 若存在一条从 $H^+ \cap V_{\text{Min}}$ 到 F 的边。令 vv' 为使 $w(vv') + \sup\Sigma^N(v') - \phi_H(v)$ 最小的这条边。我们断言，$\sup_N(v) = \sup_N(v') + w(vv')$（见引理 5a），因此可以将 v 加入 F，然后返回步骤 4。 ii. 若不存在从 $H^+ \cap V_{\text{Min}}$ 到 F 的边。此时，Max 可以在 H^+ 中采取最优对局，同时确保博弈始终留在其中，因此 H^+ 上的上确界值为 ∞：我们找到了 G 中属于 Max 的一部分获胜区域。通过标准回溯，我们计算集合 $A \supseteq H^+$，其中包含在 G 中 Max 能确保博弈到达 H^+ 的顶点。注意，ϕ_H 在 H^+ 上构成约化势，并且在回溯步骤中很容易将其扩展为 A 上的约化势。因此，我们可以在 $G \setminus A$ 上递归（见引理4）；显然，这确实是一个子博弈。 b. 否则，H^+ 为空。 i. 若 H^- 非空，则存在一条从 $H^- \cap V_{\text{Max}}$ 指向 F 的边（见引理 5b 的第一部分）。令 vv' 为使 $w(vv') + \sup\Sigma^N(v') - \phi_H(v)$ 最大的这条边。我们断言，$\sup\Sigma^N(v) = \sup\Sigma^N(v') + w(vv')$（见引理 5b 的第二部分），因此可以将 v 加入 F，然后返回步骤 4。 ii. 否则，H 为空。因此，我们已完成对 G 上 $\sup\Sigma^N$ 值的计算，且这些值均为有限值。令 G' 为在 G 上执行相应的势函数约简后得到的集合；注意，$N_{G'}$ 和 $P_{G'}$ 均包含于 N_G 中（见引理7），然后在 G' 上递归。

由于每次发生 5(a)i 或 5(b)i 时，F 的规模都会增大，因此经过对严格更小的博弈进行线性次递归调用后，必定会到达 5(a)ii 或 5(b)ii。若到达 5(a)ii，则下一次递归调用针对的是一个严格更小的博弈。若到达 5(b)ii，则（见引理7）下一次递归调用针对博弈 G'；此时 $N_{G'}$ 或 $P_{G'}$ 为空，于是 G' 是约简的，算法随之停止；或者 $N_{G'}$ 和 $P_{G'}$ 的规模都小于 N_G 和 P_G 中较小者的规模，从而保证算法终止。

我们还在第 5 节给出了更正式的伪代码。

4 主要引理的证明

接下来我们证明该算法所用的三个主要引理。第一个引理说明，在某些假设下如何将约化势函数拼接起来，从而证明步骤 5(a)ii 中递归调用的合理性。

▶**引理4.** 设 G 是一个博弈， A 是 *Min* 的陷阱集， ϕ_A 是定义在 A上的正向约化势函数， $G' = G \setminus A$ 是一个假定为 *Max* 的陷阱集的集合， ϕ' 是定义在 G'上的约化势。设 δ 是一个常数，使得对于每个 $v' \in G'$ 和 $a \in A$ ，我们都有

$$\delta \geq -w(v'a) - \phi_A(a) + \phi'(v'),$$

并设 $\phi = (\phi_A + \delta) \uplus \phi'$ 是一个势函数，它在 A 上与 $\phi_A + \delta$ 一致，在 G'上与 ϕ'一致。则 ϕ 是 G的一个约化势。

证明。 首先，注意到 G'_{ϕ_A} 和 $G'_{\phi_A+\delta}$ 是相同的博弈，因此 $G'_{\phi_A+\delta}$ 是约简的，且两个博弈中的获胜区域相同。我们断言 $ZN_\phi = ZN_{\phi'}$ 和 $ZP_\phi = ZP_{\phi'} \cup A$。我们首先观察到，在 G_ϕ中，Max 有一个策略，从 $ZP_{\phi'} \cup A$出发，确保所看到的第一条边是 ≥ 0 并朝向 $ZP_{\phi'} \cup A$。

事实上，从 A 出发，Max 甚至可以强制走一条通向 A的 ≥ 0 边，这是因为 ϕ_A （ $\phi_A + \delta$也一样）是正向递减的， A 是 Min 的陷阱集，并且 ϕ 在 A上与 $\phi_A + \delta$一致。现在从 $ZP_{\phi'}$出发，Max 有一种策略，要么走到一条通向 $ZP_{\phi'}$ 、其 ϕ修正权重为 ≥ 0的边（因为 ϕ' 在 G' 中是递减的，且与 ϕ一致），要么走到一条通向A的边 $v'a$ 。后一种情况下，边 $v'a$ 的 ϕ 修正权重为 $w(v'a) + \phi_A(a) + \delta - \phi'(v')$，根据我们对 δ的假设，这一权重是 ≥ 0 。

同样，我们也应证明，从 $ZN_{\phi'}$出发，Min 有一种策略，能够确保首先走到一条通向 $ZN_{\phi'}$、权重为 ≤ 0 的 ϕ边。显然，这是因为 G' 是 Max 的陷阱集， ϕ' 在 G'上是递减的，并且 ϕ 与 ϕ' 在 G'上一致。◀

下一条关键引理说明，在给定约化势时如何得到最优逃逸边。

▶**引理 5.** 设 G 为一个博弈，设 F 为包含 N 的顶点集，且使得 $\sup\Sigma^N$在 F上有限，设 H 由从 G 中移除 F 得到，并假设 H 是一个子博弈。设 ϕ_H 为 H 的约化势，并设 H^+ 和 H^- 表示获胜区域。

a. 假设存在一条从 $H^+ \cap V_{\text{Min}}$ 到 F 的边，并设 vv' 为使$w(vv') + sup\Sigma^N(v') - \phi_H(v)$最小的这样一条边。则 $sup\Sigma^N(v) = sup\Sigma^N(v') + w(vv')$。*b.* 假设 H^+ 为空，而 H^- 非空。则存在一条从 $v \in H^- \cap V_{\text{Max}}$ 到 $v' \in F$的边 vv' 。设 vv' 为使 $w(vv') + sup\Sigma^N(v') - \phi_H(v)$最大的这样一条边。则 $sup\Sigma^N(v) = sup\Sigma^N(v') + w(vv')$。

引理5a 的证明。 由于 Min 可以选择边 vv' ，然后从 v'开始采取最优策略，因此我们有

$$\sup\Sigma^N(v) \leq w(vv') + \sup\Sigma^N(v'),$$

因为 $w(vv') \geq 0$ (这是因为 $v \notin N$ 由于 $N \subseteq F$)，因此右侧为≥ 0，所以引理 2 适用。因此，我们着重考虑另一个不等式。我们让 Max 在 H^+ 上采用这样的策略：只选择通向 H^+ 且具有 ϕ_H-修正权重 ≥ 0的边；在 F 上，他则采取最优策略。从 v 出发，这会迫使对局始终留在 $F \cup H^+$ 内，因此我们不必定义 Max 在 H^- 上的策略。我们断言，该策略从 v出发能达到所需的值。

确实，如果对局始终留在 H^+ 内，那么经过的边都具有 ϕ_H-修正权重 ≥ 0；由于不存在零权环，修正权重序列趋于 ∞，因此原始权重也同样如此。否则，对局会一直留在 H^+ 内，直到经过一条从 $H^+ \cap V_{\text{Min}}$ 到 F 的边 uu'。此时，从 v 到 u 的博弈过程前缀具有 ϕ_H-修正权重 ≥ 0，因此我们得到

$$\begin{array}{rcl} \text{value of play from } v & \geq & \text{value of prefix } + w(uu') + \sup\Sigma^N(u') \\ & = & \text{modified weight of prefix} + \phi_H(v) - \phi_H(u) + w(uu') + \sup\Sigma^N(u') \\ & \geq & \phi_H(v) + w(uu') + \sup\Sigma^N(u') - \phi_H(u) \\ & \geq & \phi_H(v) + w(vv') + \sup\Sigma^N(v') - \phi_H(v) = w(vv') + \sup\Sigma^N(v'), \end{array}$$

如前所述。 ◀

接下来给出一个技术性命题，用于证明引理 5b。

▶**引理 6**。 设 H 为一个博弈，且 *Max* 的获胜区域为空，并设 ϕ 为一个约化势。存在一个顶点 N_H，使得 $\phi(v)$最小。

证明。 由于 ϕ 是约化势，且 Max 的获胜区域为空，因此有 $P_\phi = \varnothing$，也就是说，每个 Min 顶点都有一条修正权重为 ≤ 0 的出边，且每条从 Max 顶点出发的边的修正权重均为 ≤ 0。反设顶点 v 在 N_H 中不具有最小的 $\phi(v)$ 值，并设 u 为一个满足以下条件的顶点：$\phi(u)$ 最小。我们按如下方式进行。若 $u \in V_{\text{Min}}$。则令 uu' 为一条 ϕ-修正权重为 ≤ 0的边。于是有

$$w(uu') = \underbrace{w_\phi(uu')}_{\leq 0} + \underbrace{\phi(u) - \phi(u')}_{\leq 0} \leq 0. \qquad (*)$$

若 $w(uu') < 0$，则 $u \in N_G$，这就产生了矛盾。否则，$w(uu') = 0$，因此 $\phi(u') = \phi(u)$，并从 u'继续这一过程。

如果 $u \in V_{\text{Max}}$。那么所有边 uu' 都具有 ϕ-修正权重 ≤ 0，并满足 (*)。如果它们都满足 $w(uu') < 0$，我们就有 $u \in N_G$，问题得证。否则，存在一条边 uu'，满足 $w(uu') = 0$，因此 $\phi(u') = \phi(u)$，我们从 u'继续。

上述过程必定会停止，因为不存在零权环，由此引理得证。 ◀

现在我们可以开始下一个证明了。

引理 5b 的证明。我们首先证明如下断言：若 H^+ 为空而 H^- 非空，则存在一条从 $H^+ \cap V_{\text{Max}}$ 到 F的边。事实上，我们将证明一个稍强的断言，稍后会用到：存在一个顶点 $v \in V_{\text{Max}} \cap N_H$（即从 v 到 $H = H^+$ 的所有边都是 < 0），并且有一条指向 F 的 ≥ 0 边。确实，H^- 非空，因此 N_H 非空；又因为 $N_G \subseteq F$，H 中的所有 Min 顶点都只有 ≥ 0 出边，因此不可能属于 N_H。所以N_H 恰好由那些指向 H 的边全都为 < 0的 Max 顶点组成。此外，N_H 中的一个顶点必定有一条 ≥ 0 出边，否则它就会属于 N_G，而这条边只能指向 F。

现在我们证明这一断言：如果 vv'在从 $H \cap V_{\text{Max}}$到 F的过程中最大化了 $w(vv') + \sup\Sigma^N(v') - \phi_H(v)$，那么 $\sup\Sigma^N(v) = \sup\Sigma^N(v') + w(vv')$。由于 Max 可以先走边 vv'，然后从 v'开始采取最优策略，因此有 $\sup\Sigma^N(v) \geq \sup\Sigma^N(v') + w(vv')$：如果右侧为 ≤ 0，这一点显而易见；否则则由引理 2 得出。因此，我们着重证明另一个不等式。

现在我们将引理 6 应用于 H：存在一个顶点 u，使得 $\phi_H(u)$ 在 H 的所有顶点中取最小值。回想一下，根据引理中的极大性假设，对于所有满足 $u' \in F$ 的边 uu'，都有

$$w(vv') + \sup\Sigma^N(v') - \phi_H(v) \geq w(uu') + \sup\Sigma^N(u') - \phi_H(u),$$

并且存在一条边 uu'，使得 $w(uu') \geq 0$（而且始终有 $\sup\Sigma^N(u') \geq 0$），因此我们有

$$w(vv') + \sup\Sigma^N(v') - \phi_H(v) \geq -\phi_H(u). \qquad (*)$$

我们让 Min 在 H^- 上采用这样的策略：只选择通向 H^- 且修正权重为 ϕ_H 的边；在 F 上，她则采取最优策略。我们断言，从 v 出发，该策略能达到所需的值。

事实上，只要对局仍停留在 H^- 中，经过的就只有修正权重为 ϕ_H 的边，因此，对于以某个顶点 $v' \in H^-$ 结束的博弈过程前缀，我们有

$$\begin{aligned}\text{weight of the prefix} &= \underbrace{\text{modified weight of the prefix}}_{\leq 0} + \phi_H(v) \underbrace{-\phi_H(v')}_{\leq -\phi_H(u)} \\ &\leq \phi_H(v) - \phi_H(u) \underbrace{\leq}_{\text{by } (*)} w(vv') + \sup\Sigma^N(v'),\end{aligned}$$

因此，对局值为 $\leq w(vv') + \sup\Sigma^N(v')$，如所需。（在本证明的余下部分，我们将释放符号 u。）

然而，如果对局从 H^- 离开并前往 F，那么它必定经过一条边 uu'，其中 u 属于 V_{Max}，因此对局的形式为：

$$\underbrace{v = h_0 \to h_1 \to \ldots \to h_n = u}_{\text{inside } H^-} \to \underbrace{u'}_{\text{in } F} \to \ldots,$$

其中，从 u' 开始的对局后缀遵循 Min 的策略；该策略从 u' 出发是最优的，因此其值为 $\leq \sup\Sigma^N(u')$。因此，对局值小于或等于

$$\max\big[\max_i(\text{weight of prefix up to } h_i), (\text{weight of prefix up to } u') + \sup\Sigma^N(u')\big],$$

并且我们在上文已经论证，直到 h_i 的前缀权重为 $\leq w(vv') + \sup\Sigma^N(v')$。为了给另一项建立上界，我们采用与引理5a类似的方法：直到 u 的博弈过程前缀具有 ϕ_H-修正权重 ≤ 0，因此我们得到

$$\begin{aligned}&\text{weight of prefix up to } u' + \sup\Sigma^N(u') \\ \leq\ &\text{weight of prefix up to } u + w(uu') + \sup\Sigma^N(u') \\ =\ &\text{modified weight of prefix up to } u + \phi_H(v) - \phi_H(u) + w(uu') + \sup\Sigma^N(u') \\ \leq\ &\phi_H(v) + w(uu') + \sup\Sigma^N(u') - \phi_H(u) \\ \leq\ &\phi_H(v) + w(vv') + \sup\Sigma^N(v') - \phi_H(v) = w(vv') + \sup\Sigma^N(v'),\end{aligned}$$

由此完成证明。 ◄

这个最后的简单引理证明了算法会终止。

►**引理7**。设 G 是一个博弈，且其上的 $\sup\Sigma^N$ 有限，并设 G' 由应用相应的势函数约简得到。那么 $N_{G'}$ 和 $P_{G'}$ 都包含在 N 中。

证明。设 v 是 G' 中不属于 N 的 Min 顶点。那么，对于每条出边 vv'，都有 $\sup\Sigma^N(v) \geq \sup\Sigma^N(v') + w(vv')$，且至少有一条出边使得这里取等号。因此，$\sup\Sigma^N$-修正权重全都为 ≥ 0，且其中至少一个为 $= 0$，所以 v 属于 $Z_{G'}$。Max 顶点的证明类似。◄

5 伪代码

下面给出我们算法的一些伪代码。假设所有博弈都没有零权环。给定映射 $f\colon A \to X$ 和 $g\colon B \to X$，其中 A 和 B 是互不相交的集合，我们用 $f \uplus g$ 表示从 $A \cup B$ 到 X 的映射，该映射在 A 上与 f 一致，在 B上与 g 一致。

算法 1 主递归过程

```
1  函数 ReduceGame(G)
       输入：博弈 G = (V, E, w)
       输出：一个元组 (G⁻, G⁺, φ)，其中 φ 是约化势，G⁻, G⁺ 是
             相应的获胜区域
2      (N, P, ZN, ZP) ← ComputeZones(G)
3      若 Reduced(G, N, P, ZN, ZP) 则
4          返回 (ZN,ZP,0)
5      若 |N| ≤ |P| 则
6          F ← N
7          对于所有 v, sup^N(v) ← 0
8          当 True 时执行
9              BacktrackAllPaths(G, F, sup^N)
10             H ← G| F
11             (H⁻, H⁺, φ_H) ← ReduceGame(H)
12             如果 H⁺ ≠ ∅ 则
13                 如果存在一条从 H⁺ ∩ V_Min 到 F 的边则
14                     令 vv' 为使 w(vv') + sup^N(v') − φ_H(v)最小化的这样一条边
15                     sup^N(v) ← w(vv') + sup^N(v')
16                     F ← F ∪{v}
17                     继续
18                 else
19                     (A, φ_A) ← AttractMaxAndReduce(G, H⁺, φ_H)
20                     G' = G| A
21                     (G'⁺, G'⁻, φ') ← ReduceGame(G')
22                     δ = −min_{v∈G',v'∈A} w(vv') − min_{v'∈A} φ_A(v') + max_{v∈G'} φ'(v)
23                     返回 (G'⁻, G'⁺ ∪ A, (φ_A + δ) ⊎ φ')
24             else
25                 如果 H⁻ ≠ ∅ 则
26                     令 vv' 为一条使 w(vv') + sup^N(v') − φ_H(v) 最大化的边，来自
                       从 H ∩ V_Max 到 F 中的顶点
27                     上确界^N(v) ← w(vv') + 上确界^N(v')
28                     F ← F ∪{v}
29                     继续
30                 else
31                     G' ← PotentialReduction(G, 上确界^N)
32                     返回 ReduceGame (G')
33     else
           /* Proceed symmetrically                                  */
```

12 用于均值收益博弈的对称递归算法

下面给出三个（类似的）回溯子程序的伪代码，分别用于确定一个博弈是否为约简的，以及计算一个势函数约简。它们的运行时间均为 $O(m)$。记 $v^{-1}E = \{v' \in V \mid vv' \in E\}$ 和 $Ev^{-1} = \{v' \in V \mid v'v \in E\}$。

```
函数 ComputeZones(G)
    输入：一个博弈 G = (V, E, w)
    输出：一个元组 (N, P, ZN, ZP)，分别对应于该博弈。
    N ← {v ∈ V_Min | ∃v' ∈ v^{-1}E, w(vv') < 0} ∪ {v ∈ V_Max | ∀v' ∈ v^{-1}E, w(vv') < 0}
    P ← {v ∈ V_Min | ∀v' ∈ v^{-1}E, w(vv') > 0} ∪ {v ∈ V_Max | ∃v' ∈ v^{-1}E, w(vv') > 0}
    Z ← V \ (N ∪ P)
    对所有 v ∈ Z ∩ V_Max 执行
        esc[v] ← |{v' ∈ v^{-1}E | w(vv') = 0}|
    ZN ← ∅
    Q ← N
    当 Q ≠ ∅ 时执行
        取一些 v ∈ Q
        Q ← Q \ {v}
        ZN ← ZN ∪ {v}
        对所有 v' ∈ Ev^{-1} \ (P ∪ ZN) 执行
            如果 v' ∈ V_Min 且 w(v'v) = 0 则
                Q ← Q ∪ {v'}
            else
                转义(v') ← 转义(v') − 1
                如果 转义(v') = 0 则
                    Q ← Q ∪ {v'}
    返回 (N, P, ZN, V \ ZN)
```

```
函数 Reduced(G, N, P, ZN, ZP)输入：博弈 G = (V, E, w) 及其区域 N、P、ZN和 ZP输出：如果 G 已约简，则为 True，否则为 False。
    b ← True
    b ← b和 [∀v ∈ V_Min ∩ N, ∃v' ∈ v^{-1}E ∩ ZN, w(vv') ≤ 0]
    b ← b和 [∀v ∈ V_Max ∩ N, ∀v' ∈ v^{-1}E, v' ∈ ZN]
    b ← b和 [∀v ∈ V_Max ∩ P, ∃v' ∈ v^{-1}E ∩ ZP, w(vv') ≥ 0]
    b ← b以及 7 返回 b
```

```
函数 BacktrackAllPaths(G, F, φ)
    输入：一个博弈 G = (V, E, w)，一个顶点子集 F，满足 N_G ⊆ F，以及一个
          势函数 φ，它与 F 上的 sup Σ^{N_G} 一致。
    输出：就地更新 F 和 φ，以纳入路径均通向
          F 到 F 的顶点，使得 φ 仍与 F 上的 sup Σ^{N_G} 一致。
    对所有 v ∈ V \ F 执行
        esc[v] ← |{v' ∈ v^{-1}E | v' ∉ F}|
    Q ← {v ∈ V \ F | 转义(v) = 0}
    当 Q ≠ ∅ 时执行
        取某个 v ∈ Q
        Q ← Q \ {v}
        F ← F ∪ {v}
        如果 v ∈ V_Min 那么
            φ(v) = Min_{v'∈v^{-1}E} w(vv') + φ(v')
        else
            φ(v) = max_{v'∈v^{-1}E} w(vv') + φ(v')
        对于所有 v' ∈ Ev^{-1} \ F 执行
            esc(v') ← esc(v') − 1
            如果 esc(v') = 0 则
                Q ← Q ∪ {v'}
    返回
```

```
1 函数 AttractMaxAndReduce(G, T, φ_T)
      输入： 一个博弈 G = (V, E, w)，一个顶点目标集 T 以及一个势函数 φ_T
             该势函数在 T 上是正向递减的。
      输出： 一个元组 (A, φ_A)，其中 A ⊇ T 是一个顶点集，使得 Max 可以
             迫使博弈到达 T，并且 φ_A 在 A 上是正向递减的。
2     A ← T
3     φ_A ← φ_T
4     对所有 v ∈ (V \ T) ∩ V_Min 执行
5         esc[v] ← |{v' ∈ v^{-1}E | v' ∉ T}|
6     Q ← {v ∈ (V \ T) ∩ V_Min | esc(v) = 0} ∪ {v ∈ (V \ T) ∩ V_Max | ∃z' ∈ v^{-1}E ∩ T}
7     当 Q ≠ ∅ 时执行
8         取某个 v ∈ Q
9         Q ← Q \ {v}
10        A ← A ∪ {v}
11        如果 v ∈ V_Min 则
12            φ(v) = Min_{v'∈v^{-1}E} w(vv') + φ_A(v')
13        else
14            取一些 v' ∈ v^{-1}E ∩ A
15            令 φ_A(v) = w(vv') + φ_A(v')
16        对于所有 v' ∈ Ev^{-1}，执行
17            如果 v' ∈ V_Min 则
18                转义(v') ← 转义(v') − 1
19                如果 转义(v') = 0 则
20                    Q ← Q ∪ {v'}
21            else
22                Q ← Q ∪ {v'}
23    返回 (A, φ_A)
```

```
1 函数 AttractMinAndReduce(G, T, φ_T)
      /* Symmetric                                                    */
```

```
1 函数 PotentialReduction (G, φ)
      输入： 博弈 G = (V, E, w) 和势函数 φ。
      输出： 通过执行相应的操作得到的博弈 G' = (V, E, w_φ)
             势函数约简。
2     对于 vv' ∈ E，执行
3         w_φ(vv') ← w(vv') + φ(v') = φ(v);
4     返回 (V, E, w_φ)
```

6 优化及变体

我们为该算法提出了一些优化与变体。

6.1 优化：更好的 F 初始化

给定一个博弈 G，考虑顶点集 S，从该集合中的顶点出发，Min 可以确保在出现权重为 >0 的边之前访问 N。换言之， S 是最大的顶点集，从 S 出发，Min 可以确保要么永远只出现权重为 ≤ 0 的边，要么这些边会一直出现，直到到达 N。（注意， $N \subseteq S$。）显然， S 可以在多项式时间内计算出来（通过与上述过程类似的回溯过程），此外，s up_{Σ^N} 在 S 上取值 0。第一个优化将算法 1 中的第 6 行替换为 $F \leftarrow$ `BacktrackInitialisationMin`(G, N, P)，其中用于计算 S 的这一子过程可以按如下方式实现。

```
1  函数 BacktrackInitialisationMin(G, N, P)
       输入： 一个博弈 G = (V, E, w) 及其区域 N, P。
       输出： 顶点集 S，从这些顶点出发，Min 可以确保在……之前看到 N
                任意 > 0 边。
2      A ← ∅
3      对所有 v ∈ V_Min \ (P ∪ N) 执行
4          转义[v] ← |{v' ∈ v^{-1}E | v' ∉ P 且 w(vv') = 0}|
5      Q ← P
6      当 Q ≠ ∅ 执行
7          取一些 v ∈ Q
8          Q ← Q \ {v}
9          A ← A ∪ {v}
10         对所有 v' ∈ Ev^{-1} \ N 执行
11             如果 v' ∈ V_Min 且 w(vv') = 0 那么
12                 esc(v') ← esc(v') − 1
13                 如果 esc(v') = 0 那么
14                     Q ← Q ∪ {v'}
15             else
16                 Q ← Q ∪ {v'}
17     返回 V \ A
```

自然地，在对偶情形中也采用对称的过程 `BacktrackInitialisationMax`。

6.2 优化：一次固定多个顶点

接下来的优化更为复杂。其思路是通过增加一个回溯过程来改进第 14–16 行和第 26–28 行，以便一次可能将多个顶点加入 F，而不是仅加入一个。这个额外的多项式时间计算有望减少许多（代价高昂的）递归调用。

更准确地说，我们不执行第 14–16 行，而是按如下方式进行。假设存在一条从 $H^+ \cap V_{\mathrm{Min}}$ 到 F 的边，并令 m 表示对于 vv' 这样的边， $w(vv') + \mathrm{sup}_{\Sigma^N}(v') - \phi_H(v)$ 的最小值。若边 vv' 从 H^+ 到 F，且 $w(vv') + \mathrm{sup}_{\Sigma^N}(v') - \phi_H(v) \leq m$，则称边 vv' 为一条良好逃逸边（注意，从 $H^+ \cap V_{\mathrm{Max}}$ 到 F 的边也可能是良好逃逸边）。接着，我们计算顶点集 S，它由 H^+ 中满足以下条件的顶点组成：在 G 中，Min 能确保在遇到任何修正权重为 >0 的边或离开 H^+ 的边之前，先看到一条良好逃逸边。换言之， S 是最大的顶点集合 $\subseteq H^+$，使得从 S 出发，Min

能确保对局在仍处于H^+时只经过修正权重为 ≤ 0 的边，并且要么永远停留在 H^+ 中，要么通过一条良好逃逸边离开。于是，对于 S中的每个顶点 v，下面的引理断言 $\sup\Sigma^N(v) = m + \phi_H(v)$，因此我们可以固定 $\sup\Sigma^N(v)$，并将 S 中的所有顶点加入 F。对于第 26–28 行，我们采用类似的方法。

因此，我们确立了引理 5 的一个推广，内容如下。

▶**引理 8**。设 G 是一个博弈，设 F 是一个包含 N 的顶点集，并且上确界Σ^N 在 F 上有限。设 H 是从 G 中移除 F 后得到的，并假设 H 是一个子博弈。设 ϕ_H 是 H 的一个约化势，并令 H^+ 和 H^- 表示获胜区域。

a. 假设存在一条从 $H^+ \cap V_{\mathrm{Min}}$ 到 F 的边，并令 m 表示对于 vv' 中这样一条边，$w(vv') + \sup\Sigma^N(v') - \phi_H(v)$ 的最小值。若边 vv' 从 H^+ 指向 F，且 $w(vv') + \sup\Sigma^N(v') - \phi_H(v) \leq m$，则称其为良好逃逸边。令 S 为 H^+ 中这样一个顶点集：从其中的顶点出发，在 G 中，*Min* 可以确保在遇到任何具有 ϕ_H-修正权重 > 0 或离开 H^+ 的边之前，先遇到一条良好逃逸边。那么，对于 S 中的每个顶点 v，都有 $\sup\Sigma^N(v) = m + \phi_H(v)$。

b. 假设 H^+ 为空，而 H^- 不为空。那么存在一条 vv' 边从 $v \in H^- \cap V_{\mathrm{Max}}$到 $v' \in F$，并令 m表示对于 vv' 这样的边，$w(vv') + \sup\Sigma^N(v') - \phi_H(v)$ 的最大值。若一条边 vv' 从 H^- 到 F，且$w(vv') + \sup\Sigma^N(v') - \phi_H(v) \geq m$，则称其为一条良好逃逸边。令 S为 H^-中的顶点集：从这些顶点出发，在 G中，*Max* 可以确保在看到任何具有 ϕ_H-修正权重 < 0的边或离开 H^-的边之前，先看到一条良好逃逸边。那么，对于 S中的每个顶点 v，都有 $\sup\Sigma^N(v) = m + \phi_H(v)$。

证明。我们从第一条开始。首先证明，对于每个顶点 $v \in S$，都有 $\sup\Sigma^N(v) \leq m + \phi_H(v)$。考虑一个 Min 策略 σ，该策略确保在 S中，在看到具有 ϕ_H-修正权重 > 0 的任何边或离开 H^+ 且从F离开的任何边之前，先看到一条良好逃逸边；对于 $\sup\Sigma^N$而言，该策略是最优的。令 $\sup\Sigma^N_\sigma(v)$表示在假设 σ的情况下 v的值，即与 σ一致、从 v 出发的对局的上确界值；如果对于每个顶点 v，同时与 σ和 τ 一致的唯一对局的值都与 $\sup\Sigma^N_\sigma(v)$相同，则称 Max 策略 τ 是针对 σ 的最优策略。

现在我们证明以下断言：存在一个 Max 策略 τ，它对于σ 是最优的，并且对每个顶点 $v \in S$，都有 $\sup\Sigma^N_\sigma(v) = w(vv') + \sup\Sigma^N_\sigma(v')$，其中 $vv' = \tau(v)$（若 $v \in V_{\mathrm{Max}}$），而 $vv' = \sigma(v)$（若 $v \in V_{\mathrm{Min}}$）。注意，对于 $v \in V_{\mathrm{Min}} \cap S$，所需的等式成立（无论 τ如何），因为 $w(vv') \geq 0$（因为 $v \notin N$，而 $N \subseteq F$），并且由引理 2 可知。考虑一个顶点 $v \in V_{\mathrm{Max}} \cap S$。由于 $v \notin N$，我们知道它有一条出边 vv'，其权重为 ≥ 0，因此由引理 2 可得 $\sup\Sigma^N_\sigma(v) \geq w(vv') + \sup\Sigma^N_\sigma(v')$。

- 如果这条边是最优边，那么这里就是等式，因此我们令 $\tau(v) = vv'$，问题便得以解决。
- 否则，还存在另一条最优边 vv''，因此它满足

$$w(vv'') + \sup\Sigma^N_\sigma(v'') > w(vv') + \sup\Sigma^N_\sigma(v') \geq 0,$$

因此，由引理 2 可得 $\sup\Sigma^N_\sigma(v) = w(vv'') + \sup\Sigma^N_\sigma(v'')$，并令 $\tau(v) = vv''$。因此断言得证；我们用 τ 表示这样的 Max 策略。

考虑从给定顶点 $v \in S$ 出发、同时与 σ 和 τ 一致的唯一对局；我们需要证明其值为 $\leq m + \phi_H(v)$。

- 如果该对局永远停留在 S 中。注意，该对局是一个套索，即具有如下形式

$$v \to \ldots \to u \to \ldots \to u \to \ldots \to u \to \ldots.$$

由上述断言可得

$$\sup\Sigma^N_\sigma(u) = \text{ weight of the cycle based on } u + \sup\Sigma^N_\sigma(u).$$

由于不存在零权环，这意味着 $\sup\Sigma^N_\sigma(u) = +\infty$。然而，由于该对局与 σ 一致，且没有经过良好逃逸边，因此它只能经过修正权重为 ≤ 0 的边，其中权重为ϕ_H，所以其值不可能为 ∞。

- 否则，对局会离开 S；由于它与 σ一致，因此必定经由一条良好逃逸边离开，设该边为 uu'。根据上述断言，我们知道

$$\begin{aligned}\sup\Sigma^N_\sigma(v) &= \text{weight of prefix up to u } + w(uu') + \sup\Sigma^N_\sigma(u') \\ &= \text{modified weight of prefix up to u } + \phi_H(v) - \phi_H(u) + w(uu') + \sup\Sigma^N(u') \\ &\leq \phi_H(v) - \phi_H(u) + w(uu') + \sup\Sigma^N(u') \leq \phi_H(v) + m,\end{aligned}$$

因为前缀的修正权重为 ≤ 0 （根据 σ 的定义），且 $\sup\Sigma^N_\sigma(u') = \sup\Sigma^N(u')$ 成立，这是因为 σ 从 F 出发是最优的。

这就证明了对于每个 $v \in S$，都有 $\sup^N(v) \leq m + \phi_H(v)$。

为得到另一个不等式，固定一个顶点 $u \in V_{\text{Min}} \cap H^+$，它有一条良好逃逸边作为出边 uu'；根据引理5a，我们已经知道 $\sup\Sigma^N(u) = w(uu') + \sup\Sigma^N(u')$，而根据 m 的定义，这与 $\phi_H(u) + m$ 相同。考虑通过向 G 添加边而得到的博弈 [6] G'

$$u \xrightarrow{1/2+\phi_H(u)-\phi_H(v)} v.$$

注意， G' 也没有零权环（这就是上文使用 $1/2$ 而非 0 的原因）。由于新边的 ϕ_H-修正权重为 $1/2$，在 G' 中， H^+ 仍由 ϕ_H 正向约简，因此引理5a也适用于 G'。由于在 G' 中， G 的良好逃逸边 uu' 在 G' 中仍是一条良好逃逸边（我们没有添加任何逃逸边），引理表明它仍是 G' 中 $\sup\Sigma^N$ 的最优边。因此，在 G 和 G' 中，$\sup\Sigma^N$ 的值相同（因为最优策略不会使用这条新边），并且

$$\underbrace{\sup\Sigma^N(u)}_{\text{in } \mathbb{Z}} \leq 1/2 + \underbrace{\phi_H(u) - \phi_H(v) + \sup\Sigma^N(v)}_{\text{in } \mathbb{Z}},$$

因此

$$\sup\Sigma^N(u) \leq \phi_H(u) - \phi_H(v) + \sup\Sigma^N(v).$$

由此得到所需的界：

$$\sup\Sigma^N(v) \geq \sup\Sigma^N(u) + \phi_H(v) - \phi_H(u) = m + \phi_H(v),$$

至此完成第一条的证明。

现在我们聚焦于第二条。根据引理 5b，存在一条从 $H^- \cap V_{\text{Max}}$ 到 F的边。对于第二个陈述，我们先证明，对每个顶点 $v \in S$，都有 $\sup\Sigma^N(v) \geq m + \phi_H(v)$。考虑一个 Max 策略 τ，它保证在 S上，在看到任何权重为 < 0 的修正权重为 ϕ_H的边，或任何离开 H^-和 F的边之前，先看到一条良好逃逸边；对于 $\sup\Sigma^N$而言，这是最优的。考虑一个顶点 $v \in S$，以及一个从 v出发、与 τ一致的对局。如果对局始终留在 S中，那么根据 τ的定义，只会看到修正权重为 ≥ 0的边

[6] 为了本证明的目的，我们允许有理数权重。

，因此值为 ∞，因为不存在零权环。否则，对局会一直留在 S中，直到遇到一条良好逃逸边 uu'。那么，由于从 u' 出发时 τ 是最优的，我们得到

$$\begin{aligned} & \text{value of the play} \\ \geq\ & \text{weight of prefix up to u} + w(uu') + \sup\Sigma^N(u') \\ =\ & \underbrace{\text{modified weight of prefix up to u}}_{\geq 0} + \phi_H(v) \underbrace{-\phi_H(u) + w(uu') + \sup\Sigma^N(u')}_{\geq m} \\ \geq\ & \phi_H(v) + m. \end{aligned}$$

尚需证明，对于每个 $v \in S$，都有 $\sup^N(v) \leq m + \phi_H(v)$；对于这个不等式，我们采用与上一条相同的方法。固定一个顶点 $u \in V_{\text{Max}} \cap H^-$，它有一条良好逃逸边作为出边 uu'；我们已由引理 5b 知道，$\sup^N(u) = \phi_H(u) + m$。考虑由 G 添加边后得到的博弈 G'

$$u \xrightarrow{-1/2+\phi_H(u)-\phi_H(v)} v.$$

注意，G' 也没有零权环。由于新边的修正权重为 $\phi_H - 1/2$，在 G' 中，H^- 对 ϕ_H 仍是负向约简的，因此引理 5b 也适用于G'。由于在 G' 中，G 里的最优逃逸边 uu' 在 G' 中仍是最优逃逸边，引理表明它仍是 G' 中 $\sup\Sigma^N$ 的最优边。因此，在G 和 G' 中，$\sup\Sigma^N$ 的值相同，而且

$$\underbrace{\sup\Sigma^N(u)}_{\text{in } \mathbb{Z}} \geq -1/2 + \underbrace{\phi_H(u) - \phi_H(v) + \sup\Sigma^N(v)}_{\text{in } \mathbb{Z}},$$

因此

$$\sup\Sigma^N(u) \geq \phi_H(u) - \phi_H(v) + \sup\Sigma^N(v),$$

由此得到所需的界：

$$\sup\Sigma^N(v) \leq \sup\Sigma^N(u) + \phi_H(v) - \phi_H(u) = m + \phi_H(v),$$

证明至此完成。 ◀

我们省略了 S 的计算伪代码，因为它与例如前一次优化中的回溯类似。

6.3 变体：在上确界Σ^N 和下确界Σ^P之间进行选择

在第 5 行，我们根据是否满足$|N| \leq |P|$，选择计算 $\sup\Sigma^N$ 或 $\inf\Sigma^P$。这一选择不会影响算法的正确性或终止性；无论采用何种方式在两者之间进行选择（例如始终选择 $\sup\Sigma^N$），都能得到有效的算法。也可以设想其他选择策略，例如相反的策略：每当 $|N| \geq |P|$时，优先选择 $\sup\Sigma^N$ 而非 $\inf\Sigma^P$。假设采用第 6.1 节中的优化，一种或许有意义的选择策略是：每当 $|S_N| \geq |S_P|$时，优先选择 $\sup\Sigma^N$ 而非 $\inf\Sigma^P$；其中 S_N（分别为 S_P）是这样一个顶点集：从该集合中的顶点出发，Min（分别为 Max）可以确保在出现权重为 > 0（分别为 < 0）的任何边之前，先看到 N（分别为 P）。

6.4 变体：记忆势函数

在算法 1 的执行过程中，就在第 11 行对较小的博弈 H 进行递归调用之前，我们可以对 H 任意进行势函数约简，而不影响算法的正确性或终止性。一种有意义的选择是，应用与势函数 $\phi_{H_{\text{pred}}}$ 相关联的势函数约简；该势函数是在 while 循环的上一次迭代中计算得到的。从直观上说，这意味着我们不丢弃上一次迭代中完成的工作，而是重新利用它。直观而言，如果 H_{pred}（即 while 循环上一次迭代中的博弈 H）与 H 相似（也就是说，只移除了一个顶点；若采用上述优化，则可能移除少数几个顶点），那么我们可以预期，势函数 $\phi_{H_{\text{pred}}}$ 已经很接近使 H 成为约简的状态，因此执行势函数更新应能缩短运行时间。（请注意，采用这一变体后，我们克服了 Zielonka算法的主要弱点：每次递归调用之后，大量信息都会被丢弃。）

形式上，我们添加一个新变量，记为 ϕ_{pred}，其中包含一个势函数，初始化为 0，并在第 11 行之前添加 $H \leftarrow \texttt{ReduceGame}(H, \phi_{\text{pred}})$，在第 11 行之后添加 $\phi_{\text{pred}} \leftarrow \phi_H$。

6.5 变体：惰性版本

在这最后一个（较为激进的）变体中，我们建议一旦检测到某个顶点将离开 N_G，就结束 $\sup\Sigma^N$ 的计算。为此，每当我们在第 16 行、第 28 行或从第 9 行开始的子过程中将顶点 v' 加入 F 时，我们检查其属于 N 的前驱 v，并判断 vv' 的新权重（即 $w(vv') + \sup^N(v')$）是否会使 v 退出 N（若 v 是 Max 顶点且 $w(vv') + \sup^N(v') \geq 0$，或者 v 是 Min顶点、$w(vv') + \sup^N(v') \geq 0$，并且不存在从 v 出发的其他权重为 < 0 的出边，则会发生这种情况）。在这种情况下，我们快进到第 31 行，即立即执行势函数更新（对应于当前势函数 $\sup\Sigma^N$，此时它对应于 F 上的 $\sup\Sigma^N$-值，而在其他地方的值为 0）以及下一行的递归调用。终止性仍然得到保证，因为 N 的规模会减小。

7 讨论

我们介绍了一种求解均值收益博弈的新算法。与现有算法相比，我们认为该算法具有若干新颖之处：它是递归的、基于回溯的（用奇偶博弈的术语来说，可以称为“基于吸引集”），而且是对称的。人们或许会想称它为奇偶博弈中 Zielonka 算法 [13]的均值收益版本，因为两者具有这些共同特征；但我们的算法更为复杂，在应用于转换为均值收益博弈的奇偶博弈时，也与 Zielonka 算法有所不同。

我们认为，我们的算法有望实现次指数运行时间。这可能需要加入两项优化，并采用第6节中的一个或多个变体。迄今为止，我们尚未能对该算法进行有意义的分析，因此将此留作一个开放问题。

快速实现似乎也表明，我们的算法可能具有实际应用价值；这一问题我们也留待未来工作（以及更有资历的程序员）进一步探讨。

参考文献

1 Henrik Björklund 和 Sergei G. Vorobyov。求解均值收益博弈的组合强次指数策略改进算法。离散应用数学，155(2):210–229，2007。

20 一种用于平均收益博弈的对称递归算法

2 Patricia Bouyer、Uli Fahrenberg、Kim G. Larsen、Nicolas Markey 和 Jiří Srba。具有能量约束的加权定时自动机中的无限运行。载于 FORMATS’08，LNCS 第 5215 卷，第 33–47 页。Springer-Verlag，2008年9月。3 Lubos Brim、Jakub Chaloupka、Laurent Doyen、Raffaella Gentilini 和 Jean-François Raskin。更快的均值收益博弈算法。Formal Methods Syst. Des.，38(2)：97–118，2011年。4 Michaël Cadilhac、Antonio Casares 和 Pierre Ohlmann。快速值迭代：能量博弈高效算法的统一方法。载于 TACAS 2025，LNCS 第 15697 卷，第 323–342 页。Springer，2025年。5 Arindam Chakrabarti、Luca de Alfaro、Thomas A. Henzinger 和 Mariëlle Stoelinga。资源接口。载于 EMSOFT 2003，LNCS 第 2855 卷，第 117–133 页。Springer，柏林、海德堡，2003年。6 Dani Dorfman、Haim Kaplan 和 Uri Zwick。一种用于能量博弈和均值收益博弈的更快的确定性指数时间算法。载于 ICALP 2019，LIPIcs 第 132 卷，第 114:1–114:14 页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2019年。7 Andrzej Ehrenfeucht 和 Jan Mycielski。图上的位置博弈。Notices of the American Mathematical Society，20：A–334，1973年。8 Andrzej Ehrenfeucht 和 Jan Mycielski。均值收益博弈的位置策略。International Journal of Game Theory，109(8)：109–113，1979年。9 V. A. Gurvich、A. V. Karzanov 和 L. G. Khachiyan。有向图上的循环博弈及求极小极大环路均值的算法。USSR Computational Mathematics and Mathematical Physics，28：85–91，1988年。10 Bruno Loff 和 Mateusz Skomra。确定性折扣博弈与均值收益博弈的平滑分析。载于 ICALP 2024，LIPIcs 第 297 卷，第 147:1–147:16 页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2024年。11 Pierre Ohlmann。GKK 算法在简单均值收益博弈上速度最快。载于 CSR 2022，LNCS 第 13296 卷，第 269–288 页。Springer，2022年。12 Sven Schewe。求解奇偶博弈与收益博弈的最优策略改进算法。载于 CSL 2008，LNCS 第 5213 卷，第 369–384 页。Springer，2008年。13 Wieslaw Zielonka。有限着色图上的无限博弈及其在无限树上的自动机中的应用。Theoretical Computer Science，200(1-2)：135–183，1998年。14 Uri Zwick 和 Mike Paterson。图上均值收益博弈的复杂性。Theoretical Computer Science，158(1-2)：343–359，1996年。